From Worst-Case to Beyond: Fair, Efficient, and Truthful Decision-Making Algorithms

A dissertation submitted towards the degree Doctor of Engineering of the Faculty of Mathematics and Computer Science of Saarland University

by Golnoosh Shahkarami

Saarbrücken / 2026

Day of Colloquium:	19. June 2026
Dean of the Faculty:	Prof. Dr. Jan Reineke
Chair of the Committee:	Prof. Dr. Martina Maggio
Reporters:	Prof. Dr. Kurt Mehlhorn
	Prof. Dr. Elias Koutsoupias
	Prof. Dr. Anupam Gupta
Academic Assistant:	Dr. Evangelos Kipouridis

UNIVERSITÄT
DES
SAARLANDES

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Declaration of original authorship

I hereby declare that this dissertation is my own original work except where otherwise indicated. All data or concepts drawn directly or indirectly from other sources have been correctly acknowledged. This dissertation has not been submitted in its present or similar form to any other academic institution either in Germany or abroad for the award of any other degree.

Saarbrücken, ***02/2026***

gez. / signed

(*Name of the PhD candidate*)

Abstract

Abstract. Algorithms increasingly play a central role in decision-making, from selecting representatives and allocating shared resources to scheduling tasks in large-scale systems. Designing such algorithms raises a range of challenges. In some settings, outcomes should be fair and representative, as studied in social choice theory. In others, participants may act strategically, requiring mechanisms that ensure truthful behavior. In many applications, decisions must be made online, without full knowledge of future inputs. These challenges are unified by the presence of incomplete information.

This thesis studies algorithmic decision-making under incomplete information across social choice, mechanism design, and online algorithms. It combines classical worst-case analysis with beyond-worst-case approaches that exploit additional structure, including potentially incorrect predictions. The thesis develops algorithms that achieve strong performance guarantees while respecting constraints such as fairness, truthfulness, and robustness. A central theme is understanding which forms of additional information are truly useful, how they interact with algorithmic guarantees, and when fundamental limitations cannot be overcome. Together, the results advance the theoretical foundations of decision-making algorithms and provide a unified perspective on fairness, incentives, uncertainty, and prediction in algorithmic systems.

Zusammenfassung. Algorithmen spielen eine zunehmend zentrale Rolle in Entscheidungsprozessen, etwa bei der Auswahl von Repräsentanten, der Zuteilung gemeinsamer Ressourcen oder der Planung von Aufgaben in großen Systemen. Die Entwicklung solcher Algorithmen ist mit unterschiedlichen Herausforderungen verbunden. In einigen Kontexten sollen Entscheidungen fair und repräsentativ sein, etwa bei Wahlen. In anderen handeln Beteiligte strategisch, sodass Mechanismen erforderlich sind, die wahrheitsgemäßes Verhalten sicherstellen. In vielen Anwendungen müssen Entscheidungen zudem online getroffen werden, ohne Kenntnis zukünftiger Eingaben. Allen diesen Problemen ist gemeinsam, dass Entscheidungen unter unvollständiger Information getroffen werden.

Diese Dissertation untersucht algorithmische Entscheidungsfindung unter unvollständiger Information bei Wahlen, im Mechanismendesign und bei Online-Algorithmen. Sie verbindet klassische Worst-Case-Analyse mit Beyond-Worst-Case-Ansätzen, die zusätzliche Struktur nutzen, etwa in Form möglicherweise unzuverlässiger Vorhersagen. Ziel ist es, Algorithmen zu entwickeln, die trotz Fairness-, Anreiz- und Robustheitsanforderungen starke Leistungszusagen bieten. Ein zentrales Thema ist, welche zusätzliche Information tatsächlich hilfreich ist, wie sie algorithmische Garantien beeinflusst und wann grundlegende Grenzen bestehen.

Acknowledgments

I feel incredibly fortunate to have been supervised by Kurt. I am deeply grateful to him for always being there whenever I needed support, scientific or otherwise. Thank you for trusting me, encouraging me, and giving me the space to grow into an independent researcher.

I am thankful to all the people who guided and supported me throughout my PhD, as well as to my collaborators. In particular, I would like to thank Antonios, who introduced me to learning-augmented algorithms and was endlessly patient during my first attempts at doing research and writing papers. I also thank Vasilis, who supervised me through our project. He trusted me to start a collaboration based on a single email, and our meetings, all of which were online, were always engaging and deeply motivating. I am also grateful to Ulrike, Bhaskar, and Javier for their thoughtful feedback on my thesis and presentations, their encouragement and insightful discussions, and for making this journey both more enjoyable and much less daunting.

I am very grateful to Elias and Anupam for agreeing to serve on my PhD committee, and for their time, support, and insightful questions. My thanks also go to Martina for chairing my defense, and to Evangelos for serving as my Academic Assistant.

Many thanks go to all current and former members of the Algorithms and Complexity group at MPII, as well as to the amazing IST team. I am especially grateful for the support of our wonderful administration, Elisa, Heike, Gabi, and Christina. I was lucky to be surrounded by incredible colleagues and friends. During my first years living in Germany, this included Karl, Christoph, Ian, Hans, Andreas, Pieter, Atila, Cosmina, Roohani, and Hana. In more recent years, I benefited greatly from the support of Danupon, and from interactions with Guy, Sander, Karol, Evangelos, Rex, Aryan, and Nidhi. Each of them has left a lasting mark on my PhD and on my life.

I am also deeply thankful to my Persian friends. I thank Sepehr for being a true friend throughout all these years, and Rozhin, Mandana, Elnaz, and Farzane for their friendship, kindness, and the many joyful moments we shared. They gave me a sense of home, not only by celebrating Persian traditions together, but also by always being there when I needed them.

To my partner, Andrea, thank you for standing by me through many challenges, both during my PhD and beyond it. Your support made all the difference. You make this place feel like home, and our many joyful moments together have meant a great deal to me. I am also grateful to his family, Fiorella, Carlo, and Marco, for their warmth and support.

Finally, I would like to thank my brother, Farshad, and my sister-in-law, Mahtab, for their love and encouragement. Above all, I thank my parents, Baba (Mohsen) and Maman (Mansoore), who dedicated their lives to giving us the best possible one. None of this would have been possible without their love, trust, and unwavering support. Mersi!

Preface

During my PhD, I had the opportunity to work on a broad range of topics. To give an overview of my research contributions, I list all of my publications below.

At the time of writing, I have published fourteen conference and journal papers. In addition, this thesis includes results from work that is currently under submission.

1 Akrami, H., B. R. Chaudhury, M. Hoefer, K. Mehlhorn, M. Schmalhofer, G. Shahkarami, G. Varricchio, Q. Vermande, and E. van Wijland (2022). "Maximizing Nash Social Welfare in 2-Value Instances." In: *AAAI*. AAAI Press, pp. 4760–4767.

2 Akrami, H., B. R. Chaudhury, M. Hoefer, K. Mehlhorn, M. Schmalhofer, G. Shahkarami, G. Varricchio, Q. Vermande, and E. van Wijland (2026). "Maximizing Nash Social Welfare in Two-Value Instances: Delineating Tractability." In: *Math. Oper. Res.* 51, pp. 853–876.

3 Akrami, H., K. Mehlhorn, M. Seddighin, and G. Shahkarami (2023). "Randomized and Deterministic Maximin-Share Approximations for Fractionally Subadditive Valuations." In: *NeurIPS*, pp. 58821–58832.

4 Amiri, S. A., A. Popa, M. Roghani, G. Shahkarami, R. Soltani, and H. Vahidi (2020). "Complexity of Computing the Anti-Ramsey Numbers for Paths." In: *MFCS*. Vol. 170. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 6:1–6:14.

5 Amiri, S. A., A. Popa, M. Roghani, G. Shahkarami, R. Soltani, and H. Vahidi (2025). "Complexity of Computing the Anti-Ramsey Numbers for Paths." In: *Theor. Comput. Sci.* 1044, p. 115271.

6 Antoniadis, A., P. Jabbarzade, and G. Shahkarami (2022). "A Novel Prediction Setup for Online Speed-Scaling." In: *SWAT*. Vol. 227. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 9:1–9:20.

7 Antoniadis, A., A. Shahheidar, G. Shahkarami, and A. Soltani (2026). "A Switching Framework for Online Interval Scheduling with Predictions." In: *AAAI*. AAAI Press, pp. 36129–36136.

8 Babashah, N., H. Karimi, M. Seddighin, and G. Shahkarami (2025a). "Distortion of Multi-Winner Elections on the Line Metric: The Polar Comparison Rule." In: *SAGT*. Vol. 15953. Lecture Notes in Computer Science. Springer, pp. 441–464.

9 Babashah, N., H. Karimi, M. Seddighin, and G. Shahkarami (2025b). "On the Distortion of Multi-Winner Elections on the Line Metric." In: *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, pp. 2423–2425.

10 Balkanski, E., V. Gkatzelis, and G. Shahkarami (2024). "Randomized Strategic Facility Location with Predictions." In: *NeurIPS*, pp. 35639–35664.

11 Bampis, E., B. Escoffier, T. Gouleakis, N. Hahn, K. Lakis, G. Shahkarami, and M. Xefteris (2023). "Learning-Augmented Online TSP on Rings, Trees, Flowers and (Almost) Everywhere Else." In: *ESA*. Vol. 274. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 12:1–12:17.

12 Bonifaci, V., E. Facca, F. Folz, A. Karrenbauer, P. Kolev, K. Mehlhorn, G. Morigi, G. Shahkarami, and Q. Vermande (2022). "Physarum-Inspired Multi-Commodity Flow Dynamics." In: *Theor. Comput. Sci.* 920, pp. 1–20.
13 Borst, S., G. Shahkarami, and R. Vaish (2026). "Fair Division Meets Scheduling: Approximately Envy-Free Interval Scheduling." Under review.
14 Cembrano, J. and G. Shahkarami (2026). "Metric Distortion in Peer Selection." In: *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM.
15 Gouleakis, T., K. Lakis, and G. Shahkarami (2023). "Learning-Augmented Algorithms for Online TSP on the Line." In: *AAAI*. AAAI Press, pp. 11989–11996.

A central theme of this thesis is learning-augmented algorithms, which connect classical worst-case analysis with data-driven approaches by incorporating predictions. Several of my works pursue this direction, including papers on online optimization [6, 7, 11, 13, 15] and mechanism design with predictions [10].

More broadly, my research can be grouped into the following areas. Social choice theory, including fair division [1, 2, 3] and voting rules [8, 9, 14]. Mechanism design [13]. Online algorithms, including online scheduling [6, 7], online TSP [11, 15], and graph problems [4, 5, 12]. In addition, my work on fair scheduling [13] naturally lies at the intersection of scheduling and fair division.

Scope of the thesis. To maintain focus while still reflecting the breadth of my research, this thesis includes only a subset of these works. From social choice theory, I include papers on voting rules, with a particular emphasis on committee selection. From online problems, I include only scheduling-related work. Papers on graph problems are not included. The papers whose results and content appear in this thesis are [6, 7, 8, 9, 10, 13, 14].

On collaboration. All works included in this thesis were carried out in close collaboration with my co-authors. Contributions are equally shared among the authors of each paper. The ordering of authors in all publications follows the alphabetical convention.

Contents

CHAPTER 1

Introduction

1.1 Decision-Making with Incomplete Information

Algorithms increasingly make decisions that affect individuals, organizations, and society at large. They determine how resources are allocated among users, how representatives are elected, how tasks are scheduled in data centers, where public infrastructure should be placed, and how services are delivered in real time. In all these settings, an algorithm must transform limited input into principled decisions while balancing objectives such as efficiency, fairness, and reliability. Despite their diversity, these decision-making problems share a fundamental challenge: they must operate under incomplete information.

Strategic Behavior and Incentives. Incomplete information arises in several distinct but related ways. In some settings, the information needed to make an optimal decision exists but is deliberately obscured. In mechanism design, for example, individuals may strategically misreport their preferences if doing so benefits them. If an algorithm naively trusts such reports, the resulting outcome may be inefficient or unfair. This raises a central question: how can we design rules that align individual incentives with desirable collective outcomes? Ensuring that agents have no incentive to manipulate the system through truthful or strategyproof mechanisms is essential in such environments.

Uncertainty About the Future. In other settings, incomplete information is not the result of strategic behavior, but of uncertainty about the future. Online decision-making problems exemplify this challenge. Tasks or requests arrive sequentially over time, and decisions must often be made immediately and irrevocably. Here, the difficulty lies in acting without knowing what inputs will arrive next. Classical online algorithms provide worst-case guarantees for such problems, but these guarantees are often pessimistic and driven by highly adversarial scenarios.

Limited Preference Information. A third manifestation of incomplete information appears in collective decision-making problems such as voting and committee selection. While individuals may have rich, nuanced preferences, they typically express them in a limited form, for instance through rankings rather than precise numerical values. This restriction is often unavoidable in practice, yet it forces decision-making procedures to operate with only partial access to the underlying information. Understanding how much efficiency or fairness is lost due to this limitation is a central concern in social choice theory.

Leveraging Predictive Information. Importantly, incomplete information does not always mean a complete absence of guidance. In many modern systems, additional signals

about the unknown input are available. Historical data, forecasts, or machine-learned models can provide predictions about future events, hidden preferences, or desirable outcomes. While such predictions are inherently imperfect, they open the possibility of moving beyond purely worst-case reasoning. This motivates the study of algorithms with predictions, which are designed to exploit predictive information when it is informative while retaining strong guarantees when predictions are inaccurate or misleading. A recurring challenge is how to benefit from predictions when they are helpful without becoming fragile when they are wrong.

Scope and Questions of This Thesis. This thesis explores these questions across three domains of algorithmic decision-making. In mechanism design, we examine how decision rules can be designed to be robust in the presence of strategic behavior and how additional information interacts with incentive constraints. In social choice theory, we study how collective decisions can be made from limited preference information and how efficiency and fairness can be evaluated under such constraints. In online algorithms, we investigate how decisions can be made effectively under uncertainty about future inputs and how predictions can be used to improve performance despite irrevocable choices.

Across these domains, the central goal is to understand the fundamental trade-offs imposed by incomplete information. This thesis seeks to identify which forms of additional information are truly useful, to distinguish between limitations that can be overcome and those that are inherent, and to clarify how principles such as fairness, robustness, and randomness shape what can ultimately be achieved.

Reading guide. The remainder of this introduction elaborates on these themes by first presenting the relevant research areas and problems in a largely non-technical manner, with the aim of orienting readers who are unfamiliar with some of the settings. Readers primarily interested in the broader insights of the thesis may wish to proceed directly to Section 1.4, which distills the main conceptual messages and highlights emerging research directions, before consulting the outline of the thesis.

1.2 Research Areas

Each problem studied in this thesis can be classified within one or more established research areas in theoretical computer science. This section introduces these areas at a high level, with the goal of familiarizing the reader with the conceptual frameworks that underlie the work. Rather than focusing on specific models or results, it provides a common language for understanding the types of problems considered and the overarching goals pursued within each area. Each research area corresponds to a different source of incomplete information, such as strategic behavior, limited preference elicitation, or uncertainty about future inputs. Before introducing these areas in turn, we first discuss the learning-augmented perspective, which cuts across all of them and provides a unifying lens for the thesis.

1.2.1 Algorithms with Predictions

Worst-case analysis has long served as the backbone of theoretical computer science. By evaluating an algorithm's performance on the most challenging inputs, it provides guaran-

tees that hold universally, shielding us from adversarial or highly irregular instances. This perspective has proven extraordinarily powerful: it yields clean mathematical guarantees, fosters elegant algorithmic design, and remains indispensable in safety-critical domains where reliability is paramount. Yet its strength is also a limitation. Worst-case analysis treats all inputs as equally likely and evaluates algorithms according to their behavior on a single, possibly pathological instance, even when such instances are unlikely to arise in practice.

In many real-world systems, this pessimism is unwarranted. Applications often provide auxiliary information about the instance at hand—historical data, statistical forecasts, or machine-learned predictions—that can be remarkably informative. Pure worst-case analysis ignores this structure entirely, thereby potentially overlooking opportunities to design algorithms that combine practical efficiency with theoretical guarantees. This observation has motivated a broad line of work on *beyond worst-case analysis* (Roughgarden, 2021), which seeks to retain rigorous guarantees while better reflecting typical inputs.

The *learning-augmented*, or *algorithms with predictions*, framework embodies one such approach. Here, an algorithm receives a prediction about the uncertain or unseen part of the input—for instance, future job arrivals in a scheduling problem, hidden cardinal values in a collective decision, or an estimated optimal outcome produced by a learned model. Crucially, these predictions are not trusted blindly: they may be approximate, biased, or even adversarially wrong. As a result, a learning-augmented algorithm must carefully balance the use of predictive information with the need for robust fallback behavior.

This balance is commonly captured through three complementary dimensions, illustrated schematically in Figure 1.1: consistency, robustness, and smoothness. Together, these criteria formalize the tension between exploiting predictions when they are helpful and protecting performance when they fail.

- **Consistency:** Near-optimal performance when the prediction is accurate.
- **Robustness:** Classical worst-case guarantees even when predictions fail completely.
- **Smoothness:** Graceful degradation in performance as the prediction error increases.

These criteria encapsulate the central ethos of learning-augmented design: *preserving the strengths of worst-case guarantees while exploiting additional structure provided by predictions.* A key modeling choice is the prediction itself—what information is predicted, how error is measured, and how predictions interact with algorithmic structure. Different applications naturally give rise to different prediction models, and the achievable trade-offs between consistency and robustness often depend critically on these choices.

The learning-augmented paradigm provides a unifying lens for many problems involving incomplete information. Across domains such as strategic decision-making, collective choice under limited input, and online computation with uncertain futures, predictions offer a principled way to inject structure without abandoning reliability. Understanding when and how such information can be used effectively is central to modern algorithm design and motivates several of the research directions explored in this thesis.

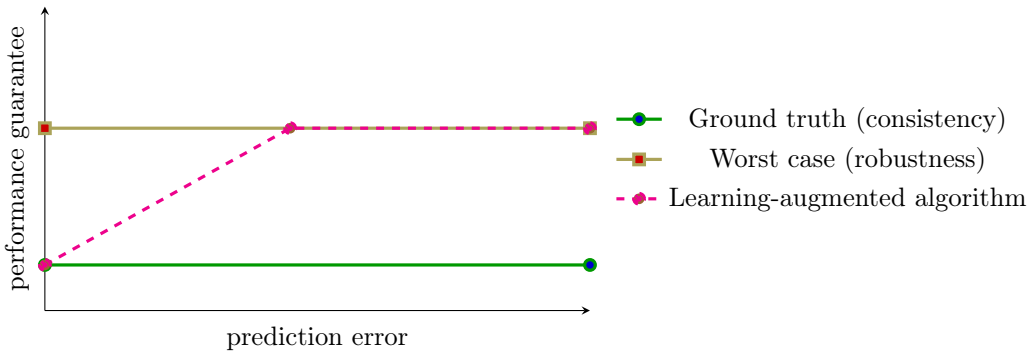


Figure 1.1: Consistency, robustness, and smoothness in learning-augmented algorithms. The figure is schematic and illustrates typical performance trends as a function of the prediction error, rather than representing specific data points. A learning-augmented algorithm matches ideal performance under perfect predictions, degrades gradually as the prediction error increases, and retains a worst-case guarantee when predictions are uninformative.

1.2.2 Mechanism Design and Truthfulness

Mechanism design studies how to make decisions in environments where the relevant information is held by self-interested individuals. Unlike classical algorithm design, where the input is assumed to be given correctly, mechanism design confronts the fact that participants may benefit from misreporting their preferences. If an algorithm simply trusts the information it receives, strategic manipulation can distort outcomes and undermine both efficiency and fairness. The central question of mechanism design is therefore how to design rules that lead to desirable outcomes even when agents act in their own interest.

A foundational concept in this area is *truthfulness*, also known as strategyproofness. A mechanism is truthful if each agent's best option is to report their private information honestly, regardless of what others do. This notion captures incentive compatibility in the sense that agents have no incentive to deviate from truthful reporting. This requirement fundamentally shapes what can be achieved: while it prevents manipulation, it also imposes strong constraints on the design space and often rules out optimal solutions altogether.

Consider an auction in which bidders must report how much an item is worth to them. In some auction formats, bidders can benefit from misreporting their value. For example, if the item is sold to the highest bidder at the price they bid, a bidder may have an incentive to bid less than their true value. In contrast, if the winner pays the second-highest bid, reporting the true value becomes the best strategy. Mechanism design seeks rules of the latter kind, where, by design, honesty is aligned with self-interest.

At a high level, a mechanism can be viewed as a rule that maps reported information to an outcome, as illustrated in Figure 1.2. The designer specifies which outcomes are feasible and how they are selected based on agents' reports. Truthfulness ensures that this mapping remains stable under strategic behavior, but achieving it typically comes at a cost. In many settings, especially those without monetary transfers, impossibility results show that no truthful mechanism can simultaneously satisfy all desirable properties. Even

Figure 1.2: A mechanism takes agents' reported private information as input and produces an outcome according to fixed rules. Truthfulness requires that reporting honestly is always in an agent's best interest.

when approximation is allowed, the resulting worst-case guarantees can be discouragingly weak.

These difficulties are particularly apparent in the *prior-free* setting, where no assumptions are made about how agents' private information is distributed. While this setting offers strong robustness guarantees, it often leads to pessimistic conclusions about achievable performance. As a result, mechanism design faces a persistent tension between incentive compatibility and efficiency.

From an algorithmic perspective, this tension becomes especially relevant in large-scale or dynamic environments, where decisions must be computed efficiently and repeatedly. Examples include allocating resources, assigning tasks, or determining facility locations based on agents' reported preferences. In such settings, the challenge is not only to enforce truthful behavior, but to do so while maintaining reasonable performance and computational tractability.

This is where additional information can play a crucial role. In many practical applications, the designer may have access to partial signals about agents' preferences or about desirable outcomes, derived from historical data or predictive models. While such information cannot be trusted outright, it can nevertheless guide decision-making in meaningful ways. In mechanism design, the key question is whether these signals can be used to mitigate the limitations imposed by truthfulness, without creating new opportunities for manipulation or undermining robustness.

Understanding how predictive information interacts with incentive constraints is therefore a central challenge in modern mechanism design. It highlights why strategic behavior, incomplete information, and uncertainty must be studied together, and why careful algorithmic design is essential when incentives and limited information coexist.

1.2.3 Social Choice and Fairness

Social choice theory studies how groups make decisions. Whenever several individuals must agree on a common outcome—such as electing a representative, choosing a committee, or selecting a shared option—their preferences need to be combined in a principled way. This problem arises naturally in democratic societies, but also in much smaller settings, such as academic committees, student councils, or groups of friends deciding how to act collectively. At its core, social choice asks how individual preferences can be aggregated into a collective decision that is reasonable and defensible.

A familiar example is an election. Each voter may have a clear opinion about which candidate they prefer, yet the group must select a single winner or a small set of winners. To make participation simple and scalable, voters are typically asked to provide rankings rather than precise numerical values. While this simplifies preference elicitation, it also limits the information available to the decision rule. As a result, voting rules must

operate under significant informational constraints, and even natural procedures may lead to outcomes that some voters perceive as unsatisfactory.

This observation motivates much of the theoretical study of voting. Rather than asking whether a voting rule always produces the best possible outcome, social choice theory examines which properties can be guaranteed when only limited preference information is available. Key questions include how well different rules represent diverse opinions, how minority preferences are treated, and how far the chosen outcome may deviate from an ideal one that would be selected if richer information were known.

Closely related to voting is the area of *fair division*, which focuses on how resources should be allocated among individuals. Classic examples include dividing a cake among several people or splitting time or land—settings in which resources are divisible and can be shared fractionally. In such cases, it is often possible to design procedures that guarantee strong notions of fairness, ensuring that each participant feels adequately treated.

Many real-world situations, however, involve *indivisible* items, such as dividing an inheritance that includes a house, a car, or a family heirloom. Here, items cannot be split without losing their value, and the problem becomes fundamentally more challenging. To see this, consider the simplest possible example: two agents and a single indivisible item that both value. No matter how the item is allocated, one agent receives it and the other does not, making certain ideal notions of fairness impossible to satisfy.

This simple example illustrates a central insight of fair division. When resources are indivisible, perfect solutions may not exist, even in the most basic settings. As a result, the field has developed a range of relaxed notions that aim to capture equitable treatment while respecting these limitations. Some focus on envy, asking whether an agent strongly prefers someone else's allocation; others focus on proportionality, asking whether each agent receives a reasonable share relative to what they could guarantee for themselves; and still others balance fairness and efficiency through global welfare measures. Overall, these approaches allow us to reason systematically about achievable solutions and unavoidable trade-offs.

Taken together, voting and fair division offer complementary perspectives on group decision-making. Both confront conflicting preferences, limited information, and structural constraints that rule out ideal outcomes. By understanding where perfect solutions are impossible and which compromises are principled, social choice theory provides a foundation for designing and evaluating decision-making procedures that affect many individuals at once.

1.2.4 Online Algorithms and Efficiency

Online algorithms study decision-making in settings where information is revealed gradually over time. Unlike classical offline algorithms, which have access to the full input in advance, online algorithms must act without knowing what will happen next. This lack of foresight is not an artificial constraint but a defining feature of many real-world systems, where waiting for complete information is impossible or impractical. As a result, decisions must often be made immediately and cannot be revised later, even if new information arrives.

A simple way to see this challenge is through scheduling. Consider a system that must process tasks using limited resources, such as assigning jobs to machines, deliveries to vehicles, or computations to servers. In many applications, tasks arrive over time, and when a task appears, the system must decide when and how to process it without knowing which tasks will arrive in the future. Decisions made early can strongly constrain what is possible later: committing resources to one task may block the execution of more urgent tasks that arrive shortly thereafter, while excessive caution can leave valuable capacity unused. Understanding which performance losses are unavoidable under such uncertainty, and which can be mitigated by careful algorithmic design, is a central theme in the study of online scheduling.

Scheduling problems provide a central model for studying these trade-offs. They arise whenever scarce resources must be shared among competing tasks while optimizing an objective, such as minimizing completion time, reducing energy consumption, or maximizing throughput. Examples range from production planning and service systems to data processing and cloud computing. Despite the diversity of applications, these problems share a common structure: tasks compete for limited resources, and decisions about their execution have long-lasting consequences.

The study of online algorithms asks how well such decisions can be made in the face of uncertainty. Since optimal solutions are typically unattainable without full knowledge of the future, the goal is to design algorithms that perform provably well across all possible scenarios. This leads to worst-case performance guarantees that compare an online decision process to an ideal benchmark that has complete foresight. While these guarantees provide robustness, they also reveal fundamental limitations, as uncertainty about future inputs can force online algorithms to make conservative choices that are suboptimal in hindsight.

At the same time, real systems often exhibit structure that worst-case analysis ignores. While the future is unknown, it is rarely completely unpredictable. Historical data, operational constraints, or empirical patterns may provide partial guidance about what is likely to happen next. A major goal in modern online algorithm design is to understand how such information can be incorporated to improve efficiency, without sacrificing the robustness that makes online algorithms reliable in the first place.

Together, online algorithms and scheduling offer a principled framework for reasoning about efficiency under uncertainty. They capture the tension between immediate action and future flexibility, and they provide tools for understanding how algorithmic decisions should be made when information arrives over time.

1.3 Research Problems

This thesis investigates a collection of problems that arise in strategic decision-making, social choice, and online scheduling, each highlighting different facets of reasoning under incomplete information. The following subsections introduce the four problem domains studied in the thesis, emphasizing the modeling choices and conceptual challenges that motivate the results developed in later chapters.

1.3.1 Facility Location

The facility location problem is a classical model for collective decision-making in which a group of agents must agree on the location of a shared facility (Procaccia and Tennenholtz, 2013). Each agent is associated with a position in a metric space, and the quality of a chosen location is measured by how far it lies from the agents. Depending on the application, the goal may be to minimize the total distance to all agents, capturing overall efficiency, or to minimize the maximum distance to any agent, protecting the worst-off individual. The model also admits natural variants, such as placing multiple facilities or considering *obnoxious* facilities that agents wish to avoid, each introducing additional complexity and trade-offs.

The strategic version of the problem adds an important layer of difficulty. Agents may misreport their locations if doing so leads to a more favorable outcome for them. As a consequence, mechanisms that simply optimize based on reported information can be manipulated. This motivates the design of *truthful* mechanisms, which ensure that reporting the true location is always an agent's best course of action.

A further dimension of variation concerns the nature of the mechanism itself. Some mechanisms are deterministic, producing the same outcome for a given set of reports, while others employ randomization to balance competing objectives or to mitigate worst-case behavior. Each of these design choices interacts in subtle ways with truthfulness and performance guarantees, and different variants of the problem highlight different aspects of this interaction.

A central challenge in strategic facility location is that worst-case analysis can be overly pessimistic. Truthfulness constraints may force mechanisms to perform poorly on carefully constructed instances, even though such instances may be rare in practice. At the same time, in many real-world settings, partial information about agents' locations or about desirable facility placements may be available from historical data, forecasts, or learned models. Incorporating such information raises natural questions about how predictions can be used to improve performance without sacrificing robustness.

These considerations make strategic facility location a natural research problem to study in the learning-augmented framework.

1.3.2 Committee Selection

Social choice theory asks how a group should make a collective decision from individual preferences. In a *single-winner* election, voters rank candidates and a voting rule selects one winner. In many applications, however, the outcome is not a single alternative but a *committee*: a set of k candidates to serve as representatives, reviewers, or decision-makers. The multi-winner setting raises an immediate conceptual question: what does it mean for a committee to be "good" for the electorate? Depending on the context, one may prioritize overall efficiency (serving voters well on average) or fairness (ensuring that no voter is left far behind).

A widely used way to formalize these goals is through a *metric* model. Voters and candidates are assumed to lie in a common metric space, and a voter's (dis)utility for a candidate is identified with their distance. The cost of a committee is then obtained by aggregating distances in two steps: first *for* each voter, and then *across* voters. This naturally yields four objective functions, obtained by combining two standard choices at

each level. For a voter, one can use the *additive* cost (sum of distances to all committee members) or the *q-cost* (distance to the qth closest committee member), the latter capturing more local notions of representation. Across voters, one can aggregate costs in a *utilitarian* way (sum over voters) or an *egalitarian* way (maximum over voters). These four combinations provide a clean vocabulary for discussing different notions of representational quality in committee selection.

The main difficulty, of course, is informational: voting rules typically observe only *ordinal* preferences, not the underlying distances. The *metric distortion* framework quantifies how costly this limitation can be. Fix a voting rule and an objective (one of the four above). For a given instance, we observe only the voters' rankings; there may be many different metric spaces and embeddings of voters and candidates that induce exactly these same rankings. The distortion of the rule is defined as the worst-case ratio between the social cost of the committee it selects and the optimal social cost, *taken over all metrics consistent with the observed rankings*. In other words, distortion measures the maximum loss in welfare that can occur because the rule must act on rankings alone, without knowing which underlying metric generated them.

A structurally important special case is *peer selection*, where voters and candidates coincide. Here, a group must elect a committee from among its own members—examples include hiring or promotion committees, self-governance in organizations, and leadership selection in communities. This restriction introduces additional structure (each agent is a candidate as well), and it is natural to ask whether that structure can alleviate the limitations captured by distortion. At the same time, peer selection remains an ordinal decision problem: even with additional structure, the rule still only sees rankings, and the underlying metric remains unknown. Understanding how distortion behaves in this special case provides insight into which barriers are truly inherent to ordinal information and which stem from the generality of the setting.

1.3.3 Energy-Efficient Scheduling

Energy efficiency has become a central concern in modern computing systems. Data centers, cloud platforms, and large-scale computing infrastructures must process growing workloads while limiting energy consumption, both for economic and environmental reasons. A common technological response to this challenge is *dynamic speed scaling*, which allows processors to adjust their operating speed over time. Running at a higher speed enables tasks to be completed more quickly, but at a disproportionately higher energy cost, whereas running at a lower speed saves energy but reduces processing capacity. As a result, scheduling decisions directly influence not only performance but also energy usage.

These trade-offs are particularly pronounced in deadline-based scheduling problems. Jobs arrive over time, each with a release time, a deadline, and a required amount of processing. The scheduler must decide, at every moment, which job to run and at what speed, ensuring that all jobs are completed before their deadlines. The objective is to balance timeliness and energy consumption, typically by minimizing a nonlinear energy cost that reflects processor behavior. While classical online algorithms provide worst-case guarantees for such problems, these guarantees are often pessimistic and driven by highly adversarial scenarios.

In practice, however, historical data, usage patterns, or forecasting models may offer partial information about upcoming jobs, their sizes, or their timing. This observation motivates the study of energy-efficient scheduling algorithms that can make use of such information while remaining reliable when it turns out to be inaccurate. The central challenge is to understand how predictions can guide speed and scheduling decisions to reduce energy consumption, without compromising robustness to uncertainty.

Energy-efficient scheduling thus provides a clean and practically relevant setting to study decision-making under uncertainty. It highlights the tension between immediate action and future flexibility, and illustrates how predictive information can improve efficiency when used carefully. At the same time, it exposes fundamental limits on what can be achieved when future inputs are only partially known, making it a natural problem domain to explore in this thesis.

1.3.4 Online Interval Scheduling

Online interval scheduling is one of the most fundamental models for decision-making under uncertainty. Intervals, representing tasks or requests, arrive over time, and the algorithm must decide whether to accept or reject each one before knowing what will arrive next. Accepted intervals must not overlap, reflecting limited resources that cannot be shared simultaneously. Unlike the energy-efficient scheduling problem discussed earlier, which is a minimization problem, interval scheduling is inherently a *maximization* problem: the goal is to accept as much value as possible while maintaining feasibility. This value may correspond to the total processing time, the number of serviced requests, or overall system throughput. Because of its simplicity and broad applicability, interval scheduling has become a canonical model for studying the power and limitations of online algorithms.

A first natural setting considers a single machine with *irrevocable* decisions. Once an interval is accepted, it must be scheduled to completion, and any overlap makes the schedule infeasible. The objective in this case is to maximize the total length, or proportional weight, of the accepted intervals. This setting captures scenarios in which commitments cannot be undone, such as reserving time slots or allocating exclusive resources. While classical online algorithms achieve strong worst-case guarantees, these guarantees are often dictated by carefully constructed adversarial instances. In many practical environments, however, some information about future intervals is available through historical data or predictive models, motivating algorithms that can take advantage of such signals without sacrificing robustness.

A second line of inquiry extends the model in two important directions: multiple machines and revocable decisions. Here, intervals may be assigned to any of m identical machines, and previously accepted intervals may be revoked before completion. The primary objective shifts to maximizing the number of accepted intervals, reflecting settings where servicing more requests is more important than their individual durations. This extension naturally connects interval scheduling to ideas from fair division. Machines can be viewed as agents competing for intervals, and one may require that the allocation remains fair, for instance by ensuring that no machine is significantly worse off than another.

Imposing fairness as a hard constraint fundamentally changes the nature of the problem. It restricts the set of admissible schedules and introduces new trade-offs between efficiency and equity. Studying these trade-offs in an online setting raises basic questions about what performance is achievable when decisions must be made sequentially, information is incomplete, and fairness requirements must be respected at all times. Through its many variants, online interval scheduling provides a flexible and expressive framework for exploring how uncertainty, predictions, revocability, and fairness interact in algorithmic decision-making.

1.4 General Messages of the Thesis

While each chapter of this thesis addresses a specific problem, many of the most important insights only become visible when these results are viewed together. Across different models and application domains, recurring technical themes emerge, revealing common principles that shape algorithmic decision-making under incomplete information. This section highlights a small number of such general messages, each of which is developed across one or more chapters of the thesis. Taken together, these messages articulate the conceptual contribution of the thesis beyond its individual chapters.

1.4.1 Understanding What Predictions Truly Matter

Predictions in learning-augmented algorithms can take many different forms. They may suggest a specific outcome, reveal partial information about the input, or forecast future events. A central message of this thesis is that, for any given problem, it is important to study *which* prediction settings are available and to understand the power and limitations of each.

Necessary and Sufficient Predictions. This matters because predictive information is rarely free. Richer predictions typically come with a higher cost: they may require stronger modeling assumptions, more detailed data collection, or more complex learning pipelines. At the same time, having access to more information does not automatically translate into better algorithmic guarantees. For many problems, performance cannot be improved beyond certain barriers even if the prediction is very informative. This makes it crucial to identify the *minimum* predictive information needed to achieve a desired guarantee, and to distinguish information that is genuinely valuable from information that is redundant.

Incomparable Prediction Models. A further complication is that prediction settings are often not directly comparable. Some forms of information differ in nature rather than strength, so one prediction model is not necessarily a refinement of another. Consequently, progress in the area is not only about designing algorithms for a fixed prediction model, but also about clarifying how guarantees change as the underlying prediction assumptions change.

In this thesis, these themes are illustrated through a combination of complementary results. In some settings, upper and lower bounds are used to characterize the exact power of specific prediction models, showing both what can be achieved and what remains impossible even with stronger information. In other settings, the contribution lies in introducing novel prediction models that capture different sources of information and

are fundamentally incomparable with previously studied ones. Together, these results emphasize that progress in learning-augmented algorithms depends as much on choosing and understanding the right prediction model as on the design of the algorithm itself. These themes are developed in Chapters 2, 5 and 6.

1.4.2 A Unifying Technique for Robustness

Beyond the question of what information is useful lies the question of how it should be used algorithmically. A recurring methodological insight in the study of learning-augmented algorithms is the idea of combining two fundamentally different decision-making approaches. One approach actively exploits predictions and can achieve excellent performance when the available information is accurate. The other ignores predictions altogether and follows a classical worst-case strategy, guaranteeing reliable behavior under all circumstances. While each approach is well understood in isolation, the main challenge lies in merging them in a principled and problem-dependent way.

Switching Between Algorithms. Different problem settings give rise to different ways of combining these two approaches. In online algorithms, a common technique is *switching*: the algorithm initially follows the prediction-driven strategy and switches to a robust alternative once the prediction error exceeds a certain threshold. In some settings, switching can be applied multiple times, allowing the algorithm to return to the optimistic strategy if the quality of the prediction improves.

Resource Splitting. In problems with divisible resources, a different form of combination becomes possible. Rather than switching over time, one can split the resource itself and apply different strategies to different parts. For example, part of the resource may be allocated according to a prediction-based rule, while the remainder is handled by a robust baseline algorithm. This approach naturally arises in settings such as online divisible allocation and scheduling, where time or capacity can be continuously divided.

Randomized Combination. Randomization offers yet another avenue for combining strategies. By randomizing over decisions, an algorithm can effectively blend multiple strategies and benefit from their behavior in expectation. This can provide robustness without requiring explicit switches or rigid resource partitioning, although it introduces its own technical challenges.

Across all these approaches, the central difficulty is to combine algorithms in a way that respects the constraints of the problem while achieving the desired guarantees: strong performance when predictions are accurate, and robustness when they are not. The precise form of the unifying technique depends on the structure of the problem and the type of uncertainty involved.

While switching has been widely studied in minimization settings, this thesis adapts it to maximization objectives and complements it with resource-splitting and randomized approaches in other problems. These ideas are developed in Chapters 2, 5 and 6. The overarching message is that robustness in learning-augmented algorithms is often achieved not by discarding predictions, but by carefully controlling their influence through principled combinations with worst-case methods.

1.4.3 Fairness as a Lens on Efficiency

Fairness plays a central role throughout this thesis, not as a secondary consideration, but as a defining element of how efficiency is interpreted and measured. In many algorithmic settings, fairness cannot be imposed without cost. Instead, it shapes which objectives are considered meaningful, which solutions are acceptable, and which trade-offs are unavoidable. Viewing fairness through this lens provides a clearer understanding of what algorithmic efficiency truly represents in decision-making problems that affect multiple participants.

Egalitarian Objectives. One way fairness enters algorithmic models is through the choice of the objective function. Rather than optimizing average performance, an algorithm may focus on protecting the worst-off individual, for example by minimizing the maximum cost experienced by any participant. Such egalitarian objectives embed fairness directly into the notion of efficiency, prioritizing the avoidance of extreme outcomes over aggregate optimality. In this perspective, efficiency is measured not by how well the system performs overall, but by how well it treats its most disadvantaged participant.

Balanced Efficiency Objectives. Fairness can also be reflected indirectly through the way efficiency is measured. In many optimization problems, solutions that perform well on average may do so by placing a disproportionate burden on a small subset of participants. Objectives that reward more balanced behavior, by discouraging extreme disparities across agents or over time, naturally limit such outcomes. In this case, fairness is not imposed as an explicit requirement, but emerges from an efficiency objective that favors stability and balance rather than uneven performance.

Fairness as a Constraint. A third approach treats fairness as an explicit restriction on the set of admissible solutions. Instead of being optimized, fairness acts as a constraint that determines which outcomes are acceptable at all. This viewpoint is common in fair division, where envy-based notions require that no participant be placed at a significant disadvantage relative to others. When such constraints are imposed on optimization problems, they fundamentally alter the feasible solution space and force a direct trade-off between fairness and efficiency.

These perspectives are complementary. They show that fairness can influence algorithm design at different levels: through the objective being optimized, through a preference for balanced solutions, or through explicit constraints that restrict feasibility. Across the problems studied in this thesis, these perspectives appear in different forms: fairness is embedded in the objective through egalitarian measures in Part I, emerges implicitly through balanced performance criteria in Chapter 5, and is enforced explicitly as a constraint in Chapter 7.

1.4.4 Limits That Shape What Is Possible

A further central message of this thesis is that understanding what *cannot* be achieved is just as important as identifying what is possible. In multiple problem settings, formal lower bounds show that certain performance guarantees are fundamentally out of reach, regardless of the algorithmic technique employed. These results are not artifacts of poor algorithm design; rather, they reflect inherent limitations imposed by incomplete information, strategic behavior, or online uncertainty.

Limits That Persist Across Models. One particularly instructive example arises in committee selection. Even in highly structured settings, such as peer selection on a line metric where voters and candidates coincide and preferences are strongly constrained, it is impossible to always select a committee that is optimal under standard measures of quality. The persistence of these limitations in such restricted environments demonstrates that they are not caused by overly general models, but are intrinsic to the use of ordinal information itself. In this sense, additional structure alone is often insufficient to overcome the loss incurred by limited preference information. Similar impossibility phenomena appear in other parts of the thesis. In online and learning-augmented settings, lower bounds show that even with access to predictive information, algorithmic performance cannot exceed certain thresholds.

Limits That Do Not Generalize. At the same time, these limits are not always universal. For instance, in the randomized single-facility location problem, a lower bound that holds on the line metric does not extend to more general Euclidean settings. Understanding precisely when and where such barriers apply is therefore essential for identifying opportunities for improved algorithmic performance.

Viewed together, these negative results play a constructive role. They delineate the boundary between achievable and unattainable guarantees, prevent misguided attempts to optimize beyond inherent limits, and help focus attention on meaningful directions for improvement. Rather than serving as endpoints, lower bounds are used throughout the thesis as tools for understanding how algorithmic design must adapt to the constraints imposed by incomplete information. Across several chapters, they clarify which limitations are fundamental and which arise from specific modeling assumptions.

1.4.5 When Randomness Opens New Doors

Randomization plays a subtle but important role across several problems studied in this thesis. In some settings, deterministic algorithms face inherent limitations: once every decision is fixed in advance, there is only so much performance one can guarantee. Allowing the algorithm to make randomized choices can relax these constraints and, in some cases, lead to significantly better guarantees. From this perspective, randomness can fundamentally change the landscape of what is achievable.

Randomization Is Not Always Easier. The effect of randomization, however, is not uniform across problems. In certain settings, randomization simplifies the design of algorithms and leads to cleaner or stronger guarantees than those attainable deterministically. In other settings, it introduces additional challenges, both in algorithm design and in analysis, and does not necessarily eliminate existing barriers. Understanding when randomness is genuinely beneficial, and when it merely shifts difficulties elsewhere, is therefore an important part of the overall picture.

Ex-Ante and Ex-Post Guarantees. Randomization also raises a methodological distinction in how guarantees are evaluated. Some guarantees hold only in expectation, averaging over the algorithm's random choices, while others hold for every possible outcome of the random process. Balancing these two notions is often delicate: strong expected performance may come at the cost of variability across outcomes, whereas outcome-wise guarantees may limit the gains from randomization. Careful algorithm design is required to benefit from randomness while maintaining meaningful control over performance.

Lower Bounds Under Randomization. From a theoretical standpoint, randomness complicates the study of limitations as well. Proving lower bounds against randomized algorithms is typically more challenging than in the deterministic case, as it requires reasoning about distributions over decisions rather than fixed strategies. As a result, negative results in randomized settings often provide particularly strong evidence of inherent difficulty.

This thesis explores the role of randomization in both strategic and online environments. In the strategic facility location problem, randomization helps overcome barriers faced by deterministic mechanisms, while preserving incentive constraints. In online interval scheduling, randomized algorithms offer new ways to balance robustness, adaptability, and performance. Taken together, these results illustrate that randomness, when used thoughtfully, can expand the space of feasible solutions, but only within the limits imposed by the underlying problem.

1.5 Emerging Directions Motivated by the Thesis

Beyond the individual results presented in each chapter, the thesis highlights broader directions that connect existing research areas in new ways. Rather than proposing entirely new models, these directions extend and combine established frameworks, revealing structural connections that have not been systematically explored.

Metric Distortion in Peer Selection. One such direction concerns the study of metric distortion in committee selection when voters and candidates coincide. Previous work on metric distortion has largely focused on elections with disjoint voter and candidate sets, while related questions in peer selection have often been approached through alternative parameters such as decisiveness. By treating voter–candidate overlap as a defining structural feature and analyzing metric distortion under multiple objective functions, the thesis offers a complementary perspective that broadens the scope of the metric distortion framework.

Fairness as Constraint and Trade-Off. A second, more general direction arises from the interaction between fairness notions and optimization problems. While fairness has been studied extensively in allocation and matching settings, it is typically treated either as a primary objective or as a domain-specific requirement. The thesis highlights a different viewpoint, in which fairness notions, particularly envy-based notions that cannot be captured by efficiency objectives alone, are incorporated as explicit constraints or as dimensions along which trade-offs with efficiency are studied. This perspective goes beyond any single problem and suggests a systematic way of integrating fairness considerations into a wide range of optimization problems, including online and scheduling settings.

Treating fairness as a constraint or as an explicit trade-off changes the nature of the algorithmic questions being asked. Rather than optimizing a single objective, one must understand how much efficiency can be retained while satisfying meaningful fairness requirements, and which compromises are unavoidable.

These directions are not intended as definitive frameworks, but as starting points for further investigation. They illustrate how ideas developed in one area can be adapted and reinterpreted in another, and how such cross-pollination can lead to new questions and refined models. In this sense, the thesis contributes not only concrete technical results, but

also modeling perspectives that may prove useful beyond the specific problems studied here.

1.6 Outline

In this thesis, we study a collection of problems in mechanism design, social choice, and online scheduling, with a particular focus on decision-making under incomplete information. The following outline briefly summarizes the contributions of each chapter.

Part I – Mechanism Design and Social Choice

Chapter 2 – Strategic Facility Location with Predictions: This chapter revisits the strategic facility location problem in the learning-augmented framework, focusing on truthful mechanisms for minimizing the egalitarian social cost. We study both the one-dimensional and Euclidean settings and investigate how randomization and different types of predictions affect achievable performance. For the line metric, we establish tight upper and lower bounds that fully characterize the trade-off between consistency and robustness for truthful randomized mechanisms, even under the strongest possible predictions. In two dimensions, we design a truthful randomized mechanism that leverages predictions about extreme agents to improve consistency while maintaining robustness, and we complement this with new lower bounds that highlight fundamental differences from the one-dimensional case. This chapter is based on joint work with Eric Balkanski and Vasilis Gkatzelis, appeared at NeurIPS 2024.

Chapter 3 – Distortion of Multi-Winner Elections on the Line Metric: This chapter investigates the problem of selecting representative committees when voters and candidates lie on a line metric, a setting that offers both structure and interpretability. We study the distortion of voting rules under additive costs, where a voter's disutility is the sum of distances to all elected committee members. While prior work has primarily focused on q-cost models, we shift attention to this more global notion of representational quality. To this end, we introduce the *Polar Comparison Rule* and analyze its performance under utilitarian additive cost. We show that it achieves a distortion of at most 2.33 for all committee sizes $k > 2$, improving upon the previously best-known upper bound of 3. For $k = 2$ and $k = 3$, we further establish tight bounds of 2.41 and 2.33, respectively. Our analysis also yields lower bounds that depend on the parity of k and illuminates how distortion behaves for both small and large committees. Finally, we extend these results to the egalitarian additive cost. This chapter is based on joint work with Negar Babashah, Hasti Karimi, and Masoud Seddighin, appeared at AAMAS 2025 and SAGT 2025.

Chapter 4 – Metric Distortion in Peer Selection: In this chapter, we initiate the study of metric distortion in the peer selection setting, where voters and candidates coincide and the goal is to elect a committee of size k based solely on agents' ordinal preferences. Even on the line metric, this setting reveals a surprisingly rich landscape: while one might expect the additional structure to substantially simplify the problem, many of the fundamental hardness phenomena known from the general case persist. We

examine four natural objective functions arising from the combination of additive versus q-cost aggregation with utilitarian versus egalitarian social cost. Our results show that, although distortion lower bounds remain strictly larger than one across all objectives, several cases admit improved constants compared to the general setting. For utilitarian cost under additive aggregation, selecting the k middle agents yields distortion between 1 and 2, while for q-cost we obtain positive guarantees for the case $q = k = 2$ despite broader impossibility results. Under egalitarian cost, choosing the extreme agents achieves an optimal distortion of 2 for additive aggregation and for q-cost when $q > k/3$, whereas no bounded distortion is achievable for $q \leq k/3$. Altogether, the chapter demonstrates that peer selection on the line metric retains inherent difficulties yet offers strictly better performance guarantees in the cases where positive results are possible. This chapter is based on joint work with Javier Cembrano, appeared at AAMAS 2026.

Part II – Online Scheduling

Chapter 5 – Online Speed-Scaling with Predictions: This chapter addresses the classical deadline-based online speed-scaling problem through the perspective of learning-augmented algorithm design. Motivated by the increasing importance of energy-aware scheduling in modern data centers, we examine how predictions of future system load, derived for instance from historical traces, can be harnessed to improve energy consumption. While pure machine-learned predictors may perform poorly under significant error, and traditional worst-case online algorithms tend to be overly conservative, the chapter explores how to bridge this gap by designing algorithms that achieve low energy usage when predictions are accurate, remain robust when predictions deviate considerably, and exhibit *smoothness*, meaning that their performance deteriorates gracefully with increasing prediction error. This chapter is based on joint work with Antonios Antoniadis and Peyman Jabbarzade Ganje, appeared at SWAT 2022.

Chapter 6 – Online Interval Scheduling with Predictions: In this chapter, we examine the classical online interval scheduling problem in the irrevocable setting, where each interval must be accepted or rejected upon arrival and the objective is to maximize the total length of non-overlapping intervals. We investigate this problem in a learning-augmented framework, seeking algorithms that meaningfully exploit predictions while retaining strong guarantees in the face of adversarial inputs. The chapter introduces a general framework for combining prediction-based strategies with classical online algorithms, illuminating the fundamental trade-off between consistency under accurate predictions and robustness against misleading ones. These results are complemented by lower bounds demonstrating the tightness of the framework in specific regimes, as well as by randomized algorithms that interpolate smoothly between prediction-guided and robust behavior. This chapter is based on joint work with Antonios Antoniadis, Ali Shahheidar, and Abolfazl Soltani, appeared at AAAI 2026.

Chapter 7 – Interval Scheduling under Approximate Envy-Freeness: The final chapter studies interval scheduling from the perspective of fair allocation. We model machines as agents and intervals as items, and require schedules to satisfy envy-freeness up to one item (EF1) across machines, while benchmarking efficiency against the offline

optimum computed without fairness. In the offline setting, we show that in the unweighted regime there exists a polynomial-time EF1 algorithm that achieves a 2/3 fraction of the unconstrained optimum, and we prove matching lower bounds showing that a loss of 3/2 is unavoidable. We then turn to the online setting, where intervals arrive over time, rejections are irrevocable, and accepted intervals may be revoked. In the unweighted regime, we present the Greedy-Balanced algorithm, which maintains EF1 at all times and achieves a competitive ratio of $2 - 1/m$, and we show that this guarantee is optimal for deterministic algorithms. Together, these results separate the efficiency loss due to fairness constraints from the additional loss caused by online uncertainty. This chapter is based on joint work with Sander Borst and Rohit Vaish, currently under review.

Taken collectively, these chapters highlight the breadth of challenges that arise in decision-making under incomplete information and the diverse techniques required to address them.

PART I

Mechanism Design and Social Choice

CHAPTER 2

Strategic Facility Location with Predictions

Chapter overview. This chapter studies strategic facility location under incomplete information through the lens of learning-augmented algorithm design. It investigates how imperfect predictive information can be incorporated into truthful mechanisms while preserving worst-case performance guarantees. The results are based on joint work with Eric Balkanski and Vasilis Gkatzelis that appeared at NeurIPS 2024.

2.1 Introduction

In the classic facility location problem the goal is to determine the ideal location for a new facility, taking as input the preferences of a group of n agents. Due to the wide variety of applications that this problem captures, it has received a lot of attention from different perspectives. A notable example is the strategic version of this problem which is motivated by the fact that the participating agents may be able to strategically misreport their preferences, leading to a facility location choice that they prefer over the one that would be chosen if they were truthful. In the strategic facility location problem, the goal is to design *truthful* mechanisms, i.e., mechanisms that elicit the agents' true preferences by carefully removing any incentive for the agents to lie (Moulin, 1980; Procaccia and Tennenholtz, 2013). This problem has played a central role in the broader literature on mechanism design without money and a lot of prior work has focused on optimizing the quality of the returned location subject to the truthfulness constraint (see Section 2.1.2 for a brief overview). However, this work has mostly focused on worst-case analysis, often leading to unnecessarily pessimistic impossibility results.

Aiming to overcome the limitations of worst-case analysis, a surge of work during the last few years has focused on the design of algorithms in the *learning-augmented framework* (Mitzenmacher and Vassilvitskii, 2022). According to this framework, the designer is provided with some machine-learned prediction regarding the instance at hand and the goal is to leverage this additional information in the design process. However, crucially, this information is not guaranteed to be accurate and can, in fact, be arbitrarily inaccurate. The goal is to achieve stronger performance guarantees whenever the prediction happens to be accurate (known as the *consistency* guarantee), while at the same time maintaining some worst-case guarantee even if the prediction is arbitrarily inaccurate (known as the *robustness* guarantee).

Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 and C. Xu and Lu, 2022 recently introduced the learning-augmented framework to mechanism design problems involving self-interested strategic agents with private information, and both of these works studied the strategic facility location problem. Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 considered both the egalitarian and the utilitarian social cost objectives and provided tight bounds on the best-feasible robustness and consistency trade-off for truthful mecha-

nisms augmented with a prediction regarding what the optimal facility location would be. However, the achievable trade-offs between robustness and consistency may heavily depend on the specific type of prediction that the mechanism is augmented with: note that the consistency guarantee only binds if the prediction is accurate, so more refined predictions bind on fewer instances (the subset of instances where even the refined prediction is accurate) and could, therefore, enable the design of learning-augmented mechanisms with improved guarantees. This leaves open the question of whether mechanisms equipped with more refined predictions can, indeed, achieve better trade-offs between robustness and consistency. Also, Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 restricted their attention to deterministic mechanisms, leaving open the possibility for truthful randomized mechanisms to achieve even stronger guarantees. In this work we significantly expand our understanding of this problem by studying both the power of randomization and the power of alternative predictions.

2.1.1 Our Contributions and Techniques

We revisit the problem of designing truthful learning-augmented mechanisms for the strategic facility location problem. These mechanisms ask each agent to report their preferred location in some metric space (which is private information) and they are also augmented with some (unreliable) prediction related to this private information. Using the agents' reported locations, along with the prediction, the mechanism chooses a facility location in this metric space, aiming to (approximately) minimize some social cost measure. Our focus in this paper is on the egalitarian social cost, i.e., the maximum distance between an agent's preferred location and the location chosen for the facility. Our goal is to evaluate the performance of mechanisms that can leverage randomization, as well as the power of different types of predictions.

We first consider the well-studied single-dimensional version of the problem, where all agents lie on a line and the mechanism needs to choose a facility location on that line. Prior work on the strategic facility location problem without predictions, showed that for this class of instances no deterministic truthful mechanism can achieve an approximation better than 2 and no randomized truthful[1] mechanism can achieve an approximation better than 1.5; both of these results are shown to be tight (Procaccia and Tennenholtz, 2013). In the learning-augmented framework, Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 showed there exists a truthful deterministic mechanism provided with a prediction regarding the optimal facility location that achieves the best of both worlds: a perfect consistency of 1 and an optimal robustness of 2.

Our main result for the single-dimensional version shows that even if the mechanism is provided with the strongest possible prediction (i.e., a prediction regarding every agent's preferred location), any randomized truthful mechanism that is $(1 + \delta)$-consistent for some $\delta \in [0, 0.5]$ can be no better than $(2 - \delta)$-robust. This implies that the previously proposed deterministic learning-augmented mechanism that is 1-consistent and 2-robust and the randomized non-learning-augmented mechanism that is 1.5-robust (and hence also 1.5-consistent) are both Pareto optimal among all randomized mechanisms, even if they are augmented with the strongest possible predictions. Beyond these two extreme

[1] A truthful randomized mechanism guarantees truthfulness in expectation. See Section 7.2 for more details.

points, this result proves a lower bound for the achievable trade-off between robustness and consistency for any randomized mechanism, even with the strongest predictions. We complement this bound by observing that this trade-off can, in fact, be achieved by a truthful randomized mechanism that chooses between the optimal learning-augmented deterministic mechanism and the optimal non-augmented randomized mechanism, thus verifying that our bound fully characterizes this Pareto frontier.

We then consider the two-dimensional case, for which much less is known about randomized mechanisms. The best-known truthful randomized mechanism is the Centroid, introduced by P. Tang, Yu, and Zhao, 2020, which guarantees a $2-1/n$ approximation. We provide a truthful randomized mechanism that is equipped with a prediction regarding the identities of the most extreme agents (those who would incur the maximum cost in the optimal solution) and, using this prediction, our mechanism achieves 1.67 consistency and 2 robustness. The idea is to use these predictions to select at most three extreme agents, apply the Centroid mechanism to them, and guarantee a 1.67 approximation if the predictions are accurate not only for these three agents but for all other agents as well, utilizing the properties of the Euler line.

An interesting observation is that the lower bound of 1.5 from the single-dimensional case does not extend to two dimensions, so prior work does not imply any lower bound for two dimensions! We prove a lower bound of 1.118 for all truthful randomized mechanisms and then that no deterministic mechanism can simultaneously guarantee better than 2 consistency and better than $1+\sqrt{2}$ robustness, even if it is augmented with the strongest predictions. Similarly, no truthful randomized mechanism can simultaneously guarantee 1 consistency and better than 2 robustness. The former result proves the optimality of a previously introduced 1-consistent and $1+\sqrt{2}$-robust truthful deterministic mechanism provided only with a prediction regarding the optimal facility location (Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022). We also show that the latter is tight.

2.1.2 Related Work

Strategic facility location. Moulin (1980) provided a characterization of deterministic truthful facility location mechanisms on the line and introduced the median mechanism, which returns the median of the agent location profile $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$. The median mechanism is known to be truthful, providing an optimal solution for the Utilitarian Social Cost and achieving a 2-approximation for the Egalitarian Social Cost, as demonstrated by Procaccia and Tennenholtz, 2013. Notably, Procaccia and Tennenholtz, 2013 showed that this 2-approximation factor represents the best performance achievable by any deterministic truthful mechanism. Building on this, Border and Jordan, 1983 extended these results to the Euclidean space by employing median schemes independently in each dimension. Subsequently, Barberà, Gul, and Stacchetti, 1993 further generalized this outcome to any L_1-norms. Other settings explored include general metric spaces (Alon, Feldman, Procaccia, and Tennenholtz, 2010) and d-dimensional Euclidean spaces (S. Goel and Hann-Caruthers, 2020; Meir, 2019; El-Mhamdi, Farhadkhani, Guerraoui, and Hoang, 2023; Walsh, 2020), as well as circles (Alon, Feldman, Procaccia, and Tennenholtz, 2010; Meir, 2019) and trees (Alon, Feldman, Procaccia, and Tennenholtz, 2010; Feldman and Wilf, 2013). Fundamental results in truthful facility location often focus on characterizing the space of truthful mechanisms. For the one-dimensional case, Moulin's characteriza-

tion (Moulin, 1980) reveals that all deterministic truthful mechanisms belong to the "general median mechanisms (GCM)" family. For the two-dimensional case, a similar characterization was provided by H. Peters, Stel, and Storcken, 1993. For a comprehensive review of previous work on this problem, see the survey by H. Chan, Filos-Ratsikas, B. Li, M. Li, and C. Wang, 2021.

Learning-augmented algorithms. Worst-case analysis on its own is often not informative enough, and several alternative measures have been proposed to overcome its limitations (Roughgarden, 2021). Learning-augmented algorithms, or algorithms with predictions (Mitzenmacher and Vassilvitskii, 2022), aim to address these limitations by incorporating predictions into the algorithm's design. Lykouris and Vassilvitskii, 2021 studied the online caching problem and introduced two main metrics, consistency and robustness, to evaluate the performance of these algorithms. These initial results were improved by subsequent works (Antoniadis, Coester, Eliáš, Polak, and Simon, 2023; Chledowski, Polak, Szabucki, and Żołna, 2021; Rohatgi, 2020; A. Wei, 2020). Various online problems have been explored using learning-augmented algorithms, including scheduling problems (Antoniadis, Jabbarzade, and Shahkarami, 2022; Bamas, Maggiori, Rohwedder, and Svensson, 2020; Lattanzi, Lavastida, Moseley, and Vassilvitskii, 2020; Mitzenmacher, 2020; Purohit, Svitkina, and R. Kumar, 2018), online selection and matching problems (Antoniadis, Gouleakis, Kleer, and Kolev, 2020; Dütting, Lattanzi, Leme, and Vassilvitskii, 2021), the online knapsack problem (Im, R. Kumar, Qaem, and Purohit, 2021), online Nash social welfare maximization (Banerjee, Gkatzelis, Gorokh, and Jin, 2022), and several online graph algorithms (Azar, Panigrahi, and Touitou, 2022). The online version of the facility location problem, introduced by Meyerson, 2001, involves points arriving online, requiring us to assign them irrevocably to either an existing facility or open a new facility. The objective is to minimize the distance of each agent to the assigned facility along with the cost of opening the facilities. Almanza, Chierichetti, Lattanzi, Panconesi, and Re, 2021, S. H.-C. Jiang, E. Liu, Lyu, Z. G. Tang, and Y. Zhang, 2022, and Fotakis, Gergatsouli, Gouleakis, and Patris, 2021 studied this problem augmented with predictions regarding the location of the optimal facility for each incoming point. The literature also covers learning-augmented algorithms for classic data structures (Kraska, Beutel, Chi, Dean, and Polyzotis, 2018), Bloom filters (Mitzenmacher, 2018), and a broader framework of online primal-dual algorithms (Bamas, Maggiori, and Svensson, 2020). Berger, Feldman, Gkatzelis, and Tan, 2024 explore the metric distortion problem, which is closely related to the facility location problem for general metric spaces, and introduce a learning-augmented algorithm that achieves the optimal robustness-consistency trade-off. We direct interested readers to a curated and frequently updated list of papers in this area (Lindermayr and Megow, n.d.).

Learning-augmented mechanism design. The framework of learning-augmented mechanism design was first introduced by Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 and C. Xu and Lu, 2022. Agrawal, Balkanski, Gkatzelis, Ou, and Tan, 2022 focused on the strategic facility location problem, proposing mechanisms that leverage predictions to improve performance while maintaining truthfulness. Their work achieves the best possible consistency and robustness trade-off given a prediction of the optimal facility location. They also evaluated the performance of their mechanisms as a function of the

prediction error. More recently, Christodoulou, Sgouritsa, and Vlachos, 2024 provided performance bounds based on an alternative measure of prediction error. Barak, A. Gupta, and Talgam-Cohen, 2024 studied this problem assuming that the predictions are regarding each agent's location, but a small fraction of these predictions may be arbitrarily inaccurate. Meanwhile, Q. Chen, Gravin, and Im, 2024 evaluated non-truthful mechanisms with respect to their price of anarchy. Istrate and Bonchiş, 2022 and J. Fang, Q. Fang, W. Liu, and Nong, 2024 studied the obnoxious facility location version of this problem, aiming to design truthful mechanisms. Apart from the facility location problem, other works in learning-augmented mechanism design have been conducted in various contexts, including strategic scheduling (Balkanski, Gkatzelis, and Tan, 2023; C. Xu and Lu, 2022), auctions (Balkanski, Gkatzelis, and Tan, 2023; Balkanski, Gkatzelis, Tan, and C. Zhu, 2024; Caragiannis and Kalantzis, 2024; Gkatzelis, Schoepflin, and Tan, 2025; Lu, Wan, and J. Zhang, 2024; C. Xu and Lu, 2022), bicriteria mechanism design (Balcan, Prasad, and Sandholm, 2023), graph problems with private input (Colini-Baldeschi, Klumper, Schäfer, and Tsikiridis, 2024), and equilibrium analysis (Gkatzelis, Kollias, Sgouritsa, and Tan, 2022; Istrate, Bonchiş, and Bogdan, 2024). Balkanski, Gkatzelis, Tan, and C. Zhu, 2024 brought together the line of work on learning-augmented mechanism design with the literature on online algorithms with predictions by studying online mechanism design with predictions.

2.2 Preliminaries

In the single facility location problem, there are n strategic agents with a location profile denoted by $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$, where x_i corresponds to the location of agent i. A mechanism $f(\mathbf{x})$ outputs a, potentially randomized, location for the facility. The cost incurred by agent i is the expected Euclidean distance $\mathbb{E}[d(x_i, f(\mathbf{x}))]$ between the facility location $f(\mathbf{x})$ and their location x_i.

Two renowned cost functions considered in this scenario are Egalitarian Social Cost and Utilitarian Social Cost. In this work, the goal is to optimize the Egalitarian Social Cost $C(f, \mathbf{x}) = \mathbb{E}[\max_{x_i \in \mathbf{x}} d(x_i, f(\mathbf{x}))]$, representing the expected maximum cost experienced by a single agent for each possible outcome of $f(\mathbf{x})$. The optimal location for the facility in the two-dimensional Euclidean space corresponds to the center of the smallest circle that encloses all the points, denoted by $o(\mathbf{x})$.

To minimize the social cost, a mechanism needs to ask the agents to report their preferred locations, $\mathbf{x} \in \mathbb{R}^{2n}$, and then use this information to determine the facility location $f(\mathbf{x}) \in \mathbb{R}^2$. However, the preferred location x_i of each agent i is private information, and they can choose to misreport their preferred location if that can reduce their own cost. A mechanism $f : \mathbb{R}^{2n} \to \mathbb{R}^2$ is considered truthful when no individual agent can benefit by misreporting their location, i.e., for all instances $\mathbf{x} \in \mathbb{R}^{2n}$, every agent $i \in [n]$, and every deviation $x_i' \in \mathbb{R}^2$, we have that $d(x_i, f(\mathbf{x})) \leq d(x_i, f(\mathbf{x}_{-i}, x_i'))$, where $\mathbf{x}_{-i} = \langle x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n \rangle$ is the vector of the locations of all agents except agent i.

In the context of randomized mechanisms, we can distinguish between two forms of truthfulness: universally truthful and truthful in expectation. A universally truthful mechanism involves a randomization over deterministic truthful mechanisms, with weights that may depend on the input. On the other hand, a mechanism is considered truthful in

expectation if truth-telling yields the agent the maximum expected value. Specifically, a mechanism $f : \mathbb{R}^{2n} \to \mathbb{R}^2$ is truthful in expectation if for all instances $\mathbf{x} \in \mathbb{R}^{2n}$, every agent $i \in [n]$, and every deviation $x'_i \in \mathbb{R}^2$, we have that $\mathbb{E}[d(x_i, f(\mathbf{x}))] \leq \mathbb{E}[d(x_i, f(\mathbf{x}_{-i}, x'_i))]$.

We focus on mechanisms that are both unanimous and anonymous. A mechanism is unanimous if, when all points $\mathbf{x}$ are located at the same position ($x_i = x_j$ for all $i, j \in [n]$), it places the facility at that specific location, i.e., $f(\mathbf{x}) = x_i$. To ensure bounded robustness, a mechanism must be unanimous, as the optimal cost is zero when the facility is at the same location as all points, whereas placing it elsewhere incurs a positive cost. A mechanism is anonymous if its outcome is independent of the agents' identities, meaning it remains unchanged under any permutation of the agents. Finally, we also assume the mechanism is scale-independent, i.e., if we multiply every coordinate of every agent by the same factor, the coordinate of the chosen facility location are scaled in the same way; this captures the fact that it is only the relative distances that really matter in this problem.

Learning-augmented algorithms encompass a class of algorithms that enhance their decision-making process by integrating pre-computed predictions or forecasts. These predictions, obtained from various sources, like statistical models, serve as inputs without requiring real-time learning from new data. For instance, in the single facility location problem, one might consider predictions $\hat{\mathbf{x}}$ regarding all of the agent's preferred locations, a prediction F^* regarding the optimal facility location, or a prediction regarding the identities of the most extreme agents $\hat{\mathbf{e}}$ (the ones that suffer the maximum cost in the optimal solution), which, as demonstrated in this paper, proves to be quite useful. We denote mechanisms enhanced with predictions in general as $f(\mathbf{x}, *)$. Specifically, for each prediction setting like $\hat{\mathbf{x}}$ or F^*, we denote a mechanism enhanced with $\hat{\mathbf{x}}$ and F^* by $f(\mathbf{x}, \hat{\mathbf{x}})$ and $f(\mathbf{x}, F^*)$, respectively. We define mechanisms enhanced with other kinds of predictions in a similar way.

The effectiveness of learning-augmented mechanisms is evaluated using their consistency and robustness. If $\mathbf{x} \lhd *$ denotes all instances $\mathbf{x}$ for which prediction $*$ is accurate, a mechanism is

- α-*consistent* if it achieves an α-*approximation* when the prediction is correct, i.e.,

$$\max_{\mathbf{x},*:\mathbf{x}\lhd *} \left\{ \frac{C(f(\mathbf{x}, *), \mathbf{x})}{C(o(\mathbf{x}), \mathbf{x})} \right\} \leq \alpha.$$

- β-*robust* if it maintains a β-*approximation* regardless of the quality of the prediction, i.e.,

$$\max_{\mathbf{x},*} \left\{ \frac{C(f(\mathbf{x}, *), \mathbf{x})}{C(o(\mathbf{x}), \mathbf{x})} \right\} \leq \beta.$$

2.3 Results for the Line

In this section, we provide a lower bound regarding the performance of any randomized mechanism for the line, even if it is equipped with the strongest type of prediction, i.e., a prediction $\hat{\mathbf{x}}$ regarding the preferred location of every agent.

Theorem 2.1. *No mechanism for the line that is truthful in expectation and guarantees $1+\delta$ consistency for some $\delta \in [0, 0.5]$ can also guarantee robustness better than $2-\delta$, even if it is provided with full predictions $\hat{\mathbf{x}}$ containing each of the agents' locations.*

The proof consists of two main parts. In Subsection 2.3.1, we show that, for any instance involving two agents, we can without loss of generality restrict our attention to a class of mechanisms that we call OnlyM mechanisms. In Subsection 2.3.2, we then show the desired lower bound for OnlyM mechanisms.

2.3.1 The Reduction to OnlyM Mechanisms

We introduce the following notations specific to this section. Let x_L be the leftmost reported location and x_R be the rightmost reported location on the line. Therefore, we have $x_L \leq x_R$. Let M denote the midpoint of these two extreme points, i.e., $M = (x_L + x_R)/2$, which would also correspond to the optimal facility location. For simplicity, we sometimes write $f(x_1, \ldots, x_n)$, dropping the angle brackets $\langle\rangle$. OnlyM is the class of mechanisms that, whenever they choose a location within the interval (x_L, x_R), then this location is always M.

Definition 2.2 (OnlyM mechanisms)**.** *A mechanism f for the line is an OnlyM mechanism if $P[f(\mathbf{x}) \in (x_L, x_R) \setminus \{M\}] = 0$.*

The main lemma for the proof of Theorem 2.1 is the following reduction that allows to restrict our attention to OnlyM mechanisms.

Lemma 2.3. *For any problem instance involving two agents with reported locations $\mathbf{x} = \langle x_L, x_R\rangle$ on the line, and any randomized truthful in expectation mechanism achieving α consistency and β robustness over this class of instances, there exists a randomized OnlyM mechanism that is truthful in expectation and achieves the same consistency and robustness guarantees.*

The remainder of Section 2.3.1 is devoted to the proof of Lemma 2.3. Consider any randomized mechanism $f(x_L, x_R)$ and let $p_\ell = \mathbb{P}[f(x_L, x_R) \in (x_L, M)]$ and $p_r = \mathbb{P}[f(x_L, x_R) \in (M, x_R)]$ represent the probabilities that this mechanism chooses a facility location in (x_L, M) and (M, x_R), respectively. If $p_\ell = p_r = 0$, the mechanism already satisfies the desired property, and the proof is complete. Otherwise, if $p_\ell > 0$, define $\pi_\ell = \mathbb{E}[f(x_L, x_R) \mid f(x_L, x_R) \in (x_L, M)]$ and, if $p_r > 0$, define $\pi_r = \mathbb{E}[f(x_L, x_R) \mid f(x_L, x_R) \in (M, x_R)]$ as the expected locations returned by the mechanism when restricted to (x_L, M) and (M, x_R), respectively.

Since π_ℓ lies in (x_L, M) and π_r lies in (M, x_R), we can express these two points as convex combinations of x_L, M, and x_R: $\pi_\ell = q_\ell x_L + (1-q_\ell)M$ and $\pi_r = q_r x_R + (1-q_r)M$ for some $q_\ell, q_r \in (0, 1)$. We then modify the original mechanism f to a new OnlyM mechanism f' defined as

$$\mathbb{P}[f'(x_L, x_R) = x] = \begin{cases} \mathbb{P}[f(x_L, x_R) = x] & \text{if } x < x_L \text{ or } x > x_R \\ 0 & \text{if } x \in (x_L, M) \cup (M, x_R) \\ \mathbb{P}[f(x_L, x_R) = x] + q_\ell p_\ell & \text{if } x = x_L \\ \mathbb{P}[f(x_L, x_R) = x] + (1-q_\ell)p_\ell + (1-q_r)p_r & \text{if } x = M \\ \mathbb{P}[f(x_L, x_R) = x] + q_r p_r & \text{if } x = x_R \end{cases}$$

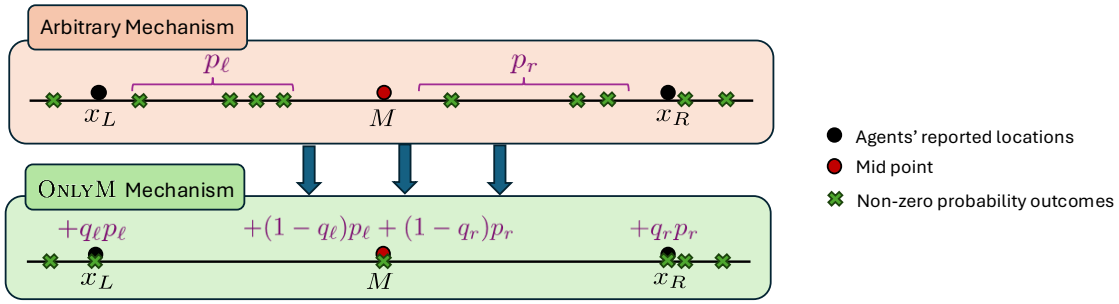


Figure 2.1: The original mechanism $f(.)$ and the new OnlyM mechanism $f'(.)$ in the proof of Lemma 1, where $p_\ell = \mathbb{P}[f(x_L, x_R) \in (x_L, M)]$ and $p_r = \mathbb{P}[f(x_L, x_R) \in (M, x_R)]$, $\pi_\ell = \mathbb{E}[f(x_L, x_R) \mid f(x_L, x_R) \in (x_L, M)]$ and $\pi_r = \mathbb{E}[f(x_L, x_R) \mid f(x_L, x_R) \in (M, x_R)]$, and q_ℓ and q_r are such that $\pi_\ell = q_\ell x_L + (1 - q_\ell)M$ and $\pi_r = q_r x_R + (1 - q_r)M$.

The construction of f' is illustrated in Figure 2.1. To show that mechanism f' achieves the same consistency and robustness guarantees as mechanism f, we show that their expected costs are equal on all instances with two agents.

Lemma 2.4. *For all mechanisms f for the line and instances $\mathbf{x} = \langle x_L, x_R \rangle$ with two agents, the expected costs of f and f' over $\mathbf{x}$ are equal, i.e., $C(f, \mathbf{x}) = C(f', \mathbf{x})$.*

Proof. First, note that

$$\begin{aligned}
&\mathbb{E}\left[\max_{x_i \in \{x_L, x_R\}} d(x_i, f(\mathbf{x})) \mid f(\mathbf{x}) \in (x_L, M)\right] \cdot \mathbb{P}[f(\mathbf{x}) \in (x_L, M)] \\
&= \mathbb{E}\left[d(x_R, f(\mathbf{x})) \mid f(\mathbf{x}) \in (x_L, M)\right] \cdot p_\ell \\
&= d(x_R, \pi_\ell) \cdot p_\ell.
\end{aligned}$$

Similarly, we have

$$\mathbb{E}\left[\max_{x_i \in \{\mathbf{x}\}} d(x_i, f(\mathbf{x})) \mid f(\mathbf{x}) \in (M, x_R)\right] \cdot \mathbb{P}[f(\mathbf{x}) \in (M, x_R)] = d(\pi_r, x_L) \cdot p_r,$$

which implies that

$$\begin{aligned}
C(f', \mathbf{x}) - C(f, \mathbf{x}) &= \mathbb{E}\left[\max_{x_i \in \{x_L, x_R\}} d(x_i, f'(\mathbf{x}))\right] - \mathbb{E}\left[\max_{x_i \in \{x_L, x_R\}} d(x_i, f(\mathbf{x}))\right] \\
&= d(x_R, x_L) \cdot \big(\mathbb{P}[f'(\mathbf{x}) \in \{x_L, x_R\}] - \mathbb{P}[f(\mathbf{x}) \in \{x_L, x_R\}]\big) \\
&\quad + d(x_R, M) \cdot \big(\mathbb{P}[f'(\mathbf{x}) = M] - \mathbb{P}[f(\mathbf{x}) = M]\big) \\
&\quad - d(x_R, \pi_\ell) \cdot p_\ell - d(x_L, \pi_r) \cdot p_r \\
&= d(x_R, x_L) \cdot (q_\ell p_\ell + q_r p_r) + d(x_R, M) \cdot ((1 - q_\ell)p_\ell + (1 - q_r)p_r) \\
&\quad - d(x_R, \pi_\ell) \cdot p_\ell - d(x_L, \pi_r) \cdot p_r \\
&= d(x_R, x_L) \cdot (q_\ell p_\ell + q_r p_r) + d(x_R, M) \cdot ((1 - q_\ell)p_\ell + (1 - q_r)p_r) \\
&\quad - d(x_R, x_L) \cdot q_\ell p_\ell - d(x_R, M) \cdot (1 - q_\ell)p_\ell \\
&\quad - d(x_L, x_R) \cdot q_r p_r - d(x_L, M) \cdot (1 - q_r)p_r \\
&= 0.
\end{aligned}$$

□

Next, to show that mechanism f' is also truthful in expectation, we first show that the costs of the agents do not change between f and f'.

Lemma 2.5. *For all mechanisms f for the line and instances $\mathbf{x} = \langle x_L, x_R \rangle$ with two agents, the cost of the agent at location x_L is identical under both f and f', i.e., $\mathbb{E}\left[d(x_L, f(\mathbf{x}))\right] = \mathbb{E}\left[d(x_L, f'(\mathbf{x}))\right]$. Similarly, we have $\mathbb{E}\left[d(x_R, f(\mathbf{x}))\right] = \mathbb{E}\left[d(x_R, f'(\mathbf{x}))\right]$.*

Proof. By definition of f', we have

$$\begin{aligned}
&\mathbb{E}\left[d(x_L, f(\mathbf{x}))\right] - \mathbb{E}\left[d(x_L, f'(\mathbf{x}))\right] \\
&= d\left(x_L, \mathbb{E}\left[f(\mathbf{x}) \mid f(\mathbf{x}) \in (x_L, M)\right] = \pi_\ell\right) \cdot \mathbb{P}\left[f(\mathbf{x}) \in (x_L, M)\right] \\
&\qquad + d\left(x_L, \mathbb{E}\left[f(\mathbf{x}) \mid f(\mathbf{x}) \in (M, x_R)\right] = \pi_r\right) \cdot \mathbb{P}\left[f(\mathbf{x}) \in (M, x_R)\right] \\
&\qquad + d\left(x_L, M\right) \cdot \left(\mathbb{P}\left[f(\mathbf{x}) = M\right] - \mathbb{P}\left[f'(\mathbf{x}) = M\right]\right) \\
&\qquad + d\left(x_L, x_R\right) \cdot \left(\mathbb{P}\left[f(\mathbf{x}) = x_R\right] - \mathbb{P}\left[f'(\mathbf{x}) = x_R\right]\right) \\
&= d(x_L, q_\ell x_L + (1 - q_\ell)M) \cdot p_\ell + d(x_L, q_r x_R + (1 - q_r)M) \cdot p_r \\
&\qquad - d(x_L, M)((1 - q_\ell)p_\ell + (1 - q_r)p_r) - d(x_L, x_R)(q_r p_r) \\
&= d(x_L, M)(1 - q_\ell)p_\ell + d(x_L, x_R)q_r p_r + d(x_L, M)(1 - q_r)p_r \\
&\qquad - d(x_L, M)(1 - q_\ell)p_\ell - d(x_L, M)(1 - q_r)p_r - d(x_L, x_R)q_r p_r \\
&= 0.
\end{aligned}$$

□

Next, we use the previous lemma to show that mechanism f' preserves the truthful in expectation guarantee of f.

Lemma 2.6. *If a mechanism f for the line is truthful in expectation over instances with two agents, then f' is also truthful in expectation over instances with two agents.*

Proof. Assume that f is truthful in expectation over instances with two agents. Assume that one of the agents deviates and reports a false location in mechanism f'. Due to symmetry, and without loss of generality, assume that the agent located at x_L deviates and reports a false location x'_L. We need to show

$$\mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right] \geq \mathbb{E}\left[d(x_L, f'(x_L, x_R))\right].$$

Since the original mechanism $f(x_L, x_R)$ is truthful in expectation, we have

$$\mathbb{E}\left[d(x_L, f(x'_L, x_R))\right] \geq \mathbb{E}\left[d(x_L, f(x_L, x_R))\right]. \tag{2.1}$$

Combining these two inequalities with Lemma 2.5, it suffices to show

$$\mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right] \geq \mathbb{E}\left[d(x_L, f(x'_L, x_R))\right].$$

To prove the above inequality, we consider two main cases. First, we focus on the scenario where the agent located at x_L deviates to the right, i.e., $x_L < x'_L$. Let $M' = \frac{x'_L + x_R}{2}$. Assume $x'_L \leq x_R$, define $p'_\ell = \mathbb{P}[f(x'_L, x_R) \in (x'_L, M')]$ and $p'_r = \mathbb{P}[f(x'_L, x_R) \in$

$(M', x_R)]$. If $p'_\ell = p'_r = 0$, then $f'(x'_L, x_R) = f(x'_L, x_R)$, and hence $\mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right] = \mathbb{E}\left[d(x_L, f(x'_L, x_R))\right]$. Therefore, the proof is complete in this case.

Next, consider the distribution where the mechanism $f(x'_L, x_R)$ returns a random facility location within the intervals (x'_L, M') if $p'_\ell > 0$, or within (M', x_R) if $p'_r > 0$. The expected locations within these intervals are $\pi'_\ell \in (x'_L, M')$ and $\pi'_r \in (M', x_R)$, which can be expressed as $\pi'_\ell = q'_\ell x'_L + (1 - q'_\ell)M'$ and $\pi'_r = q'_r x_R + (1 - q'_r)M'$, where q'_ℓ and q'_r are the respective convex coefficients.

In the first case, where $x_L < x'_L$, we show that the cost to the left agent is the same across the two mechanisms. The key idea is that the difference between the two mechanisms, in terms of their returned locations, lies within the intervals (x'_L, M') and (M', x_R), both on the right side of x_L. Thus, the agent's cost is fully determined by the expected locations in these intervals. Specifically we have

$$
\begin{aligned}
&\mathbb{E}\left[d(x_L, f(x'_L, x_R))\right] - \mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right] \\
&= d\left(x_L, \mathbb{E}\left[f(x'_L, x_R) \mid f(x'_L, x_R) \in (x'_L, M')\right] = \pi'_\ell\right) \cdot \mathbb{P}\left[f(x'_L, x_R) \in (x'_L, M')\right] \\
&\qquad + d\left(x_L, \mathbb{E}\left[f(x'_L, x_R) \mid f(x'_L, x_R) \in (M', x_R)\right] = \pi'_r\right) \cdot \mathbb{P}\left[f(x'_L, x_R) \in (M', x_R)\right] \\
&\qquad + d\left(x_L, x'_L\right) \cdot \left(\mathbb{P}\left[f(x'_L, x_R) = x'_L\right] - \mathbb{P}\left[f'(x'_L, x_R) = x'_L\right]\right) \\
&\qquad + d\left(x_L, M'\right) \cdot \left(\mathbb{P}\left[f(x'_L, x_R) = M'\right] - \mathbb{P}\left[f'(x'_L, x_R) = M'\right]\right) \\
&\qquad + d\left(x_L, x_R\right) \cdot \left(\mathbb{P}\left[f(x'_L, x_R) = x_R\right] - \mathbb{P}\left[f'(x'_L, x_R) = x_R\right]\right) \\
&= d(x_L, q'_\ell x'_L + (1 - q'_\ell)M') \cdot p'_\ell + d(x_L, q'_r x_R + (1 - q'_r)M') \cdot p'_r \\
&\qquad - d(x_L, x'_L) q'_\ell p'_\ell - d(x_L, M')((1 - q'_\ell)p'_\ell + (1 - q'_r)p'_r) - d(x_L, x_R) q'_r p'_r \\
&= d(x_L, x'_L) q'_\ell p'_\ell + d(x_L, M')(1 - q'_\ell)p'_\ell + d(x_L, x_R) q'_r p'_r + d(x_L, M')(1 - q'_r)p'_r \\
&\qquad - d(x_L, x'_L) q'_\ell p'_\ell - d(x_L, M')((1 - q'_\ell)p'_\ell + (1 - q'_r)p'_r) - d(x_L, x_R) q'_r p'_r \\
&= 0.
\end{aligned}
$$

Note that in the case of $x_R < x'_L$, the same arguments hold with a minor change in notation. Define $p'_\ell = \mathbb{P}[f(x'_L, x_R) \in (M', x'_L)]$ and $p'_r = \mathbb{P}[f(x'_L, x_R) \in (x_R, M')]$. Consider the expected locations π'_ℓ and π'_r within the intervals (M', x'_L) and (x_R, M'), respectively.

Focusing our attention on the more interesting case where $x'_L < x_L$, we aim to show that $\mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right] \geq \mathbb{E}\left[d(x_L, f(x'_L, x_R))\right]$. If we focus on the interval (x'_L, M') and assume $x_L \leq M'$, the expected location of the facility returned by the two mechanisms, $f(x'_L, x_R)$ and $f'(x'_L, x_R)$, conditioned on the facility being in the interval (x'_L, M'), is the same; this is true by construction (our reduction maintains the expected location π'_ℓ between the reported location and the midpoint). The crucial difference between the two mechanisms is that although $f(x'_L, x_R)$ may return any facility location in the (x'_L, M') interval, the mechanism $f'(x'_L, x_R)$ returns the same expected location using only x'_L and M' in its support. We show that the cost to an agent located at any point x_L in the (x'_L, M') interval is weakly higher in $f'(x'_L, x_R)$ than it is in $f(x'_L, x_R)$, thus maintaining the truthfulness guarantee. Intuitively, this is true because the two mechanisms return the same expected location π'_ℓ in that interval, but $f'(x'_L, x_R)$ only returns the extreme points of the interval, which hurt the agent at x_L the most.

More formally, we replace any probability assigned to a point in (x'_L, M') with a convex combination between x'_L and M', and each time we do this, we weakly increase the cost to

an agent who is located at any point in (x'_L, M'). Specifically, for any outcome $y \in (x'_L, M')$ of the mechanism $f(x'_L, x_R)$ and any $x_L \in (x'_L, M')$, if we have $y = w \cdot x'_L + (1-w) \cdot M'$ then $d(x_L, y) < w \cdot d(x_L, x'_L) + (1-w) \cdot d(x_L, M')$.

Since $\pi'_\ell = q'_\ell x'_L + (1-q'_\ell)M'$, at the end of this process, we will end up with $q'_\ell p'_\ell$ probability increase on x'_L, and $(1-q'_\ell)p'_\ell$ probability increase on M'. Therefore, we have

$$\begin{aligned}\mathbb{E}\left[d(x_L, f(x'_L, x_R))\right] &< d(x_L, x'_L)q'_\ell p'_\ell + d(x_L, M')\left((1-q'_\ell)p'_\ell + (1-q'_r)p'_r\right) + d(x_L, x_R)q'_r p'_r \\ &= \mathbb{E}\left[d(x_L, f'(x'_L, x_R))\right].\end{aligned}$$

In the case where $x_L > M'$, similar arguments apply in the interval (M', x_R). □

The proof of Lemma 2.3 then immediately follows from Lemma 2.4 and Lemma 2.6.

2.3.2 The Lower Bound for OnlyM Mechanisms

Equipped with Lemma 2.3, we can now prove Theorem 2.1.

Proof of Theorem 2.1. We start by proving the result for two-agent instances on the line using any OnlyM mechanism. Then, using Lemma 2.3, the result follows for any randomized mechanism.

First, note that if the chosen facility location y is at distance $d(M, y)$ from the point $M = (x_L + x_R)/2$, then its egalitarian social cost is equal to $C(o(\mathbf{x}), \mathbf{x}) + d(M, y)$. To verify this fact, assume without loss of generality that this location is on the left of M and note that its distance from the agent located at x_R is $d(x_R, M) + d(M, y)$. As a result, the expected social cost of a mechanism f with agent locations $\mathbf{x}$ is

$$C(f, \mathbf{x}) = C(o(\mathbf{x}), \mathbf{x}) + \mathbb{E}[d(M, f(\mathbf{x}))]. \tag{2.2}$$

Now, assume that there exists a mechanism that is $(1+\delta)$-consistent and better than $(2-\delta)$-robust. For robustness, this would imply that for every instance $\mathbf{x}$, irrespective of the prediction, the mechanism must guarantee that

$$\frac{C(f, \mathbf{x})}{C(o(\mathbf{x}), \mathbf{x})} < 2 - \delta \;\Rightarrow\; \frac{\mathbb{E}[d(M, f(\mathbf{x}))]}{C(o(\mathbf{x}), \mathbf{x})} < 1 - \delta. \tag{2.3}$$

For this mechanism to be truthful in expectation, it must ensure that no agent has an incentive to misreport their location. Consider the instance $\mathbf{x} = \langle x_L, x_R \rangle$, and note that the agent at x_L has the option to misreport their location as $x'_L = x_L - d(x_L, x_R)$, which would shift the new midpoint M' to x_L in the new instance $\mathbf{x}' = \langle x'_L, x_R \rangle$. This deviation would double the optimal cost, i.e., $C(o(\mathbf{x}'), \mathbf{x}') = 2 \cdot C(o(\mathbf{x}), \mathbf{x})$. Inequality (2.3), then guarantees that the expected cost for this agent after the deviation would be at most $2(1-\delta) \cdot C(o(\mathbf{x}), \mathbf{x})$ since we have

$$\frac{\mathbb{E}[d(M', f(\mathbf{x}'))]}{C(o(\mathbf{x}'), \mathbf{x}')} = \frac{\mathbb{E}[d(x_L, f(\mathbf{x}'))]}{2 \cdot C(o(\mathbf{x}), \mathbf{x})} < 1 - \delta. \tag{2.4}$$

As a result, to ensure that this agent will not misreport, the mechanism needs to ensure that the expected cost of the agent if they report the truth is strictly less than $2(1-\delta) \cdot C(o(\mathbf{x}), \mathbf{x})$. If we let $P(\leq L) = \mathbb{P}[f(\mathbf{x}) \leq x_L]$ denote the probability that the chosen

location is weakly on the left of x_L, and $P(M) = \mathbb{P}[f(\mathbf{x}) = M]$ denote the probability that the chosen location is M, then the expected cost of the agent located at x_L is at least $C(o(\mathbf{x}), \mathbf{x})P(M) + 2C(o(\mathbf{x}), \mathbf{x})(1 - P(M) - P(\leq L))$ because the mechanism is an OnlyM mechanism. This is even if we assume that the only chosen facility locations weakly on the left of x_L (and weakly on the right of x_R, respectively) are exactly on x_L (and exactly on x_R, respectively). Therefore, the mechanism needs to always satisfy

$$\begin{aligned} & C(o(\mathbf{x}), \mathbf{x})(P(M) + 2(1 - P(M) - P(\leq L))) < 2(1 - \delta)C(o(\mathbf{x}), \mathbf{x}) \\ \Rightarrow \quad & P(M) + 2(1 - P(M) - P(\leq L)) < 2(1 - \delta) \\ \Rightarrow \quad & P(\leq L) > \delta - \frac{P(M)}{2}. \end{aligned} \tag{2.5}$$

If we consider the same instance $\mathbf{x} = \langle x_L, x_R \rangle$, and assume that the mechanism is also provided with accurate predictions $\hat{x}_L = x_L$ and $\hat{x}_R = x_R$ regarding the agent locations, to guarantee $1 + \delta$ consistency, Inequality (2.2) implies that

$$\frac{C(f(\mathbf{x}, \hat{\mathbf{x}} = \mathbf{x}), \mathbf{x})}{C(o(\mathbf{x}), \mathbf{x})} \leq 1 + \delta \;\Rightarrow\; \frac{\mathbb{E}[d(M, f(\mathbf{x}, \hat{\mathbf{x}}))]}{C(o(\mathbf{x}), \mathbf{x})} \leq \delta. \tag{2.6}$$

Finally, if we once again consider the same instance with inaccurate predictions $\hat{x}_L = x_L$ and $\hat{x}_R = x_R + d(x_L, x_R)$, we observe that in this case the agent located at x_R would have the option to instead report $x''_R = x_R + d(x_L, x_R) = \hat{x}_R$, and the predictions would then appear to be accurate, forcing the mechanism to satisfy Inequality (2.6) in order to maintain the required consistency bound. Since the true location x_R would now coincide with the middle point M'' of the misreported instance $\mathbf{x}'' = \langle x_L, x''_R \rangle$, this would yield the agent located at x_R who misreported an expected cost of $2\delta \cdot C(o(\mathbf{x}), \mathbf{x})$ since we have

$$\frac{\mathbb{E}[d(M'', f(\mathbf{x}'', \hat{\mathbf{x}}))]}{C(o(\mathbf{x}''), \mathbf{x}'')} = \frac{\mathbb{E}[d(x_R, f(\mathbf{x}, \hat{\mathbf{x}}))]}{2 \cdot C(o(\mathbf{x}), \mathbf{x})} \leq \delta. \tag{2.7}$$

As a result, to ensure that this agent will not misreport, the mechanism needs to ensure that the expected cost of the agent if they report the truth is at most $2\delta \cdot C(o(\mathbf{x}), \mathbf{x})$. If we once again let $P(\leq L)$ denote the probability that the chosen location is weakly on the left of x_L, and $P(M)$ denote the probability that the chosen location is M, then the expected cost of the agent located at x_R is at least $C(o(\mathbf{x}), \mathbf{x})P(M) + 2C(o(\mathbf{x}), \mathbf{x})P(\leq L)$. This is even if we once again assume that the only chosen facility locations weakly on the left of x_L (and weakly on the right of x_R, respectively) are exactly on x_L (and exactly on x_R, respectively). Therefore, the mechanism needs to always satisfy

$$P(M) + 2P(\leq L) \leq 2\delta \quad \Rightarrow \quad P(\leq L) \leq \delta - \frac{P(M)}{2}.$$

However, this contradicts Inequality (2.5), so we conclude that no mechanism can simultaneously guarantee $(1 + \delta)$ consistency and a robustness better than $(2 - \delta)$. □

2.3.3 The Hardness Result for the Line Is Tight

We observe that the lower bound regarding the robustness and consistency trade-off shown in Theorem 2.1 is actually tight. Specifically, it can be achieved by an appropriate randomization between the optimal deterministic learning-augmented mechanism and the optimal non-learning-augmented randomized mechanism.

Proposition 2.7. *For any $\delta \in [0, 0.5]$, there exists a randomized mechanism on the line that is truthful in expectation, $1 + \delta$-consistent, and $2 - \delta$-robust.*

Proof. We show that the bound of the robustness consistency trade-off in Theorem 2.1 is tight. To verify the tightness for $\delta = 0.5$, note that it is possible to achieve a robustness of 1.5 (and therefore also a consistency of 1.5) using the following randomized truthful in expectation mechanism of Procaccia and Tennenholtz (2013):

Mechanism 1 (LRM)**.** *Given an input* $\mathbf{x}$*, the mechanism returns x_L with probability* $1/4$*, x_R with probability* $1/4$*, and M with probability* $1/2$*.*

On the other extreme, to verify the tightness of the theorem for $\delta = 0$, note that it is possible to achieve a robustness of 2 with a consistency of 1 using the following truthful deterministic learning-augmented mechanism of Agrawal, Balkanski, Gkatzelis, Ou, and Tan (2022):

Mechanism 2 (MinMaxP)**.** *Given an input* $\mathbf{x}$ *and a prediction F^* regarding the optimal facility location, the mechanism returns F^* if $F^* \in [x_L, x_R]$, x_L if $F^* < x_L$, and x_R if $F^* > x_R$.*

In fact, for any $\delta \in [0, 0.5]$, we can achieve a robustness of $2 - \delta$ and a consistency of $1 + \delta$ by randomly choosing which one of these two mechanisms to run. Specifically, we can achieve these bounds by running the LRM mechanism with probability 2δ and the MinMaxP mechanism with probability $1 - 2\delta$. Note that since the decision regarding which one of the two truthful mechanisms to run is independent of the agents' reports, the resulting randomized mechanism is truthful as well. □

As a result, the bound of Theorem 2.1 precisely captures the optimal robustness consistency trade-off over all truthful in expectation mechanisms for instances on the line.

2.3.4 Other Prediction Settings

To gain a more complete picture of different prediction settings, we study an alternative strong prediction that is not strictly stronger than the predicted optimal facility location F^*. Specifically, we consider a setting where predictions are available for all pieces of information except for one of the extreme locations. We show that these predictions are not helpful, even without any robustness constraints, to improve the consistency guarantee alone.

Theorem 2.8. *Given a prediction set that provides the identities of all n agents, there exist $n-1$ agents such that, even if we have their exact locations, there is no deterministic truthful mechanism on the line that is better than 2-consistent, and there is no randomized mechanism on the line that is truthful in expectation and better than 1.5-consistent.*

Proof. We first address the deterministic case with two agents whose locations are given by $\mathbf{x} = \langle x_L, x_R \rangle$. Assume we have a prediction $\hat{x}_L$, which predicts the location of the leftmost agent. We will show that even if $\hat{x}_L$ is accurate, there is no truthful mechanism with a consistency better than 2.

We use the same argument as in Theorem 3.2 of Procaccia and Tennenholtz, 2013. Assume, for contradiction, that $f : \mathbb{R}^2 \to \mathbb{R}$ is a truthful mechanism with a consistency less than 2. Consider the location profile $\mathbf{x} = \langle x_L = 0, x_R = 1 \rangle$ and the prediction $\hat{x}_L = 0$. To achieve consistency better than 2, the mechanism needs to choose a facility location $y = f(\mathbf{x})$ such that $y \in (0, 1)$.

Now consider an alternative location profile $\mathbf{x}' = \langle x'_L = 0, x'_R = y \rangle$ with the prediction $\hat{x}_L = 0$. The optimal facility location for this profile is at $y/2$, yielding a maximum cost of $y/2$. To maintain consistency better than 2, the mechanism should place the facility within the interval $(0, y)$. However, if this were to occur, the rightmost agent would have an incentive to report a location of 1 instead of y, as this lie would place the facility exactly at y, rather than within $(0, y)$. This contradicts the truthfulness of the mechanism.

This argument extends to cases with arbitrary n by situating all other agents at 0 in both profiles $\mathbf{x}$ and $\mathbf{x}'$, with accurate predictions for them at 0. Similar reasoning holds true.

For the randomized setting, we use the idea from the proof of Theorem 3.4 in Procaccia and Tennenholtz, 2013. We first focus on the case with two agents and then extend the result to any number of agents. Let f be a randomized truthful in expectation mechanism. Consider the location profile $\mathbf{x} = \langle x_L = 0, x_R = 1 \rangle$. We have $f(\mathbf{x}) = \mathcal{P}$, where $\mathcal{P}$ is a probability distribution over $\mathbb{R}$. There exists $x_i \in \mathbf{x}$ such that $\mathbb{E}[d(x_i, \mathcal{P})] \geq 1/2$. If this agent's location prediction is unavailable, we can prove that consistency better than 1.5 cannot be achieved.

Assume $\mathbb{E}[d(x_R, \mathcal{P})] \geq 1/2$ and that we have an accurate prediction $\hat{x}_L = 0$ for the leftmost agent x_L. Consider the profile $\mathbf{x}' = \langle x'_L = 0, x'_R = 2 \rangle$, with an accurate prediction $\hat{x}_L = 0$. For truthfulness, the expected distance from the location 1 should still be $1/2$, otherwise the rightmost agent would lie in profile $\mathbf{x}$. We know that if $\mathbb{E}[d(M, \mathcal{P})] = \Delta$, then the expected maximum cost is $\Delta + 1$. With $\Delta \geq 1/2$, the expected cost is at least 1.5, whereas the optimal cost is 1, resulting in a consistency of at least 1.5.

This argument extends to arbitrary n by situating all other agents at $1/2$ in both profiles $\mathbf{x}$ and $\mathbf{x}'$, with accurate predictions for them at $1/2$. Similar reasoning holds true. □

2.4 Results for the Plane

We now consider instances in the Euclidean plane, with a location profile $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where $x_i \in \mathbb{R}^2$ for each agent i.

2.4.1 Impossibility Results

Before considering the robustness and consistency guarantees achievable by randomized mechanisms augmented with different types of predictions, we start off by reducing the gap on what is known for mechanisms without predictions.

Since the problem of designing good facility location mechanisms in two dimensions is "harder" than the one-dimensional case, it may seem counterintuitive at first, but the lower bound that prior work proved for all one-dimensional instances does not extend to two dimensions. The reason is that the design space for two-dimensional mechanisms is much richer, and the known lower bounds apply only to the more restricted class

of one-dimensional mechanisms. For example, consider a simple instance with just two agents in two dimensions. For this two-dimensional instance, the mechanism can return a location for the facility that is not on the line containing the two agents' locations, which it, of course, cannot do in one dimension.

Note that, in terms of social cost alone, returning such a location is always Pareto-dominated by returning an appropriate location on the line (e.g., its projection onto the line). However, returning a location that is not on this line also allows the designer to affect the incentives of the participating agents in new and non-trivial ways that are impossible in the one-dimensional case. Specifically, returning points that are not on the line allows us to induce previously impossible cost vectors. For a simple example, in a one-dimensional instance involving two agents at distance 1 from each other, the only facility location that yields the same cost to both agents is the midpoint between them, leading to a cost vector of $(0.5, 0.5)$. However, in two dimensions, we can induce a cost vector of (c, c) for any $c \geq 0.5$ by simply returning the appropriate point on the interval's perpendicular bisector.

This additional flexibility could potentially provide the designer with novel ways to ensure that the agents do not misreport their locations; for example, the classic technique of "money burning," used to achieve incentive compatibility without monetary payments, heavily depends on the designer's ability to penalize and reward agents based on their reports. Although we conjecture that truthful mechanisms are always better off returning a facility on the line containing the agents' locations in this instance, proving such a result appears non-trivial. Due to this additional flexibility in two dimensions, proving inapproximability results for this broader class of mechanisms is a more challenging problem and seems to require new techniques and insights.

Theorem 2.9. *Any randomized mechanism that is truthful in expectation in the Euclidean plane has an approximation ratio of at least* 1.118*.*

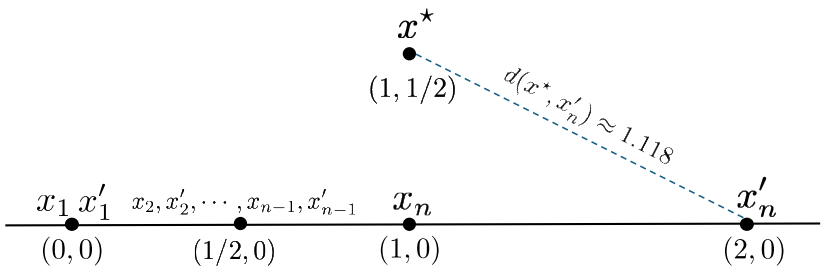
Proof. Let $f(\mathbf{x}) = \mathcal{P}$ be a randomized truthful in expectation mechanism, where $\mathcal{P}$ is a probability distribution over $\mathbb{R}^2$. Consider the location profile $\mathbf{x} = \langle x_1 = (0,0), x_2 = \cdots = x_{n-1} = (1/2, 0), x_n = (1,0)\rangle$. There exists an x_i (either x_1 or x_n) such that $d(x_i, f(\mathbf{x})) = \mathbb{E}_{y\sim\mathcal{P}} d(x_i, y) \geq 1/2$. Without loss of generality, assume that $d(x_n, f(\mathbf{x})) \geq 1/2$. Now consider the location profile $\mathbf{x}' = \langle x'_1 = (0,0), x'_2 = \cdots = x'_{n-1} = (1/2, 0), x'_n = (2,0)\rangle$. To maintain truthfulness, we must have $d(x_n = (1,0), f(\mathbf{x}')) \geq 1/2$, preventing the agent at location $x_n = (1,0)$ in the profile $\mathbf{x}$ from having an incentive to lie about being at location $x'_n = (2,0)$.

Extending the result of Theorem 3.4 in Procaccia and Tennenholtz, 2013 from the line to the Euclidean metric, if we have $d(o(\mathbf{x}), f(\mathbf{x})) \geq \Delta$, then the expected maximum cost is at least

$$\sqrt{\Delta^2 + C(o(\mathbf{x}), \mathbf{x})^2}.$$

Therefore, the expected cost of $\mathbf{x}'$ is at least $\sqrt{\frac{1}{4} + \left(\frac{2}{2}\right)^2}$, resulting in a $\sqrt{1.25} \approx 1.118$ approximation, as the optimum cost is $\frac{d(x'_1, x'_n)}{2} = 1$. □

Our following two results provide impossibility results for both deterministic and randomized mechanisms augmented with perfect predictions. For deterministic mechanisms, Agrawal, Balkanski, Gkatzelis, Ou, and Tan (2022) showed that there is a mechanism that,



.

Figure 2.2: Instances $\mathbf{x} = \langle x_1 = (0,0), x_2 = \cdots = x_{n-1} = (1/2,0), x_n = (1,0)\rangle$, and $\mathbf{x}' = \langle x'_1 = (0,0), x'_2 = \cdots = x'_{n-1} = (1/2,0), x'_n = (2,0)\rangle$ in the proof of Theorem 2. It is assumed that $d(x_n, f(\mathbf{x})) \geq 1/2$ and $x^\star \in \arg\min_{y:d(x_n,y)\geq 1/2} C(y, \mathbf{x})$

given a prediction about the optimal facility location, is 1-consistent and $(1+\sqrt{2})$-robust. Our impossibility result for deterministic mechanisms shows that stronger predictions do not help: even with predictions about the location of each agent, there is no deterministic mechanism that achieves a robustness better than $1+\sqrt{2}$ and a consistency that improves over the best approximation achievable without predictions.

Theorem 2.10. *For any $\delta > 0$, there is no deterministic truthful mechanism that is $(2-\delta)$-consistent and $(1+\sqrt{2}-\delta)$-robust, even if it is provided with full predictions $\hat{\mathbf{x}}$ (the location of each agent).*

The next result shows that there is no hope of achieving the best of both worlds in the randomized setting; to obtain 1 consistency, we would need to sacrifice robustness.

Theorem 2.11. *For any $\delta > 0$, there is no randomized mechanism that is truthful in expectation, 1-consistent, and $(2-\delta)$-robust, even for two-agent instances and even if it is provided with full predictions $\hat{\mathbf{x}}$ (the location of each agent). This is tight, i.e., there exists a randomized mechanism that is truthful in expectation, 1-consistent, and 2-robust for two-agent instances.*

2.4.2 Positive Results Using Extreme ID Prediction

We now turn to positive results, and provide a learning-augmented randomized mechanism that is provided with a new type of prediction: the prediction does not provide us with any actual location, but instead only provides us with the identities $\hat{\mathbf{e}} = \langle e_1, \cdots, e_k\rangle$ of the k agents who would suffer the maximum cost in the optimal solution (we refer to them as "extreme agents"), i.e., $\{e_1, \ldots, e_k\} = \arg\max_{e_i \in [n]} d(x_{e_i}, o(\mathbf{x}))$. Note that the smallest circle that encloses all the points is the *circumcircle* of the locations of the extreme agents. Therefore, the center of this circle is $o(\mathbf{x})$, the optimum solution.

We propose a mechanism leveraging predictions derived from IDs of extreme agents. The main idea is to return the centroid of the extreme agents, denoted by $\mathcal{G}$ with a probability of half, and each of the extreme points with a probability of $1/2k$ to prevent misreporting incentives.

P. Tang, Yu, and Zhao (2020) run this mechanism over all agents (not only the extreme ones) and achieve a $2-1/n$ approximation. We improve the approximation factor (in case

Input : Location profile $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$, Predictions $\hat{\mathbf{e}} = \langle e_1, \cdots, e_k \rangle$
Output : Probability distribution $\mathcal{P}$ on location of the facility

With probability 1/2:
return the centroid $\mathcal{G} = \frac{x_{e_1} + \cdots + x_{e_k}}{k}$
With probability $1/2k$:
return each point $x_{e_1}, \cdots, x_{e_k}$

Algorithm 2.1: Centroid Mechanism on Extreme Agents

of having good predictions) to $2 - 1/3 \approx 1.67$ by running this mechanism only on extreme agents. We use their ideas to maintain truthfulness and we use new techniques to show the approximation guarantees for an arbitrarily number of agents. Moreover, as long as any returned location falls within the minimum enclosing circle of the agents predicted to be extreme (which is the case for Mechanism 2.1), this ensures an approximation factor of 2 in case of having bad predictions.

First, we argue that $k \geq 2$, meaning that there are at least two extreme agents on the minimum enclosing circle. Otherwise, one can find a smaller circle containing all the points. As a warm-up, we first consider the $k = 2$ case and propose a randomized mechanism that is truthful in expectation and achieves 1.5 consistency and 2 robustness for any number of agents.

Theorem 2.12. *Assume there are only two extreme agents, i.e., $k = 2$. Then, given predictions $\hat{\mathbf{e}} = \langle e_1, e_2 \rangle$ that provides the IDs of the only two extreme agents, there exists a randomized mechanism that is truthful in expectation and achieves* 1.5 *consistency and* 2 *robustness for any number of agents.*

Proof. If we only have two agents x_{e_1} and x_{e_2} located on the minimum enclosing circle, $x_{e_1}x_{e_2}$ represent a diameter of this circle; otherwise, we can find a smaller circle containing all the points. Therefore, the middle point of these two locations, $\frac{x_{e_1} + x_{e_2}}{2}$, is the optimum location of the facility. Mechanism 2.1 runs LRM mechanism on the diameter of the minimum enclosing circle, meaning that it returns the optimum location with a probability of 1/2 and each of the reported locations of the extreme agents with a probability of 1/4, resulting in a 1.5 consistency.

In terms of truthfulness, since no other agent besides x_{e_1} and x_{e_2} impacts the mechanism, it suffices to show that they do not have an incentive to lie. Agents x_{e_1} and x_{e_2} lack such incentive for reasons similar to those in Mechanism 1. Additionally, as any returned location falls within the minimum enclosing circle, it ensures a robustness factor of 2. □

We then show that it is sufficient to only consider the case where we have exactly three extreme agents on the minimum enclosing circle, meaning $k = 3$. If we have more than three agents located on the minimum enclosing circle, a continuous perturbation of the points makes the probability of having at least four points lying on a circle infinitesimally small (Berg, Cheong, Kreveld, and Overmars, 2008). For $k = 3$, we use the properties of the Euler line to prove the 1.67 consistency.

Theorem 2.13. *Given a prediction set that provides the IDs of the extreme agents, there exists a randomized mechanism that is truthful in expectation and achieves* 1.67 *consistency and* 2 *robustness for any number of agents.*

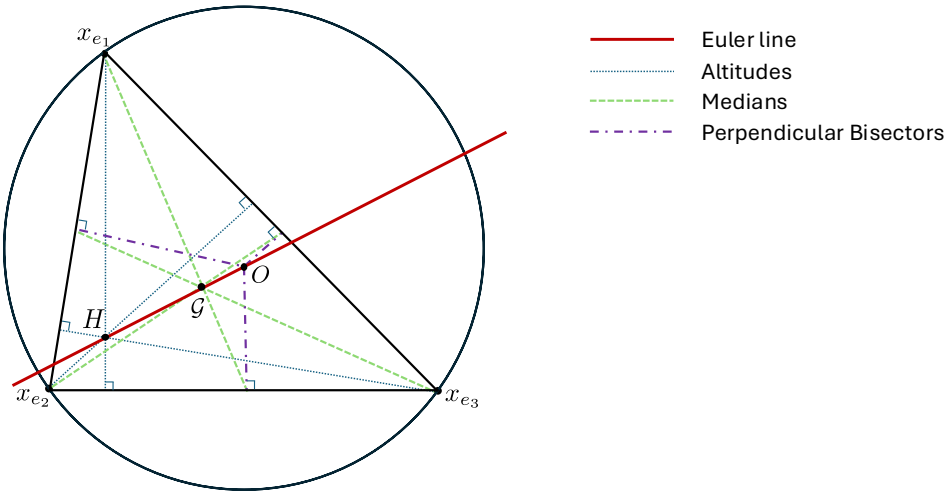


Figure 2.3: $\mathcal{G} = (x_{e_1}+x_{e_2}+x_{e_3})/3$ is the centroid of $x_{e_1}, \ldots, x_{e_3}$, which is the intersection of the three medians. H is the orthocenter, which is the intersection of the three altitudes, and O is the center of the circumcircle, which is the intersection of the three perpendicular bisectors. In any triangle, the circumcenter (O), the centroid $(\mathcal{G})$, and the orthocenter (H) are collinear, forming the Euler line. Moreover, $d(O, \mathcal{G}) = \frac{1}{2}d(\mathcal{G}, H)$.

Proof. Let us first consider instances with only three extreme agents. Mechanism 2.1 is truthful since other agents apart from $x_{e_1}, x_{e_2}, x_{e_k}$ cannot influence the result and Mechanism 2.1 is equivalent to the mechanism of P. Tang, Yu, and Zhao (2020) over agents $e_1, \ldots, e_k$ and since this mechanism is truthful in expectation, agents $e_1, \ldots, e_k$ cannot benefit from misreporting their locations. Since the mechanism returns the reported locations, rather than their predictions, and the centroid is guaranteed to be inside the circumcircle, we can conclude that the mechanism has a robustness factor of 2. The technical aspect of this proof involves establishing the consistency guarantee by considering the Euler line.

Euler line. Given three arbitrary points $x_1, \ldots, x_3$, their circumcircle is the smallest circle that encloses the three points, their centroid is $(x_1+x_2+x_3)/3$, and their orthocenter is the point where the three altitudes (the perpendicular line segments from a vertex to the line that contains the opposite side) intersect. In any triangle, the center of the circumcircle (O), the centroid $(\mathcal{G})$, and the orthocenter (H) are collinear, forming the Euler line. One property of the Euler line is that $\mathcal{G}$ is positioned midway between O and H. Additionally, $d(O, \mathcal{G}) = d(\mathcal{G}, H)/2$, implying $d(O, \mathcal{G}) = d(O, H)/3$.

Let R denote the radius of the circumcircle of $x_{e_1}, x_{e_2}, x_{e_3}$ and assume without loss of generality that the optimum cost is $R = 1$. For any $j \in [n]$, we have

$$d(x_j, \mathcal{G}) \leq d(x_j, O) + d(O, \mathcal{G}) \leq R + d(O, \mathcal{G})$$

where the first inequality is by triangle inequality and the second is because all the points are within the circumcircle of the locations of the extreme agents when the predictions

are correct. Since H lies within the circumcircle, we infer $d(O, H) \leq R$, conclusively showing that $d(O, \mathcal{G}) \leq R/3$. Since R is the optimum cost and any agent can reach O by paying that cost, the cost of returning $\mathcal{G}$ for any agent is upper bounded by $R + d(O, \mathcal{G})$, which is at most $1.34R$. Consequently, the approximation of Mechanism 2.1 in case of having accurate predictions is less or equal than

$$\frac{1}{2k}\frac{1}{R}\sum_{i=1}^{k}\max_{j\in[n]} d(x_{e_i}, x_j) + \frac{1}{2}\frac{1}{R}\max_{j\in[n]} d(x_j, \mathcal{G}) \leq 1 + \frac{1}{2}\frac{R + d(O, \mathcal{G})}{R} \leq 1 + \frac{1}{2}\left(1 + \frac{1}{3}\right) \approx 1.67.$$

where the first inequality is since $d(x_{e_i}, x_j) \leq d(x_{e_i}, O) + d(O, x_j) \leq 2R$ when the predictions are correct. Next, we show how we can modify the mechanism to achieve the same approximation guarantees as having three extreme agents while maintaining truthfulness.

As mentioned previously, the key concept involves perturbing the instance before requesting predictions to have at most three extreme agents. Given any instance $\mathbf{x} = \langle x_1, \cdots, x_n\rangle$, define a perturbed instance $\tilde{\mathbf{x}} = \langle \tilde{x}_1, \cdots, \tilde{x}_n\rangle$, which is the result of a continuous perturbation of $\mathbf{x}$. Consider the set of predictions for the IDs of extreme agents in $\tilde{\mathbf{x}}$. Although the extreme agents of the perturbed instance $\tilde{\mathbf{x}}$ might differ from those of the original instance $\mathbf{x}$, their costs remain very close to the maximum cost in case of having good predictions. Therefore, the circumcircle of them is a good representation of the minimum enclosing circle and results in the same approximation guarantees.

Since we run the mechanism on the main instance $\mathbf{x}$, truthfulness holds as before. Note that since we ask for the ID of extreme agents of the perturbed instance $\tilde{\mathbf{x}}$, we have consistency guarantees if predictions are accurate based on the perturbed instance $\tilde{\mathbf{x}}$, and in any case we can ensure the robustness factor of 2. □

2.5 Conclusion

The problem of designing randomized facility location mechanisms in the Euclidean space is very natural and several open questions remain, even without predictions. While numerous studies have addressed this problem in restricted spaces, there remains a gap regarding between the best possible approximation guarantee in $(1.118, 2 - 1/n)$. The possibility of a truthful in expectation mechanism achieving better than a $2 - 1/n$ approximation is intriguing. Obtaining a stronger lower bound than 1.118 would also be very interesting.

Augmenting these mechanisms with predictions introduces a new dimension to the problem. While we have explored various prediction settings and provided both positive and negative results, many of the current findings in Euclidean space lack tightness. Finding tight results for different types of predictions would enable meaningful comparisons and help identify which prediction strategy is most effective in different scenarios.

CHAPTER 3

Distortion of Multi-Winner Elections on the Line Metric

Chapter overview. This chapter studies multi-winner elections under limited preference information, focusing on the selection of representative committees in metric spaces. It analyzes how much efficiency can be lost when decisions must be based solely on ordinal rankings, and how this loss depends on the committee size and the underlying objective. This chapter is based on joint work with Negar Babashah, Hasti Karimi, and Masoud Seddighin, which appeared at AAMAS 2025 and SAGT 2025.

3.1 Introduction

Imagine a city council election where residents must select a committee of three representatives from nine candidates. Voters have cardinal costs for each candidate but submit ordinal rankings based on these costs. The problem is how to use these rankings to form a reasonable committee.

To make sense of the above scenario, we need to address two key questions. First, what defines a "reasonable" committee? Second, given the limitations of ordinal rankings, can we achieve—or approximately achieve—this objective without access to cardinal values?

What counts as a "reasonable" outcome depends on what we expect from the committee and how we balance efficiency and fairness. To explore this, let us set aside the limitations of ordinal rankings for now and look at the example in Figure 3.1. The goal is to select a committee of three candidates, where each voter's cost for a candidate is simply their distance from that candidate. If we aim to minimize the *utilitarian social cost*—that is, the total cost across all voters for all committee members—then choosing b_1, b_2, b_3 gives the best result. But this outcome might seem unfair. An alternative is to select a_1, b_1, c_1, which increases the total cost slightly but ensures that each voter's favorite candidate is included in the committee.

Several well-studied criteria for multi-winner voting rules capture these trade-offs by specifying different objectives based on individual voter costs. These include minimizing the distance from each voter to their nearest committee member (X. Chen, M. Li, and C. Wang, 2020; Y. Cheng, Z. Jiang, Munagala, and K. Wang, 2020), the distance to their qth nearest member (Caragiannis, Nath, Procaccia, and Shah, 2017), or the utilitarian social cost of the committee (A. Goel, Hulett, and Krishnaswamy, 2018). In this paper, we focus on two classic notions from welfare economics: the utilitarian social cost and the egalitarian social cost, defined as the maximum total cost any individual voter incurs for the entire committee.

The second challenge is the lack of cardinal information. If cardinal values were available, finding the optimal outcome under any of the above criteria would be straight-

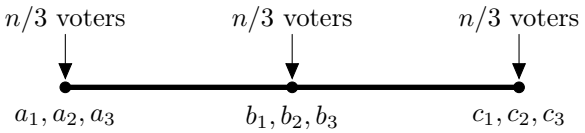


Figure 3.1: An instance with n voters and 9 candidates. The optimal committee of size $k = 3$ depends on how voter cost is defined in terms of the committee members.

forward. However, assuming access to such detailed information is often unrealistic. Voters may find it difficult to precisely quantify their preferences, and requiring cardinal inputs may lead to inconsistencies, reluctance, or increased costs in data collection. For these and other reasons, ordinal rankings are preferred, despite offering less detailed information about preferences.

The efficiency gap due to the lack of cardinal information is captured by the term *distortion* (Procaccia and Rosenschein, 2006). In single-winner elections, the distortion of a voting rule f is defined as the worst-case ratio (across all instances) between the social cost of the candidate selected by f and that of the optimal candidate, which is based on hidden cardinal values.[1]

Over the past decade, distortion has been the subject of extensive study. In general, without any assumptions on voter costs, there exist instances where distortion can be large. As a result, much of the literature has focused on introducing structure or constraints on the cost functions to ensure bounded distortion. Among these, the metric framework has received particular attention. This setting is not only mathematically well-behaved—leading to many elegant and positive results—but also captures a wide range of real-world scenarios, such as spatial models of voting, recommendation systems, and facility location.

For single-winner elections, metric distortion is very well studied. The lower bound for any deterministic rule is 3 (Anshelevich, Bhardwaj, Elkind, Postl, and Skowron, 2018), and rules such as Plurality Veto (Kizilkaya and Kempe, 2022) and Plurality Matching (Gkatzelis, Halpern, and Shah, 2020) achieve this bound. Tight bounds have also been established for many other voting rules (Anshelevich, Bhardwaj, Elkind, Postl, and Skowron, 2018).

In this paper, we examine the distortion of selecting a committee of k candidates under the social additive cost objective. The additive cost objective is particularly relevant when all committee members influence the outcome of the committee equally. For example, in a conference, if the reviewers' combined ratings determine the final score of a paper, selecting reviewers whose expertise aligns closely with the paper minimizes the total distance between the paper's field and the reviewers' expertise. Note that committee selection with an additive cost objective is equivalent to a randomized voting rule that applies a uniform distribution over the selected candidates with a fixed support size. Thus, the outcome of this rule can also be interpreted as a randomized rule.

In recent years, several studies have focused on distortion in multi-winner elections. For the utilitarian additive cost objective in the metric framework, it has been shown by A. Goel, Hulett, and Krishnaswamy (2018) that repeatedly applying any single-winner

[1]We refer to Section 7.2 for a formal definition of distortion.

rule with distortion δ results in a distortion of at most δ. For instance, applying the Plurality Veto rule (Kizilkaya and Kempe, 2022) k times yields a committee of size k with distortion at most 3. Additionally, for the q-cost objective, Caragiannis, Shah, and Voudouris (2022) show that for $q \leq k/3$, distortion is unbounded; for $k/3 < q \leq k/2$, it is tightly bounded by n; and for $q \geq k/2$, it is tightly bounded by 3. However, for the case $q \geq k/2$, they provide a voting rule with distortion 3 and exponential running time, and another rule with distortion 9 and polynomial running time. Kizilkaya and Kempe (2022) resolve this gap by providing a polynomial-time voting rule with distortion 3 for $q \geq k/2$ using Plurality Veto.

We focus on the setting where both voters and candidates are positioned on a line metric. The line metric is an important special case of a metric space, which has been widely studied (Black, 1948; Fotakis and Gourvès, 2022; Fotakis, Gourvès, and Monnot, 2016; Ghodsi, Latifian, and Seddighin, 2019; Miyagawa, 2001; Moulin, 1980; Voudouris, 2023). For example, the line metric can model political preferences along a spectrum ranging from liberal to conservative. In this setting, we improve the classic distortion bound of 3 for the utilitarian additive cost to approximately 2.33.

3.1.1 Our Contributions and Techniques

The problem of selecting a committee of k candidates based on voters' ordinal preferences is both classical and practically significant. However, choosing more than one candidate introduces new challenges in minimizing the social cost. For minimizing the utilitarian additive cost, choosing consecutive candidates leads to a committee with lower distortion compared to other configurations. Although the median voter seems to be a good representative—and indeed, if selecting a single candidate, the nearest candidate to either the left or right of the median voter would be optimal—returning the entire committee based solely on the preference list of the median voter cannot achieve a distortion better than 3. This is because the median voter might prefer all candidates on one side of their location to all candidates on the other side. Therefore, it is important to choose candidates from both sides of the median voter.

To leverage this insight, we design a new voting rule, the *Polar Comparison Rule*, which prioritizes committees that include candidates from both sides of the median voter, thereby increasing the likelihood of selecting better candidates. We first introduce this voting rule for selecting a committee of size two. Specifically, the rule selects the top-ranked candidate of the median voter and then compares the two closest candidates to the median voter on opposite sides of each other that have not yet been picked. Based on the ratio of voters who prefer one candidate to the other, the rule selects the candidate that is more preferred, while introducing a bias toward the candidate on the opposite side of the previously chosen candidate. We then extend this voting rule to select a committee of size three and generalize it to any committee size by iteratively applying the rules for $k = 3$ and $k = 2$.

The *Polar Comparison Rule* achieves upper bounds of $1 + \sqrt{2} \approx 2.41$ and $7/3 \approx 2.33$ distortion for $k = 2$ and $k = 3$, respectively, and we show that these bounds are tight. We further extend these results by showing how different voting rules can be effectively combined to improve the distortion bounds for various committee sizes. Specifically, we generalize our rule to maintain a distortion of $7/3$ for committee sizes divisible by three.

	$k=2$	$k>2$		
Upper Bound	2.41 [T. 3.14]	$2.33 \quad (k \overset{3}{\equiv} 0)$ $2.33 + \frac{4(\sqrt{2}-4/3)}{k} \quad (k \overset{3}{\equiv} 1)$ [T. 3.18] $2.33 + \frac{2(\sqrt{2}-4/3)}{k} \quad (k \overset{3}{\equiv} 2)$		
	$k=2$	$2<k<\frac{m}{2}$		$k \geq \frac{m}{2}$
Lower Bound	2.41 [T. 3.17]	$1+\sqrt{1+\frac{2}{k}} \quad (k \overset{2}{\equiv} 0)$ [T. 3.21] $2+\frac{1}{k} \quad (k \overset{2}{\equiv} 1)$		$1+\frac{m-k}{3k-m}$ [T. 3.22]

Table 3.1: Summary of bounds for the *Polar Comparison Rule*. Note that for $k=3$, the bounds match 2.33. Moreover, an improved upper bound of 2.41 for $k=4$ is derived in Theorem 3.18.

For committee sizes where $k \equiv 1 \pmod 3$, the distortion becomes $7/3 + 4(\sqrt{2} - 4/3)/k$, and for committee sizes where $k \equiv 2 \pmod 3$, the distortion becomes $7/3+2(\sqrt{2}-4/3)/k$.

Additionally, we complement these results with lower bounds. For each lower bound, we provide two instances where voters share the same preference profile and demonstrate a constraint that any voting rule must satisfy. We establish a lower bound of $2 + 1/k$ for odd values of k and $1 + \sqrt{1 + 2/k}$ for even values of k, when $k < m/2$. Furthermore, for larger committee sizes where $k \geq m/2$, we derive a lower bound of $1 + (m-k)/(3k-m)$.

To establish our results, we rely on several techniques that provide valuable insights beyond the specific context of this problem. One useful technique involves determining the order of candidates on the line metric after removing all candidates that are Pareto-dominated.

Another key observation arises from analyzing how voter movements affect distortion. Specifically, we note that in any instance, moving a single voter to either the right or the left results in a new instance with worse distortion. While this observation does not directly aid our proofs, it prompts us to explore movements that yield instances with guaranteed properties relative to the distortion of the initial instance. We show that there exists a specific point on the line such that moving all voters toward this point results in a new instance where the distortion does not significantly decrease.

To apply this technique effectively, we define certain obstacles that restrict how far voters can move toward this point. Each voter moves until encountering an obstacle, at which point they stop. This approach allows us to precisely determine the structure of the resulting instance. Importantly, the placement of these obstacles depends on the voting rule being analyzed. For our *Polar Comparison Rule*, we position these obstacles at the midpoint between the two candidates being compared. This ensures that the

resulting instance aligns with the properties of the rule and facilitates accurate analysis of distortion.

Moreover, we study the egalitarian additive cost and prove that any committee selecting candidates between the two extreme voters achieves a distortion of 2. We first show that the maximum cost is incurred by either the leftmost or rightmost voter and then bound both the cost of our selected committee and the optimal one in terms of the distance between these two extreme voters.

Organization of the Paper. The remainder of this paper is organized as follows. Section 3.2 introduces the preliminaries and key definitions. In Section 3.3, we explain how to determine the exact order of candidates on the line and present properties of the optimal candidates. Section 3.4 discusses the process of moving voters to construct worst-case instances. In Section 3.5, we provide a warm-up by explaining the high-level idea of our voting rule for 2-winner elections. Section 3.6 extends these results to general values of k, detailing how to combine different voting rules, extending our *Polar Comparison Rule* for $k = 3$, and establishing an upper bound of roughly $7/3$, along with lower bounds for general k. In Section 3.7, we analyze the egalitarian additive cost. Finally, Section 3.8 discusses extensions of our rule and outlines open questions for future research. Note that Sections 3.3–3.6 focus on the utilitarian additive cost, while Section 3.7 examines the egalitarian additive cost.

3.1.2 Related Work

As mentioned, distortion was first introduced by Procaccia and Rosenschein (2006) in the non-metric framework. Since then, several lower and upper bounds have been established for distortion across different voting rules, in various scenarios, and within both non-metric and metric frameworks. For a comprehensive overview of distortion results, we refer to Anshelevich, Filos-Ratsikas, Shah, and Voudouris (2021). In this section, we group the related studies into two main areas.

Single-Winner Voting. For the metric framework, Anshelevich, Bhardwaj, Elkind, Postl, and Skowron (2018) established a general lower bound of 3 on the distortion of any voting rule. They also proved upper bounds on the distortion of several voting rules, including Majority, Borda, and Copeland, with the best rule being Copeland, which has a distortion of 5. Later, Munagala and K. Wang (2019) generalized this result to uncovered set rules and improved the best upper bound to 4.236, and Gkatzelis, Halpern, and Shah (2020) further reduced it to 3. Randomized single-winner voting rules have also been well studied in the metric framework (Charikar and Ramakrishnan, 2022; Fain, A. Goel, Munagala, and Prabhu, 2019; Pulyassary and Swamy, 2021). The current best-known upper bound on the distortion of a randomized voting rule, achieved by Charikar, Ramakrishnan, K. Wang, and Wu (2024), uses a randomization over other voting rules that can achieve a distortion of at most 2.753. In the non-metric framework, Caragiannis and Procaccia (2011) show that any voting rule has a distortion lower bound of $\Omega(m^2)$, with simple rules like Plurality having a distortion of $O(m^2)$.

Multi-Winner Voting. In the study of metric distortion in multi-winner voting, various objective functions have been proposed to capture the value each voter derives from the elected committee (Elkind, Faliszewski, Skowron, and Slinko, 2017; Faliszewski, Skowron, Slinko, and Talmon, 2017). A. Goel, Hulett, and Krishnaswamy (2018) show that for the additive cost function, applying a single-winner voting rule with distortion δ iteratively k times results in a k-winner committee with the same distortion δ. X. Chen, M. Li, and C. Wang (2020) studied the min-cost objective in the SCV scenario, where each voter casts a vote for a candidate. In the min-cost objective, each voter's cost is determined by the nearest elected candidate. They propose a voting rule with a tight distortion of 3 and a randomized rule with a distortion of $3 - 2/m$. Another objective, introduced by Caragiannis, Shah, and Voudouris (2022), is the q-cost objective, where a voter's cost for a committee is defined as the distance to their qth closest member in the committee. In the metric setting, Caragiannis, Shah, and Voudouris (2022) show that for $q \leq k/3$ distortion is unbounded, while for $k/3 < q \leq k/2$, distortion is bounded by n. For the case $q > k/2$, they present a non-polynomial voting rule that achieves the tight distortion of 3. Subsequently, Kizilkaya and Kempe (2022) propose a polynomial-time voting rule with the same distortion. The study of metric multi-winner voting rules has also been a recent area of focus in several other settings (Anshelevich and W. Zhu, 2021; Kalayci, Kempe, and Kher, 2024; Pulyassary, 2022).

A more specific setting in the metric framework considers single-peaked and 1-Euclidean preferences, where the voters and candidates lie on the real line (Black, 1948; Fotakis and Gourvès, 2022; Fotakis, Gourvès, and Monnot, 2016; Ghodsi, Latifian, and Seddighin, 2019; Miyagawa, 2001; Moulin, 1980; Voudouris, 2023). Fotakis, Gourvès, and Patsilinakos (2025) investigate the distortion of deterministic algorithms for k-committee selection on the line with the min-cost objective function, using additional distance queries. Recently, Cembrano and Shahkarami (2026) studied this problem in the context of peer selection, where the candidates and voters coincide, and provided tight results for several social cost objectives on the line metric.

For the non-metric framework, and when a voter's utility for a committee is defined as her maximum utility for its members, Caragiannis, Nath, Procaccia, and Shah (2017) proposed a rule that achieves distortion for deterministic committee selection of $O(1+m \cdot (m-k)/k)$. Another concept related to committee selection is that of stable committees and stable lotteries, which has been the subject of recent studies (Borodin, Halpern, Latifian, and Shah, 2022; Z. Jiang, Munagala, and K. Wang, 2020).

3.2 Preliminaries

Election. An instance of a committee election is a quadruple $\mathcal{E} = (V, C, k, \succ)$, where:

- $V = [n]$ is the set of voters,
- $C = [m]$ is the set of candidates,
- $k \in \mathbb{N}$ is the size of the committee, and
- $\succ = (\succ_1, \succ_2, \ldots, \succ_n) \in \mathcal{L}^n(C)$ comprises the voters' preference profiles, where $\succ_i \in \mathcal{L}(C)$ is a linear order on C for each $i \in V$.

For any pair of candidates $a, b \in C$, we let $V_{a \succ b} = \{i \in V \mid a \succ_i b\}$ denote the set of voters preferring a over b. A candidate a *defeats* b if $|V_{a \succ b}| > n/2$, and *Pareto-dominates* b if $|V_{a \succ b}| = n$.

Line metric. We assume that all voters and candidates are located on a line metric. Each voter $i \in V$ and each candidate $a \in C$ is associated with a position $x_i, x_a \in (-\infty, \infty)$, and the distance between them under metric d is given by $d(i, a) = |x_i - x_a|$. We assume that the candidates occupy distinct locations. We refer to the (unique) median voter as $\mathfrak{m}$; in the case of an even number of voters, the two consecutive median voters are denoted by $\mathfrak{m}_l$ and $\mathfrak{m}_r$. For any interval $I \subseteq \mathbb{R}$, we let $V(I) = \{i \in V \mid x_i \in I\}$ denote the voters located in I; for a single point x, we write $V(x)$ for the voters located at $\bar{x}$.

A metric d is said to be *consistent* with a preference profile $\succ \in \mathcal{L}^n(m)$, denoted by $d \rhd \succ$, if for every voter $i \in V$ and candidates $a, b \in C$, we have $d(i, a) < d(i, b)$ whenever $a \succ_i b$. As before, we may equivalently refer to the vector of locations $x = (x_i)_{i \in V}$ and write $x \rhd \succ$. A given preference profile may admit multiple consistent embeddings along the line.

Social cost. In our setting, the cost of a committee for a voter is defined as the sum of the distances from that voter to the candidates in the committee (additive cost). The cost of a committee for the entire set of voters can then be defined in two ways: either as the sum of the individual costs across all voters (utilitarian), or as the maximum cost incurred by any voter (egalitarian).

Formally, for a fixed election $\mathcal{E} = (V, C, k, \succ)$, a metric d, and a committee $S \subseteq C$ of size k, the cost of S for voter $i \in V$ is defined using a *candidate-aggregation function* $h \colon \mathbb{R}_+^k \to \mathbb{R}_+$ as

$$\mathrm{SC}(S, i; d) = h\big((d(i, a))_{a \in S}\big).$$

In this paper, we focus on the *additive cost*, given by $h(y) = \sum_{a \in S} y_a$. Accordingly, the social cost of S for all voters, determined by a *voter-aggregation function* $g \colon \mathbb{R}_+^n \to \mathbb{R}_+$, is

$$\mathrm{SC}(S, V; d) = g\big((\mathrm{SC}(S, i; d))_{i \in V}\big).$$

We study the utilitarian and egalitarian social costs, defined respectively as

$$g(y) = \sum_{i \in V} y_i \quad \text{and} \quad g(y) = \max_{i \in V} y_i.$$

Hence, for the *utilitarian additive cost*, we have

$$\mathrm{SC}(S, V; d) = \sum_{i \in V} \sum_{a \in S} d(i, a),$$

and for the *egalitarian additive cost*, we have

$$\mathrm{SC}(S, V; d) = \max_{i \in V} \sum_{a \in S} d(i, a).$$

For a single candidate $a \in C$, its social cost is $\mathrm{SC}(a, V; d) = \sum_{i \in V} d(i, a)$. When d or V are clear from the context, we may omit them from the notation; for example, we might write $\mathrm{SC}(S)$, $\mathrm{SC}(a; d)$, or simply $\mathrm{SC}(a)$. Finally, let $\mathrm{OPT} = \{o_1, \ldots, o_k\}$ be an optimal committee of size k, i.e., a committee minimizing the utilitarian social cost $\mathrm{SC}(\cdot)$, indexed such that $\mathrm{SC}(o_1) \leq \cdots \leq \mathrm{SC}(o_k)$.

Voting rules and distortion. An (n, m, k)-voting rule f takes as input a preference profile $\succ \in \mathcal{L}^n(m)$ and returns a committee S of size k, where $S = f(\succ) \in \binom{C}{k}$. We assume that f has access only to the rankings, not to the underlying metric. For an election $\mathcal{E} = (V, C, k, \succ)$ and a metric $d \rhd \succ$, the distortion of a committee $S \subseteq C$ is defined as

$$\mathrm{dist}(S, \mathcal{E}; d) = \frac{\mathrm{SC}(S, V; d)}{\min_{S' \in \binom{C}{k}} \mathrm{SC}(S', V; d)}.$$

The distortion of S with respect to $\mathcal{E}$ is its worst-case distortion over all consistent metrics:

$$\mathrm{dist}(S, \mathcal{E}) = \sup_{d \rhd \succ} \mathrm{dist}(S, \mathcal{E}; d).$$

Finally, the distortion of the rule f is

$$\mathrm{dist}(f) = \sup_{\succ \in \mathcal{L}^n(m)} \mathrm{dist}\big(f(\succ), ([n], [m], k, \succ)\big).$$

Throughout, we study the distortion achievable by voting rules under utilitarian and egalitarian additive social costs.

3.3 Properties of Candidates on a Line

In this section, we first show that the exact order of the candidates on the line can be determined under a mild assumption. We then characterize structural properties of the optimal candidates that later play a crucial role in the selection process. This section serves as a foundation for the analysis and results presented in the subsequent sections.

3.3.1 Order of Candidates

Elkind and Faliszewski (2014) show that if the voters' preference profiles are pairwise distinct, one can uniquely determine the ordering of the voters on the line, up to reversal. They also demonstrate that, given this voter ordering, the candidates can be arranged between the top preference of the leftmost voter and that of the rightmost voter. Although their result was originally used to determine whether an election is 1-Euclidean, it naturally applies to our setting as well. For completeness, however, we present a simpler method to determine the order of candidates that are not Pareto-dominated by any other candidate.

Lemma 3.1. *Under the line metric assumption, if we remove all candidates that are Pareto-dominated by another, the exact order of the remaining candidates on the line can be uniquely determined, up to reversal.*

To prove Lemma 3.1, we rely on two simple observations that connect the relative ordering of three candidates to the subsets of voters who prefer one candidate over another.

Observation 3.2. *Given three candidates a, b, and c with $x_a < x_b < x_c$, the voters who prefer a to b form a subset of those who prefer a to c; that is, $V_{a \succ b} \subseteq V_{a \succ c}$. Moreover, $V_{c \succ b} \subseteq V_{b \succ a}$.*

Input: Election instance $\mathcal{E} = (V, C, k, \succ)$
Output: Sequence $\mathcal{S}$ of non-Pareto-dominated candidates representing their order on the line

$B \leftarrow \{ e \in C \mid \exists e' \in C \setminus \{e\} : |V_{e' \succ e}| = n \}$; // Removing Pareto-dominated candidates
$C' \leftarrow C \setminus B$
Select $a, b \in C'$; // Arbitrary candidates for partitioning
$R \leftarrow \{ c \in C' \mid V_{a \succ b} \subseteq V_{a \succ c} \}$ $L \leftarrow C' \setminus R$ $i \leftarrow$ voter whose top-ranked candidate is in L $j \leftarrow$ voter whose top-ranked candidate is in R $\mathcal{S}_L \leftarrow \succ_j (L)$; // Order L by voter j's preferences
$\mathcal{S}_R \leftarrow \succ_i (R)$; // Order R by voter i's preferences
$\mathcal{S} \leftarrow \text{reverse}(\mathcal{S}_L) \cdot \mathcal{S}_R$; // Reverse L's order and concatenate with R
return $\mathcal{S}$

Algorithm 3.1: Finding the order of candidates

Observation 3.3. *Suppose no candidate Pareto-dominates another, and let $a, c \in C$ be such that $x_a < x_c$. If there exists a candidate $b \in C$ satisfying $V_{a \succ b} \subset V_{a \succ c}$, then b must lie between a and c on the line; that is, $x_a < x_b < x_c$.*

Proof. If $x_c < x_b$, then by Observation 3.2 we must have $V_{a \succ c} \subseteq V_{a \succ b}$, which is impossible since $V_{a \succ b} \subset V_{a \succ c}$. On the other hand, if $x_b < x_a$, Observation 3.2 implies that $V_{c \succ a} \subseteq V_{a \succ b}$. Thus, $V_{c \succ a} \subseteq V_{a \succ b} \subset V_{a \succ c}$. But we know that $V_{c \succ a} \cap V_{a \succ c} = \emptyset$, which means $V_{c \succ a} = \emptyset$. Hence, candidate c is dominated by a, contradicting the assumption that all Pareto-dominated candidates have been removed. This completes the proof. □

To determine the order of the candidates, we use Algorithm 3.1. The algorithm begins by removing all Pareto-dominated candidates. It then selects two arbitrary candidates, applies Observations 3.2 and 3.3, and partitions the remaining candidates into left and right subsets. Note that if a voter is positioned entirely on one side of all the candidates within a subset, their preference list can be used to establish the relative ordering of those candidates along the line. Accordingly, the algorithm identifies two voters whose top preferences lie in the left and right subsets, respectively, and then uses their preference lists to determine the final ordering of candidates.

We are now ready to complete the proof of Lemma 3.1 by showing that Algorithm 3.1 indeed recovers the correct order of the candidates on the line.

Proof of Lemma 3.1. We prove that Algorithm 3.1 returns the correct order of the candidates. Let a and b be two candidates and, without loss of generality, assume $x_a < x_b$. Let R denote the set of all candidates c such that $V_{a \succ b} \subseteq V_{a \succ c}$, and let L be the remaining candidates. Note that $a \in L$ and $b \in R$.

Claim. Every candidate in L lies to the left of every candidate in R.

Proof of claim. Suppose for a contradiction that there exist $c \in R$ and $d \in L$ with $x_c < x_d$. Since $c \in R$, we have $V_{a \succ b} \subseteq V_{a \succ c}$.

If $V_{a \succ b} \subset V_{a \succ c}$, then by the symmetric form of Observation 3.3 we must have $x_a \leq x_b \leq x_c$. Hence $x_a \leq x_b \leq x_d$, and by Observation 3.2 it follows that $V_{a \succ b} \subseteq V_{a \succ d}$, contradicting $d \notin R$.

It remains to consider the case $V_{a\succ b} = V_{a\succ c}$. If $x_c < x_a$, then Observation 3.2 gives $V_{c\succ a} \subseteq V_{a\succ b}$, and together with $V_{a\succ b} = V_{a\succ c}$ we obtain $V_{c\succ a} \subseteq V_{a\succ c}$. But $V_{c\succ a} \cap V_{a\succ c} = \emptyset$, so $V_{c\succ a} = \emptyset$, implying that a Pareto-dominates c—a contradiction to our preprocessing step. Therefore $x_a \leq x_c < x_d$, and Observation 3.3 yields $V_{a\succ c} \subseteq V_{a\succ d}$. Since $V_{a\succ b} \subseteq V_{a\succ c}$, we conclude $V_{a\succ b} \subseteq V_{a\succ d}$, so $d \in R$, again a contradiction. This proves the claim. △

We have shown that L and R are nonempty and that all candidates in L lie to the left of all candidates in R. Consider a voter i whose top-ranked candidate lies in L. Then i is located to the left of every candidate in R, so i's preference list reveals the exact order of the candidates in R along the line. Such a voter must exist; otherwise, every voter would prefer the leftmost candidate of R to every candidate in L, contradicting our assumption that no candidate Pareto-dominates another. A symmetric argument (using a voter j whose top-ranked candidate lies in R) provides the order within L.

Finally, Algorithm 3.1 outputs $\mathcal{S} = \text{reverse}(\mathcal{S}_L) \cdot \mathcal{S}_R$, which places all elements of L before all elements of R and, in particular, lists $a \in L$ before $b \in R$. Thus the algorithm recovers the correct order of the non–Pareto-dominated candidates, up to reversal, as claimed. □

3.3.2 Optimal Candidate

We now show that, on either side of the median voter, candidates that are closer to the median voter incur a lower utilitarian social cost.

Lemma 3.4. *Let $a, b \in C$ lie on the same side of the median voter (for an even number of voters, consider either $\mathfrak{m}_l$ or $\mathfrak{m}_r$ as the reference). If a is closer to the median voter than b, then $SC(a) \leq SC(b)$.*

Proof. We consider the case where candidate a lies to the left of the median voter; the other case is symmetric. We show that for an odd number of voters, $\mathrm{SC}(a)$ can be expressed as

$$\mathrm{SC}(a) \;=\; \mathrm{SC}(\mathfrak{m}) \;+\; 2 \sum_{i \in V([x_{\mathfrak{m}}, x_a]) \setminus \{\mathfrak{m}\}} d(a,i) \;+\; d(\mathfrak{m}, a),$$

where $\mathrm{SC}(\mathfrak{m})$ denotes the social cost of a hypothetical candidate located at $\mathfrak{m}$.

For an even number of voters, there are two median voters, $\mathfrak{m}_l$ and $\mathfrak{m}_r$. Without loss of generality, suppose a is closer to $\mathfrak{m}_l$ than to $\mathfrak{m}_r$. Then $\mathrm{SC}(a)$ can be expressed with respect to $\mathfrak{m}_l$ as follows:

$$\mathrm{SC}(a) \;=\; \begin{cases} \mathrm{SC}(\mathfrak{m}_l), & \text{if } x_{\mathfrak{m}_l} \leq x_a \leq x_{\mathfrak{m}_r}, \\ \mathrm{SC}(\mathfrak{m}_l) \;+\; 2\sum_{i \in V([x_a, x_{\mathfrak{m}_l}])} d(a,i), & \text{if } x_a < x_{\mathfrak{m}_l}. \end{cases}$$

Here $\mathrm{SC}(\mathfrak{m}_l)$ and $\mathrm{SC}(\mathfrak{m}_r)$ refer to the social costs of hypothetical candidates located at the two median positions. Before proving these equalities, note that Lemma 3.4 follows immediately from the above expressions. We now verify each case.

Case 1: The number of voters is odd. Partition the rest of the voters into the following subsets:

$$\begin{aligned} X &\coloneqq V((x_{\mathfrak{m}}, +\infty)), \\ Y &\coloneqq V([x_a, x_{\mathfrak{m}})) \\ Z &\coloneqq V((-\infty, x_a]) \end{aligned}$$

By the definition of the median, $|X| = |Y| + |Z|$. Therefore,

$$\begin{aligned} \mathrm{SC}(a) - \mathrm{SC}(\mathfrak{m}) &= d(\mathfrak{m}, a) + \sum_{x \in X} \big(d(a,x) - d(\mathfrak{m},x)\big) \\ &\quad + \sum_{y \in Y} \big(d(a,y) - d(\mathfrak{m},y)\big) + \sum_{z \in Z} \big(d(a,z) - d(\mathfrak{m},z)\big) \\ &= d(\mathfrak{m},a) + |X|\, d(\mathfrak{m},a) + \sum_{y \in Y} \big(2d(a,y) - d(\mathfrak{m},a)\big) + |Z|\, d(\mathfrak{m},a) \\ &= 2 \sum_{y \in Y} d(a,y) + d(\mathfrak{m},a)\big(|X| - |Y| - |Z| + 1\big) \\ &= 2 \sum_{y \in Y} d(a,y) + d(\mathfrak{m},a), \end{aligned}$$

which is the claimed expression for the odd case.

Case 2: The number of voters is even. In this case, $\mathrm{SC}(\mathfrak{m}_l) = \mathrm{SC}(\mathfrak{m}_r)$. For any a with $x_{\mathfrak{m}_l} \leq x_a \leq x_{\mathfrak{m}_r}$ we have $\mathrm{SC}(a) = \mathrm{SC}(\mathfrak{m}_l)$, since moving a candidate within $[x_{\mathfrak{m}_l}, x_{\mathfrak{m}_r}]$ does not change its social cost. Hence, assume $x_a < x_{\mathfrak{m}_l}$. Partition the voters as follows:

$$\begin{aligned} X &\coloneqq V((x_{\mathfrak{m}_r}, +\infty)), \\ Y &\coloneqq V((x_a, x_{\mathfrak{m}_l})), \\ Z &\coloneqq V((-\infty, x_a]). \end{aligned}$$

By the definition of the two medians, $|X| = |Y| + |Z|$. Thus,

$$\begin{aligned} \mathrm{SC}(a) - \mathrm{SC}(\mathfrak{m}_l) &= \sum_{x \in X} \big(d(a,x) - d(\mathfrak{m}_l,x)\big) \\ &\quad + \sum_{y \in Y} \big(d(a,y) - d(\mathfrak{m}_l,y)\big) + \sum_{z \in Z} \big(d(a,z) - d(\mathfrak{m}_l,z)\big) \\ &\quad + d(a,\mathfrak{m}_l) + \big(d(a,\mathfrak{m}_r) - d(\mathfrak{m}_r,\mathfrak{m}_l)\big) \\ &= 2 \sum_{y \in Y \cup \{\mathfrak{m}_l\}} d(a,y) + d(a,\mathfrak{m}_l)\big(|X| - |Y| - |Z|\big) \\ &= 2 \sum_{i \in V([x_a, x_{\mathfrak{m}_l}])} d(a,i), \end{aligned}$$

which is the claimed expression for the even case. This completes the proof. □

Corollary 3.5 (of Lemma 3.4)**.** *The optimal candidate denoted by o_1 is either the closest candidate to the right or the closest candidate to the left of the median voter. When n is even, the statement applies with respect to both median voters, $\mathfrak{m}_l$ and $\mathfrak{m}_r$.*

By Lemma 3.1, we can sort the candidates by their positions on the line. Corollary 3.5 further suggests that it is useful to consider an ordering consistent with the median voter's preference profile. Lemma 3.6 shows that such an ordering can be obtained. (We note that a voter at a given location may admit multiple consistent preference lists due to ties between equidistant candidates.)

Lemma 3.6. *Given an election $\mathcal{E} = (V, C, k, \succ)$, one can obtain an ordering of the candidates that is consistent with the preference list of the median voter.*

Proof. Define the *majority graph* of the election $\mathcal{E} = (V, C, k, \succ)$ as the directed graph $\mathcal{G}$ on vertex set C with an arc (a, b) iff a defeats b, i.e., $|V_{a \succ b}| > n/2$. When n is even and $|V_{a \succ b}| = |V_{b \succ a}|$, we break ties by placing the arc from the candidate that is weakly closer to the other along the line (and hence not to the right of the line); concretely, we orient the tie as (a, b) if $x_a \leq x_b$, which is well defined by Lemma 3.1. Under this tie-breaking, $\mathcal{G}$ is a tournament, and thus it admits a Hamiltonian path (Rédei, 1934).

We claim that any Hamiltonian path of $\mathcal{G}$ induces an ordering consistent with the median voter's preference. Indeed, for any pair of candidates $a, b \in C$, if the median voter prefers a to b (i.e., $a \succ_{\mathtt{m}} b$), then at least half of the voters (those on the same side of the midpoint between x_a and x_b as $\mathtt{m}$) also prefer a to b, so either a defeats b or the pair is tied and the tie is oriented in the direction of a by our rule. Hence a precedes b in $\mathcal{G}$'s Hamiltonian path, yielding an ordering that is consistent with the median voter's preference list. □

In the rest of the paper, we refer to the ordering consistent with the median voter's preference profile, obtained in Lemma 3.6, as the *Majority order*.

We now restate a lemma from Anshelevich, Bhardwaj, Elkind, Postl, and Skowron (2018), which provides an upper bound on the ratio of social costs between two candidates in any metric setting.

Lemma 3.7 (Anshelevich, Bhardwaj, Elkind, Postl, and Skowron (2018, Restated))**.** *For every pair of candidates $a, b \in C$, we have*

$$\frac{SC(a)}{SC(b)} \leq \frac{2n}{|V_{a \succ b}|} - 1.$$

Finally, we recall Corollary 3.8 from the same work, which follows directly from Lemma 3.7 and bounds the social cost of any candidate that defeats the optimal one. In particular, it implies that the distortion of the leading candidates in the Majority order is at most 3.

Corollary 3.8 (of Lemma 3.7)**.** *Let $a \in C$ be a candidate such that $|V_{a \succ o_1}| \geq |V_{o_1 \succ a}|$. Then $SC(a) \leq 3\, SC(o)$.*

3.4 Moving Voters Toward the Focal point

In this section, we present a method to upper-bound the distortion of a committee S compared to an optimal committee OPT. We show that it is possible to simplify any election instance by relocating voters, ensuring that if the distortion ratio SC(S)/SC(OPT) is initially greater than a threshold δ, it remains greater than δ in the simplified instance.

The core of this method is a shifting argument. We determine a target location called the *Focal point* and shift voters toward it. We shift and collect the voters at specific intermediate locations along their path to the Focal Point. This process results in a simpler instance where voters are located at only a few points on the line. This simplification allows us to easily verify the distortion bound: if we prove that the ratio $\mathrm{SC}(S)/\mathrm{SC}(\mathrm{OPT})$ is smaller than the threshold δ in the simplified instance, then the ratio must be smaller than δ in the initial instance as well.

We begin by defining the notion of *consecutive committees* and a simple mathematical statement in Observation 3.10.

Definition 3.9. *Let $S_1, S_2 \subseteq C$ be committees, and let $T = S_1 \cap S_2$. We say that S_1 and S_2 are* consecutive committees *if the candidates in $S_1 \setminus T$ and $S_2 \setminus T$ lie on opposite sides of T. If $T = \emptyset$, then S_1 and S_2 are consecutive if there exists a point on the line such that all candidates in S_1 lie strictly on one side of it and all candidates in S_2 lie strictly on the other.*

Observation 3.10. *Let w, z, w', z', and ρ be positive real numbers such that $w/z \geq \rho$. Then:*

(1) If $w'/z' \geq \rho$, then $(w + w')/(z + z') \geq \rho$.

(2) If $w'/z' \leq \rho$, $w' \leq w$, and $z' < z$, then $(w - w')/(z - z') \geq \rho$.

We are now ready to present Lemma 3.11, which states that given two committees S_1 and S_2 of size k, Algorithm 3.2 returns a point x^* on the line, called the Focal point, such that moving all voters toward x^* does not significantly decrease the ratio between the social costs of S_1 and S_2.

Lemma 3.11. *Consider an election $\mathcal{E} = (V, C, k, \succ)$, a metric d consistent with $\mathcal{E}$, and a threshold $\delta > 1$. Suppose we have two consecutive committees $S_1, S_2 \subseteq C$, each of size k, such that*

$$\frac{SC(S_1, V; d)}{SC(S_2, V; d)} > \delta.$$

Let x^* be the Focal point on the line returned by Algorithm 3.2. Consider a new metric d' obtained by moving each voter toward x^*. Then,

$$\frac{SC(S_1, V; d')}{SC(S_2, V; d')} > \delta.$$

Proof. Let $T = S_1 \cap S_2$ with $|T| = t$, $L = S_1 \setminus T$, and $R = S_2 \setminus T$. Based on the definition of consecutive committees and without loss of generalisy, we assume the ordering of candidates on the line as follows:

$$\underbrace{a_1 \leq \cdots \leq a_{k-t}}_{L} \leq \underbrace{c_1 \leq \cdots \leq c_t}_{T} \leq \underbrace{b_{k-t} \leq \cdots \leq b_1}_{R}.$$

We first verify that the algorithm is well-defined. Let $j^* = \lceil \frac{k(\delta-1)}{2\delta} \rceil$ and $i^* = \lceil \frac{k}{2} + \frac{k-t}{\delta-1} \rceil$ be the indices used in the algorithm's output. We show that these refer to valid candidates within R (i.e., $1 \leq j^* \leq k - t$) and T (i.e., $1 \leq i^* \leq t$), respectively.

Input : Election instance $\mathcal{E} = (V, C, k, \succ)$, metric d, consecutive committees S_1, $S_2 \subseteq C$, and threshold $\delta > 1$.
Output : A Focal point x^* on the line with respect to S_1 and S_2.

$T \leftarrow S_1 \cap S_2 = \{ c_1, c_2, \ldots, c_t \}$; // Common candidates, ordered: $x_{c_1} \leq \cdots \leq x_{c_t}$
$L = S_1 \setminus T = \{ a_1, a_2, \ldots, a_{k-t} \}$; // Left side: $x_{a_1} \leq \cdots \leq x_{a_{k-t}} \leq x_{c_1}$
$R = S_2 \setminus T = \{ b_1, b_2, \ldots, b_{k-t} \}$; // Right side: $x_{b_1} \geq \cdots \geq x_{b_{k-t}} \geq x_{c_t}$
if $t \leq \lfloor k/2 \rfloor$ *or* $\delta \leq \dfrac{k}{2t-k}$ **then**
  return $x_{b_{\left\lceil \frac{k(\delta-1)}{2\delta} \right\rceil}}$
else
  return $x_{c_{\left\lceil \frac{k}{2} + \frac{k-t}{\delta-1} \right\rceil}}$
end

Algorithm 3.2: Finding the Focal point

- If $x^* = b_{j^*}$, we have $j^* = \lceil \frac{k}{2}(1 - \frac{1}{\delta}) \rceil$. Since $\delta > 1$, $j^* \leq \lceil k/2 \rceil$. If $t \leq k/2$, then $|R| = k - t \geq k/2 \geq j^*$, so b_{j^*} exists. If $t > k/2$, the algorithm requires $\delta \leq \frac{k}{2t-k}$. Substituting this bound yields $j^* \leq \lceil \frac{k}{2}(1 - \frac{2t-k}{k}) \rceil = k - t = |R|$. Thus $b_{j^*} \in R$.
- If $x^* = c_{i^*}$, this case requires $\delta > \frac{k}{2t-k}$ and $t > \lfloor k/2 \rfloor$, which implies $\delta - 1 > \frac{2(k-t)}{2t-k}$. and $2t - k > 0$ Substituting this inequality into the expression for i^*:

$$i^* < \frac{k}{2} + \frac{k-t}{2(k-t)/(2t-k)} = \frac{k}{2} + \frac{2t-k}{2} = t.$$

  Thus $1 \leq i^* \leq t$, so $c_{i^*} \in T$.

Now consider a voter v moving a distance ε toward the Focal point x^* designated by Algorithm 3.2, resulting in metric d'. We assume that ε is sufficiently small such that v does not pass the location of any candidate in S_1 or S_2 (though v may land exactly on a candidate). If we prove that this movement keeps the ratio $\mathrm{SC}(S_1, V; d')/\mathrm{SC}(S_2, V; d')$ above δ, we can combine such movements to prove the lemma statement for any other metric caused by moving the voters toward the focal point.

For any committee S, let ΔS denote the *reduction* in the social cost of S due to this movement.

$$\Delta S = \mathrm{SC}(S, V; d) - \mathrm{SC}(S, V; d').$$

Let $N_{dir}(S)$ be the number of candidates in S located strictly in the direction of movement relative to v. The reduction is given by:

$$\Delta S = (N_{dir}(S) - (k - N_{dir}(S))) \cdot \varepsilon = (2N_{dir}(S) - k) \cdot \varepsilon.$$

Note that ΔS can be negative (indicating a cost increase) if $N_{dir}(S) < k/2$.

Let $w = \mathrm{SC}(S_1, V; d)$ and $z = \mathrm{SC}(S_2, V; d)$. We are given $w/z > \delta$. The new ratio is

$$\frac{\mathrm{SC}(S_1, V; d')}{\mathrm{SC}(S_2, V; d')} = \frac{w - \Delta S_1}{z - \Delta S_2}.$$

We verify that this ratio remains greater than δ by distinguishing the scenarios of cost changes and applying Observation 3.10:

- Both costs decrease ($\Delta S_1 > 0, \Delta S_2 > 0$): By Observation 3.10 (Item 2), the ratio remains larger than δ if the ratio of reductions satisfies $\Delta S_1/\Delta S_2 \leq \delta$.
- S_1 increases, S_2 decreases ($\Delta S_1 \leq 0, \Delta S_2 \geq 0$): The numerator increases (or stays same) while the denominator decreases (or stays the same). The ratio increases, so the condition holds trivially.
- S_1 decreases, S_2 increases ($\Delta S_1 \geq 0, \Delta S_2 \leq 0$): We must show that this case cannot happen, as the ration will strictly decrease, unless $\Delta S_1 = \Delta S_2 = 0$.
- Both costs increase ($\Delta S_1 < 0, \Delta S_2 < 0$): This corresponds to adding positive costs $-\Delta S_1$ and $-\Delta S_2$. By Observation 3.10 (Item 1), the ratio remains larger than δ if $-\Delta S_1/(-\Delta S_2) \geq \delta$.

We now analyze the movement based on the location of v relative to x^*.

Case A: Moving Right (v is to the left of x^*). Let $N_{Right}(S)$ be the number of the candidates of S lying on the right side of v. Since $S_2 = T \cup R$ lies to the right of $S_1 = L \cup T$ (with T shared), for any voter position v, the number of S_2 candidates to the right is always greater than or equal to that of S_1, i.e., $N_{Right}(S_2) \geq N_{Right}(S_1)$. Therefore, $\Delta S_2 \geq \Delta S_1$.

- If both decrease ($\Delta S_2 > \Delta S_1 > 0$), then $\Delta S_1/\Delta S_2 \leq 1 < \delta$, satisfying the condition.
- If $\Delta S_1 \leq 0$ and $\Delta S_2 \geq 0$, the ratio increases and remains larger than δ.
- The case $\Delta S_1 \geq 0$ and $\Delta S_2 \leq 0$ is not possible because $\Delta S_2 \geq \Delta S_1$, unless $\Delta S_1 = \Delta S_2 = 0$, for which the lemma's statement holds.
- If both increase ($\Delta S_1 < \Delta S_2 < 0$), we have $-\Delta S_1 \geq -\Delta S_2 > 0$. We need to prove that $-\Delta S_1/(-\Delta S_2) \geq \delta$, or equivalently

$$\frac{k - 2N_{Right}(S_1)}{k - 2N_{Right}(S_2)} \geq \delta \iff 2\left(N_{Right}(S_2)\delta - N_{Right}(S_1)\right) \geq k(\delta - 1).$$

  Now we have three cases for the position of v:

  - If v lies on the left side of c_1, then $N_{Right}(S_1) \leq k$ and $N_{Right}(S_2) = k$. Therefore $2(N_{Right}(S_2)\delta - 2N_{Right}(S_1)) \geq k(\delta - 1)$, as we wanted.
  - If v lies on the right side of b_{k-t}, the fact that v is to the left of x^* implies that $x^* \in R$. Therefore, according to Algorithm 3.2, $x^* = x_{b_{j^*}}$ where $j^* = \lceil k(\delta - 1)/(2\delta) \rceil$. Thus, $N_{Right}(S_1) = 0$ and $N_{Right}(S_2) \geq j^* \geq k(\delta - 1)/(2\delta)$. Therefore,

$$2(N_{Right}(S_2)\delta - N_{Right}(S_1)) \geq 2j^*\delta \geq k(\delta - 1).$$

  - Finally, if v lies between c_1 and b_{k-t}, we have two subcases according to the algorithm: the Focal point x^* is $x_{c_{i^*}}$ for $i^* = \lceil k/2 + (k-t)/(\delta - 1) \rceil$, or the output is $x_{b_{j^*}}$ where $j^* = \lceil k(\delta - 1)/(2\delta) \rceil$. For both subcases, assume that

there are i members of T to the left side of (or located on) v. Therefore, $N_{Right}(S_1) = t - i, N_{Right}(S_2) = k - i$.
In the first subcase, since v must lie to the left side of c_{i^*}, $i \leq i^* - 1 \leq k/2 + (k-t)/(\delta - 1)$. This implies that

$$\begin{aligned} 2(N_{Right}(S_2)\delta - N_{Right}(S_1)) &= 2(k\delta - i\delta - t + i) \\ &= 2k\delta - 2t - 2i(\delta - 1) \\ &\geq 2k\delta - 2t - 2(\delta - 1)(\frac{k}{2} + \frac{k-t}{\delta - 1}) \\ &= k(\delta - 1). \end{aligned}$$

For the second subcase, $x^* = x_{b_{j^*}}$. According to the construction of the algorithm, this happens if either $t \leq \lfloor k/2 \rfloor$ or $\delta \leq k/(2t-k)$. Moreover, since v lies on the left side of b_{k-t}, $i \leq t$. Therefore,

$$\begin{aligned} 2(N_{Right}(S_2)\delta - N_{Right}(S_1)) &= 2(k\delta - i\delta - t + i) \\ &= 2(k\delta - t - i(\delta - 1)) \\ &\geq 2(k\delta - t - t(\delta - 1)) \\ &= 2\delta(k - t). \end{aligned}$$

We want to prove that $2\delta(k-t) \geq k\delta - k$. This is equivalent to

$$\begin{aligned} 2\delta(k-t) \geq k\delta - k &\iff k(\delta + 1) \geq 2\delta t \\ &\iff \frac{\delta + 1}{\delta} \geq \frac{2t}{k} \\ &\iff \frac{1}{\delta} \geq \frac{2t - k}{k}. \end{aligned}$$

We know that in this subcase, either $t \leq \lfloor k/2 \rfloor$ or $\delta \leq k/(2t-k)$. If $2t - k \leq 0$, the above inequality holds becaue $1/\delta > 0 \geq (2t-k)/k$. Otherwise, $t > \lceil (k+1)/2 \rceil$. Therefore, we must have $\delta \leq k/(2t-k)$. Thus, $1/\delta \geq (2t-k)/k$, proving the inequality.

Case B: Moving Left (v is to the right of x^*). Since $S_1 = L \cup T$ lies to the left of $S_2 = T \cup R$, for any voter position v, the number of candidates in S_1 located strictly to the left of v is greater than or equal to that of S_2. Let $N_{Left}(S)$ denote the number of candidates of S strictly to the left of v. We have $N_{Left}(S_1) \geq N_{Left}(S_2)$, which implies $\Delta S_1 \geq \Delta S_2$. First, we show that the social cost of S_2 decreases or stays the same (i.e., $\Delta S_2 \geq 0$). This requires $N_{Left}(S_2) \geq k/2$. Since v is to the right of x^*, $N_{Left}(S_2)$ is at least the number of candidates in S_2 to the left of (or at) x^*.

- If $x^* = b_{j^*} \in R$, then $N_{Left}(S_2) \geq t + (k - t - j^* + 1)$. Since $j^* \leq \lceil k/2 \rceil$, this sum is at least $k - \lceil k/2 \rceil + 1 \geq k/2$.
- If $x^* = c_{i^*} \in T$, then $N_{Left}(S_2) \geq i^*$. Since $i^* \geq k/2$, this is at least $k/2$.

Thus $\Delta S_2 \geq 0$, and consequently $\Delta S_1 \geq 0$. We are in the scenario where both costs decrease. The condition to maintain the ratio is $\Delta S_1 / \Delta S_2 \leq \delta$, which simplifies to:

$$2(\delta N_{Left}(S_2) - N_{Left}(S_1)) \geq k(\delta - 1). \tag{3.1}$$

We analyze this inequality based on the branches of the algorithm and the position of v.

- If the algorithm returns $x^* = b_{j^*}$: Here $x^* \in R$. Since v is to the right of x^*, v must be to the right of all candidates in T and L. Therefore, $N_{Left}(S_1) = k$. For S_2, v is to the right of all of T and at least the candidates $b_{k-t}, \ldots, b_{j^*}$ in R. The number of candidates in R strictly to the left of v is at least $(k-t) - j^* + 1$. Thus, $N_{Left}(S_2) \geq t + k - t - j^* + 1 = k - j^* + 1$. Substituting these into Inequality (3.1):

$$\begin{aligned} 2(\delta N_{Left}(S_2) - N_{Left}(S_1)) &\geq 2(\delta(k - j^* + 1) - k) \\ &= 2\delta k - 2\delta j^* + 2\delta - 2k. \end{aligned}$$

  We need this to be $\geq k\delta - k$. Rearranging terms, we require:

$$k\delta + 2\delta - k \geq 2\delta j^* \iff j^* \leq \frac{k(\delta - 1)}{2\delta} + 1.$$

  Since the algorithm sets $j^* = \lceil \frac{k(\delta-1)}{2\delta} \rceil$ and $\lceil x \rceil < x + 1$, this condition always holds.

- If the algorithm returns $x^* = c_{i^*}$: Here $x^* \in T$. The voter v is to the right of c_{i^*}. We distinguish two positions for v:

  - If v lies within T (specifically, v is between c_i and c_{i+1} for some $i \geq i^*$, or between c_t and b_{k-t}): Here, the number of T candidates to the left of v is i, where $i \geq i^*$. We have $N_{Left}(S_1) = (k-t) + i$ and $N_{Left}(S_2) = i$. Inequality (3.1) becomes:

$$2(\delta i - (k - t + i)) = 2(i(\delta - 1) - (k - t)) \geq k(\delta + 1).$$

    Solving this for i:

$$2i(\delta - 1) \geq k(\delta - 1) + 2(k - t) \iff i \geq \frac{k}{2} + \frac{k - t}{\delta - 1}.$$

    Since v is to the right of c_{i^*}, we know $i \geq i^*$. The algorithm sets $i^* = \lceil \frac{k}{2} + \frac{k-t}{\delta-1} \rceil$, ensuring this condition is satisfied.

  - If v lies within R (or to the right of R): Here, v is to the right of all candidates in T. Thus $N_{Left}(S_1) = k$. For S_2, v is to the right of all T and some non-negative number of candidates in R, say $n_R \geq 0$. So $N_{Left}(S_2) = t + n_R$. Inequality (3.1) becomes:

$$2(\delta(t + n_R) - k) \geq k(\delta - 1).$$

    Since $n_R \geq 0$, we require:

$$2(\delta t - k) \geq k(\delta - 1) \iff 2\delta t - 2k \geq k\delta - k \iff \delta(2t - k) \geq k.$$

    This branch of the algorithm is only entered when $2t - k > 0$ and $\delta > \frac{k}{2t-k}$, which implies $\delta(2t - k) > k$. Thus, the inequality holds.

□

Using the fact that moving voters toward the Focal point keeps the ratio above δ by Lemma 3.11, the following lemma shows that we can gather these voters at specific points $x_1, \ldots, x_t$ and x^*. This allows us to replace the original election with a simplified instance where all voters are located at just a few positions.

Lemma 3.12. *Let $\mathcal{E} = (V, C, k, \succ)$ be an election and let d be a consistent metric. Let $S_1, S_2 \subseteq C$ be consecutive committees of size k with*

$$\frac{SC(S_2, V; d)}{SC(S_1, V; d)} > \delta,$$

and let x^* be the Focal point with respect to threshold δ, S_1, and S_2. Suppose we have:

- *A sequence of points $z_1 \leq \cdots \leq z_t \leq x^*$ and integers $0 = r_0 \leq r_1 \leq \cdots \leq r_t$ such that for each $i \in \{1, \ldots, t\}$,*

$$|V((-\infty, z_i])| \geq r_i.$$

- *A sequence of points $y_1 \geq \cdots \geq y_p \geq x^*$ and integers $0 = q_0 \leq q_1 \leq \cdots \leq q_p$ such that for each $j \in \{1, \ldots, p\}$,*

$$|V([y_j, \infty))| \geq q_j.$$

Assume that the sets of voters counted are disjoint (i.e., $r_t + q_p \leq |V|$). Construct $\mathcal{E}' = (V, C, k, \succ')$ with metric d' by relocating voters as follows:

(1) For each $i \in \{1, \ldots, t\}$, place exactly $r_i - r_{i-1}$ voters at z_i.

(2) For each $j \in \{1, \ldots, p\}$, place exactly $q_j - q_{j-1}$ voters at y_j.

(3) Place all remaining voters at x^*.

Then,

$$\frac{SC(S_2, V; d')}{SC(S_1, V; d')} > \delta.$$

Proof. We index the voters as $v_1, v_2, \ldots, v_n$ such that their positions in the original metric d are sorted non-decreasingly: $x_{v_1} \leq x_{v_2} \leq \cdots \leq x_{v_n}$. We construct the new instance by moving specific groups of voters to the locations z_i, y_j, or x^*. We show that every voter moves toward x^* (or stays fixed), which by Lemma 3.11 implies the social cost ratio remains greater than δ.

Left Side ($x_v \leq x^*$): The constraint $|V((-\infty, z_i])| \geq r_i$ implies that at least r_i voters are located at or to the left of z_i. In our sorted indexing, this means the voter v_{r_i} satisfies $x_{v_{r_i}} \leq z_i$. Consequently, for any $k \leq r_i$, the voter v_k is located at $x_{v_k} \leq x_{v_{r_i}} \leq z_i$. For each $i \in \{1, \ldots, t\}$, we select the block of voters $\{v_k \mid r_{i-1} < k \leq r_i\}$ and move them to z_i. For any such voter v_k, the movement is from x_{v_k} to z_i. Since $x_{v_k} \leq z_i \leq x^*$, this is a movement to the right, strictly toward the Focal point x^*.

Right Side ($x_v \geq x^*$): The construction for the right side is symmetric. The constraint $|V([y_j, \infty))| \geq q_j$ ensures that the last q_j voters are to the right of y_j; moving the corresponding block to y_j constitutes a movement to the left, toward x^*.

Remaining Voters: The remaining voters are those with indices k such that $r_t < k \leq n - q_p$. We move all these voters to x^*. Regardless of whether a voter v_k is originally to the left or right of x^*, moving them directly to x^* reduces their distance to x^* to zero. Thus, they are moved toward the Focal point.

Since all voters are moved toward x^* (or remain at their position), the condition of Lemma 3.11 is satisfied, and the ratio of social costs in the new instance remains strictly greater than δ. □

3.5 Warm-up: Polar Comparison Rule for $k = 2$

In this section, we provide a voting rule for the line metric in the case where the committee size is two (i.e., a 2-winner election), and we prove that the distortion of this rule is at most $1 + \sqrt{2}$. Moreover, we show that this bound is tight for the line metric. This section serves as a warm-up for the next section, where we generalize the approach to elections with larger committees.

As described in Lemma 3.4, the candidates in the optimal committee are close to the median voter(s). Thus, a natural approach is to select the committee from the candidates near the top of the Majority order, since these candidates are near the median voter(s). In Algorithm 3.3, we present a voting rule for 2-winner elections ($k = 2$), called *Polar Comparison Rule*. This voting rule first selects the top candidate in the Majority order. Let us denote this candidate by a. To select the second member, we compare the two candidates adjacent to a, denoted by b and c, where b is the second candidate in the Majority order and c is the closest candidate to a such that a lies between b and c. Without loss of generality, assume that $x_c \leq x_a \leq x_b$. The intuition behind this approach is to ensure that the output committee is favored by voters on both sides of a. First, we prove that we can identify the three candidates a, b, and c under some mild conditions using the ordinal preferences $\succ$.

Observation 3.13. *Given an election instance $\mathcal{E} = (V, C, 2, \succ)$, if there exists a candidate c immediately to the left of a, such that c is not Pareto-dominated by any candidate other than a (and possibly not Pareto-dominated at all), we can identify the three candidates a, b, and c as described above using only the ordinal preferences $\succ$.*

Proof. To identify a and b, we apply Lemma 3.6 to establish the Majority order. Next, to determine the candidate c immediately to the left of a, we refine the set of eligible candidates. This is because c might be Pareto-dominated by a. We then apply Lemma 3.1 to determine the exact order of the remaining candidates along the line. From this ordered set, we identify c as the candidate closest to a such that a lies between b and c. If no such candidate exists, then either there is no candidate to the left of a, or all candidates to the left of a are Pareto-dominated. □

We now show that our 2-Winner Polar Comparison Rule achieves a distortion bounded by $1 + \sqrt{2}$ under the utilitarian additive cost on the line metric.

Input : Instance $\mathcal{E} = (V, C, 2, \succ)$
Output : A committee of two candidates

$a, b \leftarrow$ top two candidates in the Majority order $B \leftarrow \{e \in C \mid \exists e' \in C \setminus \{e, a\} : |V_{e' \succ e}| = n\}$; // Candidates Pareto-dominated by $e' \neq a$
$C' \leftarrow C \setminus B$; // Exclude Pareto-dominated candidates
Find the closest candidate $c \in C'$ to a such that a lies between b and c ; // By Lemma 3.1
if *c does not exist* **then**
  return $\{a, b\}$
end
if $|V_{c \succ b}| \leq \frac{n}{1+\sqrt{2}}$ **then**
  return $\{a, b\}$
else
  if $|V_{b \succ a}| \geq \frac{n}{1+\sqrt{2}}$ **then**
    return $\{a, b\}$
  else
    return $\{a, c\}$
  end
end

Algorithm 3.3: 2-Winner Polar Comparison Rule

Theorem 3.14. *Let f be the voting rule that, for any election instance $\mathcal{E} = (V, C, k, \succ)$, selects a committee of size two according to Algorithm 3.3 (the 2-Winner Polar Comparison Rule). Assume that voters and candidates lie on a line metric, and let d be any metric consistent with $\succ$. Then, under the utilitarian additive social cost,*

$$\mathrm{dist}(f(\succ), \mathcal{E}; d) \leq 1 + \sqrt{2}.$$

In particular, the distortion of f satisfies

$$\mathrm{dist}(f) \leq 1 + \sqrt{2}.$$

To prove Theorem 3.14, we use Observation 3.15, which states that the voters who prefer candidate a over b are exactly those whose locations lie on the half-line containing a and bounded by the midpoint between a and b, namely $\overline{m}_{ab}$.

Observation 3.15. *For two candidates a and b, if $x_a \leq x_b$, then $V_{a \succ b} = V((-\infty, \overline{m}_{ab}])$, where $\overline{m}_{ab}$ is the midpoint of x_a and x_b. Similarly, if $x_a \geq x_b$, then $V_{a \succ b} = V([\overline{m}_{ab}, \infty))$.*

As an intermediate step, we prove that if the median voter lies between the top two candidates in the Majority order, then selecting these two candidates yields a distortion of at most 2. This lemma will be instrumental in the proof of Theorem 3.14.

Lemma 3.16. *Let $\mathcal{E}$ be an election where a and b are the top two candidates in the Majority order. If $\mathfrak{m}$ lies between a and b, then for any pair of candidates a' and b',*

$$\frac{SC(\{a, b\})}{SC(\{a', b'\})} \leq 2.$$

Proof. Fix an instance $\mathcal{E}$. Without loss of generality, assume that $x_a \leq x_b$. It suffices to prove the lemma for the case that $\{a', b'\} = \{o_1, o_2\}$ is the optimal committee of size 2. Note that by Corollary 3.5, the optimal candidate, o_1, is either a or b. Without loss of generality, assume that $a = o_1$. If $b \neq o_2$, by Lemma 3.4, $x_{o_2} \leq x_a \leq x_b$ and also $\mathrm{SC}(a) \geq \mathrm{SC}(\mathfrak{m})$. Therefore,

$$\begin{aligned} \frac{\mathrm{SC}(\{a,b\})}{\mathrm{SC}(\{a,o_2\})} &\leq \frac{\mathrm{SC}(\{\mathfrak{m},b\})}{\mathrm{SC}(\{\mathfrak{m},o_2\})} \\ &= 1 + \frac{\mathrm{SC}(b) - \mathrm{SC}(o_2)}{\mathrm{SC}(\mathfrak{m}) + \mathrm{SC}(o_2)}. \end{aligned} \tag{3.2}$$

Next, we transform instance $\mathcal{E}$ to a new instance $\mathcal{E}'$, where the positions of voters are changed by a metric d' in which the ratio in Equation (3.2) does not decrease. This new instance will help us to upper bound the ratio in Equation (3.2). The new positions, changed by a metric d' are defined as follows:

$$x'_i = \begin{cases} x_{o_2} & \text{if } x_i \in (-\infty, x_{\mathfrak{m}}) \\ x_{\mathfrak{m}} & \text{if } x_i \in [x_{\mathfrak{m}}, +\infty)\,. \end{cases} \tag{3.3}$$

Note that in comparison to $\mathcal{E}$, the voters in $\mathcal{E}'$ are moved to new locations according to Equation (3.3). Therefore, there are two types of movements. We show that none of these movements reduce the ratio in Equation (3.2).

- $(-\infty, \mathbf{x_{\mathfrak{m}}}) \to \mathbf{x_{o_2}}$: If at first we move every voter in $(-\infty, x_{o_2})$ to x_{o_2}, this move does not change the value of $\mathrm{SC}(b) - \mathrm{SC}(o_2)$, but decreases $\mathrm{SC}(\mathfrak{m}) + \mathrm{SC}(o_2)$. Afterwards, if we move all voters in $(x_{o_2}, x_{\mathfrak{m}})$ to x_{o_2}, the new value of $\mathrm{SC}(\mathfrak{m}) + \mathrm{SC}(o_2)$ remains intact, while the value of $\mathrm{SC}(b) - \mathrm{SC}(o_2)$ is increased. Therefore, this move does not decrease the total ratio in Equation (3.2).

- $[\mathbf{x_{\mathfrak{m}}}, +\infty) \to \mathbf{x_{\mathfrak{m}}}$: If at first we move every voter in $(x_b, +\infty)$ to x_b, this move does not change the value of $\mathrm{SC}(b) - \mathrm{SC}(o_2)$, but decreases $\mathrm{SC}(\mathfrak{m}) + \mathrm{SC}(o_2)$. Afterwards, if we move all voters in $(x_{\mathfrak{m}}, x_b]$ to $x_{\mathfrak{m}}$, the new value of $\mathrm{SC}(\mathfrak{m}) + \mathrm{SC}(o_2)$ is decreased, while the value of $\mathrm{SC}(b) + \mathrm{SC}(o_2)$ is unchanged. Therefore, this move does not reduce the total ratio in Equation (3.2).

Now it remains to calculate the ratio in the new instance $\mathcal{E}'$. Note that since b appears earlier than o_2 in the Majority order, $|V_{b \succ o_2}| \geq n/2$, see Figure 3.2. For brevity, let $s = d(o_2, \mathfrak{m})$ and $r = d(\mathfrak{m}, b)$. Note that $s \geq r$. Therefore,

$$\begin{aligned} \frac{\mathrm{SC}(\{a,b\})}{\mathrm{SC}(\{a,o_2\})} &\leq \frac{\mathrm{SC}(\{\mathfrak{m},b\})}{\mathrm{SC}(\{\mathfrak{m},o_2\})} \\ &\leq \frac{\frac{n}{2}(s + r + (s + r))}{ns} \\ &\leq 2. \end{aligned}$$

□

Sketch of Proof of Theorem 3.14. We analyze different cases of the optimal committee and the output of f. In some cases, we apply Lemmas 3.7 and 3.16 to establish an upper

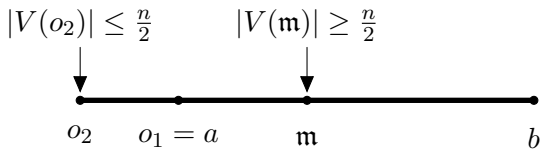


Figure 3.2: If a and b are the top two candidates in the Majority order and $\{o_1, o_2\}$ is the optimal committee of size 2, by assuming that $a = o_1$ and moving the location of voters, one can prove that $\text{dist}(\{a, b\}) \leq 2$.

bound on the distortion. In other cases, we identify the Focal point and move voters according to Lemma 3.12 to construct a new instance, for which the social cost ratio does not exceed $1 + \sqrt{2}$. □

Now, we present the proof of Theorem 3.14, which establishes the upper bound of $1 + \sqrt{2}$ on the distortion of the 2-Winner Polar Comparison Rule.

Proof of Theorem 3.14. To prove the theorem, we begin by assuming the contrary, i.e., there exists an instance $\mathcal{E} = (V, C, 2, \succ)$ and metric d, such that $\text{dist}(f(\succ), \mathcal{E}; d) > 1+\sqrt{2}$. Assume the top two candidates in the Majority order are a and b, and without loss of generality, $x_a \leq x_b$. Let c be the nearest candidate to a that is not Pareto-dominated by any other candidate and satisfies $x_c \leq x_a$.

If such a candidate c does not exist, it implies that a and b Pareto-dominate all candidates to the left of a. Therefore, by Lemma 3.6, either $\{a, b\}$ is the optimal committee, or b is the optimal candidate. In the latter case, by Corollary 3.8, $\text{SC}(a) \leq 3\text{SC}(b)$. Thus,

$$\text{dist}(f(\succ), \mathcal{E}; d) = \frac{\text{SC}(a) + \text{SC}(b)}{2\text{SC}(b)} \leq 2 < 1 + \sqrt{2}.$$

Therefore, we assume that such a candidate c exists. By Corollary 3.5, we observe that $o_1 \in \{a, b, c\}$. Furthermore, by Lemma 3.4, either $o_2 \in \{a, b, c\}$, or we can lower bound SC(OPT) by one of $2\text{SC}(b)$ and $2\text{SC}(c)$. Thus, we can consider the following cases for the optimal candidates:

$$(x_{o_1}, x_{o_2}) \in \{(x_b, x_b), (x_c, x_c), (x_a, x_c), (x_c, x_a), (x_a, x_b), (x_b, x_a)\}.$$

We evaluate the distortion of the voting rule on a case-by-case basis and establish that it is always at most $1+\sqrt{2}$. Throughout the proof, we denote by $\overline{m}_{bc}$ the midpoint of x_b and x_c, and by $\overline{m}_{ab}$ the midpoint of x_a and x_b. We furthermore introduce a parameter α, which we fix to $\sqrt{2}$, since this choice minimizes the upper bound on the distortion in every case.

Case.1 $(x_{o_1}, x_{o_2}) = (x_b, x_b)$.

Our voting rule, f, selects a and then chooses between c and b. Since b is the optimal candidate, by Corollary 3.5, the median point, $\mathfrak{m}$, must lie between a and b. Therefore, for the case that our voting rule selects a and b, by Lemma 3.16, we have:

$$\text{dist}(f(\succ), \mathcal{E}; d) \leq 2 < 1 + \sqrt{2}.$$

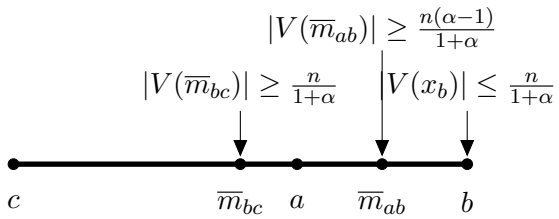


Figure 3.3: The instance for Case 1, when the optimal candidates are both located on x_b and $d(c, a) \geq d(a, b)$. The output committee is $\{a, c\}$. In the new metric d', the voters are distributed across three locations, as illustrated in the figure.

Now consider the case that f has chosen a and c. By the construction of Algorithm 3.3, this happens when $|V_{c\succ b}| > n/(1+\alpha)$ and $|V_{b\succ a}| < n/(1+\alpha)$. Therefore, by Observation 3.15, we have that $|V((-\infty, \overline{m}_{bc}))| > n/(1+\alpha)$, and $|V((-\infty, \overline{m}_{ab}))| \geq n\alpha/(1+\alpha)$. The distortion of f for instance $\mathcal{E}$ in this case is:

$$\mathrm{dist}(f(\succ), \mathcal{E}; d) = \frac{\mathrm{SC}(a) + \mathrm{SC}(c)}{2\mathrm{SC}(b)}.$$

Note that for this case, based on Algorithm 3.2, the Focal point would be the point x_b. Here, again, we consider two cases. The first is when x_a resides between $\overline{m}_{bc}$ and x_b, which means $d(c, a) \geq d(a, b)$. Consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and metric d', where there are at least $n/(1+\alpha)$ voters on $\overline{m}_{bc}$, at least $n(\alpha-1)/(1+\alpha)$ voters on $\overline{m}_{ab}$, and at most $n/(1+\alpha)$ voters on x_b, see Figure 3.3.

Since $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 1+\alpha$, $|V((-\infty, \overline{m}_{bc}))| > n/(1+\alpha)$, $|V((-\infty, \overline{m}_{ab}))| \geq n\alpha/(1+\alpha)$, and $x^* = x_b$, based on Lemma 3.12, we have

$$\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 1 + \alpha.$$

Now it remains to calculate distortion of committee $\{a, c\}$ in $\mathcal{E}'$. For brevity, let $r = d(a, \overline{m}_{bc})$, $s = d(\overline{m}_{ab}, a)$. Note that $d(\overline{m}_{ab}, b) = s$ and $d(\overline{m}_{bc}, c) = 2s + r$. We can calculate $\mathrm{dist}(f(\succ), \mathcal{E}'; d')$ as follows:

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a; d') + \mathrm{SC}(c; d')}{2\mathrm{SC}(b; d')} \\
&= \frac{\frac{2s}{1+\alpha} + \frac{s(\alpha-1)}{1+\alpha} + \frac{r}{1+\alpha} + (2s+r) + \frac{(r+s)\alpha}{1+\alpha} + \frac{s}{1+\alpha}}{2(\frac{s\alpha}{1+\alpha} + \frac{r+s}{1+\alpha})} \\
&= \frac{4s + 2r}{2(s + \frac{r}{1+\alpha})} \\
&\leq 1 + \alpha \\
&= 1 + \sqrt{2}.
\end{aligned}$$

This completes the proof for the subcase that x_a resides between $\overline{m}_{bc}$ and x_b.

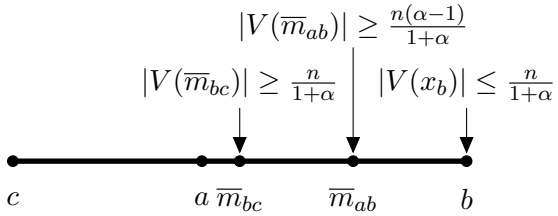


Figure 3.4: A modified instance for Case 1, when the optimal candidates are x_b and $d(c,a) < d(a,b)$. The output committee is $\{a,c\}$. In the new metric d', voters are distributed across three locations, as illustrated in the figure.

In the other subcase, the output of our voting rule is the committee $\{a,c\}$, but x_a is located on the left side of $\overline{m}_{bc}$, indicating that $d(c,a) < d(a,b)$. We again consider the instance $\mathcal{E}' = (V, C, 2, \succ')$ and metric d', where there are at least $n/(1+\alpha)$ voters on $\overline{m}_{bc}$, at least $n(\alpha-1)/(1+\alpha)$ voters on $\overline{m}_{ab}$, and at most $n/(1+\alpha)$ voters on x_b, see Figure 3.4. Since $\text{dist}(f(\succ), \mathcal{E}; d) > 1+\alpha$, also $|V((-\infty, \overline{m}_{bc}))| > n/(1+\alpha)$, $|V((-\infty, \overline{m}_{ab}))| \geq n\alpha/(1+\alpha)$, and $x^* = x_b$, based on Lemma 3.12, we have

$$\text{dist}(f(\succ), \mathcal{E}'; d') > 1+\alpha.$$

Now it remains to calculate distortion of committee $\{a,c\}$ in $\mathcal{E}'$ and d'. For brevity, let $s = d(\overline{m}_{ab}, b)$, and $r = d(\overline{m}_{bc}, a)$. One can conclude that $d(a, \overline{m}_{ab}) = s$, and $d(c, \overline{m}_{bc}) = 2s - r$. Therefore,

$$\begin{aligned}
\text{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\text{SC}(a; d') + \text{SC}(c; d')}{2\text{SC}(b; d')} \\
&\leq \frac{\left(r + (s-r)(\frac{\alpha}{1+\alpha}) + s(\frac{1}{1+\alpha})\right) \times 2 + (2s - 2r)}{2(s(\frac{\alpha}{1+\alpha}) + (s-r)\frac{1}{1+\alpha})} \\
&= \frac{(s-r)\frac{\alpha}{1+\alpha} + \frac{s}{1+\alpha} + s}{s(\frac{\alpha}{1+\alpha}) + (s-r)\frac{1}{1+\alpha}} \\
&\leq \frac{(s-r) + s\alpha}{(s-r)\frac{1}{1+\alpha} + s(\frac{\alpha}{1+\alpha})} \\
&= 1+\alpha \\
&= 1+\sqrt{2}.
\end{aligned}$$

This completes the proof for this case.

Case.2 $(x_{o_1}, x_{o_2}) = (x_c, x_c)$.

Our voting rule first selects a and then chooses between c and b. The case that $\{a,c\}$ are selected is similar to Case 1 and results in distortion at most 2. For the case that $\{a,b\}$ is selected, we know that $|V_{c \succ b}| \leq n/(1+\alpha)$. Furthermore, by

Corollary 3.5, $\mathfrak{m}$ must lie between c and a. Thus, by Lemma 3.4, $\mathrm{SC}(a) \leq \mathrm{SC}(b)$. Therefore, the distortion of instance $\mathcal{E}$ in this case is

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}; d) &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b)}{2\mathrm{SC}(c)} \\ &\leq \frac{2\mathrm{SC}(b)}{2\mathrm{SC}(c)} \\ &= \frac{\mathrm{SC}(b)}{\mathrm{SC}(c)}.\end{aligned}$$

Since $|V_{c\succ b}| \leq n/(1+\alpha)$, we have $|V_{b\succ c}| > n\alpha/(1+\alpha)$. Thus, based on Lemma 3.7,

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}; d) &\leq \frac{\mathrm{SC}(b)}{\mathrm{SC}(c)} \\ &\leq \frac{2n}{\frac{n\alpha}{1+\alpha}} - 1 \\ &= 1 + \frac{2}{\alpha} \\ &= 1 + \sqrt{2}.\end{aligned}$$

Case.3 $(x_{o_1}, x_{o_2}) \in \{(x_a, x_b), (x_b, x_a)\}$.

The outcome would be optimal if our voting rule chooses b over c. Thus, assume that the output of our voting rule is $\{a, c\}$. By the construction of Algorithm 3.3, this happens when $|V_{c\succ b}| > n/(1+\alpha)$. Therefore, by Observation 3.15, we have that $|V((-\infty, \overline{m}_{bc}))| > n/(1+\alpha)$. In this instance, the distortion would be

$$\mathrm{dist}(f(\succ), \mathcal{E}; d) = \frac{\mathrm{SC}(a) + \mathrm{SC}(c)}{\mathrm{SC}(a) + \mathrm{SC}(b)}.$$

Note that for this case, based on Algorithm 3.2, the Focal point would be the point x_b. Consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and metric d', where there are at least $n/(1+\alpha)$ voters on $\overline{m}_{bc}$, and at most $n\alpha/(1+\alpha)$ voters on x_b, see Figure 3.5. Since $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 1+\alpha$, $|V((-\infty, \overline{m}_{bc}))| > n/(1+\alpha)$, and $x^* = x_b$, based on Lemma 3.12, we can conclude that

$$\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 1 + \alpha.$$

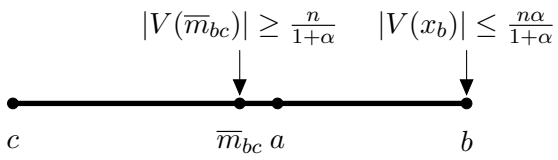


Figure 3.5: A modified instance for Case 3. The optimal candidates are located on (x_a, x_b). The output committee is $\{a, c\}$. In the new metric d', the voters are distributed across two locations, as illustrated in the figure.

Now, it remains to calculate the distortion of committee $\{a, c\}$. For brevity, let $s = d(a, b)$, and $r = d(\overline{m}_{bc}, a)$. Note that $d(c, \overline{m}_{bc}) = r + s$. Therefore,

$$\begin{aligned}
\operatorname{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a; d') + \mathrm{SC}(c; d')}{\mathrm{SC}(a; d') + \mathrm{SC}(b; d')} \\
&\leq \frac{s\alpha + r + (r+s)(2\alpha+1)}{s\alpha + r + r + s} \\
&= 1 + \frac{2(r+s)\alpha}{s(\alpha+1) + 2r} \\
&\leq 1 + \frac{2(r+s)\alpha}{2s + 2r} \\
&= 1 + \alpha \\
&= 1 + \sqrt{2},
\end{aligned}$$

which completes the proof.

Case.4 $(x_{o_1}, x_{o_2}) \in \{(x_a, x_c), (x_c, x_a)\}$.

The outcome would be optimal if our voting rule chooses c over b. Thus, assume that the output of our voting rule is $\{a, b\}$. By the construction of the Algorithm 3.3, this happens either when $|V_{c\succ b}| \leq n/(1+\alpha)$, or when $|V_{c\succ b}| > n/(1+\alpha)$ and $|V_{b\succ a}| \geq n/(1+\alpha)$. For now, let us consider the first subcase that $|V_{c\succ b}| \leq n/(1+\alpha)$.

In this instance, the distortion would be:

$$\begin{aligned}
\operatorname{dist}(f(\succ), \mathcal{E}; d) &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b)}{\mathrm{SC}(a) + \mathrm{SC}(c)} \\
&\leq \frac{\mathrm{SC}(b)}{\mathrm{SC}(c)} \\
&\leq \frac{2n}{\frac{n\alpha}{1+\alpha}} - 1 \\
&= 1 + \frac{2}{\alpha} \\
&= 1 + \sqrt{2}.
\end{aligned}$$

Note that the second inequality follows from Lemma 3.7, since $|V_{b\succ c}| \geq n\alpha/(1+\alpha)$.

Now consider the second subcase, where $|V_{c\succ b}| > n/(1+\alpha)$, and also $|V_{b\succ a}| \geq n/(1+\alpha)$. By Observation 3.15, it follows that $|V([\overline{m}_{ab}, \infty))| \geq n/(1+\alpha)$. Furthermore,

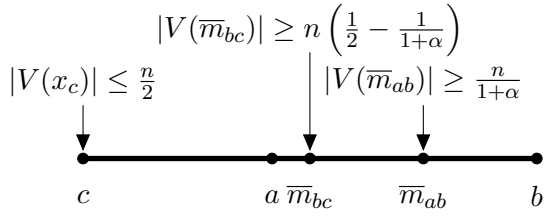


Figure 3.6: A modified instance for Case 4, when the optimal candidates are located on (x_a, x_c), $|V_{c\succ b}| > n/(1+\alpha)$, and also $|V_{b\succ a}| \geq n/(1+\alpha)$. The output committee is $\{a, b\}$. In the new metric d', the voters are distributed across three locations, as illustrated in the figure.

since $b \succ_{\mathfrak{m}} c$, thus $|V([\overline{m}_{bc}, \infty))| \geq n/2$. In this case, the output of our voting rule is $\{a, b\}$, and based on Algorithm 3.2, the Focal point would be the point x_c. Consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and metric d', where $|V(x_c)| \leq n/2$, $|V(\overline{m}_{ab})| \geq n/(1+\alpha)$, and $|V(\overline{m}_{bc})| \geq n\,(1/2 - 1/(1+\alpha))$, see Figure 3.6. Since $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 1 + \alpha$, $|V([\overline{m}_{ab}, \infty))| \geq n/(1+\alpha)$, $|V([\overline{m}_{bc}, \infty))| \geq n/2$, and $x^* = x_c$, based on Lemma 3.12, we have

$$\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 1 + \alpha.$$

Now it remains to calculate distortion of committee $\{a, b\}$ in $\mathcal{E}'$. For brevity, let $r = d(a, c)$, $t = d(a, \overline{m}_{bc})$, and $s = d(\overline{m}_{bc}, \overline{m}_{ab})$. Note that $d(\overline{m}_{ab}, b) = s + t$. Therefore,

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a; d') + \mathrm{SC}(b; d')}{\mathrm{SC}(a; d') + \mathrm{SC}(c; d')} \\
&\leq \frac{n(s+t) + n(\frac{1}{2} - \frac{1}{1+\alpha})s + \frac{n}{2}(r+s+t) + \frac{nr}{2} + \frac{nt}{2} + \frac{ns}{1+\alpha}}{\frac{nr}{2} + \frac{nt}{2} + \frac{ns}{1+\alpha} + \frac{n(r+t)}{2} + \frac{ns}{1+\alpha}} \\
&\leq \frac{2s + 2t + r}{r + t + \frac{2s}{1+\alpha}} \\
&\leq 1 + \alpha \\
&= 1 + \sqrt{2}.
\end{aligned}$$

This completes the proof.

□

We now show that any 2-winner voting rule in the line metric, under a utilitarian additive social cost function, must have a distortion of at least $1+\sqrt{2}$. This result matches our upper bound, making the distortion bound tight for 2-winner voting.

Theorem 3.17. *Any 2-winner voting rule in the line metric and under additive social cost, has a distortion of at least* $1 + \sqrt{2}$.

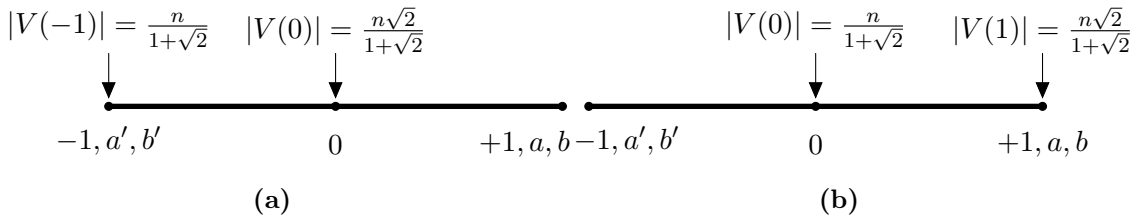


Figure 3.7: Instance $\mathcal{E}$ with two metrics d_1 (a) and d_2 (b), used for proving the lower bound in 2-winner election. a, b, a', and b' are the candidates distributed on locations -1 and $+1$.

Proof. We construct an instance $\mathcal{E} = (V, C, k, \succ)$ with two different metrics d_1 and d_2. Let X be a subset of voters of size $n_1 = n/(1+\sqrt{2})$ whose preference profile is $a' \succ b' \succ a \succ b$. Similarly, let Y be a subset of voters of size $n_2 = n\sqrt{2}/(1+\sqrt{2})$ with the preference profile $a \succ b \succ a' \succ b'$. In both metrics, assume that candidates a and b reside at point $x = 1$, and candidates a' and b' reside at point $x = -1$. In d_1, locate the voters in X at point $x = -1$, and the remaining n_2 voters in Y at point $x = 0$, as shown in Figure 3.7(a). In d_2, locate the voters in X at point $x = 0$, and the remaining n_2 voters in Y at point $x = 1$, as shown in Figure 3.7(b).

Note that in both instances, the preference profile $\succ$ is consistent with the corresponding metric. If we apply a voting rule f to $\succ$, we can consider three cases for the output:

- If f selects a' and b', distortion with respect to d_2 would be:

$$\begin{aligned}\operatorname{dist}(f(\succ), \mathcal{E}; d_2) &\geq \frac{\operatorname{SC}(\{a', b'\}; d_2)}{\operatorname{SC}(\{a, b\}; d_2)} \\ &= \frac{\operatorname{SC}(a'; d_2) + \operatorname{SC}(b'; d_2)}{\operatorname{SC}(a; d_2) + \operatorname{SC}(b; d_2)} \\ &= \frac{2n_1 + 4n_2}{2n_1} \\ &= 1 + 2\sqrt{2}.\end{aligned}$$

- If f selects a and b, distortion with respect to d_1 would be:

$$\begin{aligned}\operatorname{dist}(f(\succ), \mathcal{E}; d_1) &\geq \frac{\operatorname{SC}(\{a, b\}; d_1)}{\operatorname{SC}(\{a', b'\}; d_1)} \\ &= \frac{\operatorname{SC}(a; d_1) + \operatorname{SC}(b; d_1)}{\operatorname{SC}(a'; d_1) + \operatorname{SC}(b'; d_1)} \\ &= \frac{4n_1 + 2n_2}{2n_2} \\ &= 1 + \sqrt{2}.\end{aligned}$$

- If f selects a and a', distortion with respect to d_1 would be:

$$\begin{aligned}\operatorname{dist}(f(\succ), \mathcal{E}; d_1) &\geq \frac{\operatorname{SC}(\{a, a'\}; d_1)}{\operatorname{SC}(\{a', b'\}; d_1)} \\ &= \frac{\operatorname{SC}(a; d_1) + \operatorname{SC}(a'; d_1)}{\operatorname{SC}(a'; d_1) + \operatorname{SC}(b'; d_1)} \\ &= \frac{2n_1 + 2n_2}{2n_1} \\ &= 1 + \sqrt{2}.\end{aligned}$$

Therefore, for every possible output, the distortion in at least one of the two instances would be greater than or equal to $1 + \sqrt{2}$.

□

3.6 General Multi-Winner Voting

In this section, we generalize our voting rule to k-winner voting rules on the line metric under a utilitarian additive social cost function. We establish both upper and lower bounds on the distortion for these rules. First, we present in Theorem 3.18 an upper bound on the distortion of k-winner voting rules, which is approximately $7/3$ but may be slightly higher when k is not divisible by 3. Additionally, we derive two lower bounds. For $k \geq m/2$, the distortion of any k-winner voting rule is lower bounded by $1 + (m-k)/(3k-m)$. For $k \leq m/2$, the distortion is lower bounded by $2 + 1/k$ when k is odd and $1 + \sqrt{1 + 2/k}$ when k is even. These results provide an overview of distortion behavior for k-winner voting rules across various values of k.

Theorem 3.18. *For every integer $k > 2$, there exists a k-winner voting rule f such that for any election instance $\mathcal{E} = (V, C, k, \succ)$ where voters and candidates lie on a line metric, and for any metric d consistent with $\succ$, the distortion of f under the utilitarian additive social cost satisfies:*

$$\operatorname{dist}(f) \leq \begin{cases} 7/3 & \text{for } k \equiv 0 \pmod 3, \\ 7/3 + \frac{4(\sqrt{2}-4/3)}{k} & \text{for } k \equiv 1 \pmod 3, \\ 7/3 + \frac{2(\sqrt{2}-4/3)}{k} & \text{for } k \equiv 2 \pmod 3. \end{cases}$$

Before proving this theorem, we demonstrate how combining two multi-winner voting rules, each selecting different committee sizes, yields a k-winner voting rule with low distortion in Subsection 3.6.1. Subsequently, in Subsection 3.6.2, we generalize the Polar Comparison Rule for $k = 3$, to achieve a low distortion.

3.6.1 Combining Voting Rules

In this subsection, we present a method for combining different voting rules to construct a k-winner selection rule. We show that the distortion of the combined rule can be expressed as a function of the distortions of the individual voting rules.

Theorem 3.19. *Given integers $k, k_1, k_2, \alpha, \beta$ where $k = \alpha k_1 + \beta k_2$, if f_1 and f_2 are respectively k_1-winner and k_2-winner voting rules, with distortions at most τ_1 and τ_2, there exists a k-winner voting rule with distortion at most $(\alpha k_1 \tau_1 + \beta k_2 \tau_2)/k$.*

Proof. For the simpler case where $\beta = 0$, by running f_1 for k/k_1 iterations, each time removing the candidates that were selected as winners, we can achieve a distortion of at most τ_1 for the k-winner election, as shown by A. Goel, Hulett, and Krishnaswamy, 2018.

For the general case, we use a similar approach: assuming $\tau_2 \leq \tau_1$ without loss of generality, we fix an instance $\mathcal{E} = (V, C, k, \succ)$. Define voting rule f as follows: First, run f_1 iteratively to select αk_1 candidates $S_1 = \{a_1, \dots a_{\alpha k_1}\}$. Subsequently, apply f_2 iteratively on the remaining candidates $C \backslash S_1$, to select βk_2 candidates $S_2 = \{b, \dots, b_{\beta k_2}\}$. We prove that this voting rule, f, has the desired distortion for k-winner elections.

Recall that $\mathrm{OPT} = \{o_1, o, \dots, o_k\}$ is the optimal committee of size k, and $\mathrm{SC}(o_i) \leq \mathrm{SC}(o_j)$ for all $i \leq j$. Partition this set into two subsets $\mathrm{OPT}_1 = \{o_i \mid 1 \leq i \leq \alpha k_1\}$ and $\mathrm{OPT}_2 = \{o_j \mid \alpha k_1 < j \leq k\}$. By the assumptions of the theorem, we know that

$$\mathrm{SC}(S_1) \leq \tau_1 \mathrm{SC}(\mathrm{OPT}_1). \tag{3.4}$$

Furthermore, when running f_2, αk_1 candidates have been eliminated from C. Thus, if OPT_3 is the optimal committee of size βk_2 in the instance given to f_2, $\mathrm{SC}(\mathrm{OPT}_3) \leq \mathrm{SC}(\mathrm{OPT}_2)$. By the assumptions of the theorem, $\mathrm{SC}(S_2) \leq \tau_2 \mathrm{SC}(\mathrm{OPT}_3)$. Therefore,

$$\mathrm{SC}(S_2) \leq \tau_2 \mathrm{SC}(\mathrm{OPT}_2). \tag{3.5}$$

Using equations (3.4) and (3.5), we can conclude that

$$\begin{aligned}
\mathrm{SC}(S_1) + \mathrm{SC}(S_2) &\leq \tau_1 \mathrm{SC}(\mathrm{OPT}_1) + \tau_2 \mathrm{SC}(\mathrm{OPT}_2) \\
&= \tau_2(\mathrm{SC}(\mathrm{OPT}_1) + \mathrm{SC}(\mathrm{OPT}_2)) + (\tau_1 - \tau_2)\mathrm{SC}(\mathrm{OPT}_1) \\
&\leq \tau_2 \mathrm{SC}(\mathrm{OPT}) + (\tau_1 - \tau_2)\frac{\alpha k_1 \mathrm{SC}(\mathrm{OPT})}{k} \\
&= \frac{\beta k_2 \tau_2 + \alpha k_1 \tau_1}{k} \mathrm{SC}(\mathrm{OPT}),
\end{aligned}$$

which completes the proof. □

3.6.2 Polar Comparison Rule for $k = 3$

In this subsection, we establish an upper bound of 7/3 for distortion in the case $k = 3$. We show that this bound is tight, matching the lower bound derived in Theorem 3.21. We first define a voting rule by generalizing Polar Comparison rule for $k = 3$, and then prove the 7/3 upper bound on its distortion in Theorem 3.20.

In Algorithm 3.4, we present a voting rule for 3-winner elections, referred to as the *Polar Comparison Rule* for the case $k = 3$. In this voting rule, the first two members of the committee are a and b, the top two candidates in the Majority order. To select the third member, we compare the two candidates adjacent to a and b, denoted by b' and c. Without loss of generality, assume that b lies on the right side of a. Note that we can obtain the ordering of candidates that are not Pareto-dominated by any candidate other than a using Lemma 3.1. Therefore, we can find the closest candidate to the left side of

Input : Instance $\mathcal{E} = (V, C, 3, \succ)$
Output : A Committee of three candidates $S = \{s_1, s_2, s_3\}$

$a, b \leftarrow$ the top two candidates in the Majority order $b \leftarrow \{e \in C \mid \exists e' \in C \setminus \{e, a\} : |V_{e' \succ e}| = n\}$;
// Candidates Pareto-dominated by $e' \neq a$
$b' \leftarrow \{e \in C \mid \exists e' \in C \setminus \{e, b\} : |V_{e' \succ e}| = n\}$;
// Candidates Pareto-dominated by $e' \neq b$
$C'_1 \leftarrow C \setminus (b \cup \{a\})$ $C'_2 \leftarrow C \setminus (b' \cup \{b\})$ Find $c \in C'_1$ minimizing $d(c, a)$ such that a lies between c and b ;
// Using Lemma 3.1
Find $b' \in C'_2$ minimizing $d(b', b)$ such that b lies between a and b' ;
// Using Lemma 3.1
if *c does not exist* **then**
  return $\{a, b, b'\}$
end
if *b' does not exist* **then**
  return $\{a, b, c\}$
end
if $|V_{c \succ b'}| \geq \frac{2n}{5}$ **then**
  return $\{a, b, c\}$
else
  return $\{a, b, b'\}$
end

Algorithm 3.4: Polar Comparison Rule for 3-Winner Elections

a, namely c, and the closest candidate to the right of b, namely b'. As stated before, the goal is to either select $\{a, b, b'\}$ or $\{a, b, c\}$ as the output committee. Therefore, similar to Polar Comparison rule for 2-winner elections described in Algorithm 3.3, we compare c and b' to select the final candidate. Specifically, if $|V_{c \succ b'}| \geq 2n/5$, we choose c, otherwise, we choose b'.

Theorem 3.20. *If f is the voting rule that selects a committee of size three by Algorithm 3.4, then* $\text{dist}(f) \leq 7/3$.

Proof. To prove the theorem, we begin by assuming the contrary, i.e., there exists an instance $\mathcal{E} = (V, C, 3, \succ)$ and metric d, such that $\text{dist}(f(\succ), \mathcal{E}; d) > 7/3$. Let a and b be the first and second candidates in the Majority order of $\mathcal{E}$, respectively. Without loss of generality, assume that $x_b \geq x_a$. Moreover, let c be the closest candidate to the left side of a, and let b' be the closest candidate to the right side of b, i.e., $x_c \leq x_a \leq x_b \leq x_{b'}$.

The voting rule f, defined by Algorithm 3.4, first outputs candidates a and b, and then selects one between b' and c. Considering different cases for these two candidates and the optimal committee, we compute the distortion of our voting rule for each case separately and prove that it cannot exceed $7/3$.

Case.1 Candidate c defeats b' This implies that $|V_{c \succ b'}| \geq n/2$. Thus, b' appears later than c in the Majority order. Since $|V_{c \succ b}| \geq n/2$, the output of Algorithm 3.4 would be $\{a, b, c\}$. Therefore,

$$\text{dist}(f(\succ), \mathcal{E}) = \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(c)}{\text{SC}(o_1) + \text{SC}(o_2) + \text{SC}(o_3)}.$$

Note that a might be in the optimal committee OPT or not. If $a \notin \text{OPT}$, by Corollary 3.8 $\text{SC}(a) \leq 3\text{SC}(o_3)$. Moreover, the median voter $\mathfrak{m}$, lies between c and b. Therefore, if we hypothetically remove a from the candidates, by Lemma 3.16, $\text{SC}(b) + \text{SC}(c) \leq 2(\text{SC}(o_1) + \text{SC}(o_2))$. Thus,

$$\begin{aligned}
\text{dist}(f(\succ), \mathcal{E}) &\leq \frac{3\text{SC}(o_1) + 2\text{SC}(o_2) + 2\text{SC}(o_3)}{\text{SC}(o_1) + \text{SC}(o_2) + \text{SC}(o_3)} \\
&\leq \frac{5\text{SC}(o_1) + 2\text{SC}(o_2)}{2\text{SC}(o_1) + \text{SC}(o_2)} \\
&\leq \frac{5\text{SC}(o_1) + 2\text{SC}(o_2)}{(2 + 1/7)\text{SC}(o_1) + (1 - 1/7)\text{SC}(o_2)} \\
&\leq \frac{7}{3}.
\end{aligned}$$

If $a \in \text{OPT}$, again by eliminating a from the candidates and using Lemma 3.16, we have

$$\begin{aligned}
\text{dist}(f(\succ), \mathcal{E}) &= \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(c)}{\text{SC}(o_1) + \text{SC}(o_2) + \text{SC}(o_3)} \\
&\leq \frac{\text{SC}(o_1) + 2\left(\text{SC}(o_2) + \text{SC}(o_3)\right)}{\text{SC}(o_1) + \text{SC}(o_2) + \text{SC}(o_3)} \\
&\leq 2,
\end{aligned}$$

which completes the proof of this case.

Case.2 Candidate b' defeats c This means that $|V_{b' \succ c}| \geq n/2$. Thus, b' appears earlier than c in the Majority order. In this case, similar to the proof for $k = 2$, we consider different cases for the optimal committee $\text{OPT} = \{o_1, o_2, o_3\}$. Note that by Lemma 3.4, we only need to consider these cases for the optimal committee:

$$(x_{o_1}, x_{o_2}, x_{o_3}) \in \{(x_a, x_b, x_{b'}), (x_a, x_b, x_c), (x_a, x_c, x_c), (x_c, x_c, x_c), (x_b, x_{b'}, x_{b'})\}.$$

We consider each of these cases and prove the upper bound of 7/3 separately.

Case.2.1 $(x_{o_1}, x_{o_2}, x_{o_3}) = (x_a, x_b, x_c)$.

Our voting rule, f, first selects a, b and then chooses between b' and c. If the output is $\{a, b, c\}$, that would be the optimal committee with distortion 1. Otherwise, the output of f is $\{a, b, b'\}$. Therefore, the distortion can be expressed as

$$\text{dist}(f(\succ), \mathcal{E}) = \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(b')}{\text{SC}(a) + \text{SC}(b) + \text{SC}(c)}.$$

Note that by construction of the voting rule, $|V_{c \succ b'}|$ must not be greater than $2n/5$. Therefore, $|V_{b' \succ c}| \geq 3n/5$. Hence by Lemma 3.7, $\text{SC}(b') \leq 7/3\text{SC}(c)$. Therefore,

$$\text{SC}(a) + \text{SC}(b) + \text{SC}(b') \leq \frac{7}{3}\left(\text{SC}(a) + \text{SC}(b) + \text{SC}(c)\right).$$

This completes the proof for this case.

Case.2.2 $(x_{o_1}, x_{o_2}, x_{o_3}) = (x_a, x_c, x_c)$.

Similar to the previous cases, f chooses a candidate between c and b'. If c is selected, then,

$$\text{dist}(f(\succ), \mathcal{E}) = \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(c)}{\text{SC}(a) + 2\text{SC}(c)}.$$

Since b appears earlier than c in the Majority order, by Corollary 3.8, $\text{SC}(b) \leq 3\text{SC}(c)$. Therefore, it can be concluded that the above distortion is not greater than 2. If c is not selected,

$$\text{dist}(f(\succ), \mathcal{E}) = \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(b')}{\text{SC}(a) + 2\text{SC}(c)}.$$

By the construction of the voting rule, this happens when $|V_{b' \succ c}| \geq 3n/5$. Similar to the previous case, using Lemma 3.7, we can conclude that $\text{SC}(b') \leq 7/3\text{SC}(c)$. By Lemma 3.4, $\text{SC}(b) \leq \text{SC}(b')$. Therefore, $\text{SC}(b) \leq 7/3\text{SC}(c)$. These two facts demonstrate that the total distortion will not exceed 7/3.

Case.2.3 $(x_{o_1}, x_{o_2}, x_{o_3}) = (x_c, x_c, x_c)$.

The output of the voting rule is either $\{a, b, b'\}$ or $\{a, b, c\}$. Similar to the previous cases, the first output occurs when $|V_{b' \succ c}| \geq 3n/5$, which shows that $\text{SC}(b') \leq 7/3\text{SC}(c)$ by Lemma 3.7. Since $a \notin \text{OPT}$ and $c \in \text{OPT}$, by Lemma 3.4, one can conclude that a and c lie on different sides of $\mathfrak{m}$, the median voter. Therefore, again using Lemma 3.4, $\text{SC}(a) \leq \text{SC}(b) \leq \text{SC}(b')$. This implies that in this case,

$$\begin{aligned}
\text{dist}(f(\succ), \mathcal{E}) &= \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(b')}{3\text{SC}(c)} \\
&\leq \frac{3\text{SC}(b')}{3\text{SC}(c)} \\
&\leq \frac{7}{3}.
\end{aligned}$$

Now consider the case that the output of f is $\{a, b, c\}$. Using Corollary 3.8, one can conclude that $\text{SC}(a) \leq 3\text{SC}(c)$ and also $\text{SC}(b) \leq 3\text{SC}(c)$. Therefore,

$$\begin{aligned}
\text{dist}(f(\succ), \mathcal{E}) &= \frac{\text{SC}(a) + \text{SC}(b) + \text{SC}(c)}{3\text{SC}(c)} \\
&\leq \frac{3\text{SC}(c) + 3\text{SC}(c) + \text{SC}(c)}{3\text{SC}(c)} \\
&= \frac{7}{3}.
\end{aligned}$$

Thus, the proof of this case is completed.

Case.2.4 $(x_{o_1}, x_{o_2}, x_{o_3}) = (x_a, x_b, x_{b'})$.

Our voting rule, f, first selects a and b, and then chooses between c and b'. In the case where f selects a, b, and b', the output is the optimal committee with distortion 1. Now consider the case that f selects c. By the construction of

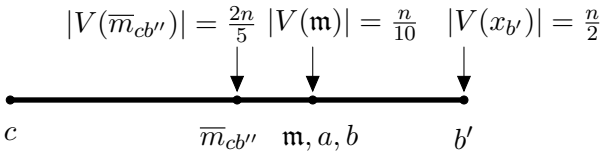


Figure 3.8: A modified instance for Case 2.4. The optimal committee is $\{a, b, b'\}$, and the output committee selected by f is $\{a, b, c\}$. Candidates a and b are moved to $\mathfrak{m}$, the median voter. Moreover, the voters are distributed across three locations, $x_{\mathfrak{m}}, x_{b'}$, and $\overline{m}_{cb''}$ as illustrated in the figure.

Algorithm 3.4, this happens when $|V_{c\succ b'}| \geq 2n/5$. Therefore, by Observation 3.15, it follows that $|V((-\infty, \overline{m}_{cb''}])| > 2n/5$. The distortion of f for instance $\mathcal{E}$ in this case is

$$\mathrm{dist}(f(\succ), \mathcal{E}) = \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(c)}{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(b')}. \tag{3.6}$$

Note that for this case, based on Algorithm 3.2, the Focal point would be $x_{b'}$.

Suppose that we move a and b to $x_{\mathfrak{m}}$. Then, by Lemma 3.4, their social cost would be reduced, which results in an increase in the distortion. Thereafter, consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and metric d', where $|V(\overline{m}_{cb''})| = 2n/5$, $|V(\mathfrak{m})| = n/10$, and $|V(x_{b'})| = n/2$, see Figure 3.8. Note that $x^* = x_{b'}$, and $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 7/3$. Moreover,

$$|V((-\infty, \overline{m}_{cb''}])| > 2n/5,$$
$$|V((-\infty, \mathfrak{m}])| \geq n/2.$$

Based on Lemma 3.12, it follows that $\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 7/3$.

Now it remains to calculate the distortion of the committee $\{a, b, c\}$ in $\mathcal{E}'$. For brevity, let $r = d(\overline{m}_{cb''}, x_{\mathfrak{m}})$, and $s = d(x_{\mathfrak{m}}, x_{b'})$. Note that $d(c, \overline{m}_{cb''}) = r + s$. We can calculate $\mathrm{dist}(f(\succ), \mathcal{E}'; d')$ as follows:

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(c)}{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(b')} \\
&= 1 + \frac{r + s + \frac{r}{10} + \frac{r+s}{2} - (\frac{s}{2} + \frac{2r}{5})}{(\frac{s}{2} + \frac{2r}{5}) \times 2 + \frac{s}{2} + \frac{2r}{5}} \\
&= 1 + \frac{\frac{6r}{5} + s}{\frac{6r}{5} + 1.5s} \\
&\leq 2.
\end{aligned}$$

This completes the proof for this case.

Case.2.5 $(x_{o_1}, x_{o_2}, x_{o_3}) = (x_b, x_{b'}, x_{b'})$.

As stated before, f first selects a and b, and then chooses between c and b'. If the voting rule outputs $\{a, b, b'\}$, $\mathrm{SC}(a) \leq 3\mathrm{SC}(b)$ and $\mathrm{SC}(b') \geq \mathrm{SC}(b)$. As a result, the distortion would be:

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}) &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(b')}{\mathrm{SC}(b) + 2\mathrm{SC}(b')} \\ &\leq \frac{4\mathrm{SC}(b) + \mathrm{SC}(b')}{2\mathrm{SC}(b) + \mathrm{SC}(b')} \\ &\leq \frac{5}{3},\end{aligned}$$

which is less than $7/3$. Conversely, if f selects c, by the construction of Algorithm 3.4, this happens when $|V_{c\succ b'}| \geq 2n/5$. Therefore, by Observation 3.15, it follows that $|V((-\infty, \overline{m}_{cb''}])| \geq 2n/5$. Furthermore, since b is the optimal candidate, by Corollary 3.5, the median voter $\mathfrak{m}$ must lie between a and b. The distortion of f in this case is

$$\mathrm{dist}(f(\succ), \mathcal{E}) = \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(c)}{\mathrm{SC}(b) + 2\mathrm{SC}(b')}. \tag{3.7}$$

Note that for this case, based on Algorithm 3.2, the Focal point would be the point $x_{b'}$.

Here we consider two cases. The first one is when $\overline{m}_{cb''}$ is between x_a and $x_{\mathfrak{m}}$.

In the first case, note that since a appears earlier than b in the Majority order, $d(a, \mathfrak{m}) \leq d(b, \mathfrak{m})$. Furthermore, suppose that we move b closer to $\mathfrak{m}$ until $d(b, \mathfrak{m}) = d(a, \mathfrak{m})$. By Lemma 3.4, this move does not decrease $\mathrm{SC}(b)$. Now, consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and a metric d' where $|V(\overline{m}_{cb''})| = 2n/5, |V(\mathfrak{m})| = n/10$, and $|V(x_{b'})| = n/2$, see Figure 3.9(a). Note that $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 7/3$ and $x^* = x_{b'}$. Moreover,

$$\begin{aligned}&|V((-\infty, \overline{m}_{cb''}])| > 2n/5, \\ &|V((-\infty, \mathfrak{m}])| \geq n/2.\end{aligned}$$

Therefore, by Lemma 3.12, it follows that $\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 7/3$.

Now it remains to calculate the distortion of committee $\{a, b, c\}$ in $\mathcal{E}'$. For brevity, let $t = d(c, a)$, $r = d(a, \overline{m}_{cb''})$, and $s = d(\overline{m}_{cb''}, x_{\mathfrak{m}})$. Note that $d(x_{\mathfrak{m}}, x_b) = r + s$, and $d(b, b') = t - 2s$. We can calculate $\mathrm{dist}(f(\succ), \mathcal{E}'; d')$ as follows:

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(c)}{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(b')} \\
&= \frac{r + \frac{s}{10} + \frac{t+r}{2} + \frac{s+r}{2} + \frac{2s}{5} + \frac{t-2s}{2} + r + t + \frac{s}{10} + \frac{t+r}{2}}{(\frac{s+r}{2} + \frac{2s}{5} + \frac{t-2s}{2}) \times 3} \\
&= \frac{1}{3} + \frac{2t + 3r + \frac{s}{5}}{\frac{3}{2} \times (t+r) - \frac{3s}{10}} \\
&= \frac{7}{3} + \frac{-t + \frac{4s}{5}}{\frac{3}{2} \times (t+r) - \frac{3s}{10}} \\
&\leq \frac{7}{3}.
\end{aligned}$$

The last inequality holds because given $d(b, b') \geq 0$, it follows that $t \geq 2s$. This completes the proof for the subcase that $\overline{m}_{cb''}$ resides between a and $x_{\mathfrak{m}}$. In the second subcase, the output of our voting rule is still the committee $\{a, b, c\}$, but $\overline{m}_{cb''}$ is located on the left side of a. Similar to the first case, in this case, $d(a, \mathfrak{m}) \leq d(b, \mathfrak{m})$. Furthermore, we move b closer to $\mathfrak{m}$ until $d(b, \mathfrak{m}) = d(a, \mathfrak{m})$. By Lemma 3.4, this will decrease $\mathrm{SC}(b)$.
Now, consider the instance $\mathcal{E}' = (V, C, 2, \succ')$, and a metric d' where $|V(\overline{m}_{cb''})| = 2n/5$, $|V(\mathfrak{m})| = n/10$, and $|V(x_{b'})| = n/2$, see Figure 3.9(b). Note that $\mathrm{dist}(f(\succ), \mathcal{E}; d) > 7/3$ and $x^* = x_{b'}$. Moreover,

$$\begin{aligned}
&|V((-\infty, \overline{m}_{cb''}])| > 2n/5, \\
&|V((-\infty, \mathfrak{m}])| \geq n/2.
\end{aligned}$$

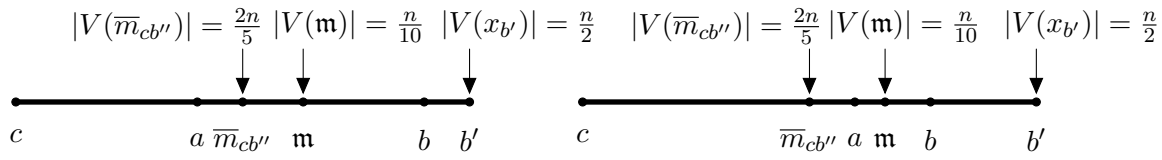
Therefore, based on Lemma 3.12, it follows that $\mathrm{dist}(f(\succ), \mathcal{E}'; d') > 7/3$.
Now it remains to calculate the distortion of committee $\{a, b, c\}$ in $\mathcal{E}'$. For brevity, let $t = d(\overline{m}_{cb''}, a)$, $s = d(a, \mathfrak{m})$, and $r = d(b, b')$. Note that $d(\mathfrak{m}, b) = s$, and $d(c, \overline{m}_{cb''}) = t + 2s + r$. We can calculate $\mathrm{dist}(f(\succ), \mathcal{E}'; d')$ as follows:

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}'; d') &= \frac{\mathrm{SC}(a) + \mathrm{SC}(b) + \mathrm{SC}(c)}{\mathrm{SC}(b) + 2\mathrm{SC}(b')} \\
&= \frac{\frac{2t}{5} + \frac{s}{10} + \frac{2s+r}{2} + (t + 2s + r) + \frac{t+s}{10} + \frac{t+2s+r}{2}}{3 \times (\frac{s+r}{2} + \frac{2(t+s)}{5})} \\
&= \frac{1}{3} + \frac{2t + \frac{21s}{5} + 2r}{\frac{6t}{5} + \frac{27s}{10} + 1.5r} \\
&\leq \frac{1}{3} + \frac{5}{3} \leq 2 < \frac{7}{3}.
\end{aligned}$$

This completes the proof for this case.

□

Now, the proof of Theorem 3.18 can be obtained as a corollary of theorems 3.14, 3.19, and 3.20.



(a) The point $\overline{m}_{cb''}$ lies between a and $\mathfrak{m}$. **(b)** Candidate a lies between $\overline{m}_{cb''}$ and $\mathfrak{m}$.

Figure 3.9: Modified instances for the first (a) and second (b) subcases of Case 2.5. The optimal candidates are located at $(x_b, x_{b'}, x_{b'})$, and the output committee selected by f is $\{a, b, c\}$. The voters are distributed across three locations: $x_{\mathfrak{m}}$, $x_{b'}$, and $\overline{m}_{cb''}$.

Proof of Theorem 3.18. Let $k_1 = 2$ and $k_2 = 3$. Moreover, let f_1 and f_2 be the Polar Comparison rules for $k = 2$ and $k = 3$, respectively. By Theorem 3.14, $\text{dist}(f_1) \leq 1 + \sqrt{2}$, and by Theorem 3.20, $\text{dist}(f_2) \leq 7/3$.

First, note that when $k = 4$, $k = 2k_1$. Therefore, by Theorem 3.19, there exists a voting rule constructed by applying f_1 twice, with distortion at most $1 + \sqrt{2}$.

For general $k > 4$, express k as $k = 3q + p$ with $0 \leq p < 3$ and $k \equiv p \pmod 3$. If $p = 0$, $k = qk_2$. Hence, by Theorem 3.19, there exists a k-winner voting rule with distortion at most $7/3$. If $p = 1$, then $k = (q-1)k_2 + 2k_1$. Applying Theorem 3.19, results in a k-winner voting rule with distortion at most

$$\frac{3(q-1) \times 7/3 + 4(1+\sqrt{2})}{3q+1} = \frac{7}{3} + \frac{4(\sqrt{2} - 4/3)}{k}.$$

If $p = 2$, then $k = qk_2 + k_1$. Therefore by Theorem 3.19, there exists a voting rule with distortion at most

$$\frac{3q \times 7/3 + 2(1+\sqrt{2})}{3q+2} = \frac{7}{3} + \frac{2(\sqrt{2} - 4/3)}{k}.$$

This completes the proof.

□

3.6.3 Lower Bounds

After establishing the upper bounds, we now turn our attention to the lower bounds. We provide two key results for lower bounds: one for $k \leq m/2$ and another for $k \geq m/2$.

Theorem 3.21. *For any $k \leq m/2$, the distortion of any k-winner voting rule f in the line metric, under additive social cost, can be lower bounded as:*

$$\text{dist}(f) \geq \begin{cases} 2 + \frac{1}{k} & \text{if } k \text{ is odd,} \\ 1 + \sqrt{1 + \frac{2}{k}} & \text{if } k \text{ is even.} \end{cases}$$

Proof. Similar to the proof for $k = 2$, we construct an instance $\mathcal{E} = (V, C, k, \succ)$ with two different metrics d_1 and d_2. Assume there are two sets of candidates, S_a and S_b, such that

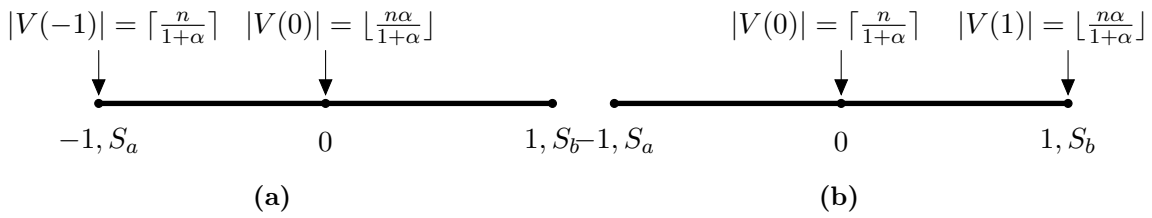


Figure 3.10: Instance $\mathcal{E}$ with two metrics, d_1 (a) and d_2 (b), used to prove the lower bound on the distortion of k-winner voting rules for $k \leq m/2$, as discussed in Theorem 3.21. S_a and S_b are two subsets of candidates located at positions -1 and $+1$. As shown in the figures, voters are divided into two groups of sizes $\lceil n/(1+\alpha) \rceil$ and $\lfloor n\alpha/(1+\alpha) \rfloor$.

$|S_a|, |S_b| \geq k$. In both metrics, locate the candidates in S_a at $x_1 = -1$ and the candidates in S_b at $x_2 = +1$. For a fixed k, define the parameter α as follows:

$$\alpha = \begin{cases} 1 & \text{if } k \text{ is odd,} \\ \sqrt{\frac{k}{k+2}} & \text{if } k \text{ is even.} \end{cases}$$

To construct $\succ$, consider a preference profile with two voter groups. The *small group*, consisting of $n_1 = n/(1+\alpha) + \varepsilon$ voters, prefers all candidates in S_a over those in S_b. The *large group*, with $n_2 = n\alpha/(1+\alpha) - \varepsilon$ voters, prefers all candidates in S_b over those in S_a. The value $\varepsilon \in [0, 1]$ is chosen to ensure n_1 and n_2 are integers.

For the first metric d_1, assume that the small group of voters lie at $x_1 = -1$ and the large group lie at $x_3 = 0$, as shown in Figure 3.10(a). For the second metric d_2, place the small group at $x_3 = 0$ and the large group at $x_2 = 1$, as shown in Figure 3.10(b). Note that in both metrics, the locations of voters are consistent with their preferences $\succ$. Furthermore, the optimal committee in $\mathcal{E}$ with metric d_1 contains k candidates from S_a, while the optimal committee in $\mathcal{E}$ with metric d_2 consists of k candidates from S_b.

Given this instance and the two metrics, we prove a lower bound on the distortion of any voting rule f. Let $S = f(\succ)$ be the output committee selected by f. Assume that S contains r candidates from S_a and $k - r$ candidates from S_b. One can observe that

$$\begin{aligned} \text{dist}(S, \mathcal{E}; d_1) &= 1 + 2 \times \frac{k-r}{k} \times \frac{n_1}{n_2}, \\ \text{dist}(S, \mathcal{E}; d_2) &= 1 + 2 \times \frac{r}{k} \times \frac{n_2}{n_1}. \end{aligned}$$

First, consider the case that $r \geq (k+1)/2$. If k is odd, $\alpha = 1$. Therefore, considering even values of n, we have $n_1 = n_2$. Thus,

$$\begin{aligned} \text{dist}(f) &\geq \text{dist}(S, \mathcal{E}; d_2) \\ &= 1 + \frac{2r}{k} \\ &\geq 1 + \frac{2(k+1)}{2k} \\ &= 2 + \frac{1}{k}. \end{aligned}$$

If k is even, it implies that $r \geq k/2 + 1$. Therefore,

$$\begin{aligned}
\text{dist}(f) &\geq \text{dist}(S, \mathcal{E}; d_2) \\
&= 1 + \frac{2r}{k} \times \frac{n_2}{n_1} \\
&= 1 + \frac{2r}{k} \times \left(\alpha - \frac{(1+\alpha)^2 \varepsilon}{n + (1+\alpha)\varepsilon} \right) \\
&\geq 1 + \frac{2r}{k} \times (\alpha - \frac{4}{n}) \\
&\geq 1 + \frac{k+2}{k} \times \sqrt{\frac{k}{k+2}} - \frac{8r}{nk} \\
&\geq 1 + \sqrt{1 + \frac{2}{k}} - \frac{8}{n}.
\end{aligned}$$

Note that as $n \to \infty$, $8/n \to 0$. Thus, given an even value for k, no voting rule can achieve a distortion better than $1 + \sqrt{1 + 2/k}$, which is the desired lower bound.

In the remaining case, we have $r < (k+1)/2$. If k is odd, then $r \leq (k-1)/2$, and therefore,

$$\begin{aligned}
\text{dist}(f) &\geq \text{dist}(S, \mathcal{E}; d_1) \\
&= 1 + \frac{2(k-r)}{k} \\
&\geq 1 + \frac{k+1}{k} \\
&= 2 + \frac{1}{k}.
\end{aligned}$$

If k is even, it implies that $r \leq k/2$. Therefore,

$$\begin{aligned}
\text{dist}(f) &\geq \text{dist}(S, \mathcal{E}; d_1) \\
&= 1 + 2 \times \frac{k-r}{k} \times \frac{n_1}{n_2} \\
&\geq 1 + 2 \times \frac{k-r}{k} \times \sqrt{\frac{k+2}{k}} \\
&\geq 1 + \sqrt{1 + \frac{2}{k}},
\end{aligned}$$

which completes the proof. □

Theorem 3.22. *For any $k \geq m/2$, the distortion of any k-winner voting rule f in the line metric, under additive social cost, can be lower bounded as*

$$\text{dist}(f) \geq 1 + (m-k)/(3k-m).$$

Proof. Similar to the proof of Theorem 3.21, we provide an instance $\mathcal{E} = (V, C, k, \succ)$ with two different metrics d_1 and d_2, having different locations for the voters. Assume there are two sets of candidates, S_a and S_b, such that $|S_a| = |S_b| = m/2$. In both metrics locate the candidates in S_a at $x_1 = -1$ and the candidates in S_b at $x_2 = +1$.

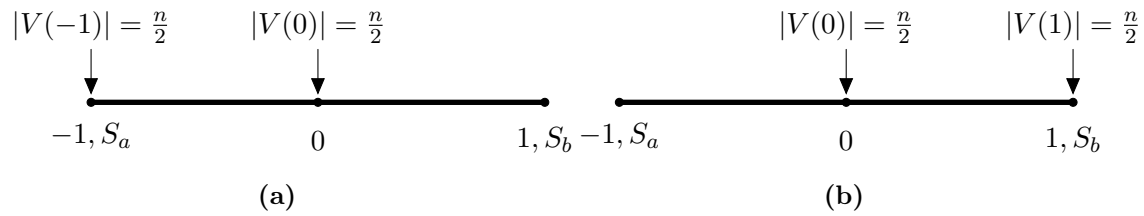


Figure 3.11: Instance $\mathcal{E}$ with two metrics, d_1 (a) and d_2 (b), used to prove the lower bound on the distortion of k-winner voting rules for $k > m/2$, as discussed in Theorem 3.22. S_a and S_b are two subsets of candidates located at positions -1 and $+1$. As shown in the figures, voters are divided into two groups of size $n/2$.

To construct $\succ$, consider a preference profile in which $n_1 = n/2$ voters prefer each candidate in S_a to the candidates in S_b, and the remaining $n_2 = n/2$ voters prefer each candidate in S_b to the candidates in S_a. For the first metric d_1, assume that the first group of voters are located at $x_1 = -1$ and the rest are located at $x_3 = 0$, as shown in Figure 3.11(a). For the second metric d_2, place one group at $x_3 = 0$ and the other group at $x_2 = 1$, as shown in Figure 3.11(b). Note that both metrics are consistent with the preference profiles $\succ$. Furthermore, the optimal committee in $\mathcal{E}$ with metric d_1 contains $m/2$ candidates from S_a and $k - m/2$ candidates from S_b, while the optimal committee in $\mathcal{E}$ with metric d_2 consists of $m/2$ candidates from S_b and $k - m/2$ candidates in S_a.

Given these instances, we prove a lower bound on any voting rule f. Let $S = f(\succ)$ be the output committee selected by f. Assume that S contains r candidates from S_a and $k - r$ candidates from S_b. One can observe that

$$\begin{aligned}\operatorname{dist}(S, \mathcal{E}; d_1) &= 1 + \frac{2(m/2 - r)}{3k - m},\\ \operatorname{dist}(S, \mathcal{E}; d_2) &= 1 + \frac{2(r - k + m/2)}{3k - m}.\end{aligned}$$

Since distortion of f can be lower bounded by both of these two values, it can also be lower bounded by their average. Therefore,

$$\begin{aligned}\operatorname{dist}(f) &\geq (\operatorname{dist}(S, \mathcal{E}; d_1) + \operatorname{dist}(S, \mathcal{E}; d_2))/2\\ &= 1 + \frac{m - k}{3k - m},\end{aligned}$$

which completes the proof. □

The results in this section suggest that as the size of the elected committee increases, both the upper and lower bounds indicate the potential for improved distortion. In particular, allowing larger values of k creates opportunities to design voting rules with lower distortion.

3.7 Egalitarian Additive Cost

In the previous sections, we studied the distortion of multi-winner elections under the utilitarian additive cost objective. The social cost of a committee in the utilitarian setting

is calculated by summing the cost of the committee across all voters. In this section, we study the egalitarian objective, focusing on the maximum cost incurred by any voter for the selected committee. This objective captures a notion of fairness.

Cembrano and Shahkarami, 2026 have studied the egalitarian additive cost objective on the line metric in peer selection, where the candidates are the voters themselves. They proved a lower bound of $3/2 - 1/k$ in this setting, which is a special case of our setting. We complement this result by proving an upper bound of 2 in Theorem 3.23.

Theorem 3.23. *If an election $\mathcal{E} = (V, C, k, \succ)$ is given, with non-Pareto-dominated candidates ordered as $x_{c_1} \leq x_{c_2} \leq \cdots \leq x_{c_m}$ on the line, then for any committee S of size $k < m - 1$ excluding c_1 and c_m, we have $\mathrm{dist}(S) \leq 2$ under the egalitarian additive cost objective.*

Proof. Fix an election $\mathcal{E}$, a consistent metric d, and a committee S such that $c_1, c_m \notin S$. Let $x_{v_1} \leq x_{v_2} \leq \cdots \leq x_{v_n}$ denote the voters' order on the line, and let $S = \{s_1, \ldots, s_k\}$, such that $x_{s_i} \leq x_{s_{i+1}}$.

Recall that the cost of the committee S for a voter i is defined as $\mathrm{SC}(S, i) = \sum_{a \in S} d(i, a)$. Since all Pareto-dominated candidates are removed, there can be at most one candidate to the left of v_1 and at most one to the right of v_n. Therefore, if $c_1, c_m \notin S$, for all $s_i \in S$, we have $x_{v_1} \leq x_{s_i} \leq x_{v_n}$.

Now, we prove that $\mathrm{SC}(S, V) = \max\{\mathrm{SC}(S, v_1), \mathrm{SC}(S, v_n)\}$. Suppose, to the contrary, that there exists a voter v^* such that

$$\mathrm{SC}(S, v^*) > \max\{\mathrm{SC}(S, v_1), \mathrm{SC}(S, v_n)\}.$$

If all candidates in S are positioned on one side of v^*, then it follows that the cost of S for either v_1 or v_n would not be less than that of v^*. Therefore, v^* lies between two consecutive candidates in S, say s_j and s_{j+1}. Without loss of generality, assume that $j \leq k/2$. Moreover, let $I = \{j+1, j+2, \ldots, k-(j+1)\}$. We have

$$\begin{aligned}
\mathrm{SC}(S, v_1) - \mathrm{SC}(S, v^*) &= \sum_{i \in I} \left(d(v_1, s_i) - d(v^*, s_i)\right) + \sum_{i \notin I} \left(d(v_1, s_i) - d(v^*, s_i)\right) \\
&\geq \sum_{i \notin I} \left(d(v_1, s_i) - d(v^*, s_i)\right) \\
&= \sum_{i=1}^{j} \Big(\big(d(v_1, s_i) + d(v_1, s_{k-i})\big) - \big(d(v^*, s_i) + d(v^*, s_{k-i})\big) \Big) \\
&= \sum_{i=1}^{j} \Big(\big(2d(v_1, s_i) + d(s_i, s_{k-i})\big) - \big(d(s_i, s_{k-i})\big) \Big) \\
&\geq 0.
\end{aligned}$$

Thus, $\mathrm{SC}(S, v^*) \leq \mathrm{SC}(S, v_1)$, which is a contradiction. Therefore,

$$\mathrm{SC}(S, V) = \max\{\mathrm{SC}(S, v_1), \mathrm{SC}(S, v_n)\} \leq \mathrm{SC}(S, v_1) + \mathrm{SC}(S, v_n).$$

Now let OPT be the optimal committee of size k, i.e., the committee with the minimum egalitarian additive cost. We have

$$\begin{aligned}
\operatorname{dist}(S, \mathcal{E}) &= \frac{\operatorname{SC}(S, V)}{\min_{S' \in \binom{C}{k}} \operatorname{SC}(S', V)} \\
&\leq \frac{\operatorname{SC}(S, v_1) + \operatorname{SC}(S, v_n)}{\max_{i \in V} \operatorname{SC}(\mathrm{OPT}, i)} \\
&\leq \frac{\operatorname{SC}(S, v_1) + \operatorname{SC}(S, v_n)}{\frac{1}{2}\big(\operatorname{SC}(\mathrm{OPT}, v_1) + \operatorname{SC}(\mathrm{OPT}, v_n)\big)} \\
&= \frac{\sum_{s_i \in S}(d(s_i, v_1) + d(s_i, v_n))}{\frac{1}{2}\sum_{s_j \in \mathrm{OPT}}(d(s_j, v_1) + d(s_j, v_n))} \\
&\leq \frac{kd(v_1, v_n)}{\frac{1}{2}kd(v_1, v_n)} \\
&= 2.
\end{aligned}$$

The last inequality holds because of the triangle inequality in the denominator of the fraction and the fact that for all $s_i \in S$, s_i lies between v_1 and v_n. This completes the proof. □

In addition to establishing a lower bound, Cembrano and Shahkarami, 2026 introduced a k-winner voting rule *k-Extremes*, which achieves an upper bound of approximately 3/2 for peer selection on the line metric under the egalitarian additive cost objective. This voting rule returns the leftmost $\lfloor k/2 \rfloor$ and the rightmost $\lceil k/2 \rceil$ candidates as the output committee.

In Observation 3.24, we show that the distortion of k-Extremes is at least 2 in the general setting.

Observation 3.24. *The distortion of the k-Extremes voting rule is at least 2 on the line metric under the egalitarian additive cost objective.*

Proof. Let $k = 2$. In Figure 3.12, candidates a, b, and c are positioned at -1, 0, and $+1$, with voters v_1 and v_2 at 0 and $+1$. The k-Extremes rule selects $\{a, c\}$, yielding $\operatorname{SC}(\{a, c\}, V) = 2$, while the optimal committee $\{b, c\}$ has $\operatorname{SC}(\{b, c\}, V) = 1$. Thus, distortion is at least 2.

This instance extends to larger k by placing at least $k/2$ candidates at -1, 0, and $+1$. k-Extremes selects those at -1 and $+1$, yielding a social cost of k. Choosing candidates at 0 and $+1$ instead reduces the social cost to $k/2$, proving a lower bound of 2. □

Finding a way to close the gap between $3/2 - 1/k$ and 2 remains an interesting future direction.

3.8 Conclusion

In this paper, we proposed a new k-winner voting rule and established distortion bounds for every committee size k in the line metric under the utilitarian additive cost objective. The proposed voting rule achieves optimal distortion for 2-winner and 3-winner elections,

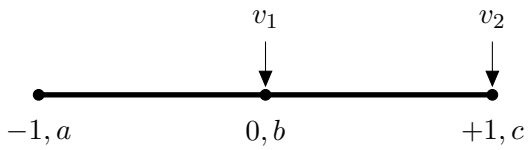


Figure 3.12: Instance $\mathcal{E}$ used to prove a lower bound on the distortion of k-Extremes in the egalitarian additive costs. The candidates a, b, and c are located on positions $-1, 0$ and $+1$. Moreover, voters v_1 and v_2 are located on 0 and $+1$, respectively.

bounds of $1 + \sqrt{2}$ and $7/3$, respectively, and an upper bound close to $7/3$ for general k-winner elections. Moreover, we proved lower bounds on the distortion of any k-winner voting rule when $k \leq m/2$. For odd values of k, this lower bound is $2 + 1/k$, and for even values of k, it is $1 + \sqrt{1 + 2/k}$. We also analyzed the egalitarian additive cost objective and established an upper bound of 2 in this setting.

We believe our bounds hold under the non-degenerate locations assumption, where voters lie at distinct locations. For both our lower bounds and upper bounds, we can distribute voters and candidates at distinct points near their original locations. There exists a small interval around point x, such that by relocating them within this interval the preferences, the optimal output, and the resulting bounds remain unchanged.

However, several open problems remain. The most significant challenge is extending our analysis to more general metric spaces. Understanding how these bounds behave in more complex settings is an important next step. Additionally, even in the line metric, tightening the bounds for general values of k remains an open question.

CHAPTER 4

Metric Distortion in Peer Selection

Chapter overview. This chapter investigates committee selection in the peer selection setting, where voters and candidates coincide and only ordinal preferences are observed. It examines how this additional structural constraint affects the distortion of social cost objectives, revealing which limitations persist and which can be alleviated relative to the general setting. This chapter is based on joint work with Javier Cembrano, which appeared at AAMAS 2026.

4.1 Introduction

A fundamental problem in social choice is the aggregation of individual preferences, expressed as rankings over a set of candidates, into a social preference consisting of a subset of elected candidates. For centuries, social choice theorists have proposed several axioms to capture desirable properties that these aggregation or *voting* rules should satisfy, usually leading to strong impossibility results (Arrow, 1963; Condorcet, 1785; Gibbard, 1973; Satterthwaite, 1975).

As an alternative approach, attempting to quantify the extent to which a certain voting rule can faithfully translate voters' preferences into the selected committee, Procaccia and Rosenschein (2006) introduced the notion of *distortion* of a rule. The underlying assumption is that a voter's (dis)affinity with a candidate can be represented by a certain cost, and voters' rankings express these cardinal preferences. The cost of a committee for a voter is then defined by aggregating the costs of the committee members, and the overall *social cost* of the committee by aggregating the costs for all voters. The distortion corresponds to the worst-case ratio between the social cost of the selected committee and that of the optimal committee, over all possible preferences and consistent metrics.

The study of the distortion of voting rules has usually focused on two ways of modeling the social cost: utilitarian and egalitarian (Caragiannis, Nath, Procaccia, and Shah, 2017; Caragiannis and Procaccia, 2011; A. Goel, Hulett, and Krishnaswamy, 2018). In the utilitarian case, the social cost is defined as the sum of the individual costs of the voters, so that all voters' costs contribute equally to the objective. In contrast, the egalitarian social cost considers the maximum individual cost among all voters, capturing a notion of fairness where no voter is excessively disadvantaged.

In voting theory, it is common to assume that voters' preferences are not fully arbitrary but exhibit some structural properties. A relevant line of work has sought structural restrictions that are natural and have powerful implications, such as single-peaked (Black, 1948) or single-crossing (Mirrlees, 1971); see Elkind, Lackner, and D. Peters (2022) for a survey. A rather general framework among these is that of *spatial* or *metric voting*, where voters and candidates are assumed to lie in a common low-dimensional metric space and voters' costs correspond to their distance to each candidate (Aziz, 2020; Enelow

and Hinich, 1984; Jessee, 2012; Merrill and Grofman, 1999). For instance, a line metric is commonly employed to capture political affinity on the left-right spectrum, whereas geographical preferences are typically modeled in two-dimensional spaces.

This structural assumption naturally fits in the metric distortion framework: the distances to candidates fully define the social cost of a committee, but the voting rules only receive their expression as preference rankings. With this structural restriction, a tight distortion bound of 3 has been shown for any single-winner deterministic voting rule (Anshelevich, Bhardwaj, Elkind, Postl, and Skowron, 2018; Gkatzelis, Halpern, and Shah, 2020; Kizilkaya and Kempe, 2022). Extending distortion to multi-winner elections requires defining how a voter's cost is aggregated over the selected committee. Two ways have been considered in the literature: the *additive cost*, where a voter's cost is the sum of their distances to all members of the committee (Babashah, Karimi, Seddighin, and Shahkarami, 2025), and the *q-cost*, where the cost is determined by their distance to their qth closest committee member (Caragiannis, Shah, and Voudouris, 2022; X. Chen, M. Li, and C. Wang, 2020).

Work on metric distortion has so far focused on the case where voters and candidates constitute disjoint sets, which forms a natural model for large-scale elections. However, in many decision-making scenarios, a group of agents seeks to elect a subset of their own members. One can think, for example, of a political organization selecting a committee. Each member ranks others according to their political affinity, and the organization aims to select a committee that represents the variety of preferences of its members. Since the voting rule only receives ordinal preferences, a small distortion constitutes a suitable objective to ensure a close-to-optimal outcome under this limited information. In general, this situation arises in the context of *peer selection*, where individuals evaluate each other to choose a group for governance, leadership, or resource allocation. Further examples include academic hiring and promotion, student representative elections, self-organized committees in cooperatives, and local governance selection.

While peer selection rules have been extensively studied in other contexts, particularly regarding the effect of strategic behavior (e.g. Alon, Fischer, Procaccia, and Tennenholtz, 2011; Caragiannis, Christodoulou, and Protopapas, 2022; Holzman and Moulin, 2013), little is known about their ability to accurately reflect agents' cardinal preferences. On the other hand, previous work on metric distortion for single-selection has often parameterized an election via its *decisiveness*, corresponding to the maximum ratio between a voter's distance to their top choice and to any other candidate (Anshelevich and Postl, 2017; Gkatzelis, Halpern, and Shah, 2020). These works have motivated this parameter by the fact that it becomes zero in the peer selection setting, as each agent becomes their own top choice. However, directly modeling a common set of voters and candidates constitutes a structural change to the problem that has not been explored so far.

4.1.1 Our Contributions and Techniques

We initiate the study of metric distortion when the set of voters and candidates coincide and bound the distortion achievable by voting rules selecting k out of n agents on the line metric for several social costs. A summary of our results appears in Table 4.1.

We start by observing a simple yet powerful property of metric voting on the line with a single set of voters and candidates, which follows from previous work (Babashah,

	additive	**q-cost** $q \leq \frac{k}{2}$	**q-cost** $\frac{k}{2} < q \leq k$
utilitarian	$1+\sqrt{1+\frac{2}{k}}$; $\frac{7}{3}+\frac{4}{k}\left(\sqrt{2}-\frac{4}{3}\right)$ [BKSS]	∞ [CSV]	3; 3 [CSV]
	1.0914; $\frac{2}{k}\left(n-2\sqrt{n(n-k)}\right)^{(*)}$ [T. 4.3]	∞ [T. 4.8]	$2-\frac{k-q}{4q-k-3}^{(**)}$; 3 [T. 4.9, CSV]
	additive	**q-cost** $q \leq \frac{k}{3}$	**q-cost** $\frac{k}{3} < q \leq k$
egalitarian	·; 2 [BKSS]	∞ [CSV]	3; 3 [CSV]
	$\frac{3}{2}-\frac{1}{k}$; $\frac{3}{2}-\frac{1}{2(k-1)}^{(*)}$ [T. 4.15]	∞ [T. 4.16]	2; 2 [T. 4.17]

Table 4.1: Summary of our and previous bounds on the distortion that voting rules can achieve in different settings. Values before and after the semicolon represent lower and upper bounds, respectively. Gray entries correspond to the previously studied setting with disjoint voters and candidates, either under a general metric [CSV] or under the line metric [BKSS]. The upper bounds for utilitarian additive and egalitarian additive social costs marked with $(*)$ correspond to the cases of even $n-k$ and even k, respectively; the slightly different bounds for the odd cases are stated in the corresponding theorems. The upper bound for utilitarian q-cost marked with $(**)$ in the text is valid for $q \geq \frac{k}{2}+1$ (slightly stronger than $q > \frac{k}{2}$). BKSS: Babashah, Karimi, Seddighin, and Shahkarami (2025); CSV: Caragiannis, Shah, and Voudouris (2022).

Karimi, Seddighin, and Shahkarami, 2025; Elkind and Faliszewski, 2014): we can fully compute the order of the agents from their rankings. This constitutes a powerful tool for the design of our mechanisms, as in the following we can always take this order as given.

Utilitarian Additive Cost. We first consider the utilitarian social cost, in which the social cost of a committee is defined as the sum of all individual costs. Intuitively, selecting k consecutive agents results in lower utilitarian social cost. In Section 4.3.1, we focus on the case of additive aggregation: the cost of a committee for a voter is given by the sum of all distances from the candidates to this voter. As a natural extension of the optimal rules for one or two agents, which select the median and closest-to-median agents, we consider a rule called Median Alternation that selects k middle agents. We show that Median Alternation provides a distortion close to 1 when k is small compared to n and approaches 2 as k goes to n. Despite its simplicity, the analysis of this rule poses significant challenges. In short, we reduce any metric to another with only two locations by showing the existence of a non-improving direction of movement for each agent, and then compute the worst-case distortion for this class.

We complement the upper bound described above with a lower bound of 1.0914 for any $k \geq 3$. This implies that, even in this seemingly simple case, we cannot always select the optimal committee.

Utilitarian q-Cost. In Section 4.3.2, we consider utilitarian q-cost, where the cost of a committee for an agent is given by the agent's distance to their qth closest candidate in the committee. In Theorem 4.8, we show that no voting rule can provide a constant distortion when $q \leq \frac{k}{2}$, implying that this known impossibility from the setting with disjoint voters and candidates and a general metric space (Caragiannis, Shah, and Voudouris, 2022) remains in place in our restricted setting. To prove this bound, we partition all but q agents into $\lfloor \frac{k}{q} \rfloor \geq 2$ sets and consider two metrics that differ in the position of the remaining q agents: relatively close to the other agents in one metric; very far in the other. Intuitively, selecting these q agents leads to an unbounded distortion in the former case but is necessary for a bounded distortion in the latter. For $q > \frac{k}{2}$, the existence of rules with distortion 3 follows from a general result by Caragiannis, Shah, and Voudouris (2022). We provide a lower bound that varies between $\frac{3}{2}$ and 2 as q varies between $\frac{k}{2} + 1$ and k, by considering three different metrics consistent with the same rankings and showing that, in one of them, there are q agents in one extreme that cannot be consistently selected. We take a closer look at the case with $k = q = 2$, where a best-possible distortion of 2 can be achieved by selecting the median agents when n is even. For odd n, we show that a rule selecting a *couple* of agents—a pair of agents who prefer each other over all other agents—among the five middle agents achieves an improved distortion of $\frac{4}{3}$, which is again best-possible. The Favorite Couple rule leverages two key principles: (1) selecting agents close to the median so as to balance overall distances from agents on each side of the median, and (2) selecting consecutive agents with a small distance between them since this distance is part of the cost of all agents. This intuition of selecting consecutive agents that are as close to each other as possible while also being close to the median in principle holds for larger k, but determining how tightly a group of k agents is clustered based solely on ordinal rankings remains a challenge.

Egalitarian Additive Cost. In Section 4.4, we turn our attention to the egalitarian social cost, where we focus on the maximum cost of a committee for a voter. We consider the simple k-Extremes rule, which selects half of the committee from each extreme. On an intuitive level, this constitutes a natural rule in this setting as it avoids that extreme voters are excessively disadvantaged. For the additive setting, we show in Section 4.4.2 that k-Extremes achieves an optimal distortion up to $O\left(\frac{1}{k}\right)$ terms. In particular, the optimal distortion of 1 is attained for $k = 2$, and distortions of $\frac{3}{2} - \frac{1}{2(k-1)}$ and $\frac{3}{2} - \frac{1}{k(k-1)}$ are achieved for even and odd $k \geq 3$, respectively, almost matching a lower bound of $\frac{3}{2} - \frac{1}{k}$. The worst-case instances involve $k+1$ agents in one extreme, a single agent in the other extreme, and k agents in the middle, which are selected in the optimal committee but cannot be detected by any rule when considering two symmetric distance metrics.

Egalitarian q-Cost. In Section 4.4.3, we show that k-Extremes attains a distortion of 2 for q-cost as long as $q > \frac{k}{3}$. To do so, we prove that the social cost of the set selected by this rule is at most the distance from the agent closest to the center to their nearest extreme, and bound the social cost of the optimal set from below by half of this distance.

We provide a matching lower bound by revisiting the instance used for the additive case. Finally, we show that no constant distortion is possible when $q \leq \frac{k}{3}$, again implying that the general impossibility result of Caragiannis, Shah, and Voudouris (2022) still holds in our setting. In the worst-case instances, we partition the agents into $\lfloor \frac{k}{q} \rfloor$ sets and consider two symmetric distance metrics where all but one set are placed at a unit distance from one another and two sets in one extreme are at the same location. We show that no rule can pick q agents from each location.

4.1.2 Related Work

Distortion of voting rules was first introduced by Procaccia and Rosenschein (2006). Since then, extensive research has been conducted to establish lower and upper bounds on the distortion of different rules under various scenarios, both within the metric and non-metric frameworks. For a comprehensive survey, we refer to Anshelevich, Filos-Ratsikas, Shah, and Voudouris (2021).

Single-Winner Voting. In the non-metric framework, Caragiannis and Procaccia (2011) showed that the distortion of any voting rule is at least $\Omega(m^2)$ and that simple rules such as Plurality achieve a distortion of at most $O(m^2)$, where m is the number of candidates. In the metric framework, Anshelevich, Bhardwaj, Elkind, Postl, and Skowron (2018) established a general lower bound of 3 on the distortion of any deterministic voting rule. They also analyzed the distortion of common voting rules such as Majority, Borda, and Copeland, showing that the latter achieves the lowest distortion of 5 among them. A. Goel, Krishnaswamy, and Munagala (2017) disproved a conjecture by Anshelevich, Bhardwaj, Elkind, Postl, and Skowron regarding a better-than-5 distortion of the Ranked Pairs rule and introduced the notion of *fairness ratio* of a rule, which captures the egalitarian social cost as a special case. These results were later improved by Munagala and K. Wang (2019), who extended the analysis to uncovered set rules and reduced the upper bound to 4.236. Gkatzelis, Halpern, and Shah (2020) closed the gap by improving this bound to 3, and showed the validity of this bound in terms of fairness ratio and thus egalitarian social cost. Randomized voting rules have also been extensively explored in the metric framework (Fain, A. Goel, Munagala, and Prabhu, 2019; Pulyassary and Swamy, 2021). The best-known upper bound for a randomized voting rule was recently obtained by Charikar, Ramakrishnan, K. Wang, and Wu (2024), who showed that a carefully designed randomization over existing and novel voting rules achieves a distortion of at most 2.753. As for lower bounds, Charikar and Ramakrishnan (2022) disproved a conjecture by A. Goel, Krishnaswamy, and Munagala (2017) regarding the existence of a randomized voting rule with distortion 2, by constructing instances whose distortion approaches 2.113 as the number of candidates grows.

Multi-Winner Voting. In the study of metric distortion for multi-winner voting, various objective functions have been proposed to capture the cost incurred by each voter for the elected committee (Elkind, Faliszewski, Skowron, and Slinko, 2017; Faliszewski, Skowron, Slinko, and Talmon, 2017). A foundational result by A. Goel, Hulett, and Krishnaswamy (2018) showed that, for the additive cost function, iterating a single-winner voting rule with distortion δ for k rounds produces a k-winner committee with

the same distortion. X. Chen, M. Li, and C. Wang (2020) studied the 1-cost objective in the metric framework when each voter casts a vote for a single candidate. They proposed a deterministic rule with a tight distortion of 3 and a randomized rule with a distortion of $3 - \frac{2}{m}$. More generally, Caragiannis, Shah, and Voudouris (2022) introduced the q-cost objective, where a voter's cost for a committee is determined by the distance to their qth closest member. They showed that the distortion is unbounded for $q \leq \frac{k}{3}$ and linear in n for $\frac{k}{3} < q \leq \frac{k}{2}$. For $q > \frac{k}{2}$, they presented a non-polynomial voting rule that achieves a distortion of 3 and a polynomial rule with a distortion of 9. They discussed how these upper bounds for $q > \frac{k}{2}$ and the unbounded distortion for $q \leq \frac{k}{3}$ carry over to egalitarian social cost, but interestingly showed that a constant distortion is possible for this objective when $\frac{k}{3} < q \leq \frac{k}{2}$. Kizilkaya and Kempe (2022) later proposed a polynomial-time rule with a distortion of 3. Recently, Babashah, Karimi, Seddighin, and Shahkarami (2025) studied the distortion of multi-winner elections with additive cost on the line, devising a rule with a distortion of roughly $\frac{7}{3}$. Caragiannis, Nath, Procaccia, and Shah (2017) studied distortion in multi-winner voting for the non-metric framework, defining a voter's utility for a committee as the highest utility derived from any of its members. They proposed a rule achieving a distortion of $1 + \frac{m(m-k)}{k}$ for deterministic committee selection when selecting k out of m candidates.

Restricted Voting Settings. A specialized setting in metric voting studies single-peaked and 1-Euclidean preferences, where both voters and alternatives are embedded on the real line (Black, 1948; Fotakis and Gourvès, 2022; Fotakis, Gourvès, and Monnot, 2016; Ghodsi, Latifian, and Seddighin, 2019; Miyagawa, 2001; Moulin, 1980; Voudouris, 2023). In particular, the work of Fotakis, Gourvès, and Patsilinakos (2025) investigated the distortion of deterministic algorithms for k-committee selection on the line under the 1-cost objective, leveraging additional distance queries.

Mechanism Design in Committee Selection. Several recent studies have explored alternative models for committee selection. The concept of stable committees and stable lotteries has been considered in various settings, focusing on fairness and individual incentives (Borodin, Halpern, Latifian, and Shah, 2022; Z. Jiang, Munagala, and K. Wang, 2020). An active area of research in the last years has focused on impartial mechanisms, where agents approve a subset of other agents and the voting rule must incentivize truthful reports while selecting well-evaluated agents (Alon, Fischer, Procaccia, and Tennenholtz, 2011; Bousquet, Norin, and Vetta, 2014; Caragiannis, Christodoulou, and Protopapas, 2022; Cembrano, Fischer, Hannon, and Klimm, 2024; Cembrano, Fischer, and Klimm, 2023; Fischer and Klimm, 2015; Holzman and Moulin, 2013; Mackenzie, 2015; Tamura, 2016). Finally, another line of work investigates distortion when agents have known locations, enabling mechanisms to explicitly consider distances in selection (Anshelevich and W. Zhu, 2021; Kalayci, Kempe, and Kher, 2024; Pulyassary, 2022).

4.2 Preliminaries

We let $\mathbb{N}$ denote the strictly positive integers and, for $n \in \mathbb{N}$, we write $[n] = \{1, \dots, n\}$ for the first n. A *linear order* $\succ$ on a set S is a complete, transitive, and antisymmetric binary relation on S; we denote the set of all linear orders on $[n]$ by $\mathcal{L}(n)$.

Election. An instance of a committee election, or simply an *election* is described by the triple $\mathcal{E} = (A, k, \succ)$, where:

- $A = [n]$ is the set of agents,
- $k \in \mathbb{N}$ is the number of agents to be selected for the committee, and
- $\succ = (\succ_1, \succ_2, \dots, \succ_n) \in \mathcal{L}^n(n)$ comprises the agents' preference profiles, where $\succ_a \in \mathcal{L}(n)$ is a linear order on $[n]$ for every $a \in [n]$.

We let $\binom{A}{k} = \{S \subseteq A \mid |S| = k\}$ denote the feasible committees for a given election; i.e., the set of all subsets of A of size k.

Line metric. A *distance metric* on A is a function $d\colon A \times A \to \mathbb{R}_+$ satisfying (i) $d(a, a) = 0$, (ii) $d(a, b) = d(b, a)$ for every $a, b \in A$, and (iii) $d(a, c) \leq d(a, b) + d(b, c)$ for every $a, b, c \in A$. In this paper, we focus on the line metric: We associate each agent $a \in A$ with a position $x_a \in (-\infty, \infty)$, and the metric d is defined by $d(a, b) = |x_a - x_b|$ for every $a, b \in A$. A metric d is said to be *consistent* with a ranking profile $\succ \in \mathcal{L}^n(n)$, denoted as $d \rhd \succ$, if for every triple of agents $a, b, c \in A$, the condition $d(a, b) < d(a, c)$ implies $b \succ_a c$.[1] Since d is fully defined by the position vector $x \in (-\infty, \infty)^A$, we often refer directly to this vector being consistent with a ranking profile $\succ \in \mathcal{L}^n(n)$ and denote it by $x \rhd \succ$. Likewise, we often exchange d by x in the definitions that follow. Finally, for a fixed election $\mathcal{E} = (A, k, \succ)$, consistent vector of locations $x \in (-\infty, \infty)^n$, and interval $I = (y, z)$ with $y < z$, we let $A(I) = \{a \in A \mid x_a \in I\}$ denote the agents with locations in I. When I is a single point $\bar{x}$, we write $A(\bar{x})$ for the agents located at this point.

Social cost. For a certain set of agents A, a committee size $k \in \mathbb{N}$, and a *candidate-aggregation function* $h\colon \mathbb{R}_+^k \to \mathbb{R}_+$, the *cost* of $S \in \binom{A}{k}$ for agent $a \in A$ is simply $\mathrm{SC}(S, a; d) = h((d(a, b))_{b \in S})$. For a set of agents A, a committee size $k \in \mathbb{N}$, and a *voter-aggregation function* $g\colon \mathbb{R}_+^n \to \mathbb{R}$, the *social cost* of $S \in \binom{A}{k}$ is $\mathrm{SC}(S, A; d) = g((\mathrm{SC}(S, a; d))_{a \in A})$. In this paper, we study a handful of candidate- and voter-aggregation functions. In terms of the voter-aggregation function $g\colon \mathbb{R}^n \to \mathbb{R}_+$, we focus on the *utilitarian social cost*, given by $g(y) = \sum_{i \in [n]} y_i$, and the *egalitarian social cost*, given by $g(y) = \max\{y_i \mid i \in [n]\}$. In terms of the candidate-aggregation function $h\colon \mathbb{R}_+^k \to \mathbb{R}_+$, we focus on the *additive social cost*, given by $h(y) = \sum_{i \in [k]} y_i$, and the *q-cost*, given by

[1] Note that this definition allows for agent-dependent tie-breaking; i.e., when $d(a, b) = d(a, c)$ agent a can rank either $b \succ_a c$ or $c \succ_a b$, independently of other agents. This assumption makes the problem in principle harder, so that our upper bounds on the distortion remain valid if a common tie-breaking rule is employed, and it allows us to construct simpler examples for lower bounds. It is not hard to see that the same lower bounds can be obtained without the assumption: Whenever a metric has ties, distances can be perturbed by a small enough constant ε so that there are no longer ties and the distortion does not improve.

$h(y) = \tilde{y}_q$, where $\tilde{y}$ is the vector with the entries of y sorted in increasing order. Thus, for example, the 1-*cost* is given by $h(y) = \min\{y_i \mid i \in [k]\}$; and the k-*cost* is given by $h(y) = \max\{y_i \mid i \in [k]\}$.

Voting rules and distortion. For $n, k \in \mathbb{N}$ with $n \geq k$, an (n,k)-*voting rule* is a function f that takes a preference profile $\succ \in \mathcal{L}^n(n)$ and returns a subset $S \in \binom{[n]}{k}$, to which we often refer as a *committee*. For an election $\mathcal{E} = ([n], k, \succ)$ and a metric d, the *distortion* $\mathrm{dist}(S, \mathcal{E}; d)$ *of* $S \subseteq A$ *under* d is the ratio between the social cost of the committee and the minimum social cost of any committee; i.e.,

$$\mathrm{dist}(S, \mathcal{E}; d) = \frac{\mathrm{SC}(S, A; d)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)}.$$

For an election $\mathcal{E} = (A, k, \succ)$, the *distortion* $\mathrm{dist}(S, \mathcal{E})$ *of a committee* $S \subseteq A$ is then defined as the worst-case distortion over all metrics consistent with the ranking profile $\succ$; i.e.,

$$\mathrm{dist}(S, \mathcal{E}) = \sup_{d \rhd \succ} \mathrm{dist}(S, \mathcal{E}; d).$$

Finally, for an (n,k)-voting rule f, the distortion of f is defined as the worst-case distortion of its output across all possible elections; i.e.,

$$\mathrm{dist}(f) = \sup_{\succ \in \mathcal{L}^n(n)} \mathrm{dist}(f(\succ), ([n], k, \succ)).$$

Throughout the paper, we study the distortion that voting rules can achieve under different social costs.

4.2.1 Computing the Order From an Election

An essential property in line metric settings is the ability to determine the order of agents based on their preferences. This result has been established in prior work. Specifically, Elkind and Faliszewski (2014) and Babashah, Karimi, Seddighin, and Shahkarami (2025) demonstrate that if voters' preference lists are pairwise distinct, it is possible to uniquely determine the ordering on the line, along with the ordering of the non-Pareto-dominated alternatives. While their setting differentiates between voters and alternatives, this result naturally extends to our context, where agents serve as both voters and candidates. In our setting, this follows directly from a simpler observation: for any three agents, their relative order on the line can be reconstructed from their preference rankings. We state this result as a lemma, which serves as a foundation for many results in this paper as it guarantees that the order of agents in any election can be uniquely identified.

Lemma 4.1 (Elkind and Faliszewski (2014), Babashah, Karimi, Seddighin, and Shahkarami (2025))**.** *For every election* $\mathcal{E} = ([n], k, \succ)$*, we can compute a permutation* $\pi \colon [n] \to [n]$ *of the agents such that, for any consistent position vector* $x \in (-\infty, \infty)^n$ *with* $x \rhd \succ$*, we have either* $x_{\pi(1)} \leq x_{\pi(2)} \leq \cdots \leq x_{\pi(n)}$ *or* $x_{\pi(n)} \leq x_{\pi(n-1)} \leq \cdots \leq x_{\pi(1)}$*.*

We briefly highlight some structural features that distinguish the peer selection setting from the general setting. First, unlike the general voting setup with disjoint voters

and candidates, the peer selection problem provides us with not only the order of the candidates but also the exact order of the voters. Second, this order can be efficiently reconstructed: we can identify extreme agents by observing who consistently appears at the bottom of others' rankings. Once such an agent is identified, their ranking determines the full order of agents along the line. Third, the mutual evaluations are more restricted, so we gain more information about relative preferences, such as how many voters prefer one candidate over another, than in the general voting setting. These features inform the design of improved mechanisms in peer selection and present new algorithmic and structural challenges.

For simplicity, whenever we fix an election throughout the paper, we will assume w.l.o.g., that the agents are already ordered, i.e., that the permutation π stated in the lemma is the identity. Hence, we denote the ordered agents by $1, \dots, n$ and informally refer to this order as *from left to right*.

4.3 Utilitarian Social Cost

Using Lemma 4.1, we know that the order of agents can be fully determined from the preference profile $\succ$. This allows us to identify the *median agent*, who minimizes the total distance to all other agents. Hence, when selecting a single agent ($k = 1$) under the utilitarian objective, the median agent is optimal. This observation holds regardless of whether the individual cost is defined additively or via the q-cost, since both notions coincide when $k = 1$.

For larger committee sizes ($k > 1$), it becomes necessary to define how a voter's distances to the selected agents are aggregated. In this section, we study two aggregation rules: one that considers the sum of all distances to selected agents in Section 4.3.1, and one that considers the distance to the qth closest selected agent in Section 4.3.2.

4.3.1 Utilitarian Additive Cost

In this section, we focus on the *utilitarian additive* objective for committee selection. This objective aims to minimize the *utilitarian additive social cost*, which is defined as the total distance from all agents to the selected committee. Formally, the *utilitarian additive social cost* of a committee $S' \in \binom{A}{k}$ is given by

$$\mathrm{SC}(S', A; d) = \sum_{a \in A} \sum_{b \in S'} d(a, b).$$

The cost of each agent $a \in A$ is the sum of their distances to all members of the selected committee S', and the overall social cost is the sum of these individual costs across all agents in A.

It is straightforward to verify that the optimal committee can be directly identified from the preference profile when the committee size is both $k = 1$ and $k = 2$. For $k = 1$, the optimal committee consists of the median agent, as discussed above. For $k = 2$, the optimal committee consists of the two median agents if n is even, or the median agent together with the agent closest to them, if n is odd. In both cases, the optimal agents can be determined solely from $\succ$, without knowledge of the underlying metric, yielding a distortion of 1.

We now introduce our voting rule for selecting a committee of size $k \geq 2$, called the Median Alternation rule. This rule can be viewed as a generalization of the *Polar Comparison* rule proposed by Babashah, Karimi, Seddighin, and Shahkarami (2025) for multi-winner elections. In contrast to the Polar Comparison rule, the Median Alternation rule (i) applies to all committee sizes k, (ii) is simpler since it does not require case distinctions or pairwise comparisons, and (iii) achieves improved distortion guarantees under peer selection.

Voting Rule 1 (Median Alternation)**.** *Compute the order of the agents $1, \ldots, n$ on the line and let $m = \lceil \frac{n+1}{2} \rceil$ denote the (upper) median agent. If $n - k$ is even, return $S = \{\frac{n-k}{2} + 1,\ \frac{n-k}{2} + 2,\ \ldots,\ \frac{n+k}{2}\}$. If $n - k$ odd and $\lceil \frac{n-k}{2} \rceil \succ_m \lceil \frac{n+k}{2} \rceil$, return $S = \{\lceil \frac{n-k}{2} \rceil, \lceil \frac{n-k}{2} \rceil + 1, \ldots, \lceil \frac{n+k}{2} \rceil - 1\}$. Otherwise, return $S = \{\lceil \frac{n-k}{2} \rceil + 1, \lceil \frac{n-k}{2} \rceil + 2, \ldots, \lceil \frac{n+k}{2} \rceil\}$.*

On an intuitive level, the rule can be understood as constructed by going through the rank list of the median(s) agent(s), selecting agents in the order reported by them but alternating between those to their left and to their right. This ensures a balanced representation of agents on both sides.

For larger committees, a key ingredient in our analysis is that there always exists an optimal committee consisting of consecutive agents. We formalize this observation in the following lemma.

Lemma 4.2. *For any election $\mathcal{E} = (A, k, \succ)$ and consistent metric $d \rhd \succ$, there exists $i \in [n - k + 1]$ such that, defining $S^* = \{i, i + 1, \ldots, i + k - 1\}$, we have $SC(S^*, A; d) = \min \{SC(S', A; d) \mid S' \in \binom{A}{k}\}$.*

Proof. Let $\mathcal{E} = (A, k, \succ)$ with $A = [n]$ and d be as in the statement, and let also $x \rhd \succ$ be a consistent position vector defining d. The result is trivial if $k = 1$, so we assume that $k \geq 2$ in what follows. We first observe that, by Lemma 2 in Babashah, Karimi, Seddighin, and Shahkarami (2025), $\mathrm{SC}(a, A; d) \leq \mathrm{SC}(b, A; d)$ holds for $a, b \in A$ are such that either (1) $a, b \geq \frac{n+1}{2}$ and $a - \frac{n+1}{2} \leq b - \frac{n+1}{2}$, or (2) $a, b \leq \frac{n+1}{2}$ and $\frac{n+1}{2} - a \leq \frac{n+1}{2} - b$. In simple words, if two agents lie on the same side of the median agent(s), the agent closer to them has a lower cost. Thus, there exist $S^* \in \binom{A}{k}$ that minimizes the social cost such that $\{m_1, m_2\} \subseteq S^*$, where $m_1 = \lfloor \frac{n+1}{2} \rfloor$ and $m_2 = \lceil \frac{n+1}{2} \rceil$ denote the median agent(s) (note that $m_1 = m_2$ if n is odd). Now, suppose that S^* is not consecutive. Since $m_1, m_2 \in S^*$, there exists an agent $a \notin S^*$ and $b \in S^*$ such that either (1) $a, b \geq \frac{n+1}{2}$ and $a - \frac{n+1}{2} \leq b - \frac{n+1}{2}$, or (2) $a, b \leq \frac{n+1}{2}$ and $\frac{n+1}{2} - a \leq \frac{n+1}{2} - b$. But then, using the result by Babashah, Karimi, Seddighin, and Shahkarami again, we obtain that $\mathrm{SC}((S^* \setminus \{b\}) \cup \{a\}, A; d) \leq \mathrm{SC}(S^*, A; d)$; i.e., we can exchange b by a and the social cost of the committee does not increase. By repeating this procedure, we reach a committee with consecutive agents and minimum social cost, as claimed in the statement. □

We now present our main result in terms of utilitarian additive social cost, regarding the distortion guaranteed by Median Alternation.

Theorem 4.3. *For every $n, k \in \mathbb{N}$ with $n \geq k \geq 2$, under the utilitarian additive social cost the distortion of Median Alternation is at most*

$$\frac{1}{k}\left(2n + \chi - 2\sqrt{n(n - k + \chi)}\right),$$

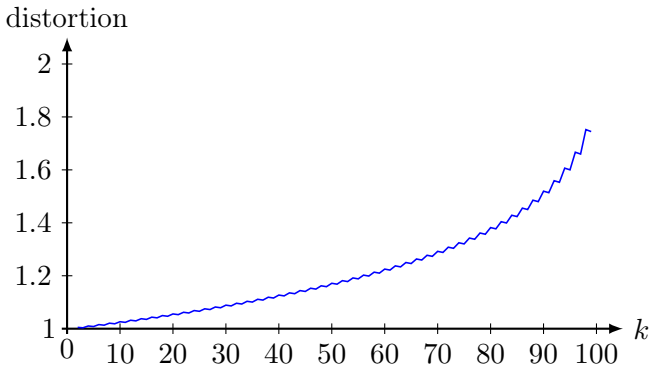


Figure 4.1: Distortion of Median Alternation stated in Theorem 4.3 for $n = 100$ and $k \in \{2, \dots, 99\}$.

where $\chi = 1$ *if* $n - k$ *is odd and* $\chi = 0$ *otherwise.*

The bound ranges between 1 and 2: it is closer to 1 when k is small relative to n, and approaches 2 as k approaches n. Figure 4.1 illustrates the bound for $n = 100$ and k between 2 and $n - 1$.

In order to prove Theorem 4.3, we will show that we can reduce any metric to another one where all agents are in one out of two locations. As a first step, we prove that an agent (or set of agents at the same location) can always be moved in one direction such that the distortion does not improve, as long as they do not pass through other agents' locations. To this end, for a position vector $x \in (-\infty, \infty)^n$, a position $\bar{x} \in (-\infty, \infty)$ such that $A(\bar{x}) \neq \emptyset$, and $\delta > 0$, we define the *shifted position vectors* $x^-(\bar{x}, \delta), x^+(\bar{x}, \delta) \in (-\infty, \infty)^n$ as follows:

$$x_a^-(\bar{x}, \delta) = x_a - \delta \text{ for every } a \in A(\bar{x}), \quad x_a^-(\bar{x}, \delta) = x_a \text{ for every } a \in A \setminus A(\bar{x}),$$
$$x_a^+(\bar{x}, \delta) = x_a + \delta \text{ for every } a \in A(\bar{x}), \quad x_a^+(\bar{x}, \delta) = x_a \text{ for every } a \in A \setminus A(\bar{x}).$$

Lemma 4.4. *Let* $\mathcal{E} = (A, k, \succ)$ *be an election with* $A = [n]$*, let* $S \in \binom{A}{k}$ *be the committee selected by* Median Alternation *on this election, and let* $x \in (-\infty, \infty)^n$ *with* $x \rhd \succ$ *be a consistent position vector. Let* $\bar{x} \in (-\infty, \infty)$ *be such that* $A(\bar{x}) \neq \emptyset$*, let* $\delta > 0$ *be such that* $A((\bar{x} - \delta, \bar{x} + \delta)) = A(\bar{x})$ *and let* $x^- = x^-(\bar{x}, \delta)$ *and* $x^+ = x^+(\bar{x}, \delta)$*. Then, for all preference profiles* $\succ^-, \succ^+$ *such that* $x^- \rhd \succ^-$ *and* $x^+ \rhd \succ^+$*, at least one of the following inequalities holds:*

$$\mathrm{dist}(S, (A, k, \succ^-); x^-) \geq \mathrm{dist}(S, \mathcal{E}; x), \quad \textit{or} \quad \mathrm{dist}(S, (A, k, \succ^+); x^+) \geq \mathrm{dist}(S, \mathcal{E}; x).$$

The proof of this lemma relies on the linearity of the objective function: If moving an agent or set of agents to the right has a certain effect on the social cost, moving them to the left has the opposite effect. Then, the ratio between the social cost of any two fixed committees must not improve in one of these directions. Since the committee selected by Median Alternation remains fixed as long as the order of agents does not change, and changing the optimal set can only lead to a worse distortion, the result follows. We now proceed with the formal proof.

Proof of Lemma 4.4. Let $\mathcal{E} = (A, k, \succ)$, S, x, $\bar{x}$, δ, x^-, x^+, $\succ^-$, and $\succ^+$ be as in the statement. We denote by d, d^-, and d^+ the distance metrics associated to x, x^-, and x^+, respectively.

We first consider an arbitrary committee $S' \in \binom{A}{k}$ and compute the difference between the social cost of this committee under metric d and under both of the other metrics. From the definition of the additive social cost, for any $a \in A$ such that $x_a < \bar{x}$ we have that

$$\begin{aligned}
\mathrm{SC}(S', a; x^-) &= \sum_{b \in S' \cap A(\bar{x})} d^-(a,b) + \sum_{b \in S' \setminus A(\bar{x})} d^-(a,b) \\
&= \sum_{b \in S' \cap A(\bar{x})} (d(a,b) - \delta) + \sum_{b \in S' \setminus A(\bar{x})} d(a,b) \\
&= \mathrm{SC}(S', a; x) - \delta \, |S' \cap A(\bar{x})|. \qquad (4.1)
\end{aligned}$$

Similarly, for any $a \in A$ such that $x_a > \bar{x}$ we have that

$$\begin{aligned}
\mathrm{SC}(S', a; x^-) &= \sum_{b \in S' \cap A(\bar{x})} d^-(a,b) + \sum_{b \in S' \setminus A(\bar{x})} d^-(a,b) \\
&= \sum_{b \in S' \cap A(\bar{x})} (d(a,b) + \delta) + \sum_{b \in S' \setminus A(\bar{x})} d(a,b) \\
&= \mathrm{SC}(S', a; x) + \delta \, |S' \cap A(\bar{x})|. \qquad (4.2)
\end{aligned}$$

Finally, for every a with $x_a = \bar{x}$, i.e., $a \in A(\bar{x})$, we have that

$$\begin{aligned}
\mathrm{SC}(S', a; x^-) &= \sum_{b \in S' \cap A((-\infty, \bar{x}))} d^-(a,b) + \sum_{b \in S' \cap A((\bar{x}, +\infty))} d^-(a,b) \\
&= \sum_{b \in S' \cap A((-\infty, \bar{x}))} (d(a,b) - \delta) + \sum_{b \in S' \cap A((\bar{x}, +\infty))} (d^-(a,b) + \delta) \\
&= \mathrm{SC}(S', a; x) + \delta \left(|S' \cap A((\bar{x}, +\infty))| - |S' \cap A((-\infty, \bar{x}))| \right). \qquad (4.3)
\end{aligned}$$

Combining eqs. (4.1) to (4.3), we obtain from the definition of utilitarian social cost that

$$\begin{aligned}
\mathrm{SC}(S', A; x^-) &= \sum_{a \in A} \mathrm{SC}(S', a; d^-) \\
&= \mathrm{SC}(S', A; x) - \delta \, |S' \cap A(\bar{x})| \left(|A(-\infty, \bar{x})| - |A(\bar{x}, +\infty)| \right) \\
&\quad - \delta \, |A(\bar{x})| \left(|S' \cap A((-\infty, \bar{x}))| - |S' \cap A((\bar{x}, +\infty))| \right).
\end{aligned}$$

One can proceed analogously for d^+ to obtain

$$\begin{aligned}
\mathrm{SC}(S', A; x^+) &= \mathrm{SC}(S', A; x) + \delta \, |S' \cap A(\bar{x})| \left(|A(-\infty, \bar{x})| - |A(\bar{x}, +\infty)| \right) \\
&\quad + \delta \, |A(\bar{x})| \left(|S' \cap A((-\infty, \bar{x}))| - |S' \cap A((\bar{x}, +\infty))| \right).
\end{aligned}$$

Hence, there exists a value $\Delta(S')$, that only depends on the committee δ, such that

$$\mathrm{SC}(S', A; x^-) = \mathrm{SC}(S', A; x) - \Delta(S'), \qquad \mathrm{SC}(S', A; x^+) = \mathrm{SC}(S', A; x) + \Delta(S'). \quad (4.4)$$

We let S^* denote an optimal committee for the metric d in what follows, i.e., a committee such that $\mathrm{SC}(S^*, A; x) = \min\left\{\mathrm{SC}(S', A; x) \mid S' \in \binom{A}{k}\right\}$. We observe that

$$\mathrm{dist}(S, (A, k, \succ^-); x^-) = \frac{\mathrm{SC}(S, A; x^-)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; x^-)} \geq \frac{\mathrm{SC}(S, A; x^-)}{\mathrm{SC}(S^*, A; x^-)} = \frac{\mathrm{SC}(S, A; x) - \Delta(S)}{\mathrm{SC}(S^*, A; x) - \Delta(S^*)}, \tag{4.5}$$

and

$$\mathrm{dist}(S, (A, k, \succ^-); x^+) = \frac{\mathrm{SC}(S, A; x^+)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; x^+)} \geq \frac{\mathrm{SC}(S, A; x^+)}{\mathrm{SC}(S^*, A; x^+)} = \frac{\mathrm{SC}(S, A; x) + \Delta(S)}{\mathrm{SC}(S^*, A; x) + \Delta(S^*)}. \tag{4.6}$$

If either $\mathrm{SC}(S^*, A; x) = \Delta(S^*)$ or $\mathrm{SC}(S^*, A; x) = -\Delta(S^*)$ holds, the distortion becomes unbounded in one of the new instances and the result follows directly. Otherwise, it follows from the simple property stated in the following claim.

Claim 4.5. *For any values $y, z \in \mathbb{R}_+$ and $w \in (-z, z)$, we have either $\frac{y+w}{z+w} \geq \frac{y}{z}$ or $\frac{y-w}{z-w} \geq \frac{y}{z}$.*

Proof. Suppose towards a contradiction that both $\frac{y+w}{z+w} < \frac{y}{z}$ and $\frac{y-w}{z-w} < \frac{y}{z}$ hold. Since $w < z$, the first inequality is equivalent to

$$z(y + w) < y(z + w) \iff zw < yw.$$

Since $w > -z$, the second inequality is equivalent to

$$z(y - w) < y(z - w) \iff yw < zw.$$

As the inequalities contradict each other, we conclude. □

Applying these properties to inequalities (4.5) and (4.6), we obtain that either

$$\mathrm{dist}(S, (A, k, \succ^-); x^+) \geq \frac{\mathrm{SC}(S, A; x) + \Delta(S)}{\mathrm{SC}(S^*, A; x) + \Delta(S^*)} \geq \frac{\mathrm{SC}(S, A; x)}{\mathrm{SC}(S^*, A; x)} = \mathrm{dist}(S, \mathcal{E}; x)$$

or

$$\mathrm{dist}(S, (A, k, \succ^-); x^-) \geq \frac{\mathrm{SC}(S, A; x) - \Delta(S)}{\mathrm{SC}(S^*, A; x) - \Delta(S^*)} \geq \frac{\mathrm{SC}(S, A; x)}{\mathrm{SC}(S^*, A; x)} = \mathrm{dist}(S, \mathcal{E}; x)$$

holds, concluding the proof. □

We can use the previous lemma to conclude that, for every election and consistent metric, Median Alternation selects a committee such that, under another metric with only two locations, the distortion does not improve. Indeed, we can iterating the argument in Lemma 4.4 to move (sets of) agents in non-extreme positions in their non-improving direction. This procedure terminates with all agents in one of the original extreme positions x_1 or x_n and that the distortion has not improved. The following lemma formally states this fact.

Lemma 4.6. *Let $\mathcal{E} = (A, k, \succ)$ be an election with $A = [n]$, let $S \in \binom{A}{k}$ be the committee selected by* Median Alternation *on this election, and let $x \in (-\infty, \infty)^n$ with $x \rhd \succ$ be a consistent position vector. Then, there exists a position vector $x' \in (-\infty, \infty)^n$ such that $x'_a \in \{x_1, x_n\}$ for every $a \in A$ and $\mathrm{dist}(S', (A, k, \succ'); x') \geq \mathrm{dist}(S, \mathcal{E}, x)$, where $\succ'$ is any preference profile such that $x' \rhd \succ'$ and $S' \in \binom{A}{k}$ is the committee selected by* Median Alternation *on the election $(A, k, \succ')$.*

Proof. Let $\mathcal{E} = (A, k, \succ)$ and x be as in the statement, where, as usual, x_1 and x_n represent the positions of the two extreme agents. To construct x' as claimed in the statement, we iteratively move agents toward the positions of the extreme agents using Lemma 4.4. Specifically, we initialize $x' = x$ and, as long as $x'_a \in (x_1, x_n)$ for some $a \in A$, we fix $\bar{x} = x_a$, we define

$$\delta^* = \max\{\delta > 0 \mid A((\bar{x} - \delta, \bar{x} + \delta)) = A(\bar{x})\},$$

and we update $x'_b \leftarrow x'_b \pm \delta^*$ for every $b \in A(\bar{x})$ and the sign that ensures not increasing the distortion $\mathrm{dist}(S, A; x')$ of S. Note that the definition of δ^* ensures both the existence of this sign, due to Lemma 4.4, and the fact that the number of different positions $|\{y \in (-\infty, \infty) \mid \exists a \in [n] : x'_a = y\}|$ is reduced in each step. Thus, the procedure terminates with a vector $x' \in (-\infty, \infty)$ such that (1) $x'_a \in \{x_1, x_n\}$ for every $a \in A$, and (2) the distortion of S under the resulting metric has not decreased. Note that, since the order of the agents has not been changed besides ties, we have either $S' = S$ if the committee selected by Median Alternation has not changed or $S' \neq S$ but $\mathrm{SC}(S', A, x') = \mathrm{SC}(S, A; x')$ if the committee has changed due to a different tie-breaking. □

We now proceed with the proof of Theorem 4.3.

Proof of Theorem 4.3. Let $\mathcal{E} = (A, k, \succ)$ be an arbitrary election, where $A = [n]$ is the set of agents. Let $d \rhd \succ$ be any consistent distance metric induced by positions $x \in (-\infty, \infty)^n$, and let S denote the committee selected by Median Alternation on this election. From Lemma 4.6, we know that there exists a new position vector $x' \in (-\infty, \infty)^n$ and associated election $\mathcal{E}' = (A, k, \succ')$, with $x' \rhd \succ'$, such that where all agents are positioned at the two extreme positions of the original instance and the distortion in $\mathcal{E}'$ is at least as bad as the distortion in $\mathcal{E}$; i.e., $x'_a \in \{x_1, x_n\}$ for every $a \in A$ and $\mathrm{dist}(S', (A, k, \succ'); x') \geq \mathrm{dist}(S, \mathcal{E}, x)$, where S' denotes the committee selected by Median Alternation on $\mathcal{E}'$. Thus, it suffices to compute the distortion for this election $\mathcal{E}'$ to bound the distortion of the voting rule. As usual, we denote by d' the metric induced by the position vector x'.

We partition the set of agents into two groups, $A = A_1 \dot{\cup} A_n$, where

$$A_1 = \{a \in A \mid x'_a = x_1\} \text{ and } A_n = \{a \in A \mid x'_a = x_n\}$$

denote the sets of agents located at positions x_1 and x_n under the position vector x', respectively. We let $S_1 = S' \cap A_1$ and $S_n = S' \cap A_n$ denote the agents selected by Median

Alternation on $\mathcal{E}'$ from agents in A_1 and A_n, respectively. Then, the social cost of S' is given by

$$\begin{aligned}\mathrm{SC}(S', A; d') &= \sum_{a\in A_1}\sum_{b\in S'} d'(x_1, x_b) + \sum_{a\in A_n}\sum_{b\in S'} d'(x_n, x_b)\\ &= |A_1|\cdot|S_n|\cdot d'(x_1, x_n) + |A_n|\cdot|S_1|\cdot d'(x_1, x_n).\end{aligned}$$

On the other hand, the optimal committee S^* clearly minimizes the total social cost by selecting as many agents as possible from the larger group between A_1 and A_n, as this cost is only incurred by agents in the smaller set. We suppose that $|A_n| \geq |A_1|$ w.l.o.g. We have two cases: either $|A_n| \geq k$ or $|A_n| < k$. In the former case,

$$\mathrm{SC}(S^*, A; d) = |A_1|\cdot k\cdot d'(x_1, x_n),$$

while in the latter case,

$$\mathrm{SC}(S^*, A; d) = |A_1|\cdot|A_n|\cdot d'(x_1, x_n) + |A_n|\cdot(k-|A_n|)\cdot d'(x_1, x_n).$$

Since $|A_n| \geq |A_1|$ implies

$$|A_1|\cdot|A_n|\cdot d'(x_1, x_n) + |A_n|\cdot(k-|A_n|)\cdot d'(x_1, x_n) \geq |A_1|\cdot k\cdot d'(x_1, x_n),$$

the social cost induced by S^* is smaller when $|A_n| \geq k$ and it suffices to bound the distortion in this case. Therefore,

$$\begin{aligned}\mathrm{dist}(f) &\leq \frac{\mathrm{SC}(S, A; d')}{\mathrm{SC}(S^*, A; d')}\\ &= \frac{|A_1|\cdot|S_n|\cdot d'(x_1, x_n) + |A_n|\cdot|S_1|\cdot d'(x_1, x_n)}{|A_1|\cdot k\cdot d'(x_1, x_n)}\\ &= \frac{|A_1|\cdot|S_n| + |A_n|\cdot|S_1|}{|A_1|\cdot k}.\end{aligned} \tag{4.7}$$

If $|S_n| = k$, we obtain $\mathrm{dist}(f) = 1$. In what follows, we thus assume $S_1 \neq \emptyset$. From the definition of the Median Alternation voting rule, we know that $|A_n|-|S_n| = |A_1|-|S_1|$ if $n-k$ is even, and $|A_n|-|S_n| = |A_1|-|S_1|-1$ if $n-k$ is odd.

We can express the previous equations simply as

$$|A_n| - |S_n| = |A_1| - |S_1| - \chi.$$

From this equality, alongside $|A_1| + |A_n| = n$ and $|S_1| + |S_n| = k$, we can express all $|A_1|$, $|S_1|$, and $|S_n|$ in terms of $|A_n|$ as follows:

$$|A_1| = n - |A_n|, \quad |S_1| = \frac{n+k-\chi}{2} - |A_n|, \quad |S_n| = |A_n| - \frac{n-k-\chi}{2}.$$

Replacing in inequality (4.7), we obtain

$$\begin{aligned}\mathrm{dist}(f) &\leq \frac{(n-|A_n|)\big(|A_n| - \frac{n-k-\chi}{2}\big) + |A_n|\big(\frac{n+k-\chi}{2} - |A_n|\big)}{(n-|A_n|)k}\\ &= \frac{2|A_n|}{k} - \frac{\chi|A_n|}{k(n-|A_n|)} - \frac{n(n-k-\chi)}{2k(n-|A_n|)}\\ &= h(|A_n|),\end{aligned} \tag{4.8}$$

where we have defined a function $h\colon \left\{\lceil \frac{n}{2} \rceil, \dots, n-1\right\} \to \mathbb{R}$, which evaluated at $|A_n|$ gives the last expression. Its first and second derivatives are given by

$$h'(y) = \frac{1}{k}\left(2 - \frac{(n-k+\chi)n}{2(n-y)^2}\right), \qquad h''(y) = -\frac{(n-k+\chi)n}{k(n-y)^3}.$$

Since $h''(y) \leq 0$ for every y in the domain of h, an upper bound for the value of h is given by its value at y^*, where y^* is such that

$$h'(y^*) = 0 \iff y^* = n - \frac{1}{2}\sqrt{(n-k+\chi)n}.$$

Combining this fact with inequality (4.8), we conclude that

$$\begin{aligned}
\operatorname{dist}(f) &\leq h(y^*) \\
&= \frac{2}{k}\left(n - \frac{1}{2}\sqrt{(n-k+\chi)n}\right) - \frac{\chi\left(n - \frac{1}{2}\sqrt{(n-k+\chi)n}\right)}{\frac{k}{2}\sqrt{(n-k+\chi)n}} \\
&\quad - \frac{n(n-k-\chi)}{k\sqrt{(n-k+\chi)n}} \\
&= \frac{1}{k}\left(2n + \chi - 2\sqrt{(n-k+\chi)n}\right),
\end{aligned}$$

which is the same as the expression in the statement. □

In terms of lower bounds, we prove that for any $k \geq 3$, there exists n such that any (n,k)-voting rule has distortion at least 1.0914. To do so, for any fixed k we consider n agents partitioned into sets A_1, A_2, A_3, with $|A_3| \leq |A_1| \leq k$ and $|A_2| + |A_3| = k$. We consider two metrics depicted in Figure 4.2, both with the agents in the same set located at the same point and $x_a \leq x_b \leq x_c$ for any $a \in A_1, b \in A_2$, and $c \in A_3$. In one metric, A_2 is equidistant to A_1 and A_3; in the other, A_2 and A_3 are located at the same point. Intuitively, a rule performs badly in the first instance if it selects many agents from A_3 and in the second instance otherwise. We formalize this distinction through a threshold, obtain bounds parameterized on this threshold for each of the corresponding cases, and obtain the overall bound by optimally picking the threshold and the number of agents in each set that maximizes the minimum distortion between the two cases.

Theorem 4.7. *For every $k \in \mathbb{N}$ with $k \geq 3$, there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n,k)-voting rule f, $\operatorname{dist}(f) \geq 1.0914$ for utilitarian additive social cost.*

Proof. Consider an arbitrary (n,k)-voting rule f. We partition the agents into three sets A_1, A_2, A_3 with $|A_1| = n_1$, $|A_2| = n_2$, and $|A_3| = k - n_2$. The natural values n_1 and n_2 will be fixed later, respecting $n_2 \leq k$ and $k - n_2 \leq n_1 \leq k$. The total number of agents is then $n = n_1 + k$. We consider the profile $\succ \in \mathcal{L}^n(n)$, where $S = f(\succ)$, and

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ for some $i, j, \ell \in [3]$ with $|i-j| < |i-\ell|$;

(ii) $b \succ_a c$ whenever $a \in A_2, b \in A_3, c \in A_1$;

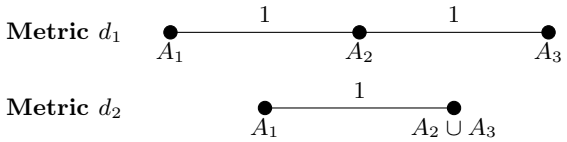


Figure 4.2: Metrics considered in the proof of Theorem 4.7. In this and all similar figures throughout the paper, the (sets of) agents are represented by circles, with the identity of the agents or sets below them, and the distances between them are written on top of the corresponding line segments. All figures consider indistinguishable metrics for a certain preference profile of the agents and thus any voting rule must select the same subsets for any of these metrics.

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A, k, \succ)$ with $A = [n]$.

In what follows, we distinguish whether or not f selects k' or more agents from A_3, where $k' \in [k - n_2]$ will be fixed later, and construct appropriate distance metrics to show that, in either case, the distortion is at least the one claimed in the statement. Intuitively, if f selects many agents from S_3 (more than k'), we will consider A_1, A_2, and A_3 from left to right, each at a distance of 1 from each other, so that picking all agents from A_2 and some agents from A_1 would lead to a lower or equal social cost (because $|A_1| \geq |A_3|$). If f selects few agents from A_3 (less than k'), we will consider A_2 and A_3 at the same location and a distance of 1 from A_1, so that picking all k agents from $A_2 \cup A_3$ would lead to a lower or equal social cost (because $|A_1| \leq |A_2 \cup A_3|$).

Formally, we first consider the case with $|S \cap A_3| \geq k'$ and define the distance metric d_1 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1$, $x_a = 1$ for every $a \in A_2$, and $x_a = 2$ for every $a \in A_3$. Note that $d_1 \rhd \succ$; see Figure 4.2 for an illustration. Since $|A_1| = n_1 \geq k - n_2 = |A_3|$ by assumption, defining $T \subset A$ such that $|A_1 \cap T| = k - k' - n_2$, $|A_2 \cap T| = n_2$, and $|A_3 \cap T| = k'$, we know that

$$\mathrm{SC}(S, A; d_1) \geq \mathrm{SC}(T, A; d_1) = n_1(n_2 + 2k') + n_2(k - n_2) + (k - n_2)(n_2 + 2(k - k' - n_2)).$$

If we consider the alternative committee S' such that $|A_1 \cap S'| = k - n_2$, $|A_2 \cap S'| = n_2$, and $|A_3 \cap S'| = 0$, we have $\mathrm{SC}(S', a; d_1) = n_2$ for every $a \in A_1$, $\mathrm{SC}(S', a; d_1) = k - n_2$ for every $a \in A_2$, and $\mathrm{SC}(S', a; d_1) = n_2 + 2(k - n_2)$ for every $a \in A_3$. We obtain

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\mathrm{SC}(S', A; d_1)} &\geq \frac{n_1(n_2 + 2k') + n_2(k - n_2) + (k - n_2)(n_2 + 2(k - k' - n_2))}{n_1 n_2 + n_2(k - n_2) + (k - n_2)(n_2 + 2(k - n_2))} \\
&= 1 + \frac{2k'(n_1 + n_2 - k)}{n_1 n_2 + n_2(k - n_2) + (k - n_2)(n_2 + 2(k - n_2))} \\
&= 1 + \frac{2k'(n_1 + n_2 - k)}{n_1 n_2 - 2k n_2 + 2k^2}. \qquad (4.9)
\end{aligned}$$

We now consider the case with $|S \cap A_3| \leq k' - 1$ and define the distance metric d_2 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1 \cup A_2$ and $x_a = 1$ for every $a \in A_3$. Note that $d_2 \rhd \succ$; see Figure 4.2 for an illustration. Since $|A_1| = n_1 \leq$

$k = |A_2 \cup A_3|$ by assumption, defining $T \subset A$ such that $|A_1 \cap T| = k - (k'-1) - n_2$ and $|(A_2 \cup A_3) \cap T| = n_2 + (k'-1)$, we know that

$$\mathrm{SC}(S, A; d_2) \geq \mathrm{SC}(T, A; d_2) = n_1(n_2 + (k'-1)) + k(k - (k'-1) - n_2).$$

If we consider the alternative committee S' such that $|A_1 \cap S'| = 0$ and $|(A_2 \cup A_3) \cap S'| = k$, we have $\mathrm{SC}(S', a; d_2) = k$ for every $a \in A_1$ and $\mathrm{SC}(S', a; d_2) = 0$ for every $a \in A_2 \cup A_3$. We obtain

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\mathrm{SC}(S', A; d_2)} &\geq \frac{n_1(n_2 + (k'-1)) + k(k - (k'-1) - n_2)}{n_1 k} \\ &= 1 + \frac{(k - n_1)(k - n_2 - k' + 1)}{k n_1}. \end{aligned} \tag{4.10}$$

We conclude that $\mathrm{dist}(f(\succ), \mathcal{E}) \geq 1 + \gamma$, where γ is the optimal value of the following optimization program:

$$\begin{aligned} \max \quad & \min\left\{ \frac{2k'(n_1 + n_2 - k)}{n_1 n_2 - 2k n_2 + 2k^2}, \frac{(k - n_1)(k - n_2 - k' + 1)}{k n_1} \right\} \\ \text{s.t.} \quad & n_1, n_2 \leq k, \\ & k' + n_2 \leq k, \\ & n_1 + n_2 \geq k, \\ & k', n_1, n_2 \in \mathbb{N}. \end{aligned} \tag{4.11}$$

Intuitively, for large enough k it is not hard to see that the above program has a strictly positive objective value, leading to a distortion strictly greater than one, while for small values of k in can be solved computationally by enumerating all possible solutions. Naturally, the larger the lower bound we choose for k, the better the bound is.

Formally, to obtain the claim bound of 1.0914, it suffices to assume $k \geq 200$. Indeed, for any fixed $k \geq 200$, we take

$$n_1 = \left\lceil \frac{3}{5}k \right\rceil, \quad n_2 = \left\lceil \frac{13}{20}k \right\rceil, \quad k' = \left\lceil \frac{1}{5}k \right\rceil.$$

These values trivially satisfy the constraints of the program (4.11). Moreover, since $k \geq 200$ we have

$$n_1 \leq \frac{3}{5}k + 1 \leq \frac{121}{200}k, \quad n_2 \leq \frac{13}{20}k + 1 \leq \frac{131}{200}k, \quad k' \leq \frac{1}{5}k + 1 \leq \frac{41}{200}k. \tag{4.12}$$

Therefore,

$$\frac{2k'(n_1 + n_2 - k)}{n_1 n_2 - 2k n_2 + 2k^2} \geq \frac{2 \cdot \frac{1}{5}k\left(\frac{3}{5}k + \frac{13}{20}k - k\right)}{\frac{121}{200}k \cdot \frac{13}{20}k - 2k \cdot \frac{13}{20}k + 2k^2} = \frac{\frac{2}{5} \cdot \frac{1}{4}}{\frac{1573 - 5200 + 8000}{4000}} = \frac{400}{4373} \approx 0.09147,$$

where the first inequality follows from (4.12) and the fact that $n_1 n_2 - 2k n_2$ is decreasing in n_2 since $n_1 < 2k$. On the other hand,

$$\frac{(k - n_1)(k - n_2 - k' + 1)}{k n_1} \geq \frac{(k - \frac{121}{200}k)(k - \frac{131}{200}k - \frac{41}{200}k)}{\frac{121}{200}k^2} = \frac{\frac{79}{200} \cdot \frac{28}{200}}{\frac{121}{200}} = \frac{2212}{24200} \approx 0.09140,$$

where the first inequality again follows from (4.12). Combining these two inequalities, we obtain $\mathrm{dist}(f(\succ), \mathcal{E}) > 1.0914$, concluding the proof for $k \geq 200$.

For values $k \leq 200$, we include in Table 4.2 feasible (and even optimal) solutions for the program (4.11) alongside their objective value. Since all objective values are strictly greater than 0.0914, this completes the proof. □

4.3.2 Utilitarian q-Cost

In this section, we study the distortion of voting rules in the context of utilitarian q-cost, in which the cost of a committee S' for an agent is given by its distance to the qth closest agent in S', and the social cost of a committee is the sum of its cost for all agents. Formally, for a set of agents A, a committee size k, a committee $S' \in \binom{A}{k}$, and a distance metric d, the social cost of the committee is given by

$$\mathrm{SC}(S', A; d) = \sum_{a \in A} \tilde{d}(a)_q,$$

where $\tilde{d}(a) \in \mathbb{R}_+^S$ contains the values $\{d(a, s) \mid s \in S'\}$ in increasing order.

Similarly to the setting with disjoint voters and candidates, the distortion of voting rules heavily depends on the value of q. The key intuition is that when q is large, the q-cost objective emphasizes the lower-ranked positions in each agent's ranking, which leads to a higher optimal cost. This makes the problem easier, as the selected committee can better approximate the worst-case distances. In contrast, when q is small, the objective focuses on the top few distances, so there may exist very good committees that are difficult to identify with only ordinal information, leading to greater potential distortion. Indeed, a result by Caragiannis, Shah, and Voudouris (2022) implies the existence of (n, k)-voting rules with distortion 3 for q-cost whenever $q > \frac{k}{2}$, since their result holds in the setting with disjoint voters and candidates and general distance metrics. We complement this result with a lower bound that ranges from $\frac{3}{2}$ to 2 as q varies between $\lceil \frac{k}{2} \rceil + 1$ and k. For $q \leq \frac{k}{2}$, Caragiannis, Shah, and Voudouris (2022) showed that no rule provides a bounded distortion; we prove that this still holds in our setting.

In Section 4.3.2, we study the case where $q = k = 2$ in further detail and achieve the best-possible distortions of $\frac{4}{3}$ and 2 for odd and even n, respectively, through natural voting rules that are able to leverage the different objectives involved in the problem.

Impossibility Results

In this section, we provide two strong impossibilities regarding distortion bounds for q-cost, analyzing the cases with $q \leq \frac{k}{2}$ and with $q \geq \lceil \frac{k}{2} \rceil + 1$ separately.

We begin with a strong impossibility for the case where we focus, for each agent, on their qth closest selected agent with $q \leq \frac{k}{2}$. We show that no constant distortion is possible in this setting, regardless of the number of agents to select.

Theorem 4.8. *For every $k \in \mathbb{N}$ with $k \geq 2$ and $q \in \mathbb{N}$ with $q \leq \frac{k}{2}$, there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n, k)-voting rule f, $\mathrm{dist}(f)$ is unbounded for utilitarian q-cost.*

k	(n_1, n_2, k')	γ	k	(n_1, n_2, k')	γ	k	(n_1, n_2, k')	γ	k	(n_1, n_2, k')	γ
3	$(2, 2, 1)$	0.1667	53	$(30, 37, 10)$	0.0998	103	$(63, 67, 21)$	0.0974	153	$(93, 99, 32)$	0.0970
4	$(2, 3, 1)$	0.1429	54	$(32, 36, 11)$	0.0995	104	$(62, 68, 22)$	0.0977	154	$(90, 104, 30)$	0.0969
5	$(3, 3, 2)$	0.1333	55	$(34, 36, 11)$	0.0996	105	$(64, 68, 22)$	0.0976	155	$(92, 104, 30)$	0.0969
6	$(3, 5, 1)$	0.1481	56	$(33, 37, 12)$	0.0996	106	$(61, 72, 21)$	0.0974	156	$(94, 103, 31)$	0.0970
7	$(3, 6, 1)$	0.1250	57	$(35, 36, 13)$	0.0992	107	$(63, 72, 21)$	0.0978	157	$(96, 103, 31)$	0.0970
8	$(4, 6, 2)$	0.1250	58	$(35, 38, 12)$	0.0986	108	$(65, 72, 21)$	0.0978	158	$(98, 101, 33)$	0.0969
9	$(4, 7, 2)$	0.1250	59	$(34, 40, 12)$	0.0997	109	$(67, 71, 22)$	0.0978	159	$(95, 106, 31)$	0.0967
10	$(6, 7, 2)$	0.1176	60	$(36, 40, 12)$	0.1000	110	$(64, 75, 21)$	0.0974	160	$(97, 105, 32)$	0.0967
11	$(5, 8, 3)$	0.1091	61	$(38, 39, 13)$	0.0992	111	$(66, 74, 22)$	0.0974	161	$(99, 105, 32)$	0.0968
12	$(7, 9, 2)$	0.1185	62	$(35, 43, 12)$	0.0995	112	$(68, 74, 22)$	0.0975	162	$(101, 104, 33)$	0.0969
13	$(5, 11, 2)$	0.1121	63	$(37, 43, 12)$	0.0992	113	$(70, 73, 23)$	0.0975	163	$(100, 104, 35)$	0.0966
14	$(9, 9, 3)$	0.1086	64	$(39, 42, 13)$	0.0992	114	$(69, 73, 25)$	0.0973	164	$(97, 109, 33)$	0.0969
15	$(8, 11, 3)$	0.1154	65	$(41, 41, 14)$	0.0991	115	$(69, 76, 23)$	0.0971	165	$(99, 110, 32)$	0.0970
16	$(10, 11, 3)$	0.1111	66	$(38, 45, 13)$	0.0986	116	$(68, 77, 24)$	0.0974	166	$(101, 108, 34)$	0.0969
17	$(9, 12, 4)$	0.1046	67	$(40, 45, 13)$	0.0986	117	$(70, 77, 24)$	0.0976	167	$(103, 107, 35)$	0.0967
18	$(9, 14, 3)$	0.1111	68	$(42, 44, 14)$	0.0986	118	$(72, 76, 25)$	0.0975	168	$(100, 112, 33)$	0.0967
19	$(11, 13, 4)$	0.1078	69	$(41, 46, 14)$	0.0990	119	$(69, 81, 23)$	0.0974	169	$(102, 112, 33)$	0.0968
20	$(13, 13, 4)$	0.1069	70	$(43, 45, 15)$	0.0987	120	$(71, 81, 23)$	0.0974	170	$(104, 111, 34)$	0.0968
21	$(12, 14, 5)$	0.1071	71	$(40, 48, 15)$	0.0982	121	$(73, 80, 24)$	0.0974	171	$(106, 110, 35)$	0.0968
22	$(12, 16, 4)$	0.1053	72	$(42, 49, 14)$	0.0991	122	$(75, 80, 24)$	0.0975	172	$(103, 114, 34)$	0.0965
23	$(14, 15, 5)$	0.1038	73	$(44, 49, 14)$	0.0989	123	$(77, 78, 26)$	0.0971	173	$(105, 114, 34)$	0.0966
24	$(13, 17, 5)$	0.1058	74	$(46, 47, 16)$	0.0987	124	$(74, 83, 24)$	0.0971	174	$(104, 114, 36)$	0.0967
25	$(15, 17, 5)$	0.1067	75	$(43, 52, 14)$	0.0985	125	$(76, 82, 25)$	0.0972	175	$(109, 113, 35)$	0.0967
26	$(15, 18, 5)$	0.1020	76	$(45, 51, 15)$	0.0984	126	$(78, 82, 25)$	0.0972	176	$(103, 118, 35)$	0.0966
27	$(14, 20, 5)$	0.1032	77	$(47, 51, 15)$	0.0984	127	$(77, 82, 27)$	0.0971	177	$(105, 119, 34)$	0.0968
28	$(16, 20, 5)$	0.1042	78	$(49, 50, 16)$	0.0986	128	$(74, 86, 26)$	0.0969	178	$(107, 118, 35)$	0.0968
29	$(18, 19, 6)$	0.1041	79	$(48, 51, 17)$	0.0981	129	$(76, 86, 26)$	0.0973	179	$(109, 118, 35)$	0.0968
30	$(17, 20, 7)$	0.1020	80	$(48, 53, 16)$	0.0979	130	$(78, 86, 26)$	0.0974	180	$(111, 116, 37)$	0.0967
31	$(17, 22, 6)$	0.1030	81	$(47, 54, 17)$	0.0982	131	$(80, 85, 27)$	0.0973	181	$(108, 121, 35)$	0.0966
32	$(19, 22, 6)$	0.1021	82	$(49, 54, 17)$	0.0986	132	$(77, 90, 25)$	0.0971	182	$(110, 120, 36)$	0.0966
33	$(21, 21, 7)$	0.1022	83	$(51, 53, 18)$	0.0983	133	$(79, 89, 26)$	0.0971	183	$(112, 120, 36)$	0.0967
34	$(20, 23, 7)$	0.1029	84	$(48, 58, 16)$	0.0982	134	$(81, 89, 26)$	0.0972	184	$(114, 119, 37)$	0.0967
35	$(22, 22, 8)$	0.1013	85	$(50, 57, 17)$	0.0983	135	$(83, 88, 27)$	0.0972	185	$(116, 117, 39)$	0.0965
36	$(22, 24, 7)$	0.1006	86	$(52, 57, 17)$	0.0983	136	$(85, 86, 29)$	0.0971	186	$(110, 123, 38)$	0.0966
37	$(21, 26, 7)$	0.1029	87	$(54, 56, 18)$	0.0983	137	$(82, 91, 27)$	0.0969	187	$(112, 123, 38)$	0.0967
38	$(23, 25, 8)$	0.1024	88	$(51, 60, 17)$	0.0979	138	$(84, 91, 27)$	0.0969	188	$(114, 122, 39)$	0.0967
39	$(25, 24, 9)$	0.1005	89	$(53, 60, 17)$	0.0978	139	$(83, 91, 29)$	0.0971	189	$(111, 127, 37)$	0.0966
40	$(22, 28, 8)$	0.1015	90	$(55, 59, 18)$	0.0979	140	$(88, 90, 28)$	0.0971	190	$(113, 127, 37)$	0.0966
41	$(24, 28, 8)$	0.1013	91	$(57, 59, 18)$	0.0980	141	$(82, 95, 28)$	0.0970	191	$(115, 126, 38)$	0.0966
42	$(26, 27, 9)$	0.1009	92	$(56, 59, 20)$	0.0978	142	$(84, 95, 28)$	0.0971	192	$(117, 126, 38)$	0.0967
43	$(23, 31, 8)$	0.1009	93	$(56, 62, 18)$	0.0974	143	$(86, 95, 28)$	0.0972	193	$(119, 125, 39)$	0.0967
44	$(27, 28, 10)$	0.1002	94	$(55, 63, 19)$	0.0981	144	$(88, 94, 29)$	0.0972	194	$(116, 129, 38)$	0.0965
45	$(27, 30, 9)$	0.1000	95	$(57, 63, 19)$	0.0982	145	$(90, 92, 31)$	0.0969	195	$(118, 129, 38)$	0.0965
46	$(26, 31, 10)$	0.1003	96	$(59, 62, 20)$	0.0980	146	$(87, 97, 29)$	0.0969	196	$(120, 128, 39)$	0.0965
47	$(28, 31, 10)$	0.1011	97	$(56, 66, 19)$	0.0978	147	$(89, 97, 29)$	0.0969	197	$(122, 128, 39)$	0.0966
48	$(30, 30, 11)$	0.1000	98	$(58, 66, 19)$	0.0978	148	$(91, 96, 30)$	0.0970	198	$(124, 125, 42)$	0.0964
49	$(27, 34, 10)$	0.0998	99	$(60, 65, 20)$	0.0978	149	$(93, 95, 31)$	0.0970	199	$(118, 132, 40)$	0.0966
50	$(29, 34, 10)$	0.1005	100	$(62, 65, 20)$	0.0979	150	$(92, 95, 33)$	0.0967	200	$(120, 133, 39)$	0.0967
51	$(31, 34, 10)$	0.1004	101	$(59, 68, 20)$	0.0974	151	$(89, 100, 31)$	0.0969			
52	$(33, 33, 11)$	0.0997	102	$(61, 68, 20)$	0.0974	152	$(91, 100, 31)$	0.0970			

Table 4.2: Optimal solutions for the program (4.11) for all integer values of k between 3 and 200, alongside their objective value.

Proof. We let k and q be as in the statement, fix $n \in \mathbb{N}$ to a large value, in particular with $n \geq 2k + q$ (we will ultimately take the limit $n \to \infty$), and consider an arbitrary (n, k)-voting rule f. We denote $p = \left\lfloor \frac{k}{q} \right\rfloor \geq 2$ and partition the agents into $p + 1$ sets $A = \dot{\bigcup}_{i=1}^{p} A_i \cup B$, such that $|A_i| \in \left\{ \left\lfloor \frac{n-q}{p} \right\rfloor, \left\lceil \frac{n-q}{p} \right\rceil \right\}$ for every $i \in [p]$ and $|B| = q$. Note that this is possible since

$$p \left\lfloor \frac{n-q}{p} \right\rfloor + q \leq n \leq p \left\lfloor \frac{n-q}{p} \right\rfloor + q.$$

We consider the profile $\succ \in \mathcal{L}^n(n)$, where

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ for some $i, j, \ell \in [p]$ with $|i - j| < |i - \ell|$;

(ii) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in B$ for some $i, j \in [p]$;

(iii) $b \succ_a c$ whenever $a, b \in B, c \in A_i$ for some $i \in [p]$;

(iv) $b \succ_a c$ whenever $a \in B, b \in A_i, c \in A_j$ for some $i, j \in [p]$ with $i > j$;

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A, k, \succ)$ with $A = [n]$.

In what follows, we distinguish whether f selects all q agents in B or not and construct appropriate distance metrics to show that, in either case, the distortion can be arbitrarily large. Intuitively, if f selects B we will consider this set to be relatively close to A_p, so that picking q agents from each set $A_1, \ldots, A_p$ would give a much lower social cost. On the contrary, if f does not select B, we will place this set extremely far from all others, so that the social cost of the selected set is huge compared to the social cost of a committee containing B.

Formally, we first consider the case with $B \subseteq S$ and define the distance metric d_1 on A given by the following positions $x \in (-\infty, \infty)^n$: $x_a = i - 1$ for every $a \in A_i$ and every $i \in [p]$, and $x_a = 2(p - 1)$ for every $a \in B$. Note that $d_1 \rhd \succ$; see Figure 4.3 for an illustration. Since $B \subseteq S$, we have that $\left| S \cap \bigcup_{i \in [p]} A_i \right| \leq k - q$. Hence, from an averaging argument, there exists $j \in [p]$ with

$$|S \cap A_j| \leq \frac{k-q}{p} = \frac{q}{k}(k - q) < q.$$

From the definition of q-cost, we thus have

$$\mathrm{SC}(S, a; d_1) \geq \min\{d_1(a, b) \mid b \in A \setminus A_j\} \geq 1 \quad \text{for every } a \in A_j. \tag{4.13}$$

On the other hand, consider the set $S = \cup_{i \in [p]} S_i$, where $S_i \subseteq A_i$ and $|S_i| \geq q$ for every $i \in [p]$. Note that this set exists because $pq = k$ and

$$|A_i| \geq \left\lfloor \frac{n-q}{p} \right\rfloor \geq \left\lfloor \frac{2k}{k} q \right\rfloor \geq q,$$

where we used our assumption $n \geq 2k + q$. From the definition of q-cost, we have that $\mathrm{SC}(S, a; d_1) = 0$ for every $a \in A_i$ and every $i \in [p]$. For each $a \in B$, we have $\mathrm{SC}(S, a; d_1) = p - 1$. Combining these facts with inequality (4.13), we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\mathrm{SC}(S, A; d_1)} \geq \frac{|A_j|}{(p-1)|B|} \geq \left\lfloor \frac{n-q}{p} \right\rfloor \frac{1}{(p-1)q} = \left\lfloor \frac{(n-q)q}{k} \right\rfloor \cdot \frac{1}{k-q}.$$

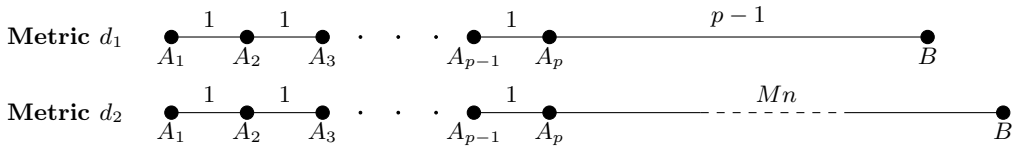


Figure 4.3: Metrics considered in the proof of Theorem 4.8. In this and all similar figures throughout the paper, the (sets of) agents are represented by circles, with the identity of the agents or sets below them, and the distances between them are written on top of the corresponding line segments. All figures consider indistinguishable metrics for a certain preference profile of the agents and thus any voting rule must select the same subsets for any of these metrics.

We now consider the case with $B \not\subseteq S$ and define the distance metric d_2 on A given by the following positions $x \in (-\infty, \infty)^n$: $x_a = i - 1$ for every $a \in A_i$ and every $i \in [p]$, and $x_a = p - 1 + Mn$ for every $a \in B$. Note that $d_2 \rhd \succ$; see Figure 4.3 for an illustration. Since $B \not\subseteq S$, we have that $|S \cap B| < q$ and thus, by the definition of q-cost, we have

$$\mathrm{SC}(S, a; d_2) \geq \min\{d_2(a, b) \mid b \in A \setminus B\} \geq Mn \quad \text{for every } a \in B. \tag{4.14}$$

On the other hand, consider the set $T = B \cup \cup_{i \in [p-1]} T_i$, where $T_i \subseteq A_i$ and $|T_i| \geq q$ for every $i \in [p-1]$. Note that this set exists because $(p-1)q = k - q$ and

$$|A_i| \geq \left\lfloor \frac{n-q}{p} \right\rfloor \geq \left\lfloor \frac{2k}{k} q \right\rfloor \geq q,$$

where we used our assumption $n \geq 2k + q$. From the definition of q-cost, we have that $\mathrm{SC}(T, a; d_2) = 0$ for every $a \in A_i$ and every $i \in [p-1]$ and $\mathrm{SC}(T, a; d_2) = 0$ for every $a \in B$. For each $a \in A_p$, we have $\mathrm{SC}(T, a; d_2) = 1$. Combining these facts with inequality (4.14), we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\mathrm{SC}(T, A; d_2)} \geq \frac{Mn|B|}{|A_p|} \geq \frac{1}{\left\lceil \frac{n-q}{p} \right\rceil} Mnq = \frac{1}{\left\lceil \frac{(n-q)q}{k} \right\rceil} Mnq.$$

Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq \left\lfloor \frac{(n-q)q}{k} \right\rfloor \cdot \frac{1}{k-q}$ if $B \subseteq S$ and $\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{1}{\left\lceil \frac{(n-q)q}{k} \right\rceil} Mnq$ otherwise, we conclude that

$$\mathrm{dist}(f) \geq \min\left\{ \left\lfloor \frac{(n-q)q}{k} \right\rfloor \cdot \frac{1}{k-q}, \frac{1}{\left\lceil \frac{(n-q)q}{k} \right\rceil} Mnq \right\},$$

which can be unbounded by taking n and M arbitrarily large. □

Next, we prove a lower bound of $2 - \frac{k-q}{4q-k-3}$ for the distortion of any voting rule for utilitarian q-cost when $\left\lceil \frac{k}{2} \right\rceil < q \leq k$ and $k \geq 3$.

Theorem 4.9. *For every $k \in \mathbb{N}$ with $k \geq 3$ and $q \in \mathbb{N}$ with $\frac{k}{2} + 1 \leq q \leq k$, there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n, k)-voting rule f, $\mathrm{dist}(f)$ is at least $2 - \frac{k-q}{4q-k-3}$ for utilitarian q-cost.*

Proof. We let k and q be as in the statement and fix $n = 2(3q - k - 2)$, and consider an arbitrary (n, k)-voting rule f. We partition the agents into four sets $A = \dot{\bigcup}_{i=1}^{4} A_i$ such that $|A_1| = |A_4| = q - 1$ and $|A_2| = |A_3| = 2q - k - 1$. Note that all these values lie between 1 and $q - 1$. Indeed, this is trivial for $|A_1|$ and $|A_4|$, whereas for $|A_2|$ and $|A_3|$ we have $2q - k - 1 \geq 2\left(\frac{k}{2} + 1\right) - k - 1 = 1$ and $2q - k - 1 \leq 2q - q - 1 = q - 1$, where we have used that q lies between $\frac{k}{2} + 1$ and k.

We consider the profile $\succ \in \mathcal{L}^n(n)$, where

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ for some $i, j, \ell \in [4]$ with $|i - j| < |i - \ell|$;

(ii) $b \succ_a c$ whenever $a \in A_2, b \in A_1, c \in A_3$;

(iii) $b \succ_a c$ whenever $a \in A_3, b \in A_4, c \in A_2$;

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A, k, \succ)$ with $A = [n]$.

In what follows, we distinguish whether f selects q or more agents from $A_1 \cup A_2$, from $A_3 \cup A_4$, or from none of them, and construct appropriate distance metrics to show that, in either case, the distortion is at least the one claimed in the statement. Intuitively, if f selects less than q agents from both $A_1 \cup A_2$ and from $A_3 \cup A_4$, we will consider $A_1 \cup A_2$ on one extreme and $A_3 \cup A_4$ on the other, so that picking q agents from any of these sets would lead to a lower social cost. If f selects q or more agents from $A_1 \cup A_2$ we will consider a metric where A_1 lies in one extreme, A_2 in the middle, and both A_3 and A_4 in the other extreme, so that picking all agents from A_4 would lead to a lower social cost. If f selects q or more agents from $A_3 \cup A_4$, we will construct a symmetric instance.

Formally, we first consider the case with $|S \cap (A_1 \cup A_2)| < q$ and $|S \cap (A_3 \cup A_4)| < q$ and define the distance metric d_1 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1 \cup A_2$ and $x_a = 2$ for every $a \in A_3 \cup A_4$. Note that $d_1 \rhd \succ$; see Figure 4.4.(b) for an illustration. It is clear that $\mathrm{SC}(S, a; d_1) = 2$ for every $a \in A$. If we consider the alternative committee $S' = A_1 \cup A_2 \in \binom{A}{k}$, we have $\mathrm{SC}(S', a; d_1) = 0$ for every $a \in A_1 \cup A_2$ and $\mathrm{SC}(S', a; d_1) = 2$ for every $a \in A_3 \cup A_4$. We obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\mathrm{SC}(S', A; d_1)} = \frac{2 \cdot n}{2\frac{n}{2}} = 2.$$

If $|S \cap (A_3 \cup A_4)| \geq q$, we define the distance metric d_2 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1 \cup A_2$, $x_a = 1$ for every $a \in A_3$, and $x_a = 2$ for every $a \in A_4$. Note that $d_2 \rhd \succ$; see Figure 4.4.(b) for an illustration. Since $|S \cap (A_1 \cup A_2 \cup A_3)| \leq (k - q) + |A_3| = q - 1 < q$, we have that $\mathrm{SC}(S, a; d_2) = 2$ for every $a \in A_1 \cup A_2$. Furthermore, since both $|A_3| < q$ and $|A_4| < q$, we have that $\mathrm{SC}(S, a; d_2) = 1$ for every $a \in A_3 \cup A_4$. If we consider an alternative committee $S' \subseteq A_1 \cup A_2 \in \binom{A}{k}$, which exists due to $|A_1 \cup A_2| = 3q - k - 2 \geq q$, we have $\mathrm{SC}(S', a; d_2) = 0$ for every $a \in A_1 \cup A_2$,

$\mathrm{SC}(S', a; d_2) = 1$ for every $a \in A_3$, and $\mathrm{SC}(S', a; d_2) = 2$ for every $a \in A_4$. Thus, we obtain

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}) &\geq \frac{\mathrm{SC}(S, A; d_2)}{\mathrm{SC}(S', A; d_2)} \\ &= \frac{2|A_1 \cup A_2| + |A_3 \cup A_4|}{|A_3| + 2|A_4|} \\ &= \frac{3(3q - k - 2)}{(2q - k - 1) + 2(q - 1)} \\ &= 2 - \frac{k - q}{4q - k - 3}.\end{aligned}$$

Analogously, if $|S \cap (A_1 \cup A_2)| \geq q$, we define the distance metric d_3 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1$, $x_a = 1$ for every $a \in A_2$, and $x_a = 2$ for every $a \in A_3 \cup A_4$. Note that $d_3 \rhd \succ$; see Figure 4.4.(b) for an illustration. Since $|S \cap (A_2 \cup A_3 \cup A_4)| \leq (k - q) + |A_2| = q - 1 < q$, we have that $\mathrm{SC}(S, a; d_3) = 2$ for every $a \in A_3 \cup A_4$. Furthermore, since both $|A_1| < q$ and $|A_2| < q$, we have that $\mathrm{SC}(S, a; d_3) = 1$ for every $a \in A_1 \cup A_2$. If we consider an alternative committee $S' \subseteq A_3 \cup A_4 \in \binom{A}{k}$, which exists due to $|A_3 \cup A_4| = 3q - k - 2 \geq q$, we have $\mathrm{SC}(S', a; d_3) = 0$ for every $a \in A_3 \cup A_4$, $\mathrm{SC}(S', a; d_3) = 1$ for every $a \in A_2$, and $\mathrm{SC}(S', a; d_3) = 2$ for every $a \in A_1$. Thus, we obtain

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}) &\geq \frac{\mathrm{SC}(S, A; d_3)}{\mathrm{SC}(S', A; d_3)} \\ &= \frac{2|A_3 \cup A_4| + |A_1 \cup A_2|}{|A_2| + 2|A_1|} \\ &= \frac{3(3q - k - 2)}{(2q - k - 1) + 2(q - 1)} \\ &= 2 - \frac{k - q}{4q - k - 3}.\end{aligned}$$

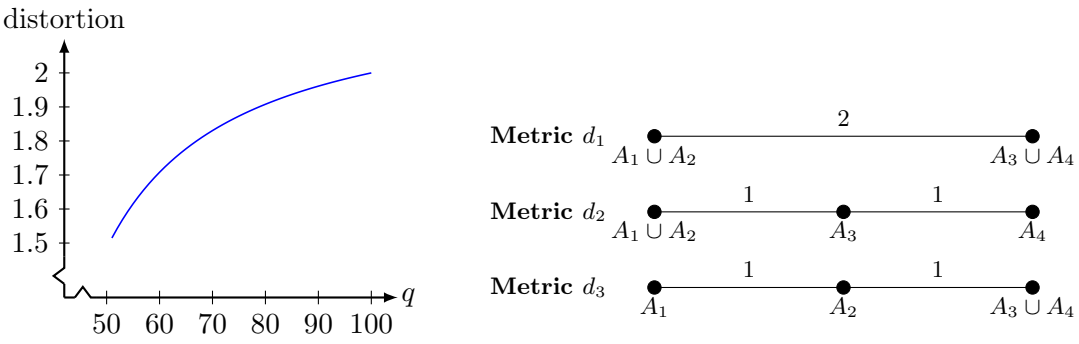
Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq 2 - \frac{k-q}{4q-k-3}$ regardless of $f(\succ)$, we conclude that $\mathrm{dist}(f) \geq 2 - \frac{k-q}{4q-k-3}$. □

The lower bound provided in this theorem increases in q and varies between $\frac{3}{2} + \frac{3}{2(k+1)}$ for $q = \frac{k}{2} + 1$ and 2 for $q = k$; Figure 4.4.(a) illustrates it for $k = 100$ and q between 51 and 100.

A Voting Rule for $k = 2$

In this section, we focus on the special case of utilitarian q-cost when $q = k = 2$. In this setting, the social cost of a committee S for an agent a is determined by the distance to the *farthest* agent in the committee S'. Formally, for a set of agents A and a committee $S' \in \binom{A}{2}$, the social cost is:

$$\mathrm{SC}(S', A; d) = \sum_{a \in A} \max_{s \in S'} d(a, s).$$



(a) Lower bounds for $k = 100$, $q \in \{51, \dots, 100\}$.

(b) Metrics considered in the proof.

Figure 4.4: Lower bound on the distortion of any rule for utilitarian q-cost stated in Theorem 4.9, and metrics used to prove it.

On an intuitive level, the goal is to select agents that are both close to each other and close to the median agent(s). In particular, note that the optimal committee always consists of two consecutive agents: For any committee of non-consecutive agents, replacing the most extreme agent among the selected one with another closer to the median cannot decrease the social cost.

A visual aid for computing the social cost of a committee is what we call *stair diagrams*, illustrated in Figure 4.5. The area below both *staircases* is a cost that every committee must incur. A specific committee $\{s_1, s_2\}$ must incur, in addition, a cost equal to the area of the rectangle whose basis is the line segment between both selected candidates and whose height is n (and potentially an additional area to reach this point from the median). The figure illustrates the common area incurred by any committee and the additional cost of two possible committees for each $n \in \{8, 9\}$. It also provides an intuition of the intrinsic difference between the cases with odd and even k. The following lemma bounds the social cost of any committee from below and provides intuition about this objective.

Lemma 4.10. *Let $\mathcal{E} = (A, k, \succ)$ be an election and $d \rhd \succ$ a consistent metric. Then, for every committee $S' = \{s_1, s_2\} \in \binom{A}{2}$,*

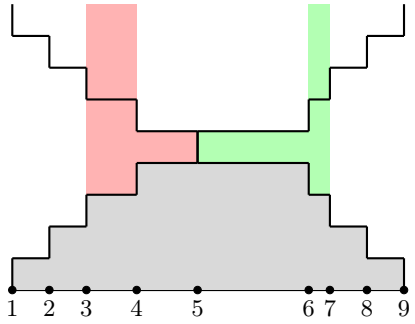
$$SC(S', A; d) \geq \begin{cases} \sum_{i=1}^{\frac{n-1}{2}} d(i, n-i+1) + \frac{n-1}{2} \cdot d(s_1, s_2) + SC\big(S', \{\frac{n+1}{2}\}; d\big) & \text{if } n \text{ is odd,} \\ \sum_{i=1}^{\frac{n}{2}} d(i, n-i+1) + \frac{n}{2} \cdot d(s_1, s_2) & \text{if } n \text{ is even.} \end{cases}$$

Proof. Let $\mathcal{E} = (A, k, \succ)$ with $A = [n]$ and d be as in the statement and $S' = \{s_1, s_2\} \in \binom{A}{k}$ an arbitrary committee. We assume that $s_1 < s_2$ w.l.o.g.. Let $i \in \{1, \dots, \lfloor \frac{n}{2} \rfloor\}$ be a fixed agent. If $i \leq s_1 < s_2 \leq n - i + 1$, we have that the cost of the committee for agents i and $n - i + 1$ is at least

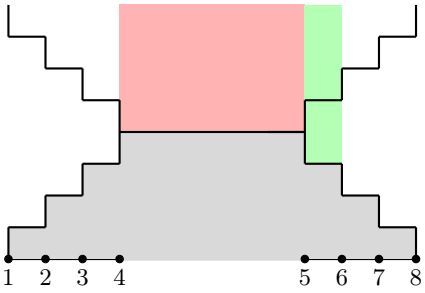
$$\mathrm{SC}(S', i; d) + \mathrm{SC}(S', n-i+1; d) = d(i, s_2) + d(s_1, n-i+1) \geq d(i, n-i+1) + d(s_1, s_2).$$

Similarly, if $s_2 < i$, we have

$$\mathrm{SC}(S', i; d) + \mathrm{SC}(S', n-i+1; d) = d(s_1, i) + d(s_1, n-i+1) \geq d(i, n-i+1) + d(s_1, s_2),$$



(a) Stair diagram for $n = 9$. The red area corresponds to the committee $\{3, 4\}$; the green area to $\{6, 7\}$.



(b) Stair diagram for $n = 8$. The red area corresponds to the committee $\{4, 5\}$; the green area to $\{5, 6\}$.

Figure 4.5: Stair diagrams for 9 and 8 agents. The common cost incurred by any committee is shown in gray; the additional cost of two specific committees is shown in red and green.

and if $s_1 > n - i + 1$,

$$\mathrm{SC}(S', i; d) + \mathrm{SC}(S', n - i + 1; d) = d(i, s_2) + d(n - i + 1, s_2) \geq d(i, n - i + 1) + d(s_1, s_2).$$

Summing up over all agents, we obtain

$$\mathrm{SC}(S', A; d) = \sum_{i=1}^{\frac{n}{2}} (\mathrm{SC}(S', i; d) + \mathrm{SC}(S', n - i + 1; d)) \geq \sum_{i=1}^{\frac{n}{2}} d(i, n - i + 1) + \frac{n}{2} d(s_1, s_2)$$

if n is even, and

$$\begin{aligned} \mathrm{SC}(S', A; d) &= \sum_{i=1}^{\frac{n-1}{2}} (\mathrm{SC}(S', i; d) + \mathrm{SC}(S', n - i + 1; d)) + \mathrm{SC}\left(S', \frac{n+1}{2}; d\right) \\ &\geq \sum_{i=1}^{\frac{n-1}{2}} d(i, n - i + 1) + \frac{n-1}{2} d(s_1, s_2) + \mathrm{SC}\left(S', \frac{n+1}{2}; d\right) \end{aligned}$$

if n is odd. □

In the remainder of this section, we establish tight distortion bounds of $\frac{4}{3}$ and 2 for odd and even values of n, respectively. To derive these bounds, we frequently utilize the inequalities above to bound the social cost of the optimal committee.

Odd number of agents We first focus on odd values of n. For $n = 3$, it is easy to see that the optimal set corresponds to the median agent and the agent that the median prefers among the others, which yields a simple rule with distortion 1. For $n \geq 5$ we introduce a voting rule called Favorite Couple. For an election $\mathcal{E} = (A, k, \succ)$, we say that agents $a, b \in A$ are a *couple* if they rank each other above all other agents; i.e., if $b \succ_a c$ and $a \succ_b c$ for every $c \in A \setminus \{a, b\}$. Note that each agent can take part in at most one couple. Favorite Couple selects the closest couple to the median when restricting to the five middle agents.

Voting Rule 2 (Favorite Couple). *For a preference profile $\succ$, compute the order from left to right $1, \dots, n$ and let $m = \frac{n+1}{2}$ be the median agent. If there is a couple among the sets $\{m-1, m\}$ and $\{m, m+1\}$, return it. Else, return $\{m+1, m+2\}$ if $m+2 \succ_m m-2$ and return $\{m-2, m-1\}$ otherwise.*

On an intuitive level, this voting rule selects two consecutive agents who are both close to each other and to the median agent. The restriction to middle agents is necessary; simply choosing an arbitrary couple can lead to a distortion of up to 2. For example, this is the case if there are n agents with distances $d(a, a+1) = 1 + a\varepsilon$ for all $a \in [n]$ and a small $\varepsilon > 0$, as the only couple is $\{1, 2\}$ with a social cost of approximately $\frac{n^2}{2}$, while the committee consisting of the median agent and any neighbor is close to $\frac{n^2}{4}$. We now show that this rule provides the best-possible distortion of $\frac{4}{3}$ for an odd number of agents. To establish the distortion of Favorite Couple, we address different cases depending on the set selected by this rule and the optimal set. In each of them, we can use the selection condition of the rule to bound the social cost of the set selected by it from above and the optimal social cost from below and achieve a ratio of at most $\frac{4}{3}$ between them. Intuitively, in situations like the one illustrated in Figure 4.5.(a), Favorite Couple can select a suboptimal committee (e.g. the red one instead of the green one), but this imposes several restriction on the distances, such as $d(3,4) \leq \min\{d(4,5), d(6,7)\}$ and $2d(3,4) \leq d(5,7)$ in this example. In particular, this implies lower bounds on the common cost incurred by any voting rule. We study and apply these inequalities carefully to obtain our guarantee.

For the lower bound, on the other hand, we consider $n = 5$ agents and two metrics illustrated in Figure 4.6. If a rule selects $\{1, 2\}$, under metric d_1 the agents' costs, from left to right, are 1, 1, 2, 4, and 4, accounting for a total cost of 12, while the optimal committee $\{4, 5\}$ induces a social cost of $4 + 3 + 2 + 0 + 0 = 9$. Similarly, note that if a rule selects $\{2, 3\}$, $\{3, 4\}$, or $\{4, 5\}$, then the social costs under metric d_2 are 8, 8, and 10, respectively, while the optimal committee $\{1, 2\}$ induces a social cost of 6. In all cases, the ratio is at least $\frac{4}{3}$, so the bound follows.

Theorem 4.11. *For every odd $n \geq 5$, Favorite Couple achieves a distortion of $\frac{4}{3}$ for utilitarian 2-cost. Moreover, there exists $n \in \mathbb{N}$ such that, for every $(n, 2)$-voting rule f, we have $\mathrm{dist}(f) \geq \frac{4}{3}$ for utilitarian 2-cost.*

Proof. We consider an arbitrary election $\mathcal{E} = (A, k, \succ)$ with $n \geq 5$ and $A = [n]$, and a consistent metric $d \rhd \succ$. We denote the five middle agents by $a_1, \dots, a_5$ from left to right, with a_3 being the median agent. We let S denote the committee selected by Favorite Couple and S^* denote the optimal committee for the metric d. We analyze two main cases, depending on whether the rule selects the median agent or not.

Case 1: $a_3 \in S$ w.l.o.g., we assume that $a_2 \succ_{a_3} a_4$, which implies that the selected committee is $S = \{a_2, a_3\}$. This implies that agents a_2 and a_3 form a couple, and both $d(a_2, a_3) \leq d(a_1, a_2)$ and $d(a_2, a_3) \leq d(a_3, a_4)$ hold. Therefore,

$$d(a_1, a_5) \geq 3 \cdot d(a_2, a_3), \quad d(a_2, a_4) \geq 2 \cdot d(a_2, a_3). \tag{4.15}$$

For each $i \leq \frac{n-1}{2}$, the joint cost of S for agents i and $n - i + 1$ is given by

$$\mathrm{SC}(S, i; d) + \mathrm{SC}(S, n-i+1; d) = d(i, a_3) + d(a_2, n-1+1) = d(i, n-i+1) + d(a_2, a_3).$$

Since the median agent incurs a cost of $\mathrm{SC}(A, a_3; d) = d(a_2, a_3)$, we obtain:

$$\begin{aligned}\mathrm{SC}(S, A; d) &= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_4) + \left(\frac{n-1}{2}\right) d(a_2, a_3) + d(a_2, a_3) \\ &= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+1}{2}\right) d(a_2, a_3) + d(a_2, a_4).\end{aligned}$$

On the other hand, by Lemma 4.10, we have:

$$\begin{aligned}\mathrm{SC}(S^*, A; d) &\geq \sum_{i=1}^{\frac{n-1}{2}} d(i, n-i+1) + \mathrm{SC}(\{a_3\}, A; d) \\ &\geq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_4) + d(a_2, a_3),\end{aligned}$$

where we used, for the second inequality, that the cost of the median agent is at least $d(a_2, a_3)$ due to the assumption that $a_2 \succ_{a_3} a_4$. Thus, the distortion is:

$$\begin{aligned}\mathrm{dist}(f) &= \frac{\mathrm{SC}(S, A; d)}{\mathrm{SC}(S^*, A; d)} \\ &\leq \frac{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+1}{2}\right) d(a_2, a_3) + d(a_2, a_4)}{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_3) + d(a_2, a_4)} \\ &\leq \frac{\left(\frac{n-3}{2}\right) \cdot 3 \cdot d(a_2, a_3) + \left(\frac{n+1}{2}\right) \cdot d(a_2, a_3) + 2 \cdot d(a_2, a_3)}{\left(\frac{n-3}{2}\right) \cdot 3 \cdot d(a_2, a_3) + d(a_2, a_3) + 2 \cdot d(a_2, a_3)} = \frac{\frac{4n-8}{2} + 2}{\frac{3n-9}{2} + 3} = \frac{4n-4}{3n-3} = \frac{4}{3},\end{aligned}$$

where the second inequality follows from inequalities (4.15) and the fact that $d(i, n-i+1) \geq d(1, 5)$ for every $i \leq \frac{n-3}{2}$. This concludes the proof for this case.

Case 2: $a_3 \notin S$ In this case, we either have $S = \{a_1, a_2\}$ or $S = \{a_4, a_5\}$; we assume the former w.l.o.g.. From the definition of Favorite Couple, this implies that $\{a_2, a_3\}$ and $\{a_3, a_4\}$ are not couples, so we must have $a_1 \succ_{a_2} a_3$ and $a_5 \succ_{a_4} a_3$. It also implies that $a_1 \succ_{a_3} a_5$, since $\{a_4, a_5\}$ would be selected otherwise. In terms of distances:

$$d(a_2, a_3) \geq d(a_1, a_2), \quad d(a_3, a_4) \geq d(a_4, a_5), \quad d(a_3, a_5) \geq d(a_1, a_3). \tag{4.16}$$

Similarly as before, the social cost of the selected committee is

$$\begin{aligned}\mathrm{SC}(S, A; d) &= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n-3}{2}\right) d(a_1, a_2) + d(a_1, a_2) + d(a_1, a_3) + d(a_1, a_4) \\ &= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+3}{2}\right) d(a_1, a_2) + d(a_2, a_3) + d(a_2, a_4).\end{aligned}$$

We now consider two cases depending on whether a_3 is in the optimal committee.

Case 2.1: $a_3 \in S^*$. If the median agent is selected in the optimal committee, we have from Lemma 4.10 that

$$\mathrm{SC}(S^*, A; d) \geq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_4) + \left(\frac{n-1}{2} + 1\right) \min\{d(a_2, a_3), d(a_3, a_4)\}. \tag{4.17}$$

We now claim that $\left(\frac{n-1}{2} + 1\right) \min\{d(a_2, a_3), d(a_3, a_4)\} \geq \frac{3}{2} d(a_1, a_3)$. Indeed, if we have $\min\{d(a_2, a_3), d(a_3, a_4)\} = d(a_2, a_3)$, this holds because $\frac{n-1}{2} + 1 \geq 3$ and, due to inequalities (4.16), $3d(a_2, a_3) \geq \frac{3}{2} d(a_1, a_3)$. If $\min\{d(a_2, a_3), d(a_3, a_4)\} = d(a_3, a_4)$, this holds because $\frac{n-1}{2} + 1 \geq 3$ and, due to inequalities (4.16), $3d(a_3, a_4) \geq \frac{3}{2} d(a_3, a_5) \geq \frac{3}{2} d(a_1, a_3)$.

Replacing in inequality (4.17), we obtain

$$\mathrm{SC}(S^*, A; d) \geq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_4) + \frac{3}{2} \cdot d(a_1, a_2) + \frac{3}{2} \cdot d(a_2, a_3).$$

Thus, the distortion is

$$\begin{aligned}
\mathrm{dist}(f) = \frac{\mathrm{SC}(S, A; d)}{\mathrm{SC}(S^*, A; d)} &\leq \frac{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+3}{2}\right) d(a_1, a_2) + d(a_2, a_3) + d(a_2, a_4)}{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_4) + \frac{3}{2} \cdot d(a_1, a_2) + \frac{3}{2} \cdot d(a_2, a_3)} \\
&\leq \frac{\left(\frac{n-3}{2}\right) \cdot 4 \cdot d(a_1, a_2) + \left(\frac{n+3}{2}\right) d(a_1, a_2) + d(a_1, a_2) + 2 \cdot d(a_1, a_2)}{\left(\frac{n-3}{2}\right) \cdot 4 \cdot d(a_1, a_2) + 2 \cdot d(a_1, a_2) + \frac{3}{2} \cdot d(a_1, a_2) + \frac{3}{2} \cdot d(a_1, a_2)} \\
&\leq \frac{\left(\frac{n-3}{2}\right) \cdot 4 + \left(\frac{n+3}{2}\right) + 1 + 2}{\left(\frac{n-3}{2}\right) \cdot 4 + 2 + \frac{3}{2} + \frac{3}{2}} \\
&= \frac{4n - 12 + n + 3 + 2 + 4}{4n - 12 + 4 + 3 + 3} = \frac{5n - 3}{4n - 2} \leq \frac{5}{4} \leq \frac{4}{3},
\end{aligned}$$

where the second inequality follows by applying inequalities (4.16) and the fact that $d(i, n-i+1) \geq d(1, 5)$ for every $i \leq \frac{n-3}{2}$. We conclude the distortion bound of $\frac{4}{3}$ for this case.

Case 2.2: $a_3 \notin S^*$. We begin by rewriting the social cost of S more conveniently as

$$\begin{aligned}
\mathrm{SC}(S, A; d) &= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n-3}{2}\right) d(a_1, a_2) + d(a_1, a_2) + d(a_1, a_3) + d(a_1, a_4) \\
&= \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+1}{2}\right) d(a_1, a_2) + d(a_1, a_3) + d(a_2, a_3) + d(a_3, a_4) \\
&\leq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+1}{2}\right) d(a_1, a_2) + d(a_3, a_5) + d(a_2, a_3) + d(a_3, a_5),
\end{aligned}$$

where the last inequality follows from inequalities (4.16). We distinguish two further cases to bound the social cost of the optimal committee from below, depending on whether the optimal committee selects agents from the left or from the right side of the median.

Case 2.2.1: $S^* \subseteq \{a_4, a_5, \dots, n\}$ If the optimal committee selects an agent on the right side of the median agent, its social cost satisfies

$$\mathrm{SC}(S^*, A; d) \geq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1)+d(a_2, a_5)+d(a_3, a_5) = \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1)+d(a_2, a_3)+2d(a_3, a_5).$$

Thus, the distortion is

$$\begin{aligned}
\mathrm{dist}(f) = \frac{\mathrm{SC}(S, A; d)}{\mathrm{SC}(S^*, A; d)} &\leq \frac{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n+1}{2}\right) d(a_1, a_2) + d(a_2, a_3) + 2d(a_3, a_5)}{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_2, a_3) + 2d(a_3, a_5)} \\
&\leq \frac{\left(\frac{n-3}{2}\right) \cdot 4 \cdot d(a_1, a_2) + \left(\frac{n+1}{2}\right) d(a_1, a_2) + d(a_1, a_2) + 2 \cdot 2 \cdot d(a_1, a_2)}{\left(\frac{n-3}{2}\right) \cdot 4 \cdot d(a_1, a_2) + d(a_1, a_2) + 2 \cdot 2 \cdot d(a_1, a_2)} \\
&\leq \frac{\left(\frac{n-3}{2}\right) \cdot 4 + \left(\frac{n+1}{2}\right) + 1 + 4}{\left(\frac{n-3}{2}\right) \cdot 4 + 1 + 4} \\
&= \frac{4n - 12 + n + 1 + 2 + 8}{4n - 12 + 2 + 8} = \frac{5n-1}{4n-2} \leq \frac{4}{3},
\end{aligned}$$

where we used inequalities (4.16) for the second inequality.

Case 2.2.2: $S^* \subseteq \{1, \dots, a_1, a_2\}$. If $S^* = S$, the distortion is trivially 1 and we conclude. Otherwise, the social cost of S^* satisfies

$$\mathrm{SC}(S^*, A; d) \geq \sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_1, a_2) + d(a_1, a_3) + d(a_1, a_4).$$

Thus, the distortion is

$$\begin{aligned}
\mathrm{dist}(f) &= \frac{\mathrm{SC}(S, A; d)}{\mathrm{SC}(S^*, A; d)} \\
&\leq \frac{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + \left(\frac{n-3}{2}\right) d(a_1, a_2) + d(a_1, a_2) + d(a_1, a_3) + d(a_1, a_4)}{\sum_{i=1}^{\frac{n-3}{2}} d(i, n-i+1) + d(a_1, a_2) + d(a_1, a_3) + d(a_1, a_4)} \\
&\leq \frac{\left(\frac{n-3}{2}\right) \cdot 4d(a_1, a_2) + \left(\frac{n-3}{2}\right) d(a_1, a_2) + d(a_1, a_2) + 2d(a_1, a_2) + 3d(a_1, a_2)}{\left(\frac{n-3}{2}\right) \cdot 4d(a_1, a_2) + d(a_1, a_2) + 2d(a_1, a_2) + 3d(a_1, a_2)} \\
&\leq \frac{4n - 12 + n - 3 + 2 + 4 + 6}{4n - 12 + 2 + 4 + 6} = \frac{5n-3}{4n} < \frac{4}{3},
\end{aligned}$$

where we used inequalities (4.16) for the second inequality. This concludes the proof of the distortion of Favorite Couple.

For the lower bound, we fix $n = 5$ and an arbitrary $(n, 2)$-voting rule f, consider the profile $\succ \in \mathcal{L}^5(5)$ defined as

$$\begin{aligned}
&1 \succ_1 2 \succ_1 3 \succ_1 4 \succ_1 5, \\
&2 \succ_2 1 \succ_2 3 \succ_2 4 \succ_2 5, \\
&3 \succ_3 2 \succ_3 1 \succ_3 4 \succ_3 5, \\
&4 \succ_4 5 \succ_4 3 \succ_4 2 \succ_4 1, \\
&5 \succ_5 4 \succ_5 3 \succ_5 2 \succ_5 1,
\end{aligned}$$

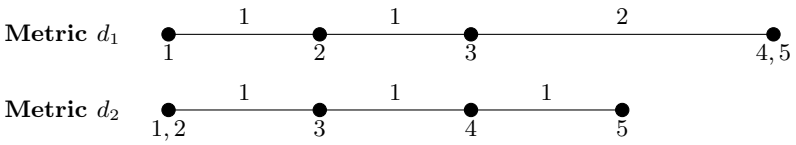


Figure 4.6: Metrics considered in the proof of Theorem 4.11.

and consider the election $\mathcal{E} = (A, 2, \succ)$ with $A = [5]$. We distinguish two cases depending on the set of agents $S = f(\succ)$ selected by the rule.

Suppose first that $S = \{1, 2\}$. We take the distance metric d_1 on A given by positions $x_1 = 0$, $x_2 = 1$, $x_3 = 2$, and $x_4 = x_5 = 4$. Note that $d_1 \rhd \succ$; see Figure 4.6 for an illustration. Since $\mathrm{SC}(\{1, 2\}, A; d_1) = 12$, and $\mathrm{SC}(\{4, 5\}, A; d_1) = 9$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\min_{S' \in \binom{A}{2}} \mathrm{SC}(S', A; d_1)} \geq \frac{12}{9} = \frac{4}{3}.$$

If $S \in \{\{2, 3\}, \{3, 4\}, \{4, 5\}\}$, we consider the distance metric d_2 on A given by positions $x_1 = x_2 = 0$, $x_3 = 1$, $x_4 = 2$, and $x_5 = 3$. Note that $d_2 \rhd \succ$; see Figure 4.6 for an illustration. Since $\mathrm{SC}(\{2, 3\}, A; d_2) = \mathrm{SC}(\{3, 4\}, A; d_2) = 8$ and $\mathrm{SC}(\{4, 5\}, A; d_2) = 10$, whereas $\mathrm{SC}(\{1, 2\}, A; d_2) = 6$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\min_{S' \in \binom{A}{2}} \mathrm{SC}(S', A; d_2)} \geq \frac{8}{6} = \frac{4}{3}.$$

Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{4}{3}$ in all these cases and sets of non-consecutive agents can only induce a larger social cost, we conclude that $\mathrm{dist}(f) \geq \frac{4}{3}$. □

Even number of agents When n is even, we show that the voting rule that selects the two median agents attains the best-possible distortion of 2.

Proposition 4.12. *For an even number of agents n, the voting rule that selects the two median agents achieves a distortion of 2 for utilitarian 2-cost. Moreover, there exists $n \in \mathbb{N}$ such that, for every $(n, 2)$-voting rule f, we have $\mathrm{dist}(f) \geq 2$ for utilitarian 2-cost.*

Proof. We consider an arbitrary election $\mathcal{E} = (A, k, \succ)$ with even $n \geq 4$ and $A = [n]$, and a consistent metric $d \rhd \succ$. Note that the assumption $n \geq 4$ is w.l.o.g. since, for $n = 2$, a distortion of 1 is trivially achieved. We let $m_1 = \frac{n}{2}$ and $m_2 = \frac{n}{2} + 1$ denote the left and right median, respectively, $S = \{m_1, m_2\}$ denote the committee selected by the rule, and S^* denote the optimal committee for the metric d. The social cost of S is

$$\mathrm{SC}(S, A; d) = \sum_{i=1}^{\frac{n}{2}} d(i, n - i + 1) + \frac{n}{2} d(m_1, m_2),$$

whereas Lemma 4.10 implies a lower bound on the social cost of the optimal committee of

$$\mathrm{SC}(S^*, A; d) \geq \sum_{i=1}^{\frac{n}{2}} d(i, n - i + 1).$$

Thus, the distortion of the voting rule is

$$\mathrm{dist}(f) = \frac{\mathrm{SC}(S, A; d)}{\mathrm{SC}(S^*, A; d)} \leq \frac{\sum_{i=1}^{\frac{n}{2}} d(i, n-i+1) + \frac{n}{2} d(m_1, m_2)}{\sum_{i=1}^{\frac{n}{2}} d(i, n-i+1)} \leq \frac{2\sum_{i=1}^{\frac{n}{2}} d(i, n-i+1)}{\sum_{i=1}^{\frac{n}{2}} d(i, n-i+1)} = 2,$$

where the second inequality follows from the fact that $d(m_1, m_2) \leq d(i, n-i+1)$ for any $i \leq \frac{n}{2}$. Thus, the voting rule achieves a distortion of at most 2.

For the lower bound, we fix $n = 4$ and an arbitrary $(n, 2)$-voting rule f, consider the profile $\succ \in \mathcal{L}^4(4)$ defined as

$$\begin{aligned} &1 \succ_1 2 \succ_1 3 \succ_1 4, \\ &2 \succ_2 1 \succ_2 3 \succ_2 4, \\ &3 \succ_3 4 \succ_3 2 \succ_3 1, \\ &4 \succ_4 3 \succ_4 2 \succ_4 1, \end{aligned}$$

and consider the election $\mathcal{E} = (A, 2, \succ)$ with $A = [4]$. We distinguish three cases depending on the set of agents $S = f(\succ)$ selected by the rule.

Suppose first that $S = \{3, 4\}$. We take the distance metric d_1 on A given by positions $x_1 = x_2 = 0$, $x_3 = 1$, and $x_4 = 2$. Note that $d_1 \rhd \succ$; see Figure 4.7 for an illustration. Since $\mathrm{SC}(\{3, 4\}, A; d_1) = 6$, and $\mathrm{SC}(\{1, 2\}, A; d_1) = 3$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\min_{S' \in \binom{A}{2}} \mathrm{SC}(S', A; d_1)} \geq \frac{6}{3} = 2.$$

If $S = \{1, 2\}$, we consider the distance metric d_2 on A given by positions $x_1 = 0$, $x_2 = 1$, and $x_3 = x_4 = 2$. Note that $d_2 \rhd \succ$; see Figure 4.7 for an illustration. Since $\mathrm{SC}(\{1, 2\}, A; d_2) = 6$ and $\mathrm{SC}(\{3, 4\}, A; d_2) = 3$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\min_{S' \in \binom{A}{2}} \mathrm{SC}(S', A; d_2)} \geq \frac{6}{3} = 2.$$

Finally, if $S = \{2, 3\}$, we consider the distance metric d_3 on A given by positions $x_1 = x_2 = 0$ and $x_3 = x_4 = 2$. Note that $d_3 \rhd \succ$; see Figure 4.7 for an illustration. Since $\mathrm{SC}(\{2, 3\}, A; d_3) = 8$ and $\mathrm{SC}(\{1, 2\}, A; d_3) = \mathrm{SC}(\{3, 4\}, A; d_3) = 4$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_3)}{\min_{S' \in \binom{A}{2}} \mathrm{SC}(S', A; d_3)} \geq \frac{8}{4} = 2.$$

Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq 2$ in all these cases and sets of non-consecutive agents can only induce a larger social cost, we conclude that $\mathrm{dist}(f) \geq 2$. □

4.4 Egalitarian Social Cost

In this section, we study the worst-case distortion achievable by voting rules in the context of peer selection with egalitarian social cost. Recall that, in this case, given a set of agents

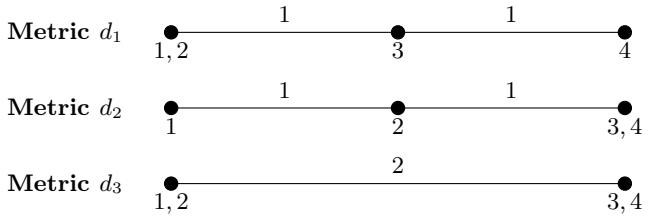


Figure 4.7: Metrics considered in the proof of Proposition 4.12 and Proposition 4.13.

A, a committee size k, and a distance metric d, the social cost of a committee $S' \in \binom{A}{k}$ corresponds to the maximum cost of this committee for some agent $a \in A$:

$$\mathrm{SC}(S', A; d) = \max\{\mathrm{SC}(S', a; d) \mid a \in A\}.$$

We will start the section with the simple case $k = 1$, where $S' = \{s\}$ for some $s \in A$ and thus $\mathrm{SC}(S', a; d)$ is simply $d(a, s)$ for every $a \in A$. In Sections 4.4.2 and 4.4.3 we explore the case of general committee size under additive and q-cost candidate-aggregation functions, respectively.

4.4.1 Warm-up: Selecting a Single Candidate

In the context of single-candidate elections, any voting rule achieves a distortion of 2, since for any election and any selected candidate, one of the extreme agents will incur a cost of at least half of its distance to the other extreme agent, and no rule can induce a social cost larger than the distance between the extreme agents. Despite its simplicity, it is not hard to see that this is the best possible distortion by considering four agents and two metrics as in Figure 4.7. If a rule selects an agent in $\{1, 2\}$, the cost incurred by agent 4 under metric d_1 is 2, while selecting agent 3 would imply a cost of at most 1 for every agent. The situation is analogous under metric d_2 if a rule selects agent 3 or 4. The following proposition formally states these facts.

Proposition 4.13. *For every $n \in \mathbb{N}$, any $(n, 1)$-voting rule has distortion 2 for egalitarian social cost. There exists $n \in \mathbb{N}$ such that, for every $(n, 1)$-voting rule f, $\mathrm{dist}(f) \geq 2$ for egalitarian social cost.*

Proof. Fix $n \in \mathbb{N}$ and an $(n, 1)$-voting rule f arbitrarily. Let $\succ \in \mathcal{L}^n(n)$ be any preference profile on $A = [n]$ and let s be the agent that f outputs for this profile, i.e., $S = f(\succ)$ and $S = \{s\}$. We denote the agents by $\{1, \ldots, n\}$ from left to right, and we let $d \rhd \succ$ be any consistent distance metric. It is clear that, on the one hand, we have

$$\mathrm{SC}(\{s\}, A; d) = \max\{d(a, s) \mid a \in A\} \leq \max\{d(a, b) \mid a, b \in A\} = d(1, n). \tag{4.18}$$

On the other hand, for every agent $b \in A$ we have that $d(1, b) + d(b, n) = d(1, n)$ and, therefore, $\max\{d(1, b), d(b, n)\} \geq \frac{d(1,n)}{2}$. This implies

$$\min_{S' \in \binom{A}{1}} \mathrm{SC}(S', A; d) = \min_{b \in A} \max\{d(a, b) \mid a \in A\} = \min_{b \in A} \max\{d(1, b), d(b, n)\} \geq \frac{d(1, n)}{2}. \tag{4.19}$$

Combining inequalities (4.18) and (4.19), we directly obtain that $\mathrm{dist}(f) \leq 2$.

For the second claim, we denote $S = f(\succ)$, and we fix $n = 4$ and an arbitrary $(n, 1)$-voting rule f, consider the profile $\succ \in \mathcal{L}^4(4)$ defined as

$$1 \succ_1 2 \succ_1 3 \succ_1 4,$$
$$2 \succ_2 1 \succ_2 3 \succ_2 4,$$
$$3 \succ_3 4 \succ_3 2 \succ_3 1,$$
$$4 \succ_4 3 \succ_4 2 \succ_4 1,$$

and consider the election $\mathcal{E} = (A, 1, \succ)$ with $A = [4]$. We distinguish two cases depending on the agent selected by f.

Suppose first that $S \in \{1, 2\}$. We take the distance metric d_1 on A given by positions $x_1 = x_2 = 0$, $x_3 = 1$, and $x_4 = 2$. It is not hard to check that $d_1 \rhd \succ$; see Figure 4.7 for an illustration. Since $\mathrm{SC}(\{1\}, A; d_1) = 2$, $\mathrm{SC}(\{2\}, A; d_1) = 2$, and $\mathrm{SC}(\{3\}, A; d_1) = 1$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\min_{a \in A} \mathrm{SC}(\{a\}, A; d_1)} \geq \frac{\mathrm{SC}(\{2\}, A; d_1)}{\mathrm{SC}(\{3\}, A; d_1)} = 2.$$

Similarly, if $S \in \{3, 4\}$, we consider the distance metric d_2 on A given by positions $x_1 = 0$, $x_2 = 1$, $x_3 = x_4 = 2$. It is not hard to check that $d_2 \rhd \succ$; see Figure 4.7 for an illustration. Since $\mathrm{SC}(\{3\}, A; d_2) = 2$, $\mathrm{SC}(\{4\}, A; d_2) = 2$, and $\mathrm{SC}(\{2\}, A; d_2) = 1$, we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\min_{a \in A} \mathrm{SC}(\{a\}, A; d_2)} \geq \frac{\mathrm{SC}(\{3\}, A; d_2)}{\mathrm{SC}(\{2\}, A; d_2)} = 2.$$

Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq 2$ both when $S \in \{1, 2\}$ and when $S \in \{3, 4\}$, we conclude that $\mathrm{dist}(f) \geq 2$. □

4.4.2 Egalitarian Additive Social Cost

In this section, we study voting rules in the context of egalitarian additive social cost, defined as the maximum over agents of the sum of the distances from the agent to all selected candidates. That is, for a set of agents A, a committee size k, and a distance metric d, the social cost of a committee $S' \in \binom{A}{k}$ is

$$\mathrm{SC}(S', A; d) = \max \left\{ \sum_{s \in S'} d(a, s) \;\middle|\; a \in A \right\}.$$

We begin with a simple observation: When $k = 2$ candidates are to be selected, a simple rule selecting both extreme candidates achieves the best-possible distortion of 1. Intuitively, this voting rule makes sense because, for any selected committee, (1) the cost of the committee is maximized for one of the extreme agents, and (2) the sum of the costs of the committee for both extreme agents is fixed (and equal to two times the distance between them). Thus, selecting both extreme agents ensures they incur the same cost and minimizes the maximum cost between them. This rule and its distortion will be covered as a special case of the rule and result we introduce in what follows.

For larger k, the above intuition about the cost of any committee being maximized for the extreme agents remains true. We state this property, which will be exploited in the development and analysis of a voting rule guaranteeing a constant distortion, in the following lemma.

Lemma 4.14. *For every set of agents $A = [n]$, committee size k, committee $S' \in \binom{A}{k}$, and distance metric d, it holds that*

$$SC(S', A; d) = \max\{SC(S', 1; d), SC(S', n; d)\}.$$

Proof. Let $A = [n]$, k, S', and d be as in the statement, and recall that we refer to the agents sorted from left to right by $\{1, \ldots, n\}$. We suppose towards a contradiction that there exists $a \in A$ such that $\mathrm{SC}(S', a; d) > \max\{\mathrm{SC}(S', 1; d), \mathrm{SC}(S', n; d)\}$; i.e.,

$$\sum_{s \in S'} d(a, s) > \max\left\{ \sum_{s \in S'} d(1, s), \sum_{s \in S'} d(s, n) \right\}. \tag{4.20}$$

We now distinguish two cases. If a has at least as many agents in S' weakly to its left as strictly to its right; i.e., $|\{s \in S' \mid s \leq a\}| \geq |\{s \in S' \mid s > a\}|$, then

$$\begin{aligned}
\sum_{s \in S'} d(s, n) &= \sum_{s \in S' : s \leq a} (d(a, s) + d(a, n)) + \sum_{s \in S' : s > a} (d(a, s) - (d(a, s) - d(s, n))) \\
&\geq \sum_{s \in S' : s \leq a} (d(a, s) + d(a, n)) + \sum_{s \in S' : s > a} (d(a, s) - d(a, n)) \\
&= \sum_{s \in S'} d(a, s) + (|\{s \in S' : s \leq a\}| - |\{s \in S' : s > a\}|) d(a, n) \\
&\geq \sum_{s \in S'} d(a, s),
\end{aligned}$$

a contradiction to inequality (4.20). Analogously, if $|\{s \in S' \mid s \leq a\}| < |\{s \in S' \mid s > a\}|$, then

$$\begin{aligned}
\sum_{s \in S'} d(1, s) &= \sum_{s \in S' : s > a} (d(1, a) + d(a, s)) + \sum_{s \in S' : s \leq a} (d(a, s) - (d(a, s) - d(1, s))) \\
&\geq \sum_{s \in S' : s > a} (d(1, a) + d(a, s)) + \sum_{s \in S' : s \leq a} (d(a, s) - d(1, a)) \\
&= \sum_{s \in S'} d(a, s) + (|\{s \in S' : s > a\}| - |\{s \in S' : s \leq a\}|) d(1, a) \\
&\geq \sum_{s \in S'} d(a, s),
\end{aligned}$$

a contradiction to inequality (4.20). □

Since for any set of agents A, committee size k, committee $S' \in \binom{A}{k}$, and distance metric d we have that

$$\mathrm{SC}(S', 1; d) + \mathrm{SC}(S', n; d) = \sum_{a \in S'} (d(1, a) + d(a, n)) = k d(1, n), \tag{4.21}$$

where the last expression does not depend on S', the previous lemma implies that the optimal committee will be the set that balances the cost for the extreme agents as much as possible. With this in mind, it is natural to consider a generalization of the rule that select both extreme agents to larger committees. The rule, which we denote k-Extremes, simply returns the $\lfloor \frac{k}{2} \rfloor$ agents closest to one extreme and the $\lceil \frac{k}{2} \rceil$ agents closest to the other extreme.

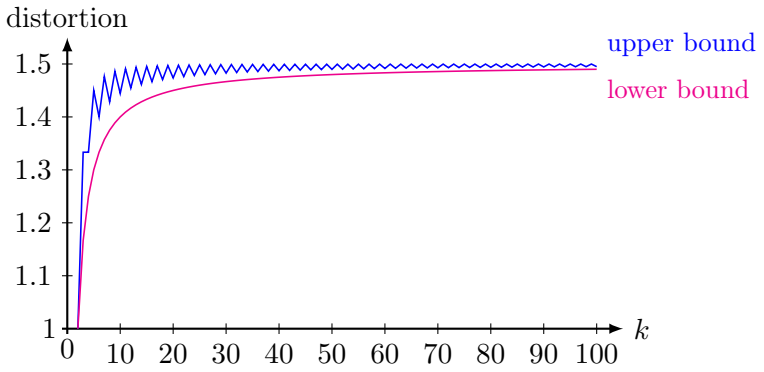


Figure 4.8: Distortion of k-Extremes and lower bound stated in Theorem 4.15 for $k \in \{2, \dots, 99\}$.

Voting Rule 3 (k-Extremes)**.** *For a preference profile $\succ$, compute the order of agents from left to right $1, \dots, n$ and return $S = \{1, \dots, \lfloor \frac{k}{2} \rfloor\} \cup \{n - \lceil \frac{k}{2} \rceil + 1, \dots, n\}$.*

The following theorem states the distortion of this voting rule. It captures the previously claimed distortion of 1 for $k = 2$, and it approaches $\frac{3}{2}$ as k grows. This is best possible up to $O(\frac{1}{k})$ terms, which vanish as k grows. The upper and lower bounds stated in this theorem are depicted in Figure 4.8.

Theorem 4.15. *For every $n, k \in \mathbb{N}$ with $n \geq k \geq 2$, k-Extremes has a distortion for egalitarian additive social cost of at most $\frac{3}{2} - \frac{1}{2(k-1)}$ if k is even and at most $\frac{3}{2} - \frac{1}{k(k-1)}$ if k is odd. Conversely, for every $k \in \mathbb{N}$ with $k \geq 3$ there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n, k)-voting rule f, $\operatorname{dist}(f) \geq \frac{3}{2} - \frac{1}{k}$ for egalitarian additive social cost.*

Proof. We first show the bound on the distortion of k-Extremes. We fix $n, k \in \mathbb{N}$ with $n \geq k \geq 2$, a linear order $\succ$ on $A = [n]$, and a consistent distance metric $d \rhd \succ$. We write $\mathcal{E} = (A, k, \succ)$ for the corresponding election and denote k-Extremes by f and the outcome by S in this part of the proof for compactness.

We claim that, if d is such that $\mathrm{SC}(S, 1; d) < \mathrm{SC}(S, n; d)$, there exists an alternative distance metric d' with $\mathrm{SC}(S, 1; d') \geq \mathrm{SC}(S, n; d')$ and $\operatorname{dist}(f(\succ), \mathcal{E}; d') \geq \operatorname{dist}(f(\succ), \mathcal{E}; d)$. Indeed, consider such d defined by positions $x \in (-\infty, \infty)^n$, and let d' be defined by positions $x' \in (-\infty, \infty)^n$, where $x'_a = x_{n+1-a}$ for every $a \in [n]$. Since f selects $\lfloor \frac{k}{2} \rfloor$ agents closest to the left-most agent and the $\lceil \frac{k}{2} \rceil$ agents closest to the right-most agent, we have

$$\mathrm{SC}(S, 1; d') \geq \mathrm{SC}(S, n; d) > \mathrm{SC}(S, 1; d) \geq \mathrm{SC}(S, n; d').$$

Furthermore, this chain of inequalities combined with Lemma 4.14 imply that $\mathrm{SC}(S, A; d') \geq \mathrm{SC}(S, A; d)$. Since $\min\{\mathrm{SC}(S', A; d') \mid S' \in \binom{A}{k}\} = \min\{\mathrm{SC}(S', A; d) \mid S' \in \binom{A}{k}\}$, this yields $\operatorname{dist}(f(\succ), \mathcal{E}; d') \geq \operatorname{dist}(f(\succ), \mathcal{E}; d)$, so the claim follows. Thanks to this claim, we can assume in what follows that $\mathrm{SC}(S, 1; d) \geq \mathrm{SC}(S, n; d)$ and thus, by Lemma 4.14, $\mathrm{SC}(S, A; d) = \mathrm{SC}(S, 1; d)$.

We distinguish three cases depending on the distances from agent 1 to other agents and show the claimed distortion for each of them. We first suppose that $d\big(1, \lfloor \frac{k}{2} \rfloor\big) \leq \frac{d(1,n)}{2}$. In this case,

$$\begin{aligned}\mathrm{SC}(S, 1; d) &= \sum_{s=1}^{\lfloor k/2 \rfloor} d(1, s) + \sum_{s=n-\lceil k/2 \rceil+1}^{n} d(1, s) \\ &\leq \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right) \frac{d(1,n)}{2} + \left\lceil \frac{k}{2} \right\rceil d(1, n) \\ &= \left(k + \left\lceil \frac{k}{2} \right\rceil - 1\right) \frac{d(1,n)}{2},\end{aligned}$$

where we used the assumption $d\big(1, \lfloor \frac{k}{2} \rfloor\big) \leq d/2$ and the fact that $d(1, 1) = 0$ for the inequality. From Lemma 4.14 and equality (4.21) we know that $\mathrm{SC}(S', A; d) \geq \frac{kd(1,n)}{2}$ for any $S' \in \binom{A}{k}$, so we obtain

$$\mathrm{dist}(f(\succ), \mathcal{E}) = \frac{\mathrm{SC}(S, 1; d)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)} \leq \frac{\big(k + \lceil \frac{k}{2} \rceil - 1\big) \frac{d(1,n)}{2}}{\frac{kd(1,n)}{2}} = \frac{3}{2} - \frac{2 - k \bmod 2}{2k},$$

which is smaller than $\frac{3}{2} - \frac{1}{2(k-1)}$ for even $k \geq 2$ and smaller than $\frac{3}{2} - \frac{1}{k(k-1)}$ for odd $k \geq 3$. Thus, we conclude the result in this case.

We next suppose that $d\big(1, \lfloor \frac{k}{2} \rfloor\big) > \frac{d(1,n)}{2}$ and $\sum_{s=2}^{\lfloor k/2 \rfloor} d(1, s) \leq \frac{k-2-k \bmod 2}{k-1} \cdot \frac{kd(1,n)}{4}$. In a similar way as before, we now have

$$\begin{aligned}\mathrm{SC}(S, 1; d) &= \sum_{s=1}^{\lfloor k/2 \rfloor} d(1, s) + \sum_{s=n-\lceil k/2 \rceil+1}^{n} d(1, s) \\ &\leq \frac{k - 2 - k \bmod 2}{k - 1} \cdot \frac{kd(1,n)}{4} + \left\lceil \frac{k}{2} \right\rceil d(1, n) \\ &= \left(3k - \frac{k - (k-2)k \bmod 2}{k - 1}\right) \frac{d(1,n)}{4},\end{aligned}$$

where the inequality follows from the assumption $\sum_{s=2}^{\lfloor k/2 \rfloor} d(1, s) \leq \frac{k-2-k \bmod 2}{k-1} \cdot \frac{kd(1,n)}{4}$ and the fact that $d(1, 1) = 0$. From Lemma 4.14 and equality (4.21) we know that $\mathrm{SC}(S', A; d) \geq \frac{kd(1,n)}{2}$ for any $S' \in \binom{A}{k}$, so we obtain

$$\begin{aligned}\mathrm{dist}(f(\succ), \mathcal{E}) &= \frac{\mathrm{SC}(S, 1; d)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)} \\ &\leq \frac{\big(3k - \frac{k-(k-2)k \bmod 2}{k-1}\big) \frac{d(1,n)}{4}}{\frac{kd(1,n)}{2}} = \frac{3}{2} - \frac{k - (k-2)k \bmod 2}{2k(k-1)},\end{aligned}$$

which corresponds to the expression in the statement.

We finally consider the case with $d\big(1, \lfloor \frac{k}{2} \rfloor\big) > \frac{d(1,n)}{2}$ and $\sum_{s=2}^{\lfloor k/2 \rfloor} d(1,s) > \frac{k-2-k \bmod 2}{k-1} \cdot \frac{kd(1,n)}{4}$. Since the distance between 1 and the right-most point among $\big\{2, \dots, \lfloor \frac{k}{2} \rfloor\big\}$, namely $d\big(1, \lfloor \frac{k}{2} \rfloor\big)$, is at least its average distance to points within this set, we know that

$$d\left(1, \left\lfloor \frac{k}{2} \right\rfloor\right) \geq \frac{1}{\lfloor \frac{k}{2} \rfloor - 1} \sum_{s=2}^{\lfloor k/2 \rfloor} d(1,s) \geq \frac{1}{\lfloor \frac{k}{2} \rfloor - 1} \cdot \frac{k-2-k \bmod 2}{k-1} \cdot \frac{kd(1,n)}{4} = \frac{kd(1,n)}{2(k-1)}. \tag{4.22}$$

Let now $S' \in \binom{A \setminus \{1\}}{k-1}$ be any set of $k-1$ agents without 1. Since $\big\{2, \dots, \lfloor \frac{k}{2} \rfloor\big\}$ are the closest agents to 1, we know that $\frac{1}{k-1} \sum_{s \in S'} d(1,s) \geq \frac{1}{\lfloor k/2 \rfloor - 1} \sum_{s=2}^{\lfloor k/2 \rfloor} d(1,s)$. Rearranging this expression and using our assumption once again, we obtain

$$\sum_{s \in S'} d(1,s) \geq \frac{k-1}{\lfloor \frac{k}{2} \rfloor - 1} \sum_{s=2}^{\lfloor k/2 \rfloor} d(1,s) \geq \frac{kd(1,n)}{2},$$

where we used inequality (4.22) for the last inequality. For any committee $S' \in \binom{A}{k}$, this implies that $\mathrm{SC}(S', 1; d) \geq \frac{kd(1,n)}{2}$, equality (4.21) implies that $\mathrm{SC}(S', 1; d) \geq \mathrm{SC}(S', n; d)$, and Lemma 4.14 implies that $\mathrm{SC}(S', A; d) = \mathrm{SC}(S', 1; d)$. Therefore,

$$\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d) = \min_{S' \in \binom{A}{k}} \mathrm{SC}(S', 1; d) = \sum_{s=2}^{k} d(1,s); \tag{4.23}$$

i.e., the optimal set in this case corresponds to $\{1, \dots, k\}$. Combining the previous expressions, we obtain the following chain of inequalities:

$$\begin{aligned}
\mathrm{dist}(f(\succ), \mathcal{E}) &= \frac{\mathrm{SC}(S, 1; d)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)} \\
&= 1 + \frac{\mathrm{SC}(S, 1; d) - \min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)}{\min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d)} \\
&\leq 1 + \frac{2}{kd(1,n)} \left(\sum_{s \in S} d(1,s) - \sum_{s=2}^{k} d(1,s) \right) \\
&= 1 + \frac{2}{kd(1,n)} \left(\sum_{s=n-\lceil k/2 \rceil + 1}^{n} d(1,s) - \sum_{s=\lfloor k/2 \rfloor + 1}^{k} d(1,s) \right) \\
&\leq 1 + \frac{2}{kd(1,n)} \cdot \left\lceil \frac{k}{2} \right\rceil \left(d(1,n) - d\left(1, \left\lfloor \frac{k}{2} \right\rfloor\right) \right) \\
&\leq 1 + \frac{2}{kd(1,n)} \cdot \left\lceil \frac{k}{2} \right\rceil \left(d(1,n) - \frac{kd(1,n)}{2(k-1)} \right) \\
&= \frac{3}{2} - \frac{k - (k-2)k \bmod 2}{2k(k-1)}.
\end{aligned}$$

Indeed, the first inequality follows from equality (4.23) and the fact that $\mathrm{SC}(S', A; d) \geq \frac{kd(1,n)}{2}$ for every $S' \in \binom{A}{k}$ due to equality (4.21), the third equality from the definition

of f, the second inequality from simple bounds on $d(1,s)$ for different values of s, and the last inequality from inequality (4.22). The other equalities come from simple calculations. Since the last expression again corresponds to the expression in the statement, we conclude.

For the lower bound, we consider any $k \in \mathbb{N}$ with $k \geq 3$, we fix $n = 2(k+1)$, and consider an arbitrary (n,k)-voting rule f. We partition the agents into four sets $A = \dot{\bigcup}_{i=1}^{4} A_i$ such that $A_1 = \{1\}$, $A_4 = \{n\}$ and $|A_2| = |A_3| = k$. We consider the profile $\succ \in \mathcal{L}^n(n)$, where $S = f(\succ)$, and

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ for some $i,j,\ell \in [4]$ with $|i-j| < |i-\ell|$;

(ii) $1 \succ_a b$ whenever $a \in A_2, b \in A_3 \cup A_4$;

(iii) $n \succ_a b$ whenever $a \in A_3, b \in A_1 \cup A_2$;

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A,k,\succ)$ with $A = [n]$.

In what follows, we distinguish whether f selects more agents from $A_1 \cup A_2$ or from $A_3 \cup A_4$ and construct appropriate distance metrics to show that, in either case, the distortion is at least the one claimed in the statement. Intuitively, if f selects more agents from $A_1 \cup A_2$ we will consider a metric where these sets lie on one extreme, $A_4 = n$ on the other extreme, and all agents A_3 in the middle, so that picking all agents from A_3 would lead to a much lower social cost. In the opposite case, we will construct a symmetric instance.

Formally, we first consider the case with $|S \cap (A_1 \cup A_2)| \geq \frac{k}{2}$ and define the distance metric d_1 on A by the following positions $x \in (-\infty,\infty)^n$: $x_a = 0$ for every $a \in A_1 \cup A_2$, $x_a = 1$ for every $a \in A_3$, and $x_n = 2$. It is not hard to check that $d_1 \rhd \succ$; see Figure 4.9 for an illustration. Since $|S \cap (A_1 \cup A_2)| \geq \frac{k}{2}$, we obtain

$$\mathrm{dist}(f(\succ),\mathcal{E}) \geq \frac{\mathrm{SC}(S,A;d_1)}{\mathrm{SC}(A_3,A;d_1)} \geq \frac{\mathrm{SC}(S,n;d_1)}{\mathrm{SC}(A_3,n;d_1)} \geq \frac{(k-1)+|S\cap(A_1\cup A_2)|}{k} \geq \frac{3}{2} - \frac{1}{k}.$$

Conversely, if $|S \cap (A_3 \cup A_4)| \geq \frac{k}{2}$, we define the distance metric d_2 on A by the following positions $x \in (-\infty,\infty)^n$: $x_1 = 0$, $x_a = 1$ for every $a \in A_2$, and $x_a = 2$ for every $a \in A_3 \cup A_4$. It is not hard to check that $d_2 \rhd \succ$; see Figure 4.9 for an illustration. Since $|S \cap (A_3 \cup A_4)| \geq \frac{k}{2}$, we obtain

$$\mathrm{dist}(f(\succ),\mathcal{E}) \geq \frac{\mathrm{SC}(S,A;d_2)}{\mathrm{SC}(A_2,A;d_2)} \geq \frac{\mathrm{SC}(S,1;d_2)}{\mathrm{SC}(A_2,1;d_2)} \geq \frac{(k-1)+|S\cap(A_3\cup A_4)|}{k} \geq \frac{3}{2} - \frac{1}{k}.$$

Since $\mathrm{dist}(f(\succ),\mathcal{E}) \geq \frac{3}{2} - \frac{1}{k}$ regardless of $f(\succ)$, we conclude that $\mathrm{dist}(f) \geq \frac{3}{2} - \frac{1}{k}$. □

4.4.3 Egalitarian q-Cost

We now turn our attention to the q-cost aggregation function of candidates, so that the social cost is the maximum over agents of the distance from each agent to its qth closest candidate; i.e.,

$$\mathrm{SC}(S',A;d) = \max\left\{\tilde{d}(a)_q \mid a \in A\right\}$$

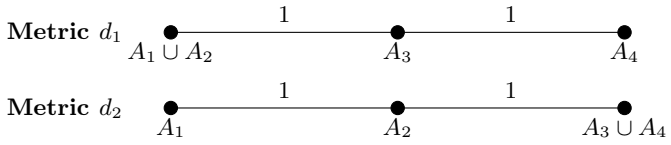


Figure 4.9: Metrics considered in the proof of Theorem 4.15.

for a set of agents A, a committee size k, a committee $S' \in \binom{A}{k}$, and a distance metric d, where $\tilde{d}(a) \in \mathbb{R}_+^{S'}$ contains the values $\{d(a,s) \mid s \in S'\}$ in increasing order.

We begin by showing that no voting rule can guarantee a constant distortion for q-cost when $q \leq \frac{k}{3}$. This implies that the unbounded distortion for this objective, previously established in the setting of disjoint voters and candidates (Caragiannis, Shah, and Voudouris, 2022), also holds in our setting.

Theorem 4.16. *For every $k, q \in \mathbb{N}$ with $\frac{k}{3} \geq q$, there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n,k)-voting rule f, $\mathrm{dist}(f)$ is unbounded for egalitarian q-cost.*

Proof. We let $k, q \in \mathbb{N}$ with $\frac{k}{3} \geq q$ be arbitrary, define $p = \left\lfloor \frac{k}{q} \right\rfloor$, and take $n = (p+1)q$. We partition the agents into $p+1 \geq 4$ sets $A = \dot{\bigcup}_{i \in [p+1]} A_i$ such that $|A_i| \geq q$ for every $i \in [p+1]$; note that this is possible since $(p+1)q \leq \left(\frac{k}{q}+1\right)q = k+q = n$. We consider any fixed (n,k)-voting rule f and the profile $\succ \in \mathcal{L}^n(n)$, where $S = f(\succ)$, and

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ with $|i-j| < |i-\ell|$ for some $i,j,\ell \in [p+1]$;

(ii) $b \succ_a c$ whenever $a \in A_i, b \in A_1, c \in A_j$ with $|i-1| = |i-j|$ for some $i,j \in [p]$;

(iii) $b \succ_a c$ whenever $a \in A_i, b \in A_{p+1}, c \in A_j$ with $|i-(p+1)| = |i-j|$ for some $i,j \in \{2,\dots,p+1\}$;

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A,k,\succ)$ with $A = [n]$. Since $(p+1)q > \frac{k}{q}q = k$, we know that there exists $j \in [p+1]$ such that $|S \cap A_j| < q$. We distinguish two cases depending on the identity of j.

If $j \notin \{p, p+1\}$, we consider the distance metric d_1 on A given by the following positions $x \in (-\infty,\infty)^n$: $x_a = i-1$ for every $a \in A_i$ and $i \in [p]$, and $x_a = p-1$ for every $a \in A_{p+1}$. It is not hard to see that $d_1 \rhd \succ$; see Figure 4.10 for an illustration. Since $|S \cap A_j| < q$ for some $j \notin \{p, p+1\}$, we have that $\mathrm{SC}(S, A_j; d_1) = 1$. On the other hand, we can define an alternative committee $S' = \bigcup_{i \in [p]} S'_i$ such that $|S'_i \cap A_i| \geq q$ for every $i \in [p]$, which is possible because $|A_i| \geq q$ for every $i \in [p]$ and $pq \leq \frac{k}{q}q = k$. Since $\mathrm{SC}(S', A; d_1) = 0$ and $\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S,A;d_1)}{\mathrm{SC}(S',A;d_1)}$, we conclude that $\mathrm{dist}(f(\succ), \mathcal{E})$ is unbounded.

If $j \in \{p, p+1\}$, we consider the distance metric d_2 on A given by the following positions $x \in (-\infty,\infty)^n$: $x_a = 0$ for every $a \in A_q$, and $x_a = i-2$ for every $a \in A_i$ and $i \in \{2,\dots,p+1\}$. It is not hard to see that $d_2 \rhd \succ$; see Figure 4.10 for an illustration. Since $|S \cap A_j| < q$ for some $j \in \{p, p+1\}$, we have that $\mathrm{SC}(S, A_j; d_2) = 1$. On the other hand, we can define an alternative committee $S' = \bigcup_{i \in \{2,\dots,p+1\}} S'_i$ such that $|S'_i \cap A_i| \geq q$ for every $i \in \{2,\dots,p+1\}$, which is possible because $|A_i| \geq q$ for every $i \in \{2,\dots,p+1\}$

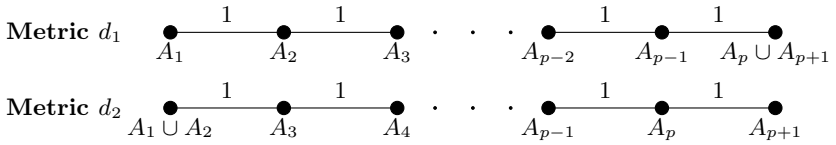


Figure 4.10: Metrics considered in the proof of Theorem 4.16.

and $pq \leq \frac{k}{q}q = k$. Since $\mathrm{SC}(S', A; d_2) = 0$ and $\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S,A;d_2)}{\mathrm{SC}(S',A;d_2)}$, we conclude that $\mathrm{dist}(f(\succ), \mathcal{E})$ is unbounded.

Since $\mathrm{dist}(f(\succ), \mathcal{E})$ is unbounded regardless of $f(\succ)$, we conclude that $\mathrm{dist}(f)$ is unbounded. □

In the context of egalitarian q-cost for $q > \frac{k}{3}$, much better results are possible. The case with $q > \frac{k}{2}$ behaves similarly to the setting where a single candidate is to be selected: Any voting rule achieves a distortion of 2 and this is best possible. When $\frac{k}{3} < q \leq \frac{k}{2}$, the best-possible distortion a voting rule can achieve is again 2, but not any rule does so. We show that k-Extremes attains it.

Theorem 4.17. *Let $n, k, q \in \mathbb{N}$ be such that $n \geq k \geq 2$ and $q > \frac{k}{3}$. If $q > \frac{k}{2}$, any (n, k)-voting rule has distortion 2 for egalitarian q-cost. If $q > \frac{k}{3}$, k-Extremes has distortion 2 for egalitarian q-cost. For every $k, q \in \mathbb{N}$ with $q > \frac{k}{3} \geq 1$, there exists $n \in \mathbb{N}$ with $n \geq k$ such that, for every (n, k)-voting rule f, $\mathrm{dist}(f) \geq 2$.*

Proof. Let $n, k \in \mathbb{N}$ be such that $n \geq k \geq 2$. Let first $q \in \mathbb{N}$ be such that $q > \frac{k}{2}$. Let f be any (n, k)-voting rule and let $\succ \in \mathcal{L}^n(n)$ be an arbitrary preference profile on $A = [n]$. We denote, as usual, agents by $\{1, \ldots, n\}$ from left to right, $S = f(\succ)$, and we let $d \rhd \succ$ be any consistent distance metric. For a committee $S' \in \binom{A}{k}$, we let $\tilde{d}(S', a) \in \mathbb{R}^k_+$ denote the vector with the values $\{d(a, s) \mid s \in S'\}$ in increasing order. It is clear that

$$\mathrm{SC}(S, A; d) = \max\{\tilde{d}(S, a)_q \mid a \in A\} \leq \max\{d(a, b) \mid a, b \in A\} = d(1, n). \tag{4.24}$$

On the other hand, for every committee $S' \in \binom{A}{k}$, if we denote the agents in S' in increasing order by $s_1, \ldots, s_k$ we have that $s_q > s_{k-q}$ because $q > \frac{k}{2}$. This implies that, for every committee $S' \in \binom{A}{k}$, we have

$$\tilde{d}(S', 1)_q + \tilde{d}(S', n)_q = s(1, s_q) + d(s_{k-q}, n) > d(1, n),$$

and thus $\max\{\tilde{d}(S', 1)_q, \tilde{d}(S', n)_q\} \geq \frac{d(1,n)}{2}$. Therefore,

$$\begin{aligned} \min_{S' \in \binom{A}{k}} \mathrm{SC}(S', A; d) &= \min_{S' \in \binom{A}{k}} \max\{\tilde{d}(S', a)_q \mid a \in A\} \\ &\geq \min_{S' \in \binom{A}{k}} \max\{\tilde{d}(S', 1)_q, \tilde{d}(S', n)_q\} \geq \frac{d(1, n)}{2}. \end{aligned} \tag{4.25}$$

Combining inequalities (4.24) and (4.25), we directly obtain that $\mathrm{dist}(f) \leq 2$.

Let now $q \in \mathbb{N}$ be such that $\frac{k}{3} < q \leq \frac{k}{2}$, $\succ \in \mathcal{L}^n(n)$ be an arbitrary preference profile on $A = [n]$, and $d \rhd \succ$ be a consistent distance metric; we consider the election $\mathcal{E} = (A, k, \succ)$.

We denote the outcome of k-Extremes for this profile by S for compactness. We denote agents by $\{1, \dots, n\}$ from left to right and, for $S' \in \binom{A}{k}$, we let $\tilde{d}(S', a) \in \mathbb{R}^k_+$ denote the vector with the values $\{d(a, s) \mid s \in S'\}$ in increasing order. We finally let $a^* \in \arg\max\{\min\{d(1, a), d(a, n)\} \mid a \in A\}$ denote the agent with maximum distance from both extreme agents, assume w.l.o.g. that this is its distance to 1, i.e., $d(1, a^*) \leq d(a^*, n)$, and write $d^* = d(1, a^*)$ for this distance. Observe that

$$\min\{d(a^*, n), d(1, a^* + 1)\} \geq \frac{d(1, n)}{2}. \tag{4.26}$$

Indeed, $d(a^*, n) \geq \frac{d(1,n)}{2}$ follows directly from the inequality $d(1, a^*) \leq d(a^*, n)$ and the equality $d(1, a^*) + d(a^*, n) = d(1, n)$. Having $d(1, a^* + 1) < \frac{d(1,n)}{2}$ would imply $\min\{d(1, a^* + 1), d(a^* + 1, n)\} > d^*$, a contradiction to the definition of a^*.

We first tackle two simple cases. If $a^* < q$, i.e., there are less than q agents between 1 and a^*, then for any committee $S' \in \binom{A}{k}$ we have $\mathrm{SC}(S', A; d) \geq \mathrm{SC}(S', 1; d) \geq d(1, a^* + 1) \geq \frac{d(1,n)}{2}$, where the second inequality follows from inequality (4.26). Since $\mathrm{SC}(S', A; d) \leq d(1, n)$ holds for any committee $S' \in \binom{A}{k}$, we know that in particular $\mathrm{SC}(S, A; d) \leq d(1, n)$ and thus $\mathrm{dist}(f) \leq 2$. Similarly, if $n - a^* < q$, i.e., there are less than q agents between $a^* + 1$ and n, then for any committee $S' \in \binom{A}{k}$ we have $\mathrm{SC}(S', A; d) \geq \mathrm{SC}(S', 1; d) \geq d(a^*, n) \geq \frac{d(1,n)}{2}$, where the second inequality follows from inequality (4.26). As before, $\mathrm{dist}(f) \leq 2$ thus follows directly.

If none of the previous cases hold, we have both $a^* \geq q$ and $n - a^* \geq 2$, so that from the definition of k-Extremes we have $|S \cup \{1, \dots, a^*\}| = \left\lfloor \frac{k}{2} \right\rfloor \geq q$ and $|S \cup \{a^* + 1, \dots, n\}| = \left\lceil \frac{k}{2} \right\rceil \geq q$. This implies that

$$\mathrm{SC}(S, A; d) \leq \max\{d(1, a^*), d(a^* + 1, n)\} \leq d^*. \tag{4.27}$$

We claim that, for every $S' \in \binom{A}{k}$, we have $\mathrm{SC}(S', A; d) \geq \frac{d^*}{2}$. Together with inequality (4.27), this would immediately imply $\mathrm{dist}(f) \leq 2$ and conclude the proof. To prove this fact, suppose for the sake of contradiction that $\mathrm{SC}(S', A; d) < \frac{d^*}{2}$ for some $S' \in \binom{A}{k}$. This is equivalent to the fact that

$$\mathrm{SC}(S', a; d) < \frac{d^*}{2} \iff \left| S' \cup \left\{ b \in A : d(a, b) < \frac{d^*}{2} \right\} \right| \geq q$$

for every $a \in A$. Since the sets $\left\{b \in A \mid d(a, b) < \frac{d^*}{2}\right\}$ for $a \in \{1, a^*, n\}$ are disjoint, we conclude that $|S'| \geq 3q > k$, a contradiction.

For the lower bound, we consider the same instances as in the proof of Theorem 4.15; we repeat the construction for completeness. Naturally, the proof of the lower bound in the end differs from the additive case. We consider any $k \in \mathbb{N}$ with $k \geq 2$, we fix $n = 2(k + 1)$, and consider an arbitrary (n, k)-voting rule f. We partition the agents into four sets $A = \dot{\bigcup}_{i=1}^{4} A_i$ such that $A_1 = \{1\}$, $A_4 = \{n\}$ and $|A_2| = |A_3| = k$. We consider the profile $\succ \in \mathcal{L}^n(n)$, where $S = f(\succ)$, and

(i) $b \succ_a c$ whenever $a \in A_i, b \in A_j, c \in A_\ell$ for some $i, j, \ell \in [4]$ with $|i - j| < |i - \ell|$;

(ii) $1 \succ_a b$ whenever $a \in A_2, b \in A_3 \cup A_4$;

(iii) $n \succ_a b$ whenever $a \in A_3, b \in A_1 \cup A_2$;

and the remaining pairwise comparisons are arbitrary. We consider the election $\mathcal{E} = (A, k, \succ)$ with $A = [n]$.

In what follows, we distinguish whether f selects more agents from $A_1 \cup A_2$ or from $A_3 \cup A_4$ and construct appropriate distance metrics to show that, in either case, the distortion is at least the one claimed in the statement. Intuitively, if f selects more agents from $A_1 \cup A_2$ we will consider a metric where these sets lie on one extreme, $A_4 = n$ on the other extreme, and all agents A_3 in the middle. This way, the selected committee gives twice the social cost as picking all agents from A_3 In the opposite case, we will construct a symmetric instance.

Formally, we first consider the case with $|S \cap (A_1 \cup A_2)| \geq \frac{k}{2}$ and define the distance metric d_1 on A by the following positions $x \in (-\infty, \infty)^n$: $x_a = 0$ for every $a \in A_1 \cup A_2$, $x_a = 1$ for every $a \in A_3$, and $x_n = 2$. It is not hard to check that $d_1 \rhd \succ$; see Figure 4.9 for an illustration. Since $|S \cap (A_1 \cup A_2)| \geq \frac{k}{2}$, we obtain $\mathrm{SC}(S, n; d_1) = 2$ and thus

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_1)}{\mathrm{SC}(A_3, A; d_1)} \geq \frac{\mathrm{SC}(S, n; d_1)}{\mathrm{SC}(A_3, n; d_1)} \geq 2.$$

Conversely, if $|S \cap (A_3 \cup A_4)| \geq \frac{k}{2}$, we define the distance metric d_2 on A by the following positions $x \in (-\infty, \infty)^n$: $x_1 = 0$, $x_a = 1$ for every $a \in A_2$, and $x_a = 2$ for every $a \in A_3 \cup A_4$. It is not hard to check that $d_2 \rhd \succ$; see Figure 4.9 for an illustration. Since $|S \cap (A_3 \cup A_4)| \geq \frac{k}{2}$, we obtain $\mathrm{SC}(S, 1; d_2) = 2$ and thus

$$\mathrm{dist}(f(\succ), \mathcal{E}) \geq \frac{\mathrm{SC}(S, A; d_2)}{\mathrm{SC}(A_2, A; d_2)} \geq \frac{\mathrm{SC}(S, 1; d_2)}{\mathrm{SC}(A_2, 1; d_2)} \geq 2.$$

Since $\mathrm{dist}(f(\succ), \mathcal{E}) \geq 2$ regardless of $f(\succ)$, we conclude that $\mathrm{dist}(f) \geq 2$. □

4.5 Conclusion

In this work, we have introduced the study of metric distortion in committee elections where voters and candidates coincide and provided a first step towards an understanding of this setting by focusing on the line metric. Our results span a variety of social costs and include both analyses of voting rules and constructions of negative instances to provide impossibility results. Although most of our results are tight, an intriguing gap remains for utilitarian q-cost when q is greater than $\frac{k}{2}$. We believe that rules with a distortion better than the current upper bound of 3 exist and their design may benefit from the insights provided by our rule for $q = k = 2$.

The study of the distortion of voting rules in more general metric spaces constitutes an interesting direction for future work, even in the general setting. The main open question concerns the design of voting rules that achieve small distortion beyond the line metric under the utilitarian additive cost.

Another challenge in the design of elections is preventing strategic behavior. A mild assumption in the context of peer selection, adopted by the growing literature on impartial selection, is that agents' primary concern is whether they are selected themselves, and a voting rule is deemed impartial if an agent cannot affect this fact by changing their

reported preferences. On the other hand, a rule is called strategyproof in the voting literature if no agent can misreport their preferences and lead to a better outcome with respect to their actual preferences. Both notions—impartiality and strategyproofness–can be readily applied to our setting, the former being a relaxed version of the latter in this case. Most of the voting rules developed in this work depend on the order of the agents and are thus strategyproof if one restricts voters' deviations to those that are consistent with this order. This constitutes a sensible way to define these axioms, as inconsistent reports could be easily detected and punished by the designer. A notable exception is the Favorite Couple rule, which does not depend exclusively on the order and is not even impartial: For instance, an agent next to the median agent could in some cases modify their ranking, reporting the median agent immediately after themselves, to create a couple and become selected. Designing impartial and strategyproof voting rules with bounded distortion for peer selection constitutes an interesting challenge for future work in the area.

PART II

Online Scheduling

CHAPTER 5

Online Speed-Scaling with Predictions

Chapter overview. This chapter studies deadline-based online speed-scaling under uncertainty about future workloads, using the learning-augmented framework. It explores how imperfect predictions of future system load can guide scheduling and speed decisions to reduce energy consumption while maintaining robust worst-case guarantees. This chapter is based on joint work with Antonios Antoniadis and Peyman Jabbarzade Ganje, which appeared at SWAT 2022.

5.1 Introduction

Energy is a major concern in society in general and computing environments in particular. Indeed, data centers alone are estimated to consume 200 terawatt-hours (TWh) per year, which is likely to increase by a factor of 15 by year 2030 N. Jones, 2018. Hardware manufacturers approach this problem by incorporating energy-saving capabilities into their hardware, with the most popular one being *dynamic speed scaling*, i.e., one can adjust the speed of the processor or device. A higher speed implies a higher energy consumption but also more processing capacity. In contrast, a lower speed incurs energy savings while being able to perform less processing per unit of time. Naturally, to take advantage of this energy-saving capability, scheduling algorithms need to decide on what speed to use at each timepoint and consider the energy consumption of the produced schedule alongside more "traditional" quality-of-service considerations.

This paper studies online, deadline-based *speed-scaling* scheduling, augmented with machine-learned predictions. More specifically, a set of jobs $\mathcal{J}$, each job $j \in \mathcal{J}$ with an associated release time r_j, deadline d_j and processing requirement w_j, arrives online and has to be scheduled on a single speed-scalable processor. A scheduling algorithm needs to decide for each timepoint t on: (i) the processor speed $s(t)$ and (ii) which job $j \in \mathcal{J}$ to execute at t ($j(t)$). Both decisions have to be made by the algorithm at any timepoint t while only having knowledge of the jobs with a release time equal to or less than t. A schedule is said to be feasible if the whole processing requirement of every job j is executed within the respective release time and deadline interval, i.e., if $\int_{t:j(t)=j} s(t)dt \geq w_j$. The energy consumption of a schedule, which we seek to minimize over all feasible schedules, is given by $\int_0^{+\infty} s(t)^\alpha dt$, where $\alpha > 1$ is a constant, which in practice is between 1.1 and 3 depending on the employed technology Critchlow, Dennard, and Schuster, 2000; Wierman, Andrew, and A. Tang, 2012. The offline setting of the problem in which the complete job set $\mathcal{J}$ including their release times, deadlines, and workloads are known in advance was solved in the seminal paper Yao, Demers, and Shenker, 1995 who gave an optimal offline algorithm called YDS. The arguably more interesting online setting in which the characteristics of a job j only become known at its release time r_j has been extensively studied by Albers, Antoniadis, and Greiner, 2015; Bansal, Bunde, H.-L. Chan,

and Pruhs, 2011; Bansal, H.-L. Chan, Katz, and Pruhs, 2012; Bansal, Kimbrel, and Pruhs, 2007; Yao, Demers, and Shenker, 1995, and the currently best known online algorithm is qOA, introduced by Bansal, H.-L. Chan, Katz, and Pruhs, 2012, achieving a competitive ratio of $4^\alpha/(2e^{1/2}\alpha^{1/4})$.

However, the purely online setting may be too restrictive in many practical scenarios for which one can predict – with reasonable accuracy – the characteristics of future jobs, for example, by employing a learning approach on historical data. *Learning augmented algorithms* is a very novel research area (arguably first introduced in 2018 by Lykouris and Vassilvitskii, 2021) trying to capture such scenarios in which predictions of uncertain quality are available for future parts of the input. The goal in learning augmented algorithms is to design algorithms that are at the same time (i) *consistent*, i.e., obtain an improved competitive ratio in the presence of adequate predictions, (ii) *robust*, i.e., there is a worst case guarantee independently of the prediction accuracy (ideally within a constant factor of the competitive ratio of the best known online algorithm that does not employ any predictions) and (iii) *smooth*, i.e., the performance guarantee degrades gracefully with the quality of the predictions.

Different Prediction Setups. How adequate some information about the instance can be predicted generally depends on the concrete application, and often even for the same computational problem different aspects may be predictable to a different degree depending on the exact setting. To this end it is useful to investigate different prediction setups, ideally covering all possible scenarios and establishing trade-offs for each.

Online Speed-Scaling with machine-learned predictions has been investigated before by Bamas, Maggiori, Rohwedder, and Svensson, 2020, who consider a prediction setup in which the release times and deadlines of jobs are known in advance[1], and the processing requirements only become known at the release time of the job but there is a prediction available on it.

While any input instance (with integer release times and deadlines) can be modeled in the way of Bamas, Maggiori, Rohwedder, and Svensson, 2020 (by considering all possible pairs of release times and deadlines and a processing requirement of zero for pairs that do not correspond to a job), this can be computationally expensive – even if these predictions can then be communicated to the algorithm more compactly.

Bamas et al. present a consistent, robust, and smooth algorithm for the specific case in which the interval length of each job is the same. They extend their consistency and robustness results to the general case, where each job can have an arbitrary interval length. However, for this more general setting, the proof of smoothness is omitted because "... the prediction model and the measure of error quickly become complex and notation-heavy.' They demonstrated the existence of a learning-augmented online algorithm for the general case, achieving $(1+\varepsilon)$ consistency and $O\left(\frac{\alpha^3}{\varepsilon^2}\right)^\alpha$ robustness.

In this paper, we explore a novel prediction setup where predictions on the release times and deadlines are provided to the algorithm. To maintain simplicity in our model, we assume that the actual processing requirement of each job $j \in \mathcal{J}$, as well as the number of jobs n, are known. For readers, it might be helpful to conceptualize our setup

[1]This is achieved by considering a job for each possible release time/deadline pair even if the "corresponding job" has zero processing volume.

as having as many unit-size jobs as the total processing volume in the instance, each with a prediction on its release time and deadline. However, it's important to note that our actual setup requires significantly fewer predictions than this simplified model.

In this context, the primary contribution of our paper is the introduction of a natural alternative prediction setup and error measure. We present an algorithm (SwP) within this setup, demonstrating desired properties of consistency, smoothness, and robustness in the general setting. Specifically, it achieves $\left(\frac{1}{1-\mu}\right)^{\alpha-1}$ consistency and $2^{\alpha-1}\alpha^{\alpha}\left(\frac{1}{\mu}\right)^{\alpha-1}$ robustness.

It should be pointed out that since this paper considers a different prediction setting than that of Bamas, Maggiori, Rohwedder, and Svensson, 2020 and in turn the considered error measures also differ. As a consequence the algorithms as well as their guarantees are incomparable. We nevertheless point the reader to Figure 5.2 for a depiction of the consistency-robustness trade-offs implied by the obtained guarantees for each of the algorithms. However one can consider the two prediction setups as complementary of each other.

Our Contribution. We show how the predictions can be used to develop an algorithm called Scheduling with Predictions (SwP), that improves upon qOA when the predictions are reasonably accurate. More formally, in Section 5.3 we show the following theorem.

Theorem 5.1. *Given parameters $\mu \in [0,1]$ and $\lambda \in [0,1/2)$, algorithm SwP achieves a competitive ratio of*

$$\left(\frac{1}{1-\mu}\right)^{\alpha-1}\left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1}$$

if $\eta \in [0,\lambda]$, and $2^{\alpha-1}\alpha^{\alpha}\left(\frac{1}{\mu}\right)^{\alpha-1}$ otherwise.

Here, η is the *error* of the prediction (defined formally later) that captures the distance between the predicted and the actual input instances, $0 \leq \lambda < 1/2$ and $0 \leq \mu \leq 1$ are two hyperparameters that can be thought of as the confidence in the prediction. There are two main problems we need to address. First, even if the predictions are slightly off, an algorithm that blindly computes optimal schedules assuming the predictions are perfectly accurate may result in an infeasible schedule. To solve this issue, we shrink all predicted availability intervals from both endpoints by a factor λ, which guarantees the existence of a feasible schedule assuming that the error η is bounded by λ. Note that $\lambda \geq \frac{1}{2}$ is pointless, since even with $\eta = 0$ the shrunk availability interval would have non-positive length. We therefore restrict our focus to $\lambda < \frac{1}{2}$.

The second problem that we need to address is that, following predictions blindly might result in an infinite competitive ratio if the predictions are far from the actual times. In other words the algorithm is not robust. To address this issue, we combine two algorithms: one that follows the predictions and another which is a classical online algorithm without predictions. How exactly these two algorithms are combined is based on our confidence in the predictions, which is captured by a confidence parameter μ (and can, for example, be determined by observing historic data).

Theorem 5.1 implies that SwP is at the same time *consistent*, *smooth* and *robust* where the exact consistency, smoothness and robustness depend on the choice of the

hyperparameters λ and μ. The main ingredients behind this result are a careful partitioning of the time-horizon as well as the jobs processing requirements as well as a novel combination technique between a consistent and a robust algorithm on the respective inputs.

Additionally, in Section 5.4 we obtain improved results for the restricted case in which all jobs have a common deadline d, and we are given predictions regarding the release times of the jobs. The corresponding algorithm is called Common-Deadline-Scheduling with Predictions (CDSwP) and obtains the following improved competitive ratio.

Theorem 5.2. *Given a parameter $\lambda \in [0, 1)$, algorithm CDSwP achieves a competitive ratio of*

$$\left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1}$$

if $\eta \in [0, \lambda]$, and $2^{\alpha}\left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1}$ otherwise.

Although restricted, this case seems to capture the difficulty of the online setting for the problem, as supported by the fact that the strongest lower bound of $e^{\alpha-1}/\alpha$ on the competitive ratio for online algorithms for the problem is proven on such an instance by Bansal, H.-L. Chan, Katz, and Pruhs, 2012.

Finally, in Section 5.5 we present an empirical evaluation of our results on a real-world data-set, which suggests the practicality of our algorithm. The actual results are preceded by Section 5.2 which contains preliminary results and observations. All omitted proofs can be found in the Appendix.

5.1.1 Related Work

Online Energy-Efficient Scheduling. As already mentioned, speed-scaling was first studied from an algorithmic point of view by Yao, Demers, and Shenker, 1995. They studied the deadline-based version of the problem also considered here, and in addition to providing the optimal offline algorithm called YDS, two online algorithms called *Optimal Available (OA)* and *Average Rate (AVR)*. OA recalculates an optimal offline schedule for the remaining instance at each release time, whereas AVR "spreads" the processing volume equally between its release time and deadline in order to determine the speed for each timepoint t. The actual schedule then is simply an *Earliest Deadline First (EDF)* schedule with these speeds. They show that AVR obtains a competitive ratio of $2^{\alpha-1}\alpha^{\alpha}$ which is essentially tight as shown by Bansal, Bunde, H.-L. Chan, and Pruhs, 2011. Algorithm OA, on the other hand, was analyzed by Bansal, H.-L. Chan, Katz, and Pruhs, 2012 who proved a tight competitive ratio of α^{α}.

The currently best known algorithm for the problem, at least for modern processors which satisfy $\alpha = 3$ is the aforementioned qOA algorithm, which for any parameter $q \geq 1$ sets the speed of the processor to be q times the speed that the optimal offline algorithm would run the jobs in the current state. Algorithm qOA attains a competitive ratio of $4^{\alpha}/(2e^{1/2}\alpha^{1/4})$, for $q = 2 - 1/\alpha \approx 1.667$.

The multiprocessor version of online, deadline-based, speed-scaling has also been studied, by Albers, Antoniadis, and Greiner, 2015; Angel, Bampis, Kacem, and Letsios, 2019 as well as other objectives, for example, flow time by Albers and Fujiwara, 2007;

Bansal, Pruhs, and Stein, 2009. We refer the interested reader to surveys by Albers, 2010; Irani and Pruhs, 2005.

Further Results on Learning Augmented Algorithms. Lykouris and Vassilvitskii, 2021 was arguably the seminal paper in the area, considering the online caching problem. Subsequently, Purohit, Svitkina, and R. Kumar, 2018 studied both the ski-rental problem and non-clairvoyant scheduling. Similar to the current work, the robustness and consistency guarantees were given as a function of a hyperparameter that is part of the input to the algorithm. Both the caching and the ski-rental problem have since been extensively studied in the literature (see for example Antoniadis, Coester, Eliáš, Polak, and Simon, 2023; Rohatgi, 2020; A. Wei, 2020 and Gollapudi and Panigrahi, 2019; S. Wang, J. Li, and S. Wang, 2020).

Several other online problems have been investigated through the lens of learning-augmented algorithms and results of similar flavor were obtained. Examples include scheduling and queuing problems by Lattanzi, Lavastida, Moseley, and Vassilvitskii, 2020; Mitzenmacher, 2020, online selection and matching problems by Antoniadis, Gouleakis, Kleer, and Kolev, 2020; Dütting, Lattanzi, Leme, and Vassilvitskii, 2021, or the more general framework of online primal-dual algorithms by Bamas, Maggiori, and Svensson, 2020. Recently, Balkanski, Perivier, Stein, and H.-T. Wei, 2023 continued this line of work and explored other energy-efficient scheduling problems with predictions. We direct the interested reader to a recent survey by Mitzenmacher, 2020 and list of papers in this area by Lindermayr and Megow, n.d.

5.2 Preliminaries

We consider *online, deadline-based speed-scaling* as described in the introduction. Given a scheduling algorithm A on the set jobs $\mathcal{J}$, the energy consumption of A on $\mathcal{J}$ is denoted by $\mathcal{E}_A(\mathcal{J})$. When clear from the context, we may write $\mathcal{E}_A$ instead of $\mathcal{E}_A(J)$ to simplify the notation.

As usual for online problems, the performance guarantees are given by employing *competitive analysis*. Following the speed-scaling literature (see for example Bansal, H.-L. Chan, Katz, and Pruhs, 2012) we use the *strict competitive ratio*. Formally, the (strict) competitive ratio of algorithm A for the online, deadline-based speed-scaling problem, on input instance I is given by

$$\max_I \frac{\mathcal{E}_{A(I)}}{\mathcal{E}_{YDS(I)}},$$

where $\mathcal{E}_{A(I)}$ is the cost that algorithm A incurs on instance I, and the maximum is taken over all possible input instances I. The competitive ratio in many cases will depend on the prediction error.

Prediction Setup. The algorithm initially gets information about the number of jobs n, the corresponding processing volumes $w_j, \forall j \in \mathcal{J}$, as well as for every job $j \in \mathcal{J}$ a prediction p_j for the release time r_j and another prediction q_j for the deadline d_j. Again, the actual values of r_j and d_j only become known at timepoint r_j. Let $R = \{r_1, \dots r_n\}$,

$D = \{d_1, \dots d_n\}$, $P = \{p_1, \dots p_n\}$, and $Q = \{q_1, \dots q_n\}$. Note that in the special case where all jobs have a common deadline d we naturally only obtain predictions for the release times.

The quality of the prediction is measured in terms of a *prediction error* η. Intuitively η measures the distance between the predicted values and the actual ones. We start by defining the individual prediction error η_i for each job $i \in \mathcal{J}$.

Definition 5.3. *Let the prediction error for job i be* $\eta_i = \max\left\{\frac{|p_i - r_i|}{q_i - p_i}, \frac{|q_i - d_i|}{q_i - p_i}\right\}$.

Note that we implicitly assume that $p_i \leq q_i$ for all $i \in \mathcal{J}$ since otherwise, it is immediately obvious that the quality of the predictions is low, and one could just run a classical online algorithm for the problem. Furthermore, if the instance has a common deadline then η_i simplifies to $\eta_i = \frac{|p_i - r_i|}{d - p_i}$.

The (total) prediction error η of an input instance is then given by $\eta = \max_i \eta_i$. We call this *max-norm-error*.

Definition 5.4. *We say that the total error η is a* max-norm-error *if η is given by the infinity norm of the vector of the respective errors for each job. More formally,*

$$\eta = \|\boldsymbol{\eta}\|_\infty = \max(\eta_1, \eta_2, \dots \eta_n).$$

Although in many cases our algorithm performs well even with an inadequate prediction for a single job, note that the optimal speed scaling schedule can be very sensitive to such errors.

Consider an instance that is worst-case for the setting without predictions (the best known lower bound on the competitive ratio for classic online algorithms is $e^{\alpha-1}/\alpha$ Bansal, H.-L. Chan, Katz, and Pruhs, 2012) and consider each job of that instance being associated with some "useless" prediction. One could then "pad" this original instance with additional jobs such that their availability interval is disjoint from the original instance. Each additional job could have a correct prediction and an availability interval that is arbitrarily long compared to their volume. This does not affect the original worst case instance, and in this way one can push the average prediction error arbitrarily close to zero, while at the same time the improvement over the worst-case lower bound attainable without predictions is arbitrarily small. This is in contrast to what one might expect from a reasonable error metric, namely if the error is arbitrarily close to zero, one should be able to substantially improve over the state of the art without predictions.

Moreover, while on one hand the sum of errors is less sensitive to individual large errors (outliers), this comes at the disadvantage that small errors may accumulate, potentially completely misclassifying instances with relatively accurate predictions.

Performance Guarantees. In the following we formalize the performance guarantees used to evaluate our algorithms.

Definition 5.5. *We say that an algorithm within the above prediction setup is:*

- Consistent, *if its competitive ratio is strictly better than that of the best online algorithm without predictions for the problem, whenever $\eta = 0$.*

- Robust, *if its competitive ratio is within a constant factor from that of the best online algorithm without predictions for the problem. Note that Robustness is independent of the prediction quality.*
- Smooth, *if its competitive ratio is a smooth function of* η.

Shrinking of Intervals. The most straightforward way to consider the predictions would arguably be to blindly trust the predictions, i.e., schedule jobs assuming that the predicted instance is the actual instance.

Consider the instance J_{PQ} (resp. J_{RD}) in which every job has the corresponding predicted (resp. actual) release time and deadline. The naive algorithm would compute the optimal offline schedule $YDS(J_{PQ})$ and try to schedule tasks according to it. If the predictions are perfectly accurate, then this clearly is an optimal schedule, and the best one can do. However, if the predictions are even slightly inaccurate, then the resulting schedule may be infeasible. Moreover, our goal is to have a robust algorithm, which cannot be obtained by following the predictions blindly. For these reasons, one has to trust the predictions more cautiously and not blindly.

One of our crucial ideas is to slightly shrink the interval between each job's release time and deadline before scheduling it. The intuition is that if the predictions are only slightly off, then a YDS schedule for the newly obtained instance will be feasible at a slight increase in energy consumption over the YDS schedule of the predicted instance. The following lemmas formalize this intuition. We note that a similar result is also presented in Bamas, Maggiori, Rohwedder, and Svensson, 2020; however, given that the actual setups are different new proofs are required. Proofs of these lemmas can be found in Appendix 5.7.

Lemma 5.6. *Consider a common deadline instance* $\mathcal{J}$*, and another common deadline instance* $\hat{\mathcal{J}}$ *constructed from* $\mathcal{J}$ *such that every job* $\hat{j}_i \in \hat{\mathcal{J}}$ *has workload* $\hat{w}_i = w_i$*,* $\hat{d} = d$*, and* $\hat{r}_i = r_i + (1 - c_i) \cdot (d - r_i)$ *for some shrinking parameter* $0 < c_i \leq 1$*. Set* $c = \max_i c_i$*. Then,*

$$\mathcal{E}_{YDS}(\hat{\mathcal{J}}) \leq (1/c)^{\alpha-1} \mathcal{E}_{YDS}(\mathcal{J}).$$

Lemma 5.7. *Consider a (general) instance* $\mathcal{J}$*, and another instance* $\hat{\mathcal{J}}$ *in which every job* $j_i \in \mathcal{J}$ *corresponds to a job* $\hat{j}_i \in \hat{\mathcal{J}}$ *with workload* $\hat{w}_i = w_i$*,* $\hat{r}_i = r_i + \frac{1-c}{2} \cdot (d_i - r_i)$ *and* $\hat{d}_i = d_i - \frac{1-c}{2} \cdot (d_i - r_i)$ *for some shrinking parameter* $0 < c \leq 1$*. Then,*

$$\mathcal{E}_{YDS}(\hat{\mathcal{J}}) \leq (1/c)^{\alpha-1} \mathcal{E}_{YDS}(\mathcal{J}).$$

It will be useful to bound the energy consumption of (the possibly infeasible for the original input instance) schedule $YDS(J_{PQ})$. We compute the energy consumption of schedule $YDS(J_{PQ})$ in the following lemma.

Lemma 5.8. *For any* $\eta \geq 0$ *there holds*

$$\mathcal{E}_{YDS(J_{PQ})} \leq (2\eta + 1)^{\alpha-1} \mathcal{E}_{YDS(J_{RD})}.$$

Proof. Consider two sets $P^* = \{p_1^*, \dots p_n^*\}$ and $Q^* = \{q_1^*, \dots q_n^*\}$, with $p_i^* = p_i - \eta_i(q_i - p_i)$ and $q_i^* = q_i + \eta_i(q_i - p_i)$.

By the definition of η_i, p_i^* and q_i^*, we have $(r_i, d_i) \subseteq (p_i^*, q_i^*)$, and therefore

$$\mathcal{E}_{YDS(J_{P^*Q^*})} \leq \mathcal{E}_{YDS(J_{RD})}.$$

By having $c = \frac{1}{(2\eta+1)}$, and $J = J_{P^*Q^*}$ $(r_i = p_i^*, d_i = q_i^*)$ in Lemma 5.7, we obtain $J' = J_{PQ}$ and therefore,

$$\mathcal{E}_{YDS(J_{PQ})} \leq (2\eta+1)^{\alpha-1}\mathcal{E}_{YDS(J_{P^*Q^*})}.$$

□

Using Lemma 5.6, we can obtain a similar result for common deadline instances.

Corollary 5.9. *In common deadline instances for any parameter $\eta \geq 0$, there holds*

$$\mathcal{E}_{YDS(J_P)} \leq (\eta+1)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_R)}.$$

The idea of shrinking intervals as described above will be useful for the general case as well as the restricted common deadline case. How much each algorithm will shrink the predicted job intervals will depend on the *confidence*. This will be denoted by a *confidence parameter* $0 < \lambda \leq 1/2$ that will be given as input to the respective algorithm. In the following, we define the *shrunk prediction set* of release times and deadlines parametrized by the confidence parameter λ, and use the above lemmas to argue about how this "shrinking" actually affects the energy consumption of the corresponding YDS-schedule.

Definition 5.10. *Let $P' = \{p_1', \dots, p_n'\}$ and $Q' = \{q_1', \dots, q_n'\}$ be the shrunk prediction set of release times and deadlines respectively in which $p_i' = \lfloor p_i + \lambda(q_i - p_i) \rfloor$ and $q_i' = \lceil q_i - \lambda(q_i - p_i) \rceil$ for all $i \in [n]$.*

We first observe that any schedule that considers the sets P' and Q' as the actual release times and deadlines of the jobs will be feasible, as long as the error η is not larger than λ.

Observation 5.11. *Under the assumption that $\eta \in (0, \lambda)$, it follows that $r_i \leq p_i'$ and $q_i' \leq d_i$ hold for every job i.*

Therefore, the schedule $YDS(J_{P'Q'})$ is feasible. Although shrinking the intervals and then running YDS is not a robust algorithm, it will be useful to bound its energy consumption when $\eta \leq \lambda$ holds.

Lemma 5.12. *For any $\eta \in [0, \lambda]$ there holds*

$$\mathcal{E}_{YDS(J_{P'Q'})} \leq \left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_{RD})}.$$

Proof.

$$\mathcal{E}_{YDS(J_{P'Q'})} \leq \frac{1}{(1-2\lambda)^{(\alpha-1)}} \cdot \mathcal{E}_{YDS(J_{PQ})} \leq \left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_{RD})}. \tag{5.1}$$

By Lemma 5.7 we have the first inequality in (5.1), and the second inequality holds because of Lemma 5.8. □

Similarly for common deadline instances, since we shrink from one side, we obtain a better competitive ratio.

Corollary 5.13. *For any $\eta \in [0, \lambda]$ in common deadline instances there holds*

$$\mathcal{E}_{YDS(J_{P'})} \leq \left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_R)}.$$

Proof.

$$\mathcal{E}_{YDS(J_{P'})} \leq \frac{1}{(1-\lambda)^{(\alpha-1)}} \cdot \mathcal{E}_{YDS(J_P)} \leq \left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_R)}.$$

By Lemma 5.6 we have the first inequality, and the second inequality holds because of Corollary 5.9. □

5.3 General Case

In this section we present algorithm ScheduleWithPredictions(λ, μ) (SwP(λ, μ) for short) for the general learning-augmented speed-scaling setting. Parameter $0 \leq \lambda < 1/2$ describes for which range of prediction errors we would like to obtain an improved competitive ratio. The smaller the λ, the smaller that range but the better the corresponding competitive ratio for $\eta < \lambda$. On the other hand, parameter $0 \leq \mu \leq 1$ allows us to set the desired trade-off between consistency and robustness. As we will see, perfect predictions and $\lambda = \mu = 0$ would give a competitive ratio of 1.

Inspired by Bamas, Maggiori, Rohwedder, and Svensson, 2020 algorithm SwP begins by partitioning each time slot $I_t = [t, t+1), t \in \mathbb{Z}$ into two parts: $I_t^\ell = [t, t+(1-\mu))$ and $I_t^r = [t+(1-\mu), t+1)$. We call I_t^ℓ the *left part*, and I_t^r the *right part* of time slot I_t. The idea is to reserve the left parts of time-slots for following the prediction, and the right parts of the time-slots are, roughly speaking, intended for safeguarding against inaccurate predictions. A key component of our algorithm consists of elegantly and dynamically distributing the processing volume of each job upon its arrival among the two parts. This distribution is crucial in order to obtain a trade-off between consistency and robustness, based on the parameters λ and μ. The algorithm consists of two steps, the *preprocessing* and the *online* step which we now describe in more detail.

Preprocessing: Partition left parts into intervals and assign jobs to them. Upon receiving the predictions (P, Q), SwP computes a YDS-schedule S' for instance (P', Q') – which is obtained by "shrinking" (P, Q) as described above. Although S' may not be feasible for the actual instance (R, D), it will be used to partition the left parts into intervals and subsequently assign each such interval of the partition to a specific job.

To this end, let $I_t(j) := [t+a_t(j), t+b_t(j)) \subseteq I_t$ be the maximal subinterval of I_t during which j is executed under S'. Note that $I_t(j)$ could be empty for some combinations of j and t. Furthermore, since by definition there are no release times or deadlines within I_t, and YDS schedules according to EDF, there can be at most one execution interval of j within I_t. Let, for every job j and left part I_t^ℓ, $I_t^\ell(j) := [t + a_t^\ell(j), t + b_t^\ell(j))$, where $a_t^\ell(j) = a_t(j)/(1-\mu)$ and $b_t^\ell(j) = b_t(j)/(1-\mu)$, be the subinterval of I_t^ℓ assigned to job j.

To obtain some intuition, scheduling the whole processing volume of each job j at a uniform speed throughout intervals $I_t^\ell(j)$ would result in a "compressed" version of $YDS(P'Q')$ where each time-slot is sped-up by a factor of $1/(1-\mu)$ to fit in the left part only, thus having an energy consumption increased by a factor of $(1/(1-\mu))^\alpha$ over that of $YDS(P'Q')$. Although (as we will see) such a compressed schedule would be consistent, it may not be robust (or even feasible) in the presence of subpar predictions. For this reason, we will eventually only schedule part of the volume of each job in the associated left parts whenever feasible, and the remaining volume will be processed on right parts.

Online Step: Job arrivals and processing SwP needs to decide exactly when each job is to be processed within each time-slot and at what speed. This is done by (i) distributing the processing volume of each job j to right parts of different time-slots I_t and associated left parts $I_t^\ell(j)$ upon its arrival, and (ii) feasibly scheduling the whole volume assigned to the current time-slot (both to its left and right part), within the time-slot itself. In the following we discuss how this is accomplished.

(i) Job Arrivals: Upon arrival of job j at r_j, let $\delta_j = w_j/(d_j - r_j)$ be its density and $\ell(j) := \sum_{t\in[r_j,d_j)} |I_t^\ell(j)|$ be the total processing time reserved for job j on the left parts during the preprocessing step that can actually be feasibly used for job j. Furthermore let $V_t(j)$ be the total volume currently (from jobs $1, 2, \dots j-1$) assigned to I_t^r, for all t (thus $V_t(1) = 0$).

The algorithm assigns some amount of volume y_j^t (to be determined later) of job j to interval I_t^r (thus $V_t(j+1) := V_t(j) + y_j^t$), for all $t \in [r_j, d_j)$, with $0 \le y_j^t \le \delta_j$. Finally the remaining volume $X_j := w_j - \sum_t y_j^t$ is assigned to the left parts $I_t^\ell(j)$ with $t \in [r_j, d_j)$, proportionally to their length, i.e., an interval $I_t^\ell(j)$ with $t \in [r_j, d_j)$ receives an $|I_t^\ell(j)|/\ell(j)$-fraction of X_j which implies that the average speed within $I_t^\ell(j)$ must be $X_j/\ell(j)$. To gain some intuition on the values of y_j^t, it is useful to think of the algorithm as waterfilling the volume of j to both the left and the right parts such that no right part receives more than δ_j amount of volume. More formally, the y_j^t, with $0 \le y_j^t \le \delta_j$ and $t \in [r_j, d_j)$ are defined such that they satisfy the following inequalities:

$$\frac{V_t(j)}{\mu} \ge X_j/\ell(j) \qquad \forall t \in [r_j, d_j) \text{ with } y_j^t = 0 \tag{5.2}$$

$$\frac{V_t(j) + y_j^t}{\mu} = X_j/\ell(j) \qquad \forall t \in [r_j, d_j) \text{ with } 0 < y_j^t < \delta_j \tag{5.3}$$

$$\frac{V_t(j) + y_j^t}{\mu} \le X_j/\ell(j) \qquad \forall t \in [r_j, d_j) \text{ with } y_j^t = \delta_j. \tag{5.4}$$

Note that the left hand side in each of the above inequalities corresponds exactly to $V_t(j+1)/\mu$ and therefore to the average speed required to process the volume assigned to t before the arrival of job $j+1$ within I_t^r. We prove the existence of such y_j^t and describe how they can be computed in Appendix 5.7.

(ii) Processing: For each $I_t, t = r_j, \dots r_{j+1} - 1$ the algorithm processes job $j' \le j$ within every $I_t^\ell(j')$ at a speed of $X_{j'}/\ell(j')$, and the assigned volume to I_t^r is processed within I_t^r at a speed of $V_t(j+1)/\mu$, with the order of the jobs within each I_t^r being determined by EDF.

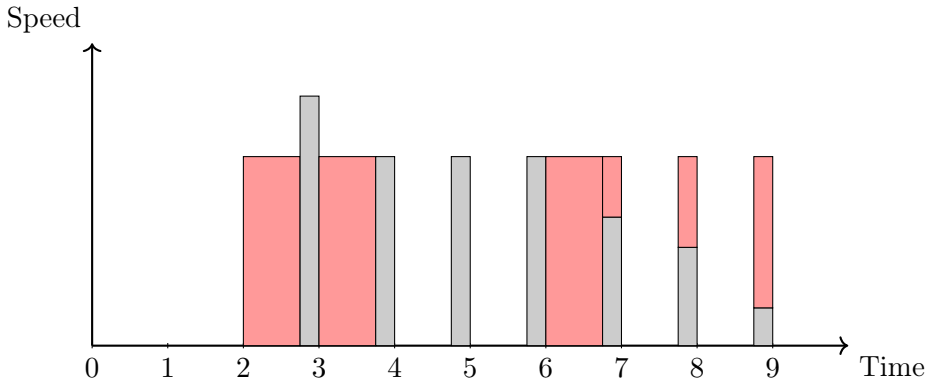


Figure 5.1: The speed profile corresponds to an instance with $\mu = 0.25$. Job i arrives at $r_i = 2$, with $d_i = 9$, and $w_i = 7$. Hence, $\delta_i = \frac{w_i}{d_i - r_i} = 1$. For this instance, $YDS(P'Q')$ runs job i only in 3 blocks, so we have $\ell(i) = 3 \cdot 0.75 = 2.25$. For the first four blocks we have $y_i^t = 0$ and inequality (5.2) holds. In the fifth and sixth blocks, $0 < y_i^t < \delta_i$ and equality (5.3) holds. And in the last block, $y_i^t = \delta_i = 1$ and inequality (5.4) holds.

The online step gets repeated upon the arrival of each job. We next show that the resulting schedule is feasible.

Lemma 5.14. *In the schedule output by* SwP(λ, μ) *a volume of* w_j *is fully processed for each job* j *within* $[r_j, d_j)$.

Proof. It is relatively easy to see that a total volume of w_j is assigned to left and right parts of I_t's with $t \in [r_j, d_j)$: Indeed, by the algorithm definition volume of w_j only gets assigned to $I_t^\ell \subset I_t$ or $I_t^r \subset I_t$ with $t \in [r_j, d_j)$. In addition, a volume of $\sum_t y_j^t$ gets assigned to the right parts and $w_j - \sum_t y_j^t$ to the left parts for a total volume assignment of w_j. It therefore remains to show that all the assigned volume is feasibly processed in the processing step.

Consider some I_t with $r_j \leq t < r_{j+1}$ and the corresponding I_t^ℓ and I_t^r. Note that by the above argument, for any such t no job with an index greater than j will assign any volume, and that only job $j' \leq j$ may be assigned to $I_t^\ell(j')$. Therefore a speed of $X_{j'}/\ell(j')$ throughout every such $I_t^\ell(j)$ is sufficient to schedule all volume assigned to it. Finally, since there are no release times or deadlines within each individual interval, the total volume of $V_t(j+1)$ can be feasibly scheduled within I_t^r at a speed of $V_t(j+1)/\mu$. □

We next show consistency and robustness of the algorithm.

Lemma 5.15 (Consistency & Smoothness)**.** *For any* $\eta \in [0, \lambda]$ *there holds*

$$\mathcal{E}_{SwP} \leq \left(\frac{1}{1-\mu}\right)^{\alpha-1} \left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1} \mathcal{E}_{YDS(RD)}.$$

Proof. We can express $\mathcal{E}_{\text{SwP}}$ as:

$$\begin{aligned}\mathcal{E}_{\text{SwP}} &= \sum_{j=1}^{n}\left(\frac{X_j^\alpha}{\ell(j)^{\alpha-1}} + \sum_{t\in[r_j,d_j)}\left(\frac{V_t(j+1)^\alpha}{\mu^{\alpha-1}} - \frac{V_t(j)^\alpha}{\mu^{\alpha-1}}\right)\right)\\ &= \sum_{j=1}^{n}\left(\frac{X_j^\alpha}{\ell(j)^{\alpha-1}} + \sum_{t\in[r_j,d_j)}\left(\frac{\left(V_t(j)+y_j^t\right)^\alpha}{\mu^{\alpha-1}} - \frac{V_t(j)^\alpha}{\mu^{\alpha-1}}\right)\right)\\ &\leq \sum_{j=1}^{n}\frac{w_j^\alpha}{\ell(j)^{\alpha-1}} = \left(\frac{1}{1-\mu}\right)^{\alpha-1}\mathcal{E}^j_{YDS(J_{P'Q'})}.\end{aligned}$$

The inequality holds by convexity of the power function and by the fact that $V_t(j+1)/\mu \leq X_j/\ell(j)$ for each t such that $y_j^t > 0$ (Equations 5.3 and 5.4). The last equality follows since for $\eta \in [0,\lambda]$, for every job j there holds $[r_j,d_j) \supseteq [p'_j,q'_j)$ (Observation 5.11), and by construction $\ell(j)$ is $1/(1-\mu)$ times the total processing time reserved for job j under $\text{YDS}(P'Q')$.

The lemma directly follows, since by Lemma 5.12,

$$\mathcal{E}_{YDS(J_{P'Q'})} \leq \left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1}\cdot\mathcal{E}_{YDS(J_{RD})}.$$

□

Lemma 5.16 (Robustness)**.** *For any instance, we have*

$$\mathcal{E}_{SwP} \leq 2^{\alpha-1}\alpha^\alpha\left(\frac{1}{\mu}\right)^{\alpha-1}\mathcal{E}_{YDS(J_{RD})}.$$

Proof. Note that by the algorithm definition there holds that $V_t(j) \geq V_t(i)$, for $j > i$ and any t, since upon each release time new volume gets assigned but volume never gets removed. We therefore have

$$\begin{aligned}\mathcal{E}_{\text{SwP}} &\leq \sum_{j=1}^{n}\left(\frac{X_j^\alpha}{\ell(j)^{\alpha-1}}\right) + \sum_t\left(\frac{V_t(n+1)^\alpha}{\mu^{\alpha-1}}\right)\\ &\leq \sum_t\left(\frac{\left(\sum_{j:t\in[r_j,d_j)}\delta_j\right)^\alpha}{\mu^{\alpha-1}}\right) = \left(\frac{1}{\mu}\right)^{\alpha-1}\mathcal{E}_{AVR}.\end{aligned}$$

The second inequality follows by the convexity of the power function and the fact that $V_t(n+1)/\mu \geq V_t(j+1)/\mu \geq X_j/\ell(j)$ for each t such that $y_j^t < \delta_j$ (Equations 5.3 and 5.2). The lemma follows by the competitive ratio of AVR Bansal, Bunde, H.-L. Chan, and Pruhs, 2011. □

Lemmas 5.15 and 5.16 together directly imply Theorem 5.1.

Theorem 5.1. *Given parameters $\mu \in [0,1]$ and $\lambda \in [0,1/2)$, algorithm SwP achieves a competitive ratio of*

$$\left(\frac{1}{1-\mu}\right)^{\alpha-1}\left(\frac{2\eta+1}{1-2\lambda}\right)^{\alpha-1}$$

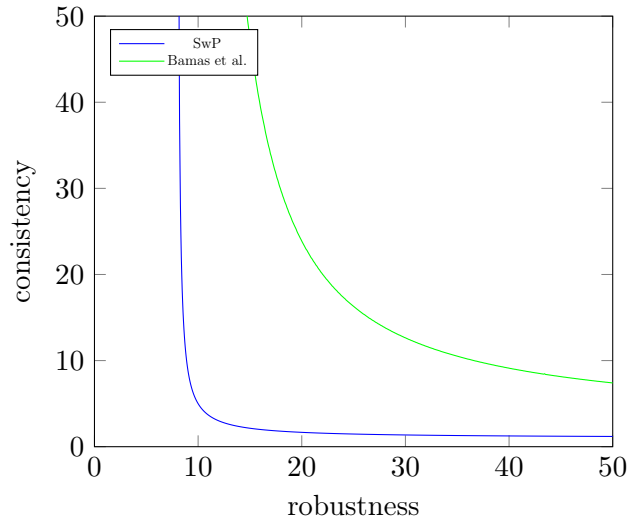


Figure 5.2: The consistency-robustness tradeoffs implied by the bounds for SwP and the algorithm of Bamas, Maggiori, Rohwedder, and Svensson, 2020 for $\alpha = 2$.

if $\eta \in [0, \lambda]$, *and* $2^{\alpha-1}\alpha^{\alpha}\left(\frac{1}{\mu}\right)^{\alpha-1}$ *otherwise.*

Note that Theorem 5.1 not only implies consistency and robustness, but also smoothness: the competitive ratio gracefully degrades as the error increases. In Figure 5.2 the consistency-robustness tradeoff for the algorithm is compared to that of Bamas, Maggiori, Rohwedder, and Svensson, 2020 for $\alpha = 2$.

5.4 All Jobs Have a Common Deadline

In this section, we present a simpler algorithm that achieves improved consistency and robustness over SwP for the special case in which all jobs have the same deadline, i.e., $d_j = d$ for all $j \in \mathcal{J}$. Since the deadline is the same for all jobs, we only consider predictions on the n release times $R = \{r_1, \ldots, r_n\}$ and denote these by a set $P = \{p_1, \ldots, p_n\}$.

We begin by analyzing a framework for combining different algorithms before presenting an algorithm in Subsection 5.4.1 that is based on combining two different algorithms; the classic online algorithm qOA that has a worst-case guarantee independent of the prediction error, and a second one, that considers the predictions and has a good performance in the case of small prediction error.

The general idea of combining online algorithms has been repeatedly employed in the past in the areas of online algorithms and online learning, see, for example, the celebrated results of Fiat, Rabani, and Ravid, 1994, Blum and Burch, 2000, Herbster and Warmuth, 1998, Littlestone and Warmuth, 1994. Such a technique has also been used in the learning augmented setting, see Antoniadis, Coester, Eliáš, Polak, and Simon, 2023 for an explicit framework for combining algorithms, and Lykouris and Vassilvitskii, 2021 as well as Rohatgi, 2020 for implicit uses of such algorithm combinations. However, as we will see, the specific problem considered in this paper allows for way more flexibility in such algorithm combinations since it is possible to simulate the parallel execution of

different algorithms by increasing the speed. This allows us to obtain a much more tailored result with at most one switch between the different algorithms and more straightforward analysis. We start with the following structural lemma.

Lemma 5.17. *Consider a partition of the job set of instance J into m job sets $J_1, J_2, \dots J_m$, and furthermore consider m schedules $C_1, C_2, \dots C_m$ with speed functions $s_1(t), s_2(t), \dots s_m(t)$ respectively, such that C_i is a feasible schedule for J_i for all $i = 1, \dots m$. Then there exists a schedule C with speed function $s_C(t) = \sum_i s_i(t)$ that is feasible for the complete job set J and has an energy consumption of $\mathcal{E}_C \leq m^{\alpha-1} \sum_i \mathcal{E}_i$, where for each i, $\mathcal{E}_i = \int_t s_i(t)^\alpha dt$ is the energy consumption of the respective schedule.*

5.4.1 CDSwP Algorithm

At a high level CDSwP(λ) (almost) follows the optimal schedule for the predicted instance as long as the prediction error is not higher than λ and switches to a classical online algorithm (i.e., one without predictions) in case the prediction error becomes higher than λ.

More formally, the algorithm can reside in one of two modes: *follow the prediction* (FtP) mode, and *recovery* mode. Initially, before the release time r_1 of the first job the algorithm is in the FtP-mode and has an associated speed-profile given by $s(FtP(0), t) = 0$ for all $t \in [0, d]$. Upon each release time r_i, $i = 1, \dots n$, and while in the FtP-mode, CDSwP(λ) does the following:

- If $\eta_i \leq \lambda$, CDSwP remains in the FtP-mode and updates the speed profile from $s(FtP(i-1), t)$ to $s(FtP(i), t)$ for $[r_i, d]$ with the help of a job instance J^i. Instance J^i consists of:
  - One job i' with release time $r_{i'} = r_i$, workload $w_{i'}$ equal to the total amount of unfinished workload at r_i workload that was released at any timepoint $t \leq r_i$, and deadline d.
  - For each job j not yet released at r_j, include job j with a release time of p'_j, a deadline of d and a volume of w_j in J^i.

  The new speed-profile $s(FtP(i), t)$ is given for any $t \in [r_i, d]$ by

$$s(FtP(i), t) := \begin{cases} s(YDS(J'), t), & \text{if } YDS(J') \text{ runs job } i' \text{ at } t, \\ 0, & \text{otherwise.} \end{cases}$$

  Algorithm CDSwP now runs at $s(FtP(i), t)$ for any $t \in [r_i, r_{i+1})$, and remains in the FtP-mode.

- Otherwise, if $\eta_i > \lambda$ then CDSwP switches to the recovery-mode, and sets $k := i$.

When in recovery-mode, the algorithm runs at speed $s(t) = s(FtP(k-1), t) + s(qOA(k), t)$ at each timepoint t until d, where $s(FtP(k-1), t)$ is the last speed-profile generated in the FtP-mode, and $s(qOA(k), t)$, is the speed that the online algorithm qOA would have at timepoint t when presented (in an online fashion) with (the actual) jobs $k, \dots n$.

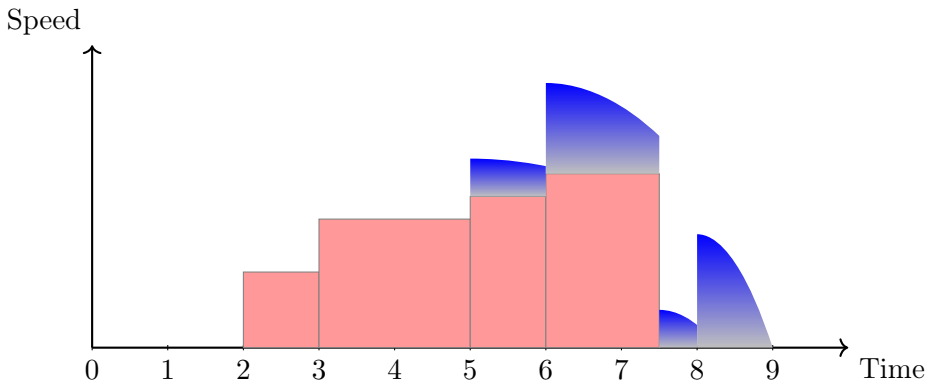


Figure 5.3: A common deadline instance with $\eta > \lambda$. The first time point with $\eta_i > \lambda$ is time 5 in which we start to run qOA for the rest of the jobs (blue part) while we continue running YDS for the jobs released before 5 (red part). At time point 7.5, the workload of the first set of jobs is finished.

Note that defining the speed at any timepoint t is sufficient in order to fully describe the algorithm. Indeed, since all jobs have a common deadline of d, it is irrelevant which job (among the active jobs) is being processed at any timepoint t. Nevertheless, to simplify the presentation we will implicitly assume in the following that at timepoint t the currently active and unfinished job with the earliest release time is the one being processed – and ties are broken arbitrarily. We first prove that the algorithm produces feasible schedules:

Observation 5.18. *Algorithm CDSwP fully processes the whole processing volume of each job w_j, within $[r_j, d]$.*

Proof. Note that by the algorithm definition, no job starts being processed before its arrival in any mode. So it suffices to show that the complete processing volume of each job is completed before its deadline. Assume first that the algorithm remains in the FtP-mode until d. By the definition of the job instances J^i, any still unfinished processing volume $w_{n'}$ will be assigned to job n' at timepoint r_n and YDS will schedule it within $[r_n, d)$ according to YDS at a speed of $w_{n'}/(d - r_n)$. So the resulting schedule is feasible in that case. If the algorithm switches to the recovery mode at some r_k, then by the above argument the speed profile $s(FtP(k-1), t)$ is sufficient to finish jobs $1, \dots k-1$, and furthermore speed profile $s(qOA(k), t)$ is feasible for jobs $k, \dots n$, by the feasibility of algorithm qOA. So the overall speed profile $s(FtP(k-1), t) + s(qOA(j), t)$ is sufficient for processing the whole volume. □

We begin by showing the following lemma which will imply consistency and smoothness.

Lemma 5.19 (Consistency & Smoothness)**.** *Under the assumption that $\eta \in [0, \lambda]$, there holds*

$$\mathcal{E}_{CDSwP} \leq \left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_R)}.$$

Before proving Lemma 5.19 we show the following intermediate result.

Lemma 5.20. *Assuming that $\eta \in [0, \lambda]$, there holds*

$$\mathcal{E}_{CDSwP} \leq \mathcal{E}_{YDS(J_{P'})}$$

Proof. Consider job instance $J^{i'}$ which consists of:

- A job j^{i-1} with release time r_i, deadline d and volume $w_{j^{i-1}} := w'_i - w_i$ equal to the total volume of jobs $1, \dots i-1$ that is still unfinished at r_i,
- Job i with release time at p'_i (and still deadline d and processing volume w_i),
- For each job j not yet released at r_j, include job j with a release time of p'_j, a deadline of d and a volume of w_i in $J^{i'}$.

Note that, instance $J^{i'}$ differs from J^i only in that job i is considered separately, and not together with all previously released jobs that are still not finished. By Observation 5.11 a YDS schedule for the former is a feasible schedule for the later, and therefore by optimality of YDS,

$$\mathcal{E}^{[r_i,\infty)}_{CDSwP(J^i)} \leq \mathcal{E}^{[r_i,\infty)}_{CDSwP(J^{i'})}, \tag{5.5}$$

where $\mathcal{E}^{[a,b]}_{A(J)}$, refers to the energy consumption that the schedule produced by algorithm A on instance J has within interval $[a, b]$.

Using this notation, we can express the total energy-consumption of the $CDSwP$ as

$$\begin{aligned}
\mathcal{E}_{CDSwP} &= \sum_{i=1}^{n} \mathcal{E}^{[r_i,r_{i+1})}_{CDSwP(J^i)} \\
&= \sum_{i=1}^{n-2} \mathcal{E}^{[r_i,r_{i+1})}_{CDSwP(J^i)} + \mathcal{E}^{[r_{n-1},r_n)}_{CDSwP(J^{n-1})} + \mathcal{E}^{[r_n,d)}_{CDSwP(J^n)} \\
&\leq \sum_{i=1}^{n-2} \mathcal{E}^{[r_i,r_{i+1})}_{CDSwP(J^i)} + \mathcal{E}^{[r_{n-1},r_n)}_{CDSwP(J^{n-1})} + \mathcal{E}^{[r_n,d)}_{CDSwP(J^{n\prime})} \\
&= \sum_{i=1}^{n-2} \mathcal{E}^{[r_i,r_{i+1})}_{CDSwP(J^i)} + \mathcal{E}^{[r_{n-1},d)}_{CDSwP(J^{n-1})} \\
&\vdots \\
&\leq \mathcal{E}^{[r_1,d)}_{CDSwP(J^1)} \leq \mathcal{E}_{YDS(J_{P'})},
\end{aligned}$$

where the inequalities follow by applying Equation (5.5). □

Now, we are ready to prove Lemma 5.19.

Proof. By combining Lemmas 5.20 and 5.13 we have,

$$\mathcal{E}_{CDSwP} \leq \mathcal{E}_{YDS(J_{P'})} \leq \left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{YDS(J_R)},$$

and the lemma directly follows. □

We note that the above proof also works in exactly the same way when only a subset A of the job set is processed.

Corollary 5.21. *Consider a set of jobs $A \subseteq \mathcal{J}$ and assume that $\eta_i \in (0, \lambda)$ holds for every job $i \in A$. Then*

$$\mathcal{E}_{CDSwP(A)} \leq \mathcal{E}_{YDS(J_{P'}(A))}.$$

We next analyze the case of inadequate predictions.

Lemma 5.22. *(Robustness) With a parameter $\eta \notin (0, \lambda)$, we have*

$$\mathcal{E}_{CDSwP} \leq 2^{\alpha} \left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}_{qOA}.$$

Proof. As in the definition of CDSwP, let k be the smallest index, such that $\eta_k > \lambda$. Hence, the algorithm switches to the recovery-mode at r_k. We partition the job set into two subsets $A = \{1, \cdots, k-1\}$ and $B = \{k, \cdots, n\}$. By Lemma 5.17, and by the fact that by Corollary 5.21 the energy consumption for set B is at most the energy consumption of qOA for the whole job instance, it suffices to upper bound the energy consumption required for set A by the total energy that $qOA(k)$ uses.

We transform the schedule obtained by CDSwP for job set A through three intermediate steps to the schedule produced by $YDS^J(R)$. Since $\mathcal{E}_{YDS^J(R)} \leq \mathcal{E}_{qOA^J(R)}$ this will imply the lemma.

Step 1: Let J^A be the job instance that contains all jobs in A, along with jobs $j = k, k+1, \ldots n$ with respective release time p'_j, deadline d and processing volume w_j.

Let $\mathcal{E}^A_{CDSwP}$, and $\mathcal{E}^A_{YDS(J^A)}$ be the energy consumptions incured while scheduling the subset of jobs A for CDSwP(J) and $YDS(J^A)$ respectively. By Corollary 5.21,

$$\mathcal{E}^A_{CDSwP} \leq \mathcal{E}^A_{YDS(J_{P'})}.$$

Let J^A_P be the job instance, consisting of the predicted release times (p_i) of jobs in set A and the "shrunk" predicted release times (p'_i) for the remaining jobs. Note that J^A_P differs from J^A only in the release-times of jobs in set A. Since $\eta_i \leq \lambda$ for any $i \in A$, there holds for any such i that $d - p'_i = 1/(1-\lambda)(d - p_i)$. By Lemma 5.6, there therefore holds

$$\mathcal{E}^A_{YDS(J_{P'})} \leq \left(\frac{1}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}^A_{YDS(J^A_P)}.$$

Consider set $P^* = \{p^*_1, \ldots p^*_{k-1}, p'_k, \ldots, p'_n\}$ with $p^*_i = p_i - \eta_i(q_i - p_i)$ for all $j \in [k-1]$. There holds

$$\mathcal{E}_{YDS(J^A_P)} \leq (1+\lambda)^{\alpha-1} \mathcal{E}_{YDS(J_{P^*})} \leq (1+\lambda)^{\alpha-1} \mathcal{E}_{YDS(J^A)}. \tag{5.6}$$

By having $c = \frac{1}{(1+\lambda)}$, and $J = J_{P^*}$ $(r_i = p^*_i)$ in Lemma 5.6, we obtain $J' = J_P$. Since we have $\eta_j < \lambda$ for all $j < k$, the first inequality in (5.6) holds. For every job $i \in A$ there

holds $(r_i, d) \subseteq (p_i^*, d)$. More specifically, a feasible schedule for J^A is feasible for J_{P^*} as well. The second inequality in (5.6) then directly follows by the optimality of YDS.

Putting things together we therefore have

$$\mathcal{E}^A_{CDSwP} \leq \left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}^A_{YDS(J^A)}. \tag{5.7}$$

Step 2: In this step, we want to compare $\mathcal{E}^A_{YDS(J^A)}$ with the energy of YDS algorithm for a new job instance in which we consider the real release times for some jobs in set B that their shrinking predictions are after their real release times.

A job instance J^l is defined, consisting of the real release times of jobs in set A, the real release times of job j in set B for which $r_j \leq p'_j$, and the shrunk prediction (p'_j) for the rest. Since moving the release times of the future jobs to the left could increase the speed (and hence increases energy) in the first part,

$$\mathcal{E}^A_{YDS(J^A)} \leq \mathcal{E}^A_{YDS(J^l)}.$$

Step 3: In the last step, we want to compare $\mathcal{E}^A_{YDS(J^l)}$ with the optimum offline algorithm (YDS) for the complete job instance J and their real release time J_R. We want to show

$$\mathcal{E}^A_{YDS(J^l)} \leq \mathcal{E}^J_{YDS(J^l)} \leq \mathcal{E}^J_{YDS(J_R)}.$$

The first inequality holds because $A \subseteq J$. Consider the difference between two job instances J_R and J^l. Since for each job i, its available time in J_R is a subset of its available time in J^l, $YDS(J_R)$ is a feasible algorithm for job instance J^l. Therefore, the second inequality holds.

So far we proved that

$$\mathcal{E}^A_{CDSwP} \leq \left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}^J_{YDS(J_R)}.$$

Since we run qOA for the job set B,

$$\mathcal{E}^B_{CDSwP} = \mathcal{E}^B_{qOA(J_R)} \leq \mathcal{E}^J_{qOA(J_R)}.$$

And by Lemma 5.17,

$$\mathcal{E}_{CDSwP} \leq 2^{\alpha-1} \cdot \left(\left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} \cdot \mathcal{E}^J_{YDS(J_R)} + \mathcal{E}^J_{qOA(J_R)}\right).$$

Since $\mathcal{E}^J_{YDS(J_R)} \leq \mathcal{E}^J_{qOA(J_R)}$,

$$\mathcal{E}_{CDSwP} \leq 2^{\alpha-1} \cdot (\mathcal{E}_{qOA})\left(\left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} + 1\right) \leq 2^{\alpha} \left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1} \cdot (\mathcal{E}_{qOA}).$$

□

Lemmas 5.19 and 5.22 together imply Theorem 5.2.

Theorem 5.2. *Given a parameter $\lambda \in [0, 1)$, algorithm CDSwP achieves a competitive ratio of*

$$\left(\frac{1+\eta}{1-\lambda}\right)^{\alpha-1}$$

if $\eta \in [0, \lambda]$, and $2^{\alpha}\left(\frac{1+\lambda}{1-\lambda}\right)^{\alpha-1}$ otherwise.

5.5 Experiments

In order to give some intuition on how the confidence parameters μ and λ affect the obtained performance guarantees of SwP, we perform some numerical experiments for different settings. Moreover, we compare our algorithm with the currently best-known online algorithm qOA, and the optimum offline algorithm YDS using real-world data. All experiments were run on a typical laptop computer, and the codes are available at Antoniadis, Jabbarzade, and Shahkarami, 2024.

We only consider $\alpha = 3$ for the experiments, as this is the typical value of α for real-world processors, see for example Critchlow, Dennard, and Schuster, 2000; Wierman, Andrew, and A. Tang, 2012. Furthermore, for qOA, we only consider $q = 2 - \frac{1}{\alpha} \approx 1.667$ since this is the value that minimizes the competitive ratio Bansal, H.-L. Chan, Katz, and Pruhs, 2012.

The input data for our experiments is the same as in Abousamra, Bunde, and Pruhs, 2013. There, jobs are generated from http requests received on the EPAs web-server. For practical reasons, we limit our input instances to the first 1000 jobs of their sample. In order to generate predictions for the input, we use a normal distribution with a mean of 0, and a standard deviation of $0.01, 0.05$, or 0.1. For each job, two samples from this distribution are taken, and each of them is scaled by the real interval length of the job. The result is then added to each job's actual release time and deadline to obtain predictions for them.

In order to illustrate the effect of parameters λ and μ, we run SwP for different combinations of these values. In particular we consider $\lambda = 0, 0.1, 0.2, 0.3$ and $\mu = 0.1, 0.2, \cdots, 1$. Our results with standard deviations 0.01, 0.05, 0.1 can be found in Figure 5.4, Figure 5.5, and Figure 5.6 respectively. The same results in terms of competitive ratio are presented in Figure 5.7. Finally, we also show the relationship between the prediction error and the competitive ratio for different values of μ and η in Figure 5.8.

To gain some intuition on the results, recall that μ denotes the portion of each block for which AVR is run. In particular, for $\mu = 1$ the SwP algorithm becomes identical to the AVR algorithm and disregards the predictions, whereas the smaller μ's value the more the predictions are trusted. This explains why the competitive ratio increases with μ. Similarly, recall that λ defines how much the predicted interval will be shrunk and that the improved competitive ratio is only proven for $\eta \leq \lambda$ but on the other hand the bigger λ gets the smaller that improvement in the competitive ratio will be. Although the best choices for λ and μ depend on the quality and/or structure of the predictions, our experiments highlight that for appropriate such choices, one can significantly improve upon the energy-consumption of qOA. To summarize, in practice the most sensible settings of λ and μ will depend on the quality as well as structure of the predictions and it may be worthwhile experimenting with different such settings.

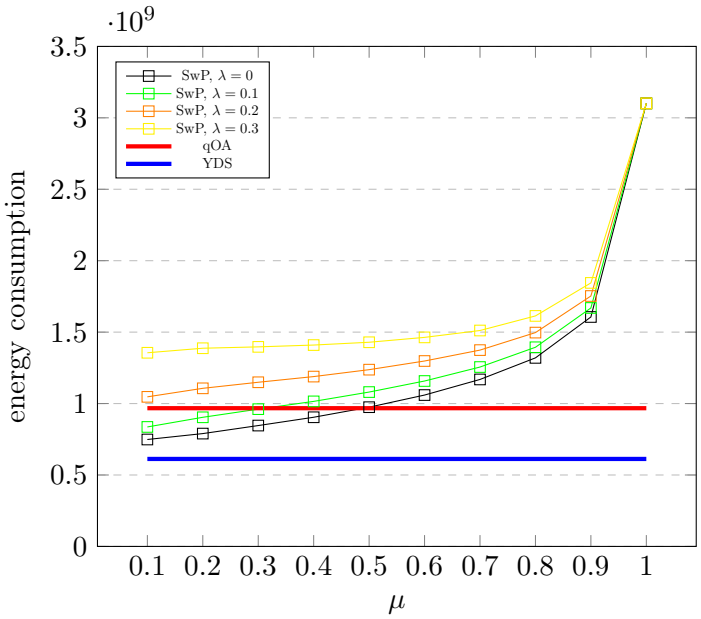


Figure 5.4: Synthetic predictions with stddev=0.01 and $\boldsymbol{\eta = 0.0370878}$. The energy consumption of SwP for different values of $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

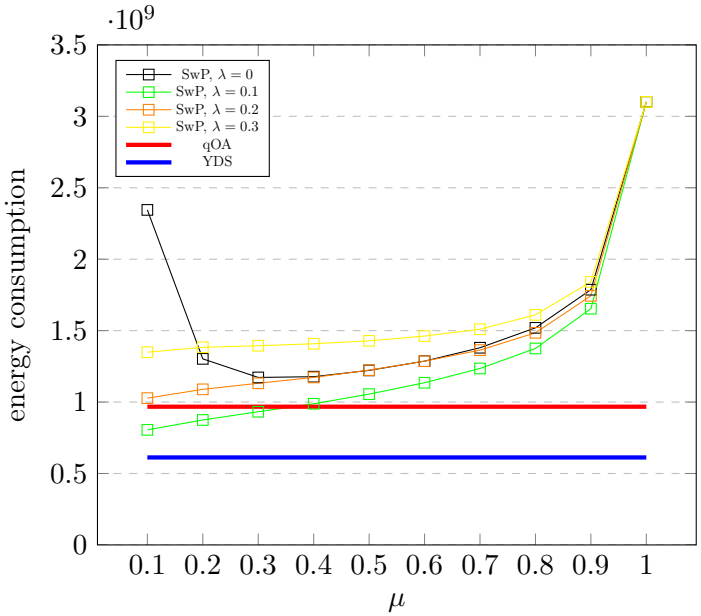


Figure 5.5: Synthetic predictions with stddev=0.05 and $\boldsymbol{\eta = 0.204602}$. The energy consumption of SwP for different values of $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

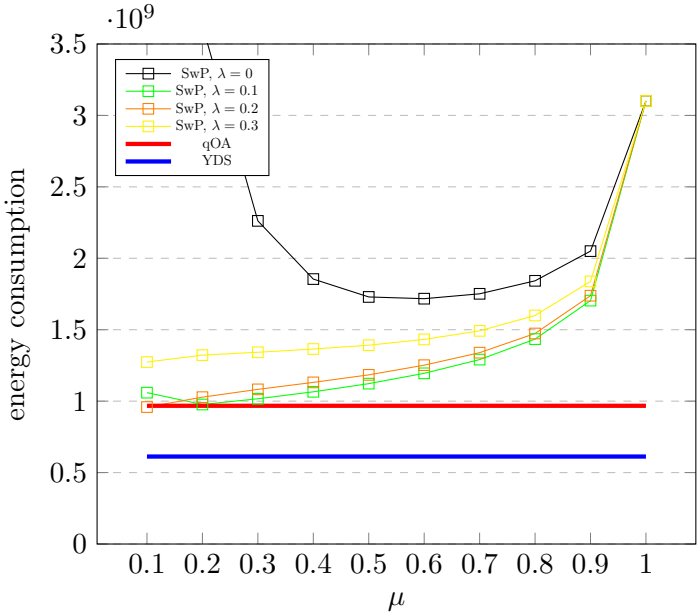


Figure 5.6: Synthetic predictions with stddev=0.1 and $\boldsymbol{\eta = 0.469849}$. The energy consumption of SwP for different values of $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

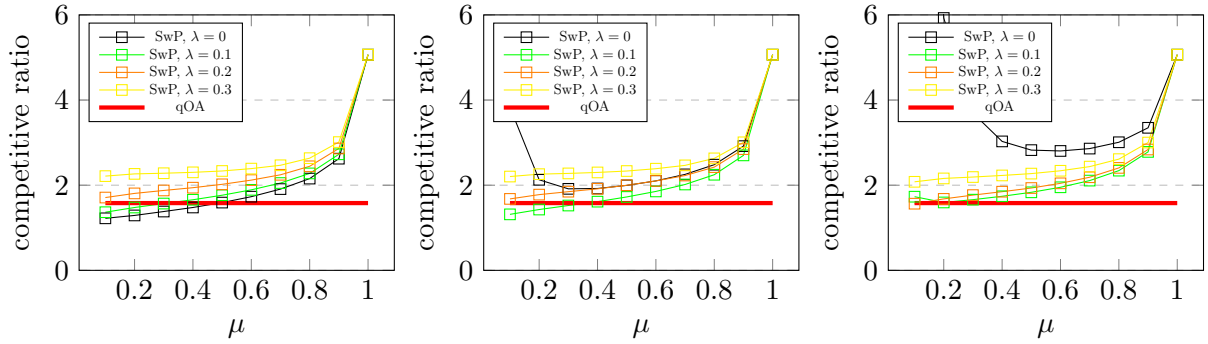


Figure 5.7: Synthetic predictions with stddev=0.01 and $\boldsymbol{\eta = 0.0370878}$, stddev=0.05 and $\boldsymbol{\eta = 0.204602}$, and stddev=0.1 and $\boldsymbol{\eta = 0.469849}$ respectively (from left to right). The respective competitive ratios of SwP for different values of $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

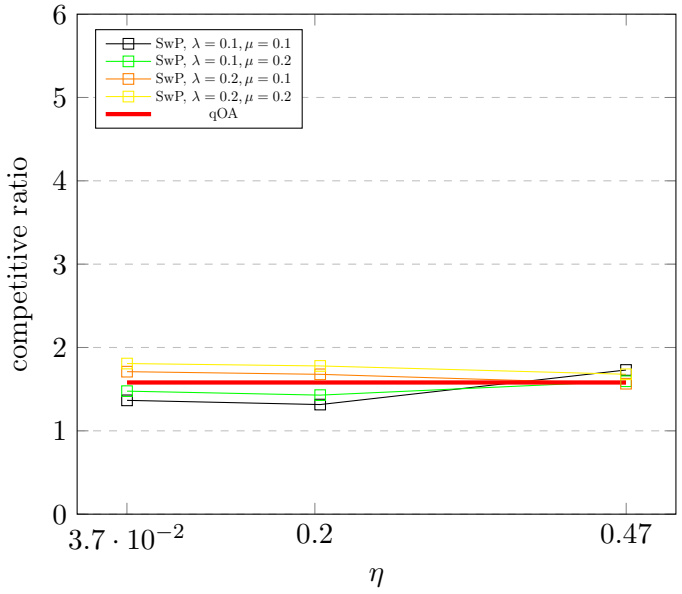


Figure 5.8: The relationship between the prediction error $\boldsymbol{\eta}$ and the obtained competitive ratio for different values of $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$.

5.6 Conclusion

In this paper, we have presented a consistent, smooth, and robust algorithm for the general classical, deadline-based, online speed-scaling problem using ML predictions for release times and deadlines.

We can remove the assumption of knowing the number of jobs n, by slightly adapting the error definition so that the prediction is considered to be inadequate if the predicted number of jobs is wrong.

It remains an interesting open question whether a similarly robust, consistent, and smooth algorithm exists for the more general setup in which the workloads of the jobs are not known in advance but predicted along with their release times and deadlines. Although we were able to extend SwP under the assumption that it satisfies a natural monotonicity property, it is unclear if that property holds in general.

5.7 Appendix

Energy of the Shrunk Instances

Lemma 5.6. *Consider a common deadline instance $\mathcal{J}$, and another common deadline instance $\hat{\mathcal{J}}$ constructed from $\mathcal{J}$ such that every job $\hat{j}_i \in \hat{\mathcal{J}}$ has workload $\hat{w}_i = w_i$, $\hat{d} = d$, and $\hat{r}_i = r_i + (1 - c_i) \cdot (d - r_i)$ for some shrinking parameter $0 < c_i \leq 1$. Set $c = \max_i c_i$. Then,*

$$\mathcal{E}_{YDS}(\hat{\mathcal{J}}) \leq (1/c)^{\alpha-1} \mathcal{E}_{YDS}(\mathcal{J}).$$

Proof. In order to prove the lemma, we start with schedule $C_{YDS}(\mathcal{J})$, in which each job j_i runs at a speed s_i. It suffices to show that there exists a feasible schedule $C'(\hat{\mathcal{J}})$ for the jobs of $\hat{J}$ in which each job $\hat{j}_i$ runs at speed of $\hat{s}_i = \frac{1}{c} \cdot s_i$. The lemma then directly follows by the definitions of the energy and the power function.

It is without loss of generality to assume that both $C_{YDS}(\mathcal{J})$ and $C'(\hat{\mathcal{J}})$ are earliest release time first schedules. Let a_i be the timepoint at which j_i starts being processed in $C_{YDS}(\mathcal{J})$, and let $\hat{a}_i = d - c(d - a_i) = d(1-c) + ca_i$. Note that by construction, feasibility of $C_{YDS}(\mathcal{J})$ and because $c < 1$, we have that $a_i \geq r_i$ and therefore $\hat{a}_i = d(1-c) + ca_i \geq d(1-c) + cr_i = \hat{r}_i$. Consider earliest release time first schedule $C(\hat{\mathcal{J}})$ in which every job j_i is processed at a speed of $\hat{s}_i$.

In the remainder of this proof, we show by induction, that in $C'(\hat{\mathcal{J}})$ every job $\hat{j}_i$ starts at $\hat{a}_i$. For the base case, it can be easily shown than in the optimal EDF schedule j_1 starts at r_1 and therefore $a_1 = r_1$. It follows that $\hat{a}_1 = \hat{r}_1$ and the job can feasibly start executing at $\hat{a}_1$. Assume that in $C'(\hat{\mathcal{J}})$ the first (according to release time) i many jobs start at their respective $\hat{a}_i$'s. Then, in order to show that $\hat{j}_{i+1}$ starts at $\hat{a}_{i+1}$ it suffices to show that $\hat{j}_i$ finishes its execution before $\hat{a}_{i+1}$. Clearly j_i finishes its execution no later than a_{i+1}. Therefore, $a_i + w_i/s_i \leq a_{i+1}$. In turn,

$$\hat{a}_i + w_i/s'_i = d_i(1-c) + ca_i + c(w_i/s_i) \leq d_i(1-c) + ca_{i+1} = \hat{a}_{i+1}.$$

and $\hat{j}_{i+1}$ can start at $\hat{a}_{i+1}$. A similar argument shows that $\hat{j}_n$ finishes before $\hat{d}$ and feasibility and therefore the proof of the lemma follows. □

Lemma 5.7. *Consider a (general) instance $\mathcal{J}$, and another instance $\hat{\mathcal{J}}$ in which every job $j_i \in \mathcal{J}$ corresponds to a job $\hat{j}_i \in \hat{\mathcal{J}}$ with workload $\hat{w}_i = w_i$, $\hat{r}_i = r_i + \frac{1-c}{2} \cdot (d_i - r_i)$ and $\hat{d}_i = d_i - \frac{1-c}{2} \cdot (d_i - r_i)$ for some shrinking parameter $0 < c \leq 1$. Then,*

$$\mathcal{E}_{YDS}(\hat{\mathcal{J}}) \leq (1/c)^{\alpha-1} \mathcal{E}_{YDS}(\mathcal{J}).$$

Proof. We may assume that $C_{YDS}(\mathcal{J})$ is an earliest deadline first schedule in which every job j_i runs at a speed of s_i. It suffices to show that there exists a feasible schedule $C'(\hat{\mathcal{J}})$ for the jobs of $\hat{J}$ in which each job $\hat{j}_i$ runs at a speed of $\hat{s}_i = \frac{1}{c} \cdot s_i$. Let the jobs be ordered by their deadlines.

Consider an earliest deadline first schedule $C'(\hat{\mathcal{J}})$ in which every job $\hat{j}_i$ runs at a speed of $\hat{s}_i$. For a job k and a job $i \leq k$, let $t_i(k)$ be the amount of time during which i is processed within (r_k, d_k). Similarly we can define $\hat{t}_i(k)$. It suffices to show that for every job j_k and every $i < k$ there holds

$$\hat{t}_i(k) \leq c \cdot t_i(k).$$

Assume for the sake of contradiction that this does not hold, and let j_k and j_i be the first pair of jobs for which it is not true. For this to be the case we must have $r_i \leq r_k \leq d_i \leq d_k$ and $\hat{r}_i \leq \hat{r}_k \leq \hat{d}_i \leq \hat{d}_k$. The reason is that if the intervals are disjoint or the interval of one job contains the other in any schedule then the inequality directly holds. So let $z = r_k - r_i$ and $\hat{z} = \hat{r}_k - \hat{r}_i$. By construction it must be that $\hat{z} \geq cz$. Therefore and by the assumption that k and i are the first such pair, more of job $\hat{j}_i$ runs in $\hat{z}$ than of job j_i in z. This leads to a contradiction since there is less of job $\hat{j}_i$ left to run after $\hat{r}_k$ □

Observation 5.11. *Under the assumption that $\eta \in (0, \lambda)$, it follows that $r_i \leq p'_i$ and $q'_i \leq d_i$ hold for every job i.*

Proof. If $p_i \geq r_i$ the observation directly follows because $p'_i \geq p_i$.

If $p_i < r_i$, let $p''_i = p_i + \lambda(q_i - p_i)$. By the assumption and the definitions of η, and η_i, we have

$$|p_i - r_i| \leq \lambda(q_i - p_i) = p''_i - p_i.$$

Then the above equation gives

$$-p_i + r_i \leq p''_i - p_i,$$

and therefore $r_i \leq p''_i$. Since $p'_i = \lfloor p''_i \rfloor$ and r_i is an integer, we can also conclude $r_i \leq p'_i$.

Same holds for the deadlines and their shrunk predictions. □

Calculating the y_i^t's

First we show the following lemma.

Lemma 5.23. *For any given $0 \leq X \leq \ell(j) \max_{t\in[r_j,d_j)}(V_t(j) + \delta_j)/\mu$ there exist values y_j^t, with $0 \leq y_j^t \leq \delta_j$ so that equations (5.2),(5.3), and (5.4) are satisfied for all $t \in [r_j, d_j)$, with X in place of X_j. Furthermore for any t, t' with $y_j^t \leq y_j^{t'}$ there holds $V_{t'}(j) \leq V_t(j)$, and $\sum y_j^t$ is a continuous and non-decreasing function in X.*

Proof. If $X/\ell(j) < \min_{t\in[r_j,d_j)} V_t(j)/\mu$, then it is easy to verify that $y_j^t = \delta_j$ for all $t \in [r_j, d_j)$ satisfies all equations. So we assume for the remainder of this proof that $\min_{t\in[r_j,d_j)} V_t(j)/\mu \leq X/\ell(j) \leq \max_{t\in[r_j,d_j)}(V_t(j) + \delta_j)/\mu$.

For any $t \in [r_j, d_j)$, let

$$y_j^t := \begin{cases} 0, & \text{if } V_t(j)/\mu \geq X/\ell(j), \\ \delta_j, & \text{if } (V_t(j) + \delta_j)/\mu \leq X/\ell(j), \\ \mu X/\ell(j) - V_t(j), & \text{otherwise.} \end{cases} \tag{5.8}$$

It is easy to verify that for the above definition of y_j^t, equations (5.2),(5.3), and (5.4) are satisfied with X in place of X_j, and that for any t, t' with $y_j^t \leq y_j^{t'}$ there holds $V_{t'}(j) \leq V_t(j)$. Finally, $\sum y_j^t$ is a continuous function as a sum of a finite number of continuous functions, and non-decreasing in X (as each y_j^t is by definition a non-increasing function of X). □

Lemma 5.24. *For any set of values $V_t(j)$, there exist values y_j^t, with $0 \leq y_j^t \leq \delta_j$ so that equations (5.2),(5.3), and (5.4) are satisfied for all $t \in [r_j, d_j)$.*

Proof. Note that it suffices to show that there exists $X_j = w_j - \sum_t y_j^t$ where the y_j^t are as defined in the proof of Lemma 5.23, since then by Lemma 5.23 the equations (5.2),(5.3), and (5.4) would hold for $X_j = w_j - \sum_t y_j^t$.

First, let $X = w_j$ and compute the values of y_j^t via (5.8). If $\sum_t y_j^t = 0$, then we have found the desired X and are done. Assume therefore, that $0 < \sum_t y_j^t \leq w_j$. By

Lemma 5.23, $\sum_t y_j^t$ is a non-decreasing and continuous function of X within $[0, w_j]$ that obtains value 0 for $X = 0$, and a value $\leq w_j$ for $X = w_j$. Equivalently the function $w_j - \sum_t y_j^t$ is non-increasing and continuous in X within $[0, w_j]$ and obtains value w_j for $X = 0$ and a value ≥ 0 for $X = w_j$. Therefore, by the intermediate value theorem there must exist an $X_j \in [0, w_j]$, such that $w_j - \sum_t y_j^t$ obtains a value of X_j, which concludes the proof of the lemma. □

Algorithm

Lemmas 5.23 and 5.24 directly imply an algorithm for identifying such values of y_j^t. In particular, since for any t, t' with $y_j^t \leq y_j^{t'}$ there holds $V_{t'}(j) \leq V_t(j)$, we can order all relevant t's by $V_t(j)$ and find (through enumeration) t', t'' such that for any $V_t(j) \geq V_t'(j)$ we have $y_j^t = 0$, for any $V_t(j) \leq V_{t''}(j)$, $y_j^t = \delta_j$ and for all other t there holds $0 < y_j^t < \delta_j$. Let N be the number of t's such that $y_j^t = \delta_j$, and $Z = w_j - N\delta_j$ be the remaining processing volume that needs to be assigned through the y_j^t's for t's with $V_{t''}(j) < V_t(j) < V_{t'}(j)$. In other words we need to find $0 < y_j^t < \delta_j$ so that $Z - \sum_t y_j^t = X_j$, and for each individual such y_j^t, we have $y_j^t = \mu X_j/\ell(j) - V_t(j)$. This implies a system of $k + 1$ equations (for some k) with $k + 1$ unknowns, that by Lemma 5.24 has a solution assuming that t', t'' were chosen correctly.

Combining Online Scheduling Algorithms

The proof of the following lemma is an adaptation of the proof of a similar result used in the analysis of the Average Rate algorithm for the problem (see Bansal, Bunde, H.-L. Chan, and Pruhs, 2011; Yao, Demers, and Shenker, 1995). However we reprove it here, since (i) we obtain a more general result, and (ii) our respective schedules do not necessarily satisfy all the properties of the corresponding schedules in Bansal, Bunde, H.-L. Chan, and Pruhs, 2011; Yao, Demers, and Shenker, 1995.

Lemma 5.17. *Consider a partition of the job set of instance J into m job sets $J_1, J_2, \ldots J_m$, and furthermore consider m schedules $C_1, C_2, \ldots C_m$ with speed functions $s_1(t), s_2(t), \ldots s_m(t)$ respectively, such that C_i is a feasible schedule for J_i for all $i = 1, \ldots m$. Then there exists a schedule C with speed function $s_C(t) = \sum_i s_i(t)$ that is feasible for the complete job set J and has an energy consumption of $\mathcal{E}_C \leq m^{\alpha-1} \sum_i \mathcal{E}_i$, where for each i, $\mathcal{E}_i = \int_t s_i(t)^\alpha dt$ is the energy consumption of the respective schedule.*

Proof. Regarding feasibility, consider a partitioning of the time horizon defined by the points $T = \cup_i \{r_i, d_i\}$. Let $t_1, t_2, \ldots$ be the points of T ordered from left to right. By definition of $s_C(t)$ we have that $\int_{t_i}^{t_{i+1}} s_C(t)dt = \sum_i \int_{t_i}^{t_{i+1}} s_i(t)$. Consider schedule C that processes in every interval (t_i, t_{i+1}) the same amount of volume for each job j as the corresponding schedule C_k with $j \in C_k$ does in this interval. The jobs inside such an interval (t_i, t_{i+1}) are processed in an arbitrary order (this is possible because the interval does not contain any release times or deadlines). Assume for the sake of contradiction that C is not feasible for J. Then there must exist a job $j \in C_k$ that misses its deadline $d_j = t_\ell$. This contradicts the feasibility of C_k since C processes exactly the same amount of job j in every interval (t_i, t_{i+1}) for $i = 0, \ldots \ell - 1$ as C_k.

With respect to the energy consumption, we have

$$\mathcal{E}_C = \int_t \left(\sum_i s_i(t) \right)^\alpha dt.$$

Note that for all i, $\int_t s_i(t)^\alpha dt \leq \mathcal{E}_i$. The lemma now follows since

$$\int_t \left(\sum_i s_i(t) \right)^\alpha dt \quad \leq \quad m^{\alpha-1} \int_t \left(\sum_i s_i(t)^\alpha \right) dt \quad = \quad m^{\alpha-1} \left(\sum_i \mathcal{E}_i \right),$$

where the first inequality follows by Jensen's inequality. □

CHAPTER 6

Online Interval Scheduling with Predictions

Chapter overview. This chapter examines online interval scheduling with irrevocable decisions in the presence of uncertainty about future arrivals. It studies how predictive information can be incorporated into online algorithms to balance consistency under accurate predictions with robustness against adversarial inputs. This chapter is based on joint work with Antonios Antoniadis, Ali Shahheidar, and Abolfazl Soltani, which appeared at AAAI 2026.

6.1 Introduction

The online interval scheduling problem is a well-studied model in algorithm design that captures the challenge of selecting a subset of non-overlapping time intervals with the goal of optimizing a given objective. Intervals arrive sequentially over time, and the algorithm must irrevocably decide, upon each arrival, whether to accept or reject the interval without knowledge of future arrivals. Each interval typically represents a task, job, or request, and one natural objective is to maximize the total length of the accepted intervals. This model arises in a wide range of practical applications, including computing resource management, manufacturing systems, and real-time service scheduling. Additional domains include vehicle routing and satellite communication scheduling (Bar-Noy, Canetti, Kutten, Mansour, and Schieber, 1999). Two surveys by Kolen, Lenstra, Papadimitriou, and Spieksma, 2007; Kovalyov, Ng, and T. C. E. Cheng, 2007 provide comprehensive overviews of interval scheduling and its many application areas.

While classical online algorithms for interval scheduling, such as greedy approaches, offer strong worst-case guarantees, their performance can be overly pessimistic in practical settings. This is further highlighted by the fact that most lower-bound proofs rely on very carefully designed and fragile instances. In many real-world systems, partial information about future intervals is often available through historical trends or forecasting techniques, including machine-learned predictions. This motivates the study of *learning-augmented algorithms* (also known as algorithms with predictions), which aim to bridge the gap between worst-case online guarantees and the performance of offline algorithms by leveraging predicted information during the decision process.

As in the traditional online setting, the standard formulation of interval scheduling requires decisions, that is acceptance or rejection of intervals, to be made irrevocably. However, it is known that no online algorithm can achieve a constant competitive ratio for this problem in general (Lipton and Tomkins, 1994; Long and Thakur, 1993). To address this limitation, a relaxed model has been considered in which accepted intervals may be revoked prior to their deadlines, while rejections remain final. Several objective functions have been considered for the classic online interval scheduling problem, including maximizing the number of accepted intervals (unit-weight) and maximizing the

total length of accepted intervals (proportional-weight). Boyar, Favrholdt, Kamali, and Larsen, 2023 studies the unit-weight case, where the goal is to maximize the number of accepted intervals using predicted information. Moreover, Karavasilis, 2025 considers a proportional-weight objective, aiming to maximize the total length of accepted intervals in a revocable setting.

This leaves a natural gap: the irrevocable setting with proportional weights, which models many practical scenarios where job values depend on duration and decisions cannot be revoked. In this work, we fill this gap by providing a comprehensive analysis of learning-augmented interval scheduling under irrevocable decisions. We present both upper and lower bounds, characterizing the trade-offs between consistency and robustness in both deterministic and randomized settings. Moreover, we propose a smooth randomized algorithm and provide both theoretical guarantees and experimental results demonstrating its effectiveness on real-world data.

6.1.1 Our Contributions and Techniques

A common design pattern in learning-augmented algorithms is to combine two strategies: one that follows the predictions and one that ignores them entirely. This allows the algorithm to adapt its behavior based on the quality of the prediction. The challenge, of course, is deciding how and when to rely on each component.

A natural approach is to switch between a predictive and a classic algorithm based on the reliability of the prediction. This technique has been used effectively in several minimization problems (Antoniadis, Coester, Eliáš, Polak, and Simon, 2023; Bamas, Maggiori, Rohwedder, and Svensson, 2020; Bansal, Coester, R. Kumar, Purohit, and Vee, 2020; Lykouris and Vassilvitskii, 2021; Purohit, Svitkina, and R. Kumar, 2018; Rohatgi, 2020; A. Wei, 2020), where the algorithm initially relies on the prediction and transitions to a backup strategy when the prediction appears unreliable, potentially continuing to switch as new evidence emerges. While switching in maximization problems poses additional challenges, it has been explored in online bipartite matching with imperfect advice (Choo, Gouleakis, Ling, and Bhattacharyya, 2024). In this paper, we apply the switching technique to irrevocable online interval scheduling, where early commitments may permanently block future high-value intervals. Therefore, the timing of the switch, as well as the behavior in the prediction trusting phase, becomes particularly delicate.

While switching is a natural way to combine two strategies—one that trusts the prediction and one that ignores it—another powerful approach arises in the randomized setting: merging strategies probabilistically. We can define a randomized algorithm that selects between a predictive and a classic strategy based on a carefully chosen probability distribution. Leveraging this idea, we design a randomized algorithm that not only maintains robustness but also achieves *smoothness*—its performance degrades gracefully as the prediction quality deteriorates.

We now summarize our main results and techniques, which build on these merging principles.

The Trust-and-Switch Framework. We introduce a general mechanism that allows algorithms to balance predictive and non-predictive behavior. The Trust-and-Switch framework takes as input a prediction model and a classic algorithm, and it

decides—based on an online evaluation of the prediction—when to switch from trusting the prediction to relying on the classic algorithm. The switch occurs at most once and is irrevocable. We show that this simple structure captures the trade-off between consistency and robustness. More specifically, it achieves 1-consistency, and if the algorithm used after the switch is θ-competitive, then the overall solution satisfies

$$|\text{OPT}| \leq \theta \cdot |\textsc{Alg}| + 2k,$$

where k is the maximum interval length.

The SemiTrust-and-Switch Framework. In addition to distinguishing between additive and multiplicative loss, our goal is to expose a tunable trade-off between consistency and robustness. This motivates a refined variant of the framework: instead of blindly following the prediction, the algorithm only follows the prediction for intervals whose length is below a threshold τ, and accepts longer intervals greedily. This hybrid approach improves robustness without sacrificing too much consistency—in effect, it interpolates between fully trusting and fully ignoring the prediction. More specifically, it achieves $(1 + \frac{k}{\tau})$-consistency, and if the algorithm used after the switch is θ-competitive, then the overall solution satisfies

$$|\text{OPT}| \leq \max(\theta, \Delta + 1) \cdot |\textsc{Alg}| + \tau,$$

where Δ is the ratio between the longest and shortest interval lengths.

Prediction Settings and Lower Bounds. Our frameworks work with a binary prediction model (denoted $\hat{\mathbf{O}}$), where predictions arrive online with the intervals and suggest whether to accept ($\hat{o}_i = 1$) or reject ($\hat{o}_i = 0$) interval I_i. We also establish lower bounds under a stronger, full-information setting (denoted $\hat{\mathbf{I}}$). Interestingly, our results show that having access to more detailed predictions, or receiving them offline, does not improve the achievable guarantees. In particular, for the Trust-and-Switch framework, we prove matching bounds for two-value instances: any $(1 + \alpha)$-consistent algorithm $\textsc{Alg}'$ with $\alpha < \frac{\lceil \Delta \rceil - 1}{\Delta}$ must incur a robustness loss of at least $\Delta \cdot |\textsc{Alg}'| + k$, even when given full predictions. This matches the guarantee of the framework for this class of instances, making the Trust-and-Switch framework optimal even when full predictions are available. Moreover, we provide a lower bound for our SemiTrust-and-Switch framework.

The SmoothMerge Algorithm. We introduce a randomized algorithm, SmoothMerge, that merges the behaviors of two strategies in a probabilistic manner. The algorithm simulates both strategies in parallel and independently. Upon the arrival of each interval, proceeds as follows: if the interval conflicts with previously accepted intervals, it is immediately rejected; otherwise, if both simulations agree on accepting (or rejecting), the algorithm follows that unanimous decision. When the simulations disagree, the algorithm accepts the interval with probabilities p_t and p_g, corresponding to the two strategies. This probabilistic merging yields both smoothness and robustness properties. Specifically, letting η denote the prediction error, we have:

$$\mathbb{E}[|\textsc{Alg}|] \geq \max\left\{|\text{OPT}| \cdot (1 - \eta) \cdot p_t \cdot (1 - p_g), \frac{p_g - p_t p_g}{1 - p_t p_g} \cdot |\textsc{Greedy}|\right\}.$$

6.1.2 Related Work

Online Interval Scheduling. The online interval scheduling problem was introduced by Lipton and Tomkins, 1994, who established the classical model in which intervals arrive sequentially and must be irrevocably accepted or rejected upon arrival. The goal is to maximize the total length of accepted, non-overlapping intervals. Long and Thakur, 1993 showed that there is no deterministic algorithm that can achieve a constant competitive ratio, and Lipton and Tomkins, 1994 showed that no randomized algorithm can achieve a competitive ratio better than $O(\log \Delta)$, where Δ is the ratio between the longest and shortest intervals, and they provided an $O((\log \Delta)^{1+\varepsilon})$-competitive algorithm.

Several extensions of online interval scheduling have been studied. In the *revocable* setting, where accepted intervals can be later removed, improved competitive guarantees are possible (Borodin and Karavasilis, 2023; Garay, Gopal, Kutten, Mansour, and M. Yung, 1997; Tomkins, 1995). Another extension of online interval scheduling is the *any order* setting (Borodin and Karavasilis, 2023), where the order of intervals can be arbitrary, and the *sorted order* setting (Lipton and Tomkins, 1994), where the intervals arrive in order of their release time. In the *weighted* setting, where intervals have arbitrary weights, no deterministic or randomized algorithm can achieve a finite competitive ratio in general (Canetti and Irani, 1995; Woeginger, 1994). However, for weight functions tied to interval length, Woeginger, 1994 identified classes (e.g., benevolent functions) that admit finite guarantees, and Seiden, 2000 gave improved randomized bounds. The complexity also varies with input structure: while disjoint path allocation is solvable in polynomial-time on trees and outerplanar graphs (Garg, Vazirani, and Yannakakis, 1997; Wagner and Weihe, 1995), it becomes NP-complete on general and series-parallel graphs (Even, Itai, and Shamir, 1976; Nishizeki, Vygen, and Zhou, 2001). For a comprehensive overview of the extensive literature on interval scheduling, we refer to the survey by Kolen, Lenstra, Papadimitriou, and Spieksma, 2007, which provides thorough coverage of algorithmic techniques and complexity results across various problem variants.

Learning-augmented Algorithms. Traditional worst-case analysis often fails to capture the practical performance of algorithms, motivating the study of beyond worst-case analysis (Roughgarden, 2021). One prominent framework in this direction is that of learning-augmented algorithms (also known as algorithms with predictions), which incorporate machine-learned advice into decision-making (Mitzenmacher and Vassilvitskii, 2022). Lykouris and Vassilvitskii, 2021 formalized this approach in the context of online caching, introducing the now-standard notions of consistency and robustness to measure how well an algorithm performs under both accurate and adversarial predictions. This framework has since been applied to a wide range of online problems, including caching (Antoniadis, Coester, Eliáš, Polak, and Simon, 2023; Lykouris and Vassilvitskii, 2021; Rohatgi, 2020; A. Wei, 2020), secretary and matching problems (Antoniadis, Gouleakis, Kleer, and Kolev, 2020; Dütting, Lattanzi, Leme, and Vassilvitskii, 2021), knapsack (Im, R. Kumar, Qaem, and Purohit, 2021), Traveling Salesman Problem (Bampis, Escoffier, Gouleakis, Hahn, Lakis, Shahkarami, and Xefteris, 2023; Chawla and Christou, 2024; Gouleakis, Lakis, and Shahkarami, 2023), online graph algorithms (Azar, Panigrahi, and Touitou, 2022), and various scheduling problems (Antoniadis, Jabbarzade, and Shahkarami, 2022; Bamas, Maggiori, Rohwedder, and Svensson, 2020; Lattanzi, Lavastida,

Moseley, and Vassilvitskii, 2020; Mitzenmacher, 2020; Purohit, Svitkina, and R. Kumar, 2018). The scope of this area also extends beyond online problems. For an up-to-date list of papers in this field, we refer the reader to a curated online repository by Lindermayr and Megow, n.d.

Online Interval Scheduling with Predictions. Finally, the closest works to our own are Boyar, Favrholdt, Kamali, and Larsen, 2023 and Karavasilis, 2025. Boyar, Favrholdt, Kamali, and Larsen, 2023 study the unit-weight interval scheduling problem, where the objective is to maximize the number of accepted intervals, given predictions about the input sequence. They provide tight bounds on the competitive ratio achievable by deterministic algorithms. Their approach combines prediction-following with greedy acceptance when it is possible without a decrease in the profit of the planned solution. More recently, Karavasilis, 2025 considered proportional-weight interval scheduling in a revocable setting, where the objective is to maximize the total length of accepted intervals and decisions can be revoked before deadlines. This work focuses on binary predictions and develops algorithms that can adaptively modify their decisions based on observed prediction quality. Our work addresses a crucial gap by studying proportional-weight interval scheduling with irrevocable decisions.

6.2 Preliminaries

The input to the *online interval scheduling problem* consists of a set $\mathcal{I}$ of n intervals. Each interval $I_j \in \mathcal{I}$ is defined by a release time r_j and a deadline d_j, with length $l_j = d_j - r_j$. Intervals arrive online: the information about interval I_j, namely its release time, deadline, and length, becomes available to the algorithm only at time r_j. Upon arrival, the algorithm must make an irrevocable decision to either accept or reject the interval, without knowledge of future intervals. The accepted intervals must be non-overlapping, and the objective is to maximize the total length of the accepted subset.

Throughout this work, we assume that intervals arrive in non-decreasing order of their release times. Without loss of generality, we label the intervals $I_1, I_2, \ldots, I_n$ in order of arrival, so that $r_1 \leq r_2 \leq \cdots \leq r_n$.

We define the following key parameters:

- $k = \max_{I_j \in \mathcal{I}} l_j$: the maximum length of any interval.
- $\Delta = \frac{k}{\min_{I_j \in \mathcal{I}} l_j}$: the ratio between the maximum and minimum interval lengths.

Given an algorithm $\mathcal{A}$, we use $\mathcal{A}(\mathcal{I})$ to denote the outcome of $\mathcal{A}$ on instance $\mathcal{I}$, and $|\mathcal{A}(\mathcal{I})|$ to denote the total length of intervals selected by it.

We define notation for subsets of intervals based on their release times and deadlines. For timepoints $t_1, t_2 \in \mathbb{R}_{\geq 0}$, let $\mathcal{I}(t_1, t_2)$ denote the subset of intervals in $\mathcal{I}$ whose release times and deadlines satisfy constraints determined by t_1 and t_2, respectively. We use the symbol $\cdot$ to indicate that no constraint is applied to that component (release time or deadline). Superscripts $\leftarrow$ and $\rightarrow$ indicate the direction of the inequality: for a timepoint t, $t^{\leftarrow}$ denotes times at or before t (i.e., $\leq t$), and $t^{\rightarrow}$ denotes times strictly after t (i.e., $> t$). For example:

- $\mathcal{I}(t_1^{\leftarrow}, \cdot)$ is the set of intervals with release time $\leq t_1$ (no constraint on deadline),
- $\mathcal{I}(\cdot, t_2^{\leftarrow})$ is the set of intervals with deadline $\leq t_2$ (no constraint on release time),
- $\mathcal{I}(t_1^{\leftarrow}, t_2^{\rightarrow})$ is the set of intervals with release time $\leq t_1$ and deadline $> t_2$.

We denote the optimal offline algorithm by OPT. In particular, $|\mathrm{OPT}(\mathcal{I})|$ denotes the total length of the optimal offline solution, and $|\textsc{Alg}(\mathcal{I})|$ denotes the total length achieved by the online algorithm under consideration.

The performance of an online algorithm is evaluated using the *competitive ratio*, which measures the worst-case performance relative to the optimal offline solution. The competitive ratio is defined as:

$$\text{competitive ratio} = \sup_{\mathcal{I}} \frac{|\mathrm{OPT}(\mathcal{I})|}{|\textsc{Alg}(\mathcal{I})|},$$

where the supremum is taken over all possible input instances $\mathcal{I}$. An online algorithm is said to be *c-competitive* if competitive ratio $\leq c$. In this case, the total length of intervals accepted by the algorithm is guaranteed to be at least $1/c$ of the total length of intervals accepted by the optimal offline solution for any input instance.

For randomized algorithms, the competitive ratio is defined based on the expected performance of the algorithm. Let $\mathbb{E}[\textsc{Alg}(\mathcal{I})]$ denote the expected total length of intervals accepted by the randomized online algorithm. The competitive ratio for randomized algorithms is defined as:

$$\text{competitive ratio (randomized)} = \sup_{\mathcal{I}} \frac{|\mathrm{OPT}(\mathcal{I})|}{\mathbb{E}[\textsc{Alg}(\mathcal{I})]}.$$

A randomized online algorithm is said to be *c-competitive* if competitive ratio (randomized) $\leq c$, meaning that the expected total length of intervals accepted by the algorithm is at least $1/c$ of the total length accepted by the optimal offline solution, for any input instance $\mathcal{I}$.

Prediction Setup. Learning-augmented algorithms represent a class of algorithms that leverage pre-computed predictions to enhance decision-making in online settings. These predictions, often derived from sources such as statistical models or machine learning systems, provide guidance to the algorithm without requiring it to learn from real-time data. In the context of the online interval scheduling problem, we consider the following prediction settings:

- **Predictions $\hat{\mathbf{O}}$:** The predictions $\hat{\mathbf{O}} = \{\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_n\}$ are a binary vector, where $\hat{o}_i \in \{0, 1\}$ for each interval $I_i \in \mathcal{I}$. The value $\hat{o}_i = 1$ indicates that the interval I_i should be accepted, while $\hat{o}_i = 0$ indicates that the interval should be rejected. This prediction represents a suggested optimal solution.
- **Predictions $\hat{\mathbf{I}}$:** The predictions $\hat{\mathbf{I}}$ aim to provide complete foresight into the characteristics of all intervals in the input. Specifically, $\hat{\mathbf{I}} = \{(\hat{r}_1, \hat{d}_1), (\hat{r}_2, \hat{d}_2), \ldots, (\hat{r}_n, \hat{d}_n)\}$, where each tuple $(\hat{r}_i, \hat{d}_i)$ contains the predicted release time $\hat{r}_i$ and deadline $\hat{d}_i$ of interval I_i. The length of each interval can be computed as $\hat{l}_i = \hat{d}_i - \hat{r}_i$. This prediction setting aims to provide the algorithm with more detailed information to guide its decisions.

Performance Metrics. We use $*$ to denote a generic prediction model, which can represent any of the specific prediction settings considered in this work ($\hat{\mathbf{O}}$ or $\hat{\mathbf{I}}$). If $\mathcal{I} \lhd *$ denotes all instances $\mathcal{I}$ for which the prediction $*$ is accurate, an algorithm is

- α-*consistent* if it is α-*competitive* when the prediction is correct, i.e.,

$$\max_{\mathcal{I},*:\mathcal{I}\lhd *} \left\{ \frac{|\mathrm{OPT}(\mathcal{I})|}{|\mathrm{ALG}(\mathcal{I},*)|} \right\} \leq \alpha,$$

  where $\mathrm{ALG}(\mathcal{I},*)$ denotes the solution returned by the algorithm using the prediction $*$, and $\mathrm{OPT}(\mathcal{I})$ denotes the optimal offline solution.

- β-*robust* if it is β-*competitive* regardless of the quality of the prediction, i.e.,

$$\max_{\mathcal{I},*} \left\{ \frac{|\mathrm{OPT}(\mathcal{I})|}{|\mathrm{ALG}(\mathcal{I},*)|} \right\} \leq \beta,$$

  where $\mathcal{I}$ denotes all possible input instances, and the algorithm's performance is bounded even when the prediction $*$ is arbitrarily inaccurate.

  We denote by Trust the algorithm that follows the prediction blindly. Given a prediction $\hat{\mathbf{O}}$, for each interval I_j, it accepts I_j if and only if $\hat{o}_j = 1$.

6.3 Warm-up: Trust-and-Switch Framework

In this section, we introduce a general framework for addressing the online interval scheduling problem with predictions. Roughly speaking, the idea is to begin by following the predictions and fall back to a classic algorithm once it is certain that the predictions are not perfectly accurate. Using this framework, we design both deterministic and randomized algorithms that utilize predictions $\hat{\mathbf{O}}$ about the optimal set of intervals.

At time r_j, the algorithm has access to the set $\mathcal{I}(r_j^{\leftarrow}, \cdot)$, that is, all intervals released up to that point, alongside with the corresponding predictions for these intervals.

Framework Trust-and-Switch. The general framework receives intervals and predictions in an online manner, along with a classic algorithm as input, and returns a robust consistent online algorithm. It consists of two phases: the *trust phase* and the *independent phase*. The algorithm begins in the *trust phase*, where it follows the predictions blindly and may transition to the *independent phase* at a designated time point if the predictions are found to be inaccurate. In the *trust phase*, decisions on whether to accept or reject an interval are made solely based on the prediction. In contrast, the *independent phase* ignores the predictions entirely and makes decisions using the given classic algorithm, without relying on any prediction.

Upon the arrival of interval I_j at time r_j, the framework evaluates the predictions, and the moment it is certain that the predictions are inaccurate, it decides to switch to the classic algorithm. Although the framework re-evaluates the accuracy of the predictions at each release time r_j, it also takes into account the whole of $\mathcal{I}(r_j^{\leftarrow}, \cdot)$.

More formally, upon the arrival of interval I_j at time r_j, the framework computes the *evaluation point*:

$$t_j := \max\left(\{r_j\} \cup \{d_i \mid I_i \in \mathcal{I}(r_j^{\leftarrow}, \cdot),\ \hat{o}_i = 1\}\right).$$

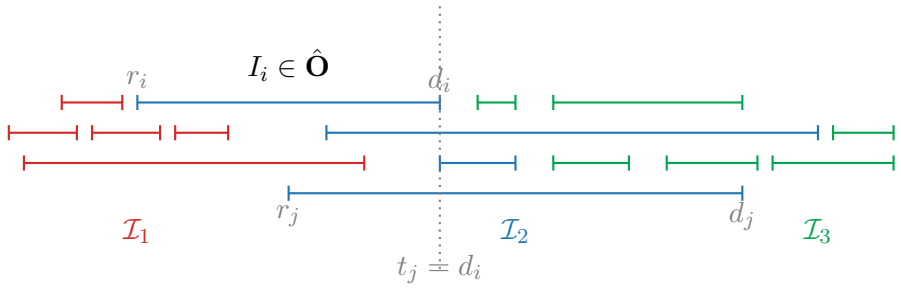


Figure 6.1: Classification of intervals when, at timepoint r_j, the framework decides to switch based on the evaluation up to $t_j = d_i$, where $I_i \in \hat{\mathbf{O}}$. The three classes $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ are illustrated with distinct visual styles in the figure.

This value ensures that if there is an accepted interval predicted to be in $\hat{\mathbf{O}}$ that is still active at time r_j, we evaluate the prediction up to the last deadline among predicted intervals in $\mathcal{I}(r_j^{\leftarrow}, \cdot)$. Otherwise, we simply consider r_j as the evaluation point.

Note that if $\hat{o}_j = 1$, then the corresponding evaluation point is $t_j = d_j$; however, if we are already certain at r_j about the inaccuracy of the predictions, the switch to the *independent phase* can take place earlier.

To evaluate the accuracy of the prediction up to the time t_j, the algorithm first verifies the consistency of $\hat{\mathbf{O}}$: in other words, whether all intervals in $\mathcal{I}(r_j^{\leftarrow}, \cdot)$ with $\hat{o}_i = 1$, including I_j, are non-overlapping. If not, the prediction $\hat{\mathbf{O}}$ is clearly invalid, and the algorithm switches as soon as possible to the *independent phase*.

Otherwise, it compares the quality of the prediction with the offline optimum up to time t_j. Specifically, it verifies whether

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| > |\textsc{Trust}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))|.$$

If this inequality holds, the algorithm transitions to the *independent phase* and starts to run the classic algorithm as soon as possible – that is: at r_j if $I_j \in \hat{\mathbf{O}}$, and at t_j otherwise – on the instance $\mathcal{I}(r_j^{\rightarrow}, \cdot)$ (resp. $\mathcal{I}(t_j^{\rightarrow}, \cdot)$) consisting of the intervals released after this point. Otherwise, it continues in the *trust phase* and accepts I_j if and only if $\hat{o}_j = 1$.

We first show that whenever the framework switches to the *independent phase*, the predicted solution is indeed suboptimal. This implies that whenever the predictions are accurate, the framework remains in the *trust phase*; this guarantees the 1-consistency of the framework.

Lemma 6.1. *The* Trust-and-Switch *framework satisfies* 1*-consistency.*

Proof. We want to show that if the predictions are accurate, the framework never switches which directly implies its 1-consistency.

Assume for contradiction that the prediction $\hat{\mathbf{O}}$ is accurate, but the framework switches at time t_j. By construction, this means that

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| > |\textsc{Trust}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))|.$$

Since t_j is either a release time or deadline of some interval in $\hat{\mathbf{O}}$, or the release time of an interval not in $\hat{\mathbf{O}}$ that does not overlap with any predicted interval, we can partition the predicted intervals into two disjoint subsets:

- $\mathcal{I}_{\text{before}} := \{I_i \in \mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}) \mid \hat{o}_i = 1\}$, i.e., predicted intervals released by r_j, that complete at or before t_j,
- $\mathcal{I}_{\text{after}} := \{I_i \in \mathcal{I}(t_j^{\rightarrow}, \cdot) \mid \hat{o}_i = 1\}$, i.e., predicted intervals released after t_j.

Now consider constructing a new feasible solution by replacing $\mathcal{I}_{\text{before}}$ with the intervals in $\text{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))$ and keeping $\mathcal{I}_{\text{after}}$ unchanged (note that $\mathcal{I}_{\text{after}}$ contains no two intersecting intervals, by the assumption on the optimality of the prediction, and furthermore does not contain any interval intersecting with $\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow})$). The total length of this new solution is

$$|\text{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j))| + \sum_{I_i \in \mathcal{I}_{\text{after}}} l_i > \sum_{I_i \in \mathcal{I}_{\text{before}}} l_i + \sum_{I_i \in \mathcal{I}_{\text{after}}} l_i = |\textsc{Trust}(\mathcal{I})|,$$

contradicting the optimality of the prediction.

Therefore, the framework does not switch when the predictions are accurate, and thus guarantees 1-consistency. □

We now analyze the robustness of the Trust-and-Switch framework. Specifically, we show that even in the presence of inaccurate predictions, the total value of the optimal solution is at most $\theta \cdot |\textsc{Alg}| + 2k$, where the additive loss accounts for following inaccurate predictions in the *trust phase* and the transition to the *independent phase*, and the multiplicative term corresponds to the use of a θ-competitive algorithm in the *independent phase*.

Theorem 6.2. *Consider the Trust-and-Switch framework Alg, and assume that the independent phase uses a θ-competitive algorithm. Then the algorithm satisfies*

$$|OPT| \leq \theta \cdot |\textsc{Alg}| + 2k.$$

Proof. Assume that the framework decides to switch to the *independent phase* at time r_j, based on the evaluation of the prediction up to t_j, and we have to wait until then to be able to switch. If the framework never switches, we set $j = n$.

To analyze the performance of the algorithm, we partition the intervals into three disjoint classes based on their relation to the switching point t_j (see Figure 6.1):

- $\mathcal{I}_1 := \mathcal{I}(\cdot, t_j^{\leftarrow})$, intervals with both release time and deadline strictly before t_j,
- $\mathcal{I}_2 := \mathcal{I}(t_j^{\leftarrow}, t_j^{\rightarrow})$, intervals with release time at or before t_j and deadline strictly after t_j,
- $\mathcal{I}_3 := \mathcal{I}(t_j^{\rightarrow}, \cdot)$, intervals with release time strictly after t_j.

We now analyze the contribution of each class to the overall value of the optimal solution and the algorithm, beginning with Class $\mathcal{I}_1$.

Note that $j > 1$, since the algorithm does not decide to switch at t_1, regardless of whether $\hat{o}_1 = 1$ or $\hat{o}_1 = 0$. We distinguish two cases based on how t_j is determined:

- **Case 1:** $t_j = d_i$ for some $i = \arg\max\{d_i \mid I_i \in \mathcal{I}(r_j^{\leftarrow}, \cdot),\ \hat{o}_i = 1\}$ and $i \neq j$.

  In this case, $t_i = t_j = d_i$. Since the framework did not switch at r_i, it must be that:

  $$\begin{aligned}|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| &\leq |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| \\ &= |\textsc{Trust}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))|.\end{aligned}$$

  Since Alg accepted interval I_i, and we decide to switch at r_j with $r_i < r_j < d_i$, we have:

  $$|\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))| = |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))|.$$

  Meanwhile, the optimal offline solution may include additional intervals w with $r_i < r_w < d_w < d_i = t_j$, whose total length is at most l_i. Thus:

  $$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))| < |\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| + l_i.$$

  Therefore,

  $$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| < |\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| + k.$$

- **Case 2:** $t_j = d_j$ and $\hat{o}_j = 1$.

  In this case, we have $t_j = d_j$. Consider the prefix $\mathcal{I}(r_{j-1}^{\leftarrow}, \cdot)$. If I_j overlaps with a previously predicted interval, we can bound the loss using the same argument as in Case 1. Therefore, we assume that the predicted intervals do not conflict with each other. In this case, we must have $t_{i-1} < r_i$. Since the framework did not switch at t_{i-1}, we have:

  $$\begin{aligned}|\mathrm{OPT}(\mathcal{I}(r_{i-1}^{\leftarrow}, t_{i-1}^{\leftarrow}))| &\leq |\textsc{Alg}(\mathcal{I}(r_{i-1}^{\leftarrow}, t_{i-1}^{\leftarrow}))| \\ &= |\textsc{Trust}(\mathcal{I}(r_{i-1}^{\leftarrow}, t_{i-1}^{\leftarrow}))|.\end{aligned}$$

  Since Trust accepts interval I_i, we have:

  $$|\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, t_i^{\leftarrow}))| = |\textsc{Alg}(\mathcal{I}(r_{i-1}^{\leftarrow}, t_{i-1}^{\leftarrow}))| + l_i.$$

  On the other hand, the optimal offline solution may include one additional interval w with $r_w \leq r_{i-1}$ and $t_{i-1} < d_w < t_i$. Its length cannot exceed k, so:

  $$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, t_i^{\leftarrow}))| < |\mathrm{OPT}(\mathcal{I}(r_{i-1}^{\leftarrow}, t_{i-1}^{\leftarrow}))| + k.$$

  Therefore,

  $$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, t_i^{\leftarrow}))| < |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, t_i^{\leftarrow}))| + (k - l_i).$$

- **Case 3:** $t_j = r_j$.

  In this case, t_j is determined by the release time of I_j, which implies that no predicted interval with $\hat{o}_i = 1$ in $\mathcal{I}(r_j^{\leftarrow}, \cdot)$ overlaps I_j. Hence, $t_{j-1} < t_j = r_j$. Because the framework did not switch at t_{j-1}, we must have

  $$\begin{aligned}|\mathrm{OPT}(\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}))| &\leq |\textsc{Alg}(\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}))| \\ &= |\textsc{Trust}(\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}))|.\end{aligned}$$

At time t_j, the framework does switch, so

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| > |\textsc{Trust}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))|.$$

The only difference between $\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow})$ and $\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow})$ is the set of intervals w whose release time is at or before t_{j-1} and whose deadline lies in the open interval (t_{j-1}, r_j). More formally,

$$\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}) \setminus \mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}) = \mathcal{I}(t_{j-1}^{\leftarrow},\ (t_{j-1}, r_j)).$$

Because all such intervals contain the point t_{j-1}, the non-overlapping constraint implies that the optimal solution can select at most one of them. Consequently,

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| \leq |\mathrm{OPT}(\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}))| + k.$$

Therefore, since ALG does not accept any new interval in the time window (t_{j-1}, t_j), we have $|\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| = |\textsc{Alg}(\mathcal{I}(r_{j-1}^{\leftarrow}, t_{j-1}^{\leftarrow}))|$, and thus

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| \leq |\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| + k.$$

So far, we have bounded the cost of the algorithm for the first class of intervals with deadlines before t_j. In both cases, we have shown that the value of the optimal solution for intervals in $\mathcal{I}_1$ is at most the value obtained by the algorithm on $\mathcal{I}_1$ plus an additive term of k:

$$|\mathrm{OPT}(\mathcal{I}_1)| \leq |\textsc{Alg}(\mathcal{I}_1)| + k.$$

We now turn to the second class of intervals, $\mathcal{I}_2$. If the algorithm switches to the *independent phase* at time t_j, where $t_j = d_i$ for some $j \neq i = \arg\max\{d_i \mid I_i \in \mathcal{I}(r_j^{\leftarrow}, \cdot),\ \hat{o}_i = 1\}$, then there may be a set of intervals whose release time is before t_j and whose deadline is after t_j. These intervals are not considered in either phase of the algorithm. This set might be non-empty only in Case 1.

Since all of these intervals contain the timepoint t_j, the non-overlapping constraint implies that the optimal solution can include at most one of them. Moreover, since the algorithm does not select any interval in this class, we have:

$$|\mathrm{OPT}(\mathcal{I}_2)| \leq |\textsc{Alg}(\mathcal{I}_2)| + k = k.$$

Finally, we consider the third class $\mathcal{I}_3$, consisting of intervals with release time at or after t_j. By construction, the algorithm switches to the *independent phase* at time t_j and runs a θ-competitive algorithm on this suffix of the instance. Therefore, we have:

$$|\mathrm{OPT}(\mathcal{I}_3)| \leq \theta \cdot |\textsc{Alg}(\mathcal{I}_3)|.$$

Since intervals in different classes might overlap with each other, we can bound the optimal value as:

$$|\mathrm{OPT}| \leq |\mathrm{OPT}(\mathcal{I}_1)| + |\mathrm{OPT}(\mathcal{I}_2)| + |\mathrm{OPT}(\mathcal{I}_3)|.$$

Putting all parts together,

$$\begin{aligned}|\text{OPT}| &\leq |\text{OPT}(\mathcal{I}_1)| + |\text{OPT}(\mathcal{I}_2)| + |\text{OPT}(\mathcal{I}_3)| \\ &\leq (|\textsc{Alg}(\mathcal{I}_1)| + k) + k + \theta \cdot |\textsc{Alg}(\mathcal{I}_3)|,\end{aligned}$$

and since $\theta > 1$, we conclude

$$|\text{OPT}| \leq \theta \cdot |\textsc{Alg}| + 2k.$$

This completes the proof. □

We note that the framework allows any deterministic or randomized algorithm to be used in the *independent phase*. In practice, it makes sense to apply the best-known classical algorithm for the given setting, whether deterministic or randomized.

Two-Value Instances

We give a tighter bound for two-value instances. All intervals in these instances have one of two distinct lengths. We refer to these as *short* and *long* intervals. More specifically, we show that the Trust-and-Switch framework provides a better robustness guarantee in this special case, due to the restricted structure of the interval lengths.

Lemma 6.3. *Consider the Trust-and-Switch framework Alg, and assume that the independent phase uses a θ-competitive algorithm. Then, on two-value instances, the algorithm satisfies*

$$|OPT| \leq \max\{\theta, 2\} \cdot |\textsc{Alg}| + k.$$

Proof. The proof follows the same structure as the proof of Theorem 6.2. Assume that the framework decides to switch to the *independent phase* at time r_j, based on the evaluation of the prediction up to t_j, and we have to wait until then to be able to switch. If the framework never switches, we set $j = n$.

To analyze the performance of the algorithm, we partition the intervals into three disjoint classes based on their relation to the switching point t_j:

- $\mathcal{I}_1 := \mathcal{I}(\cdot, t_j^{\leftarrow})$, intervals with both release time and deadline strictly before t_j,
- $\mathcal{I}_2 := \mathcal{I}(t_j^{\leftarrow}, t_j^{\rightarrow})$, intervals with release time at or before t_j and deadline strictly after t_j,
- $\mathcal{I}_3 := \mathcal{I}(t_j^{\rightarrow}, \cdot)$, intervals with release time strictly after t_j.

As before, the analysis for intervals in $\mathcal{I}_3$ remains unchanged. Our goal is to analyze the intervals in $\mathcal{I}_1$ and $\mathcal{I}_2$ together.

We again distinguish the performance of the algorithm based on how the evaluation point t_j is determined. We consider the following three cases:

- **Case 1:** $t_j = d_i$ for some $i \neq j$, where $I_i \in \hat{\mathbf{O}}$ and $d_i = \max\{d_i \mid I_i \in \mathcal{I}(r_j^{\leftarrow}, \cdot),\ \hat{o}_i = 1\}$,
- **Case 2:** $t_j = d_j$, the deadline of the current interval I_j (since $\hat{o}_j = 1$),

- **Case 3:** $t_j = r_j$, the evaluation point is the release time of I_j.

Since $\mathcal{I}_2 = \emptyset$ in Cases 2 and 3, the analysis remains unchanged in those cases, and we lose at most an additive factor of k:

$$|\mathrm{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| \leq |\textsc{Alg}(\mathcal{I}_1 \cup \mathcal{I}_2)| + k.$$

However, we now provide a refined analysis for Case 1 by jointly analyzing the intervals in Classes $\mathcal{I}_1$ and $\mathcal{I}_2$ to establish the following bound:

$$|\mathrm{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| \leq 2 \cdot |\textsc{Alg}(\mathcal{I}_1 \cup \mathcal{I}_2)| + k.$$

In this case, we have $t_j = d_i$ for some $i \neq j$. We distinguish two cases based on the length l_i.

If $\mathcal{I}_i$ is a short interval, then since the algorithm did not decide to switch at r_i, we have:

$$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| \leq |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| = |\textsc{Trust}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))|.$$

Moreover, since $\mathcal{I}_i$ is short, we have:

$$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| = |\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))| \quad \text{and} \quad |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| = |\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))|.$$

Therefore, we still have:

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))| \leq |\textsc{Alg}(\mathcal{I}(r_j^{\leftarrow}, d_i^{\leftarrow}))|,$$

and the algorithm does not decide to switch.

If $l_i = k$, then since the algorithm did not decide to switch at r_i, we have:

$$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| \leq |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| = |\textsc{Trust}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))|.$$

This implies:

$$|\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| \leq |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, r_i^{\leftarrow}))| + k.$$

Now we consider the long interval I_i selected by Alg, together with all intervals in OPT whose release times lie in $(r_i, d_i]$, that is, intervals that Alg cannot select due to a conflict with I_i.

Among these conflicting intervals, there can be at most one long interval and at most $\lceil \Delta \rceil - 1$ short intervals. Therefore, the total length of these intervals is bounded above by:

$$k + (\lceil \Delta \rceil - 1) \cdot \frac{k}{\Delta},$$

and so their contribution relative to k is:

$$\frac{k + (\lceil \Delta \rceil - 1) \cdot \frac{k}{\Delta}}{k} = 1 + \frac{\lceil \Delta \rceil - 1}{\Delta} < 2.$$

Therefore, we can bound the optimal value on $\mathcal{I}_1 \cup \mathcal{I}_2$ as:

$$\begin{aligned} |\mathrm{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| &= |\mathrm{OPT}(\mathcal{I}(r_i^{\leftarrow}, d_i^{\leftarrow}))| + |\mathrm{OPT}(\mathcal{I}((r_i, d_i], \cdot))| \\ &\leq |\textsc{Alg}(\mathcal{I}(r_i^{\leftarrow}, r_i^{\leftarrow}))| + k + 2 \cdot |\textsc{Alg}(\mathcal{I}(r_i^{\rightarrow}, d_i^{\leftarrow}))| \\ &\leq 2 \cdot |\textsc{Alg}(\mathcal{I}_1 \cup \mathcal{I}_2)| + k. \end{aligned}$$

Considering all cases, we have shown that the value of the optimal solution for intervals in $\mathcal{I}_1 \cup \mathcal{I}_2$ is at most

$$|\text{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| \leq 2 \cdot |\textsc{Alg}(\mathcal{I}_1 \cup \mathcal{I}_2)| + k.$$

Since intervals in different classes might overlap with each other, we can bound the optimal value as:

$$|\text{OPT}| \leq |\text{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| + |\text{OPT}(\mathcal{I}_3)|.$$

Putting all parts together,

$$\begin{aligned} |\text{OPT}| &\leq |\text{OPT}(\mathcal{I}_1 \cup \mathcal{I}_2)| + |\text{OPT}(\mathcal{I}_3)| \\ &\leq 2 \cdot |\textsc{Alg}(\mathcal{I}_1 \cup \mathcal{I}_2)| + k + \theta \cdot |\textsc{Alg}(\mathcal{I}_3)|. \end{aligned}$$

Since $\theta > 1$, we conclude

$$|\text{OPT}| \leq \max\{\theta, 2\} \cdot |\textsc{Alg}| + k.$$

This completes the proof. □

6.3.1 Tightness

In this final subsection, we show that the guarantees of the Trust-and-Switch framework are tight for two-value instances. First, we prove that the Greedy algorithm achieves a robustness guarantee in this setting. Then, we show that even under the strongest prediction model $\hat{\mathbf{I}}$, no algorithm with consistency better than $1 + \frac{\lceil \Delta \rceil - 1}{\Delta}$ can achieve a stronger robustness guarantee. This establishes the tightness of the Trust-and-Switch framework for two-value instances. While our lower bound extends to general instances as well, there remains a gap between the upper and lower bounds in the general setting.

Greedy Algorithm

We first bound the competitive ratio of the Greedy algorithm on two-value instances. This will allow us to instantiate the robustness guarantee in our framework. Specifically, we show that Trust-and-Switch(Greedy) satisfies 1-consistency and guarantees

$$|\text{OPT}| \leq \max\{\Delta, 2\} \cdot |\textsc{Trust-and-Switch}(\textsc{Greedy})| + k.$$

We briefly recall the definition of the Greedy algorithm below.

Greedy. The Greedy algorithm accepts an arriving interval if and only if it does not overlap with any previously accepted interval.

Lemma 6.4. *Given a two-value instance $\mathcal{I}$, let $\textsc{Greedy}(\mathcal{I})$ be the solution obtained by the Greedy algorithm. Then,*

$$|\mathit{OPT}(\mathcal{I})| \leq \max\{\Delta, 2\} \cdot |\textsc{Greedy}(\mathcal{I})|.$$

Proof. We first note that by the definition of Greedy for each interval $\mathcal{I}_i \in \mathrm{OPT}(\mathcal{I})$ there exists exactly one interval $I_j \in \textsc{Greedy}(\mathcal{I})$ with $r_j \leq r_i$ and $I_i \cap I_j \neq \varnothing$ (note that it could be that $I_i = I_j$). We define $f : \mathrm{OPT}(\mathcal{I}) \to \textsc{Greedy}(\mathcal{I})$ such that for each $I_i \in \mathrm{OPT}(\mathcal{I})$, $f(I_i)$ be exactly the interval $I_j \in \textsc{Greedy}(\mathcal{I})$ as described above.

Note that f is many-to-one and furthermore for each interval $I_j \in \textsc{Greedy}(\mathcal{I})$, $f^{-1}(I_j)$ can consist of at most one interval if I_j is short (and thus $|f^{-1}(I_j)| \leq \Delta l_s$ in this case), and $\sum_{I_i \in f^{-1}(I_j)} |I_i| < 2 \cdot k$ if I_j is long – since at most one interval of $f^{-1}(I_j)$ can be long. This gives

$$\begin{aligned}
|\mathrm{OPT}(\mathcal{I})| &= \sum_{I_i \in \mathrm{OPT}(\mathcal{I})} |I_i| \\
&= \sum_{\substack{I_i \in \mathrm{OPT}(\mathcal{I}) \\ f(I_i) \text{ short}}} |I_i| + \sum_{\substack{I_i \in \mathrm{OPT}(\mathcal{I}) \\ f(I_i) \text{ long}}} |I_i| \\
&\leq \Delta \cdot \sum_{\substack{I_i \in \mathrm{OPT}(\mathcal{I}) \\ f(I_i) \text{ short}}} |f(I_i)| + 2 \cdot \sum_{\substack{I_i \in \mathrm{OPT}(\mathcal{I}) \\ f(I_i) \text{ long}}} |f(I_i)| \\
&\leq \max\{2, \Delta\} |\textsc{Greedy}(\mathcal{I})|.
\end{aligned}$$

□

Now, using Theorem 6.2, and Lemmas 6.3, and 6.4, we can conclude the following proposition.

Proposition 6.5. *Given a two-value instance and predictions $\hat{\mathbf{O}}$, Trust-and-Switch(Greedy) satisfies 1-consistency. Moreover, it guarantees*

$$|OPT| \leq \max\{\Delta, 2\} \cdot |\textsc{Trust-and-Switch}(\textsc{Greedy})| + k,$$

independent of the quality of the predictions.

Lower Bound

We now show that the consistency–robustness trade-off achieved by the Trust-and-Switch framework is tight for two-value instances. Even with access to full information through $\hat{\mathbf{I}}$, no algorithm with sufficiently strong consistency can guarantee a better robustness bound.

Lemma 6.6. *For any $(1+\alpha)$-consistent algorithm Alg$'$ for two-value instances, with $\alpha < \frac{\lceil \Delta \rceil - 1}{\Delta}$, there exists an instance $\mathcal{I}$ such that*

$$\Delta \cdot |\textsc{Alg}'(\mathcal{I})| + k \leq |OPT(\mathcal{I})|,$$

even when provided with full predictions $\hat{\mathbf{I}}$.

Proof. We construct an adversarial input sequence. Note that this instance can be constructed even if the total number of intervals is known in advance. The key idea is to present the algorithm with a long interval that is not predicted by $\hat{\mathbf{O}}$. If this interval is

accepted, consistency is violated by introducing many short intervals and an additional long interval that do not conflict with each other but do conflict with the accepted interval. Conversely, if the long interval is rejected, the algorithm immediately incurs a loss of k, assuming that the predictions are inaccurate.

We begin by describing the predicted instance $\hat{\mathbf{I}}$ that the algorithm receives a priori. We then define the actual instance $\mathcal{I}_{ADV}$, which is constructed adaptively based on the decisions made by the algorithm ALG′.

Predicted Instance $\hat{\mathbf{I}}$. $\hat{\mathbf{I}}$ consists of $\lceil\Delta\rceil^2+2$ intervals, indexed in non-decreasing order of release time. These intervals are defined recursively as follow:

- $I_1 = [0,\ k)$ — a long interval,
- $I_2 = \left[\varepsilon,\ \varepsilon + \frac{k}{\Delta}\right)$ — the first short interval,
- $I_j = \left[d_{j-1} + \varepsilon,\ d_{j-1} + \varepsilon + \frac{k}{\Delta}\right)$ for $j = 3, \ldots, \lceil\Delta\rceil$ — a chain of short intervals following I_2,
- $I_j = \left[k - \varepsilon \cdot \frac{k}{\Delta},\ k - \varepsilon\right)$ for $j = \lceil\Delta\rceil + 1, \ldots, \lceil\Delta\rceil^2 + 1$ — a block of overlapping short intervals packed at the end of the long interval I_1,
- $I_n = [k - \varepsilon,\ 2k - \varepsilon)$ — the final long interval that touches the end of I_1.

Here, $\varepsilon > 0$ is chosen such that

$$\varepsilon < 1 + \frac{1 - \lceil\Delta\rceil}{\Delta},$$

ensuring that all intervals are non-overlapping within their respective groups. These intervals are visualized in Figure 6.2.

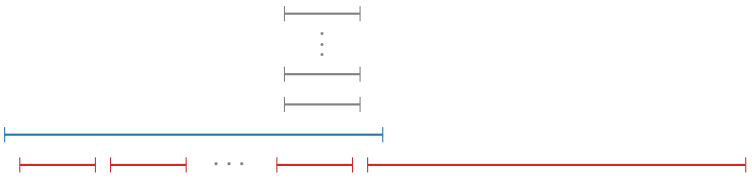


Figure 6.2: Illustration of prediction $\hat{\mathbf{I}}$. The red intervals represent the optimal solution for this set of intervals.

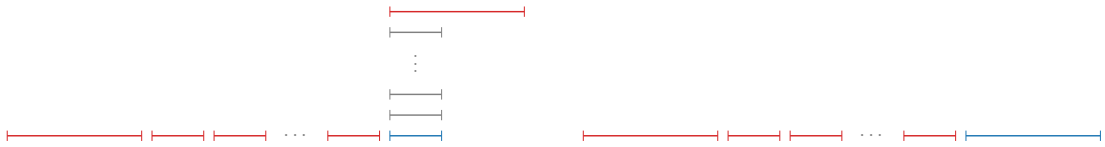
Note that the optimal solution for instance $\hat{\mathbf{I}}$ consists of

$$\bigcup_{j \in \{2, \ldots, \lceil\Delta\rceil\} \cup \{n\}} I_j,$$

and has a total value of

$$(\lceil\Delta\rceil - 1) \cdot \frac{k}{\Delta} + k.$$

We are now ready to construct the adversarial instance $\mathcal{I}_{ADV}$.



(a) The adversarial input when the Alg accepts a short interval.

(b) The adversarial input when the Alg never accepts a short interval.

Figure 6.3: The two cases of the adversarial input. In both figures, the red intervals represent the intervals in the optimal solution. In case 6.3(a), the blue interval appears only in the Alg, whereas in case 6.3(b), it is included in both the optimal solution and the Alg.

Adversarial Instance $\mathcal{I}_{ADV}$. We start by releasing the interval I_1 as described above. We distinguish two cases based on whether I_1 is selected by Alg$'$ or not (see also Figure 6.3):

- **Case 1: Alg$'$ accepts I_1**
  In this case, we set $\mathcal{I}_{ADV} = \hat{\mathbf{I}}$, and argue that Alg$'$ cannot satisfy the required consistency. Note that the prediction is perfectly accurate, and yet Alg$'$ obtains only a total value of k on $\mathcal{I}_{ADV}$. The value of the optimal solution is

  $$(\lceil\Delta\rceil - 1) \cdot \frac{k}{\Delta} + k,$$

  resulting in the following ratio between the values of OPT and Alg$'$:

  $$\frac{(\lceil\Delta\rceil - 1) \cdot \frac{k}{\Delta} + k}{k} = 1 + \frac{\lceil\Delta\rceil - 1}{\Delta}.$$

- **Case 2: Alg$'$ rejects I_1**
  In this case, the remaining part of the instance consists of $\lceil\Delta\rceil^2 + 1$ intervals that arrive sequentially. The first $\lceil\Delta\rceil^2$ intervals are short, and the final one is a long interval. The exact release times are defined adaptively based on the decisions made by Alg$'$. More specifically, for each interval I_i, if Alg$'$ does not select I_i, then I_{i+1} is released after the deadline of I_i. Otherwise, I_{i+1} and all subsequent intervals are released so that they overlap with I_i. Figure 6.4 depicts the two different possibilities for $\mathcal{I}_{ADV}$.

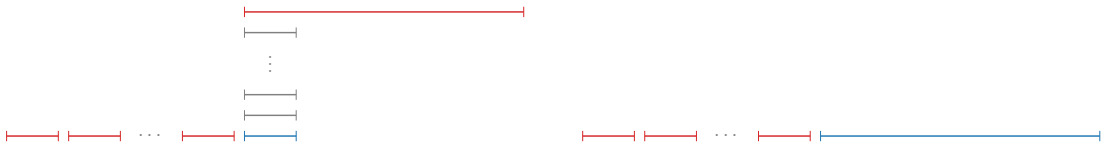
  There are two subcases to consider:

  - **Alg$'$ selects some short interval I_i.**
    By construction, this is the only interval that Alg$'$ can select, since all subsequent intervals overlap with I_i. On the other hand, the optimal algorithm can reject all short intervals and instead select the final long interval (and potentially some non-overlapping short ones). Thus, $|\text{Alg}'(\mathcal{I}_{ADV})| \leq \frac{k}{\Delta}$, while $|\text{OPT}(\mathcal{I}_{ADV})| \geq k$.
  - **Alg$'$ selects no short interval I_i.**
    In this case, Alg$'$ can only select the final long interval. However, the optimal

algorithm can select the long interval as well as all $\lceil\Delta\rceil^2$ short intervals, which do not overlap with each other. That is, $|\textsc{Alg}'(\mathcal{I}_{ADV})| \leq k$, while

$$|\mathrm{OPT}(\mathcal{I}_{ADV})| \geq \lceil\Delta\rceil^2 \cdot \frac{k}{\Delta}.$$

Overall, this gives a competitive ratio of at least Δ. Since the instance is constructed independently of the prediction, the competitive ratio between $\textsc{Alg}'(\mathcal{I}_{ADV})$ and $\mathrm{OPT}(\mathcal{I}_{ADV})$ is Δ. Additionally, $\textsc{Alg}'$ loses the first long interval, incurring an additive loss of k.



(a) The adversarial input when the Alg accepts a short interval.

(b) The adversarial input when the Alg waits for the long interval.

Figure 6.4: The two cases of the adversarial input. In both figures, the red intervals represent the intervals selected by the optimal solution. In case 6.4(a), the blue interval is only selected byAlg, whereas in case 6.4(b), it is selected by both the optimal solution and Alg. The gray intervals are not selected by neither Alg nor OPT.

□

6.4 SemiTrust-and-Switch Framework

In this section, we turn to a more general variant of the Trust-and-Switch framework that relaxes its reliance on the predictions in the *trust phase*. The goal is to achieve a better consistency–robustness trade-off. The key idea is to define a length threshold and follow the prediction only for intervals whose lengths fall below this threshold. Although this approach sacrifices 1-consistency unless the threshold is set very high, it allows for improved robustness guarantees when predictions are unreliable.

Intuitively, this method ensures that the algorithm does not miss promising long intervals in the case of inaccurate predictions, thereby improving its worst-case performance.

Framework SemiTrust-and-Switch. In this framework, the algorithm uses a length threshold τ to determine whether to follow the prediction. More formally, in the *trust phase*, the algorithm accepts an interval if either (i) the prediction suggests it should be accepted, or (ii) the interval's length exceeds the threshold τ. As in Trust-and-Switch, the algorithm evaluates the predictions over time and switches to a classical algorithm in the *independent phase* using the same strategy.

Note that the framework uses the Trust algorithm to evaluate the quality of the prediction. More specifically, it compares the value of the predicted solution with the offline optimum up to time t_j, and switches to the *independent phase* if

$$|\mathrm{OPT}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))| > |\textsc{Trust}(\mathcal{I}(r_j^{\leftarrow}, t_j^{\leftarrow}))|.$$

Since SemiTrust-and-Switch uses the same evaluation procedure and switching strategy as the Trust-and-Switch framework, we have the following observation.

Observation 6.7. *SemiTrust-and-Switch switches to the independent phase only if the predictions are inaccurate.*

We now analyze the performance of the SemiTrust-and-Switch framework as a function of the threshold parameter τ. Recall that in the *trust phase*, the algorithm accepts an interval if either the prediction suggests it should be accepted, or its length exceeds τ. If τ is set below the length of the shortest interval, the algorithm accepts all intervals greedily, behaving identically to Greedy. On the other hand, if $\tau > k$, where k is the maximum interval length, then no interval is accepted based on length alone, and the algorithm behaves identically to Trust-and-Switch(Greedy), relying fully on predictions in the *trust phase*.

The interesting regime occurs when the threshold τ lies strictly between the shortest and longest interval lengths. Specifically, we assume throughout this section that

$$\frac{k}{\Delta} < \tau < k,$$

so that the algorithm accepts long intervals (with length $> \tau$) greedily, while relying on predictions only for short intervals (with length $\leq \tau$). This setting improves robustness by ensuring that long, high-value intervals are not missed due to inaccurate predictions, while still maintaining reasonable consistency guarantees.

Lemma 6.8. *The SemiTrust-and-Switch framework satisfies $(1 + \frac{k}{\tau})$-consistency.*

Proof. Assume the predictions $\hat{\mathbf{O}}$ are accurate. We aim to show that the total length of the optimal solution is at most $(1 + \frac{k}{\tau})$ times the total length of the solution produced by SemiTrust-and-Switch. By Observation 6.7, the SemiTrust-and-Switch algorithm never switches to the *independent phase* and remains in the *trust phase* throughout the instance.

Let us compare the output of SemiTrust-and-Switch with the optimal offline solution. Note that any interval with length less than τ accepted by SemiTrust-and-Switch must have been predicted, and by the accuracy of the predictions, all such intervals also appear in the optimal solution.

We now remove these common short intervals (those with length $< \tau$) from both solutions. What remains are only long intervals (with length $\geq \tau$) in the output of SemiTrust-and-Switch. For each such long interval I_i accepted by the algorithm. Consider all intervals in the optimal solution that conflict with I_i, and group them into a class corresponding to I_i.

We now analyze one such class. Since I was accepted by the algorithm, all intervals in the optimal solution that conflict with I must start after r_i and end before d_i.

Therefore, the total length of intervals in the optimal solution within any class is at most:

$$k + l_i,$$

while the contribution of SemiTrust-and-Switch in this class is l_i. Thus, the competitive ratio in this class is at most

$$\frac{k + l_i}{l_i} = 1 + \frac{k}{l_i} \leq 1 + \frac{k}{\tau},$$

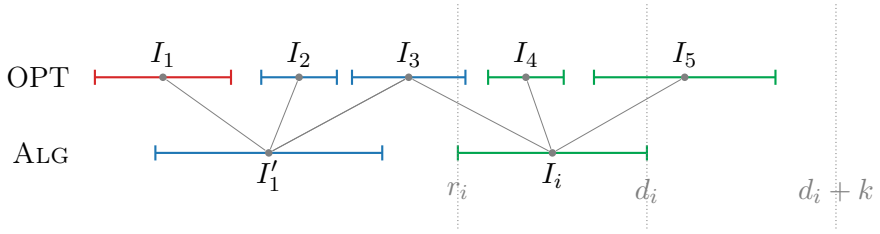


Figure 6.5: The last connected component of the conflict graph $G(\text{OPT}, \text{ALG})$ before switching. The styles of intervals represent their mapped counterparts. Note that I_1 is the only interval taken by OPT that is not mapped to any interval. All intervals I_4 and I_5, which are mapped to I_i, are contained within the range $[r_i, d_i + k]$.

since $l_i \geq \tau$. □

Theorem 6.9. *The SEMITRUST-AND-SWITCH framework ALG satisfies* $(1+\frac{k}{\tau})$*-consistency. Moreover, if the independent phase uses a* θ*-competitive algorithm, then the framework satisfies*

$$|OPT| \leq \max(\theta, \Delta + 1) \cdot |ALG| + \tau.$$

Proof. Let A and B be two feasible sets of intervals. Define conflict graph $G(A, B)$ as an undirected graph with $|A| + |B|$ nodes, where each node corresponds to an interval from either A or B. If an interval appears in both sets, it is represented by two distinct nodes, one for each set. An edge is placed between two nodes if their corresponding intervals intersect.

Now consider the graph $G(\text{OPT}, \text{ALG})$. It may consist of several connected components. For all but the last component (i.e., before the point where the framework switches), we must have $|\text{ALG}| \geq |\text{OPT}|$; otherwise, the framework would have detected suboptimality and switched earlier.

We now analyze the final component. We aim to show that within this component, we have:

$$|\text{OPT}| \leq (\Delta + 1) \cdot |\text{ALG}| + \tau.$$

For each interval I_i in ALG, we assign to I_i all intervals in OPT whose release times lie in the interval $(r_i, d_i]$. Since this forms a connected component, there is at most one interval in OPT that is not assigned to any interval in ALG. Moreover, since the algorithm did not accept this interval, its length must be less than τ. Figure 6.5 presents a visual summary.

Now consider an arbitrary interval $I_i \in \text{ALG}$ within this component. The total length of the intervals in OPT assigned to I_i cannot exceed $l_i + k$. This is because all such intervals must fit within the window $(r_i, d_i]$ and cannot overlap with I_i, which has length l_i. Therefore,

$$l_i + k \leq (\Delta + 1) \cdot l_i,$$

where we use the fact that $l_i \geq \frac{k}{\Delta}$.

Summing over all intervals in ALG and adding the at most τ contribution from the one unassigned interval in OPT, we obtain:

$$|\text{OPT}| \leq (\Delta + 1) \cdot |\text{ALG}| + \tau.$$

For the second part of the instance (after the switch), the algorithm switches and ran a θ-competitive algorithm in the *independent phase.* .

Combining both parts, we get:

$$|\text{OPT}| \leq \max(\theta, \Delta + 1) \cdot |\text{ALG}| + \tau.$$

This completes the proof.

□

6.4.1 Lower Bound

In this subsection, we show a lower bound on the robustness of algorithms with consistency better than Δ. Even with access to full information through $\hat{\mathbf{I}}$, no algorithm with consistency strictly better than Δ can guarantee a robustness bound better than $|\text{OPT}| \leq \Delta \cdot |\text{ALG}| + \frac{k}{\Delta}$.

Lemma 6.10. *Given two-value instances, no algorithm with robustness better than* $\Delta \cdot |\text{ALG}| + \frac{k}{\Delta}$ *can achieve consistency better than* Δ*, even with full predictions* $\hat{\mathbf{I}}$*.*

Proof. We want to show that for any deterministic algorithm ALG that guarantees robustness of

$$|\text{OPT}(\mathcal{I})| \leq \Delta \cdot |\text{ALG}(\mathcal{I})| + \frac{k}{\Delta} - \varepsilon$$

for all instances $\mathcal{I}$, there exists an adversarial instance $\mathcal{I}_{\text{Adv}}$ on which ALG has a competitive ratio of at least Δ.

Predicted Instance $\hat{\mathbf{I}}$. Consider a predicted instance $\hat{\mathbf{I}}$ with $\Delta^2 + 2$ intervals. The first interval is short, followed by a long interval that overlaps it. Then, Δ^2 short intervals arrive, all overlapping with each other and ending before the deadline of the initial long interval. Finally, a second long interval arrives after all others. This instance is illustrated in Figure 6.2.

Now, consider the scenario where $\hat{\mathbf{I}}$ is given to the algorithm ALG. We construct an adversarial instance $\mathcal{I}_{ADV}$ based on the algorithm's decision.

Adversarial Instance. The first interval in $\mathcal{I}_{ADV}$ is a short interval, identical to the first interval in $\hat{\mathbf{I}}$. We distinguish two cases based on the algorithm's decision for this interval.

- **Case 1: The algorithm ALG accepts the first short interval.**
  In this case, we set $\mathcal{I}_{ADV} = \hat{\mathbf{I}}$. The optimal solution has total length k, i.e., $|\text{OPT}(\mathcal{I}_{ADV})| = k$, while the algorithm selects only the short interval, so $|\text{ALG}(\mathcal{I}_{ADV})| = \frac{k}{\Delta}$. Since the prediction is accurate, this case witnesses a consistency ratio of Δ.
- **Case 2: The algorithm ALG rejects the first short interval.**
  In this case, the remaining part of the instance consists of $\lceil\Delta\rceil^2 + 1$ intervals that arrive sequentially. The first $\lceil\Delta\rceil^2$ intervals are short, and the final one is a long interval. The exact release times are defined adaptively based on the decisions made by ALG. More specifically, for each interval I_i, if ALG does not select I_i, then I_{i+1}

is released after the deadline of I_i. Otherwise, I_{i+1} and all subsequent intervals are released so that they overlap with I_i.

Figure 6.4 depicts the two possible evolutions of $\mathcal{I}_{ADV}$ under this construction. The algorithm ALG cannot achieve a competitive ratio better than Δ on the remaining part of the instance, and additionally incurs a loss of $\frac{k}{\Delta}$ due to rejecting the first short interval.

The behavior of both OPT and ALG is illustrated in Figure 6.6.

(a) The adversarial input when the ALG accepts a short interval.

(b) The adversarial input when the ALG never accepts a short interval.

Figure 6.6: The two cases of the adversarial input. In both figures, the red intervals represent the intervals in the optimal solution. In case 6.3(a), the blue interval appears only in the ALG, whereas in case 6.3(b), it is included in both the optimal solution and the ALG.

□

6.5 Randomized Setting

In this section, we study randomized algorithms for online interval scheduling with predictions. Our goal is to understand how randomness can improve performance guarantees. We first show that if we plug a randomized algorithm into the *independent phase* of the TRUST-AND-SWITCH framework, we obtain meaningful consistency–robustness trade-offs. Our main result is the SMOOTHMERGE algorithm, whose performance degrades gracefully with the quality of the prediction. The key idea is to combine two algorithms, one that uses predictions and another that is prediction-agnostic, in a smooth and controlled manner.

6.5.1 The TRUST-AND-SWITCH Framework

The TRUST-AND-SWITCH framework naturally extends to this setting by allowing a randomized algorithm to be used in the *independent phase*.

VIRTUAL ALGORITHM, introduced by Lipton and Tomkins, 1994, achieves a competitive ratio of 2 on two-value instances. Briefly, this algorithm accepts any available interval of length k whenever possible. For intervals of length 1, the algorithm flips a fair coin to decide whether to accept the interval immediately or to *virtually* accept it—that is, to defer accepting the short interval until the end of its duration.

Therefore, using Lemmas 6.1 and 6.3, we obtain the following proposition.

Proposition 6.11. *Given a two-value instance and predictions* $\hat{\mathbf{O}}$*,* Trust-and-Switch(Virtual Algorithm) *satisfies* 1*-consistency. Moreover, it guarantees*

$$|OPT| \leq 2 \cdot |\textsc{Trust-and-Switch}(\textsc{Virtual Algorithm})| + k,$$

independent of the quality of the predictions.

For general instances, Lipton and Tomkins, 1994 introduced a randomized Marriage Algorithm that achieves a competitive ratio of $O((\log \Delta)^{1+\varepsilon})$, where $\varepsilon > 0$ is an arbitrarily small constant. Therefore, using Theorem 6.2, we obtain the following proposition.

Proposition 6.12. *Given predictions* $\hat{\mathbf{O}}$*,* Trust-and-Switch(Marriage Algorithm) *satisfies* 1*-consistency. Moreover, it guarantees*

$$|OPT| \leq \max\left\{2,\ O\left((\log \Delta)^{1+\varepsilon}\right)\right\} \cdot |\textsc{Alg}| + 2k,$$

where Alg *denotes* Trust-and-Switch(Marriage Algorithm)*, independent of the quality of the predictions.*

6.5.2 A Smooth Algorithm

In this section, we introduce the SmoothMerge algorithm, which achieves smoothness while maintaining robustness. The key insight is to combine two complementary algorithms: one that follows the prediction and performs well when the predictions are good (i.e., low error), and another classic algorithm that maintains reasonable performance regardless of prediction quality. Specifically, we design a randomized algorithm by merging the Trust and Greedy algorithms.

SmoothMerge. Upon the arrival of each interval, the algorithm proceeds as follows: if the interval conflicts with any previously accepted interval, it is immediately rejected. Otherwise, if both simulated strategies agree on accepting (or rejecting) the interval, the algorithm follows that unanimous decision. When the strategies disagree, the algorithm accepts the interval with probabilities p_t and p_g, corresponding to the Trust and Greedy strategies, respectively. The full procedure is given in Algorithm 6.1.

To analyze the performance of our algorithm based on the quality of the predictions, we need to formally define prediction error.

Definition 6.13. *For a given instance* $\mathcal{I}$ *and prediction* $\hat{\mathbf{O}}$*, the* prediction error η *is defined as:*

$$\eta(\mathcal{I}, \hat{\mathbf{O}}) = \frac{|OPT(\mathcal{I})| - |\textsc{Trust}(\mathcal{I}, \hat{\mathbf{O}})|}{|OPT(\mathcal{I})|}.$$

where $|OPT(\mathcal{I})|$ *is the value of the optimal solution and* $|\textsc{Trust}(\mathcal{I}, \hat{\mathbf{O}})|$ *is the value obtained by the* Trust *algorithm following prediction* $\hat{\mathbf{O}}$*.*

The prediction error $\eta \in [0, 1]$ quantifies the reliability of predictions: $\eta = 0$ indicates perfect predictions (where Trust achieves optimal performance), while larger values indicate less reliable predictions. When $\eta = 1$, the prediction-based algorithm performs arbitrarily poorly compared to the optimum.

Input : Interval set $\mathcal{I}$ arriving in online order, prediction $\hat{\mathbf{O}}$, probabilities p_t, p_g
Output : Selected interval set S

Initialize $S \leftarrow \emptyset$
for *each interval I arriving in online order* **do**
  Let A_{TRUST} be true if TRUST accepts I, and false otherwise Let A_{GREEDY} be true if GREEDY accepts I, and false otherwise
  if *I conflicts with any interval in S* **then**
    reject I
  else
    if *A_{TRUST} and A_{GREEDY}* **then**
      $S \leftarrow S \cup \{I\}$
    else
      if *A_{TRUST} and not A_{GREEDY}* **then**
        With probability p_t, set $S \leftarrow S \cup \{I\}$
      else
        if *not A_{TRUST} and A_{GREEDY}* **then**
          With probability p_g, set $S \leftarrow S \cup \{I\}$
        else
          reject I
        end
      end
    end
  end
end
Return S

Algorithm 6.1: SMOOTHMERGE

Theorem 6.14. *SMOOTHMERGE achieves both smoothness and robustness guarantees. Specifically,*

$$\mathbb{E}[|ALG|] \geq \max\Big\{|OPT| \cdot (1-\eta) \cdot p_t \cdot (1-p_g), \frac{p_g - p_t p_g}{1 - p_t p_g} \cdot |GREEDY|\Big\},$$

where p_t and p_g are the chosen probabilities for the TRUST and GREEDY algorithms, respectively.

Proof. We start by analyzing the acceptance probabilities for intervals in both GREEDY($\mathcal{I}$) and TRUST($\mathcal{I}$). For any interval $I_i \in \mathcal{I}$, we compute the probability that Algorithm 6.1 accepts I_i. The key insight is that intervals belonging to both GREEDY($\mathcal{I}$) and TRUST($\mathcal{I}$) are accepted with probability 1, as they cannot conflict with previously selected intervals.

Let conflict graph $G(\text{GREEDY}(\mathcal{I}), \text{TRUST}(\mathcal{I}))$ be an undirected graph with $|\text{GREEDY}(\mathcal{I})|+|\text{TRUST}(\mathcal{I})|$ nodes, where each node corresponds to an interval from either GREEDY($\mathcal{I}$) or TRUST($\mathcal{I}$). If an interval appears in both sets, it is represented by two distinct nodes, one for each set. An edge is placed between two nodes if their corresponding intervals intersect.

We analyze intervals of each connected components of $G(\textsc{Greedy}(\mathcal{I}), \textsc{Trust}(\mathcal{I}))$ separately, as they are independent of each other. Consider a connected component with n intervals from $\textsc{Greedy}(\mathcal{I})$ (denoted $I_{G_0}, I_{G_1}, \ldots, I_{G_{n-1}}$) and m intervals from $\textsc{Trust}(\mathcal{I})$ (denoted $I_{T_0}, I_{T_1}, \ldots, I_{T_{m-1}}$), ordered by release time. Note that I_{G_0} has an earlier release time than I_{T_0} due to the Greedy algorithm's property of always selecting the earliest available interval.

For the earliest interval I_{G_0} in a component, we can easily find the probability of accepting this interval by

$$\mathbb{P}[I_{G_0} \in \textsc{Alg}(\mathcal{I})] = p_g.$$

For any subsequent interval I_{G_i} that conflicts with an earlier interval I_{T_j}, we can compute the probability recursively

$$\mathbb{P}[I_{G_i} \in \textsc{Alg}(\mathcal{I})] = \big(1 - \mathbb{P}[I_{T_j} \in \textsc{Alg}(\mathcal{I})]\big) \cdot p_g. \tag{6.1}$$

Similarly, for any interval $I_{T_i} \in \textsc{Trust}(\mathcal{I})$ that conflicts with an earlier interval I_{G_j},

$$\mathbb{P}[I_{T_i} \in \textsc{Alg}(\mathcal{I})] = \big(1 - \mathbb{P}[I_{G_j} \in \textsc{Alg}(\mathcal{I})]\big) \cdot p_t. \tag{6.2}$$

Solving the recurrence relations (6.1) and (6.2), we obtain:

$$\mathbb{P}[I_{T_i} \in \textsc{Alg}(\mathcal{I})] = (p_t - p_t p_g) \left(\frac{1 - (p_t p_g)^{x_i}}{1 - p_t p_g} \right),$$

where x_i is the depth in the recursion sequence, which is at most $n + m$. In another way, x_i is the number of times we should substitute the right-hand side of equation recursively, until we reach to the base case of the recursion with is $\mathbb{P}[I_{G_0} \in \textsc{Alg}(\mathcal{I})]$.

Observation 6.15. *The probability $\mathbb{P}[I_{T_i} \in \textsc{Alg}(\mathcal{I})]$ is non-decreasing with respect to the recursion depth.*

With the Observation 6.15, we have $\min_i \mathbb{P}[I_{T_i} \in \textsc{Alg}(\mathcal{I})] = \mathbb{P}[I_{T_0} \in \textsc{Alg}(\mathcal{I})]$. Therefore,

$$\mathbb{P}[I_{T_i} \in \textsc{Alg}(\mathcal{I})] \geq \mathbb{P}[I_{T_0} \in \textsc{Alg}(\mathcal{I})] = p_t \cdot (1 - p_g).$$

For intervals in $\textsc{Greedy}(\mathcal{I})$, we have $\min_i \mathbb{P}[I_{G_i} \in \textsc{Alg}(\mathcal{I})] = \mathbb{P}[I_{G_{m-1}} \in \textsc{Alg}(\mathcal{I})]$, based on equation (6.1) and Observation 6.15. Hence,

$$\begin{aligned}
\mathbb{P}[I_{G_i} \in \textsc{Alg}(\mathcal{I})] &\geq \mathbb{P}[I_{G_{n-1}} \in \textsc{Alg}(\mathcal{I})] \\
&= (1 - \mathbb{P}[I_{T_j} \in \textsc{Alg}(\mathcal{I})]) \cdot p_g \\
&\geq \left(1 - \frac{p_t - p_t p_g}{1 - p_t p_g}\right) \cdot p_g \\
&= \frac{p_g - p_t p_g}{1 - p_t p_g}.
\end{aligned}$$

The expected performance is bounded by contributions from $\textsc{Greedy}(\mathcal{I})$, which complete the proof of robustness property.

$$\begin{aligned}
\mathbb{E}[|\textsc{Alg}(\mathcal{I})|] &= \sum_{I_i \in \mathcal{I}} \ell_i \cdot \mathbb{P}[I_i \in \textsc{Alg}(\mathcal{I})] \\
&\geq \sum_{I_i \in \textsc{Greedy}(\mathcal{I})} \ell_i \cdot \mathbb{P}[I_i \in \textsc{Alg}(\mathcal{I})] \\
&\geq \sum_{I_i \in \textsc{Greedy}(\mathcal{I})} \ell_i \cdot \frac{p_g - p_t p_g}{1 - p_t p_g} \\
&= |\textsc{Greedy}(\mathcal{I})| \cdot \frac{p_g - p_t p_g}{1 - p_t p_g}. \qquad (6.3)
\end{aligned}$$

Similarly, considering contributions from $\textsc{Trust}(\mathcal{I})$:

$$\begin{aligned}
\mathbb{E}[|\textsc{Alg}(\mathcal{I})|] &= \sum_{I_i \in \mathcal{I}} \ell_i \cdot \mathbb{P}[I_i \in \textsc{Alg}(\mathcal{I})] \\
&\geq \sum_{I_i \in \textsc{Trust}(\mathcal{I})} \ell_i \cdot \mathbb{P}[I_i \in \textsc{Alg}(\mathcal{I})] \\
&\geq \sum_{I_i \in \textsc{Trust}(\mathcal{I})} \ell_i \cdot p_t \cdot (1 - p_g) \\
&= |\textsc{Trust}(\mathcal{I})| \cdot p_t \cdot (1 - p_g). \qquad (6.4)
\end{aligned}$$

From Definition 6.13, we have $|\textsc{Trust}(\mathcal{I})| = |\mathrm{OPT}(\mathcal{I})| \cdot (1 - \eta)$.

$$\mathbb{E}[|\textsc{Alg}(\mathcal{I})|] \geq |\mathrm{OPT}(\mathcal{I})| \cdot (1 - \eta) \cdot p_t \cdot (1 - p_g).$$

Therefore, combining equations (6.3) and (6.4), we have

$$\begin{aligned}
\mathbb{E}[|\textsc{Alg}(\mathcal{I})|] \geq \max \Big\{ &|\mathrm{OPT}(\mathcal{I})| \cdot (1 - \eta) \cdot p_t \cdot (1 - p_g), \\
&\frac{p_g - p_t p_g}{1 - p_t p_g} \cdot |\textsc{Greedy}(\mathcal{I})| \Big\},
\end{aligned}$$

This completes the proof. □

Figure 6.7 illustrates the trade-off between smoothness and robustness across different values of p_t and p_g. Specifically, it visualizes the maximum robustness achievable for different levels of smoothness.

When $p_t = 1$ and $p_g = 0$, the algorithm purely follows predictions, achieving perfect smoothness but no robustness. Conversely, when $p_t = 0$ and $p_g = 1$, the algorithm reduces to the standard Greedy approach, providing full robustness but no smoothness benefit. Other balanced choices of p_t and p_g offer both properties simultaneously, though with reduced coefficients. Table 6.1 illustrates the trade-offs between smoothness and robustness for different parameter choices.

This approach can be generalized to combine any pair of algorithms beyond Trust and Greedy. The key insight is to merge a prediction-dependent algorithm with a robust classical baseline. While it is natural to use a state-of-the-art classical algorithm in this role, our choice of the Greedy algorithm as the robust component is motivated by its simplicity and the tractability of its competitive analysis.

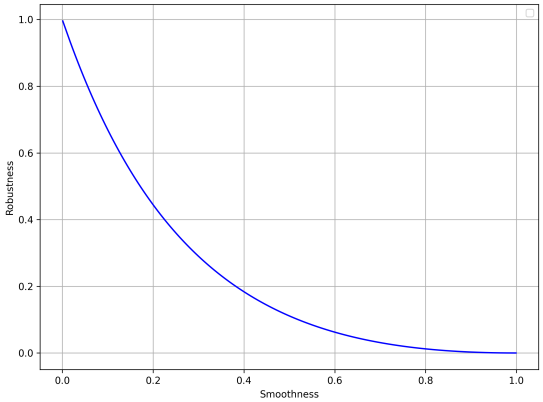


Figure 6.7: Trade-off curve between the best smoothness and robustness coefficients for SmoothMerge as p_t and p_g vary. In the smoothness expression, the factor $|\text{OPT}(\mathcal{I})| \cdot (1-\eta)$ is omitted from the plot; in the robustness expression, the factor $|\textsc{Greedy}(\mathcal{I})|$ is omitted.

p_t	p_g	Smoothness	Robustness
1	0	$(1-\eta) \cdot \|\text{OPT}(\mathcal{I})\|$	0
0	1	0	$\|\textsc{Greedy}(\mathcal{I})\|$
0.50	0.50	$0.25 \cdot (1-\eta) \cdot \|\text{OPT}(\mathcal{I})\|$	$0.33 \cdot \|\textsc{Greedy}(\mathcal{I})\|$
0.75	0.33	$0.50 \cdot (1-\eta) \cdot \|\text{OPT}(\mathcal{I})\|$	$0.11 \cdot \|\textsc{Greedy}(\mathcal{I})\|$
0.50	0.75	$0.12 \cdot (1-\eta) \cdot \|\text{OPT}(\mathcal{I})\|$	$0.60 \cdot \|\textsc{Greedy}(\mathcal{I})\|$

Table 6.1: Smoothness and robustness coefficients for different values of p_t and p_g.

Experimental Analysis

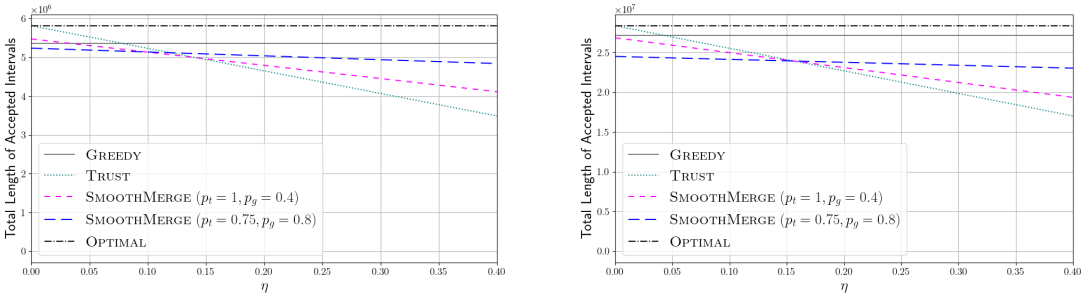
We present an experimental evaluation of SmoothMerge in comparison with Greedy Trust, and OPT. The algorithm smoothly interpolates between the behaviors of Trust and Greedy, adapting to prediction reliability through the choice of p_t and p_g.

Experimental Setup. Our evaluation uses real-world scheduling data for parallel machines (Chapin, Cirne, Feitelson, J. P. Jones, Leutenegger, Schwiegelshohn, Smith, and Talby, 1999). Table 6.2 outlines the interval scheduling inputs we generated from the benchmarks of Chapin, Cirne, Feitelson, J. P. Jones, Leutenegger, Schwiegelshohn, Smith, and Talby, 1999. For each benchmark with n intervals, we consider 1000 equally spaced values of $d \in [0, n]$. For each value of d, we randomly selected $\frac{n}{d}$ intervals and moved them all to the end of the timeline, making them conflict with each other. Then, we computed the offline optimal solution for this modified set of intervals and used this solution as the binary prediction $\hat{\mathbf{O}}$. To evaluate the SmoothMerge algorithm, we calculated the average performance over 50 runs on each prediction, using our personal laptop equipped with an Apple M3 GPU and 16GB of memory.

Figure 6.8 shows the result of SmoothMerge algorithm for different values of p_t and p_g. The results are aligned with our theoritical findings, Trust becomes worse than SmoothMergeas the error value increases, while SmoothMergedegrades gently as

Name	Input Size	k	Δ
NASA-iPSC-1993-3.1	18,065	62,643	62,643
CTC-SP2-1996-3.1	77,205	71,998	71,998

Table 6.2: Details on the benchmarks used in our experiment.



(a) NASA-iPSC-1993-3.1 dataset **(b)** CTC-SP2-1996-3.1 dataset

Figure 6.8: Performance analysis of SmoothMerge algorithm for two choices of parameters p_t and p_g, compared to OPT, Trust, and Greedy.

a function of the Greedy. We note that Greedy performs better when there is less overlap between the input intervals. In an extreme case, when no two intervals overlap, Greedy is trivially optimal.

6.6 Conclusion

We presented a systematic study of online interval scheduling with predictions in the irrevocable setting, focusing on maximizing the total length of accepted intervals. Our main contributions include the SemiTrust-and-Switch framework, which captures trade-offs between consistency and robustness. We showed that these frameworks are optimal for two-value instances in certain settings and can incorporate both deterministic and randomized algorithms. In addition, we introduced the SmoothMerge algorithm, which achieves smoothness by blending predictive and non-predictive strategies in a randomized manner.

Beyond the specific results presented here, our frameworks offer a modular approach that may be adaptable to other variants of online interval scheduling, such as revocable models, weighted objectives, or settings with resource constraints. While maximization problems present unique challenges for switching strategies, our results suggest that such approaches may be broadly useful in designing learning-augmented algorithms across a wider range of settings.

CHAPTER 7

Interval Scheduling under Approximate Envy-Freeness

Chapter overview. This chapter studies online interval scheduling from the perspective of fair allocation, modeling machines as agents and intervals as indivisible items. It investigates the trade-off between efficiency and fairness when decisions must be made sequentially and fairness constraints must be maintained at all times. This chapter is based on joint work with Sander Borst and Rohit Vaish, which is currently under review.

7.1 Introduction

Interval scheduling is a fundamental optimization problem in which one must assign time intervals, each specified by a start time and an end time, to identical machines so that intervals assigned to the same machine do not overlap, while maximizing the total value of the accepted intervals (Kolen, Lenstra, Papadimitriou, and Spieksma, 2007; Kovalyov, Ng, and T. C. E. Cheng, 2007). It arises naturally in two-sided service platforms. Consider a ride-hailing service such as Uber or Lyft, a food-delivery platform such as Deliveroo, or a freelance marketplace such as Upwork: requests for service arrive over time, each with a fixed window during which it must be handled, and the platform assigns each accepted request to one of several available workers. Since a worker cannot handle two overlapping requests, the set of requests served by any single worker must be conflict-free, which is precisely the feasibility constraint of interval scheduling, with workers playing the role of machines. On such platforms, the classical objective of maximizing the number of served requests is not the only concern. The workers who fulfill these requests are stakeholders in their own right: each accepted request is a unit of paid work, and workers expect the platform to distribute this work *fairly*. An efficiency-maximizing assignment may pile many requests onto a few workers while leaving others idle, even though all workers are equally capable. When machines represent workers, or more generally clients or organizational units sharing a common infrastructure, such concentration is naturally perceived as unfair. This tension is inherent to the two-sided structure: efficiency is measured over the requests we accept, whereas fairness is measured over the workers to whom we assign them. It motivates the central question:

> *Can one combine the efficiency of interval scheduling with the fairness guarantees studied in fair division, while respecting the underlying feasibility constraints?*

The offline and online versions of interval scheduling have both been studied extensively. In the offline problem, all intervals are known in advance; the problem dates back to the 1950s, stemming from the works of Dantzig, Ford, and Fulkerson (Dantzig and

Fulkerson, 1954; Ford and Fulkerson, 1962), and has numerous applications including crew scheduling (Beasley and Cao, 1998), telecommunications (Bar-Noy, Canetti, Kutten, Mansour, and Schieber, 1999), satellite photography (Gabrel, 1995), VLSI layout design (U. I. Gupta, Lee, and Leung, 1979), and computational biology (Z.-Z. Chen, Lin, Rizzi, Wen, D. Xu, Y. Xu, and T. Jiang, 2005). In the online problem, intervals arrive sequentially in nondecreasing order of start times, and the algorithm must decide without knowledge of future arrivals which intervals to keep (Lipton and Tomkins, 1994). We study a *revocable* online model in which accepted intervals can be revoked at any time before their completion. This model captures settings with soft reservations or provisional allocations, where a task can be preempted or withdrawn before service is completed, but once the opportunity is declined it is lost. The performance of an online algorithm is measured by the *competitive ratio* (Sleator and Tarjan, 1985), comparing it to an offline algorithm with full knowledge of the input.

More broadly, interval scheduling can be understood as the allocation of time-indexed opportunities over shared resources, which naturally raises concerns about fairness alongside the classical goal of efficiency. Our work focuses on the *unweighted* setting on identical machines, which is the simplest framework for integrating concepts from fair division and offers a relevant context for analyzing the online cost of fairness. In more complex weighted scenarios, the online interval scheduling problem does not permit a constant competitive ratio, making the unweighted setting ideal for isolating and quantifying the efficiency loss due to fairness.

From this perspective, fair division guarantees can be imposed on temporal allocation problems as explicit constraints, with interval scheduling providing a clean first instantiation.

To reason about fairness, we draw on the area of *fair division*, which provides a principled framework for allocating resources among agents (Brams and Taylor, 1996; Brandt, Conitzer, Endriss, Lang, and Procaccia, 2016; Moulin, 2004). A central notion in this literature is *envy-freeness* (Foley, 1967; Varian, 1974), which requires that no agent should prefer another agent's allocation over its own. For indivisible goods, exact envy-freeness may be impossible, so one turns to its relaxations. A well-studied notion of approximate fairness is *envy-freeness up to one item* (EF1) (Budish, 2011), which allows pairwise envy to be eliminated by removing a single item from the envied bundle. It is known that an EF1 allocation always exists for discrete goods (Lipton, Markakis, Mossel, and Saberi, 2004). In our setting, we consider intervals as the goods and machines as the agents, and EF1 requires that any envy between two machines can be resolved by hypothetically removing one interval from the schedule with a higher value.

Our goal is not to replace the scheduling objective with a fairness objective; instead, we aim to combine both perspectives. Traditional approaches to fairness in optimization usually incorporate fairness directly into the objective, often using egalitarian or max-min criteria. In contrast, we impose a fair-division guarantee as a *hard constraint* on the scheduling problem and compare the resulting efficiency to the unconstrained optimum. This approach provides a clean way to quantify the cost of fairness while preserving the combinatorial structure of interval scheduling.

In the offline setting, we measure this loss through the *price of fairness* (Bertsimas, Farias, and Trichakis, 2011; Caragiannis, Kaklamanis, Kanellopoulos, and Kyropoulou, 2012), namely the worst-case ratio between the value achieved by a fair schedule and that

of an optimal schedule without fairness constraints. In the online setting, the benchmark must additionally account for the uncertainty of future arrivals. For this reason, we formulate the *fair competitive ratio*, defined as the worst-case ratio between the value achieved by a fair online algorithm and that of an optimal offline algorithm without fairness constraints. This benchmark captures both sources of degradation: the loss imposed by fairness and the additional loss due to online uncertainty.

With this framework in place, we study interval scheduling under EF1 in both offline and online settings, establish tight worst-case guarantees, and determine how much efficiency must be sacrificed to obtain fair allocations over time.

7.1.1 Our Contributions and Techniques

At a high level, our work can be viewed from two complementary perspectives. First, we can consider interval scheduling under EF1 as a *fair allocation problem with scheduling constraints*. In this context, machines act as agents, intervals represent items, and the feasibility of a schedule is determined by the requirement that only non-conflicting items are assigned to each agent. Alternatively, we can see it as a *scheduling problem with a fairness constraint*, where the goal is to maximize total accepted weight while adhering to EF1. The first perspective links our research to the fair division literature, while the second focuses on approximation guarantees and potential efficiency loss. A central theme of our paper is to reconcile these two viewpoints by allowing for partial allocation and providing a clear understanding of the efficiency trade-offs associated with achieving fairness.

Offline Version as a Constrained Fair Allocation Problem. We start by considering the offline version of our problem. In the absence of fairness constraints, the offline interval scheduling problem on identical machines can be solved optimally via polynomial-time algorithms (Arkin and Silverberg, 1987; Bouzina and Emmons, 1996). When fairness constraints are imposed, our problem can be viewed as a fair allocation problem with feasibility constraints (Suksompong, 2021). Prior work on this topic, especially in the context of interval scheduling, has focused on *existential* questions pertaining to EF1 allocations alongside an efficiency requirement such as *maximality* (Igarashi, Manurangsi, and Yoneda, 2025; Y. Kumar, Equbal, Gurjar, Nath, and Vaish, 2024). (A maximal schedule is one where no unallocated interval can be feasibly assigned to any machine.) In contrast, we focus on optimizing a well-studied efficiency objective while allowing for partial allocations. Our goal is to maximize the total weight of accepted intervals subject to the EF1 constraint. This objective is strictly stronger than maximality, as any maximum (or efficient) allocation is also maximal. Our offline algorithms and lower bounds quantify the precise efficiency loss incurred by enforcing EF1 (i.e., the price of fairness) under scheduling constraints, yielding tight constant-factor bounds summarized in Table 7.1.

Online Arrival as an Additional Dimension. Next, we study how the fairness constraint interacts with online arrivals, introducing an additional layer of complexity beyond offline feasibility. In the online model, intervals arrive in nondecreasing order of their start times. Upon arrival, each interval must either be accepted or rejected. Once an interval is rejected, that decision is final; however, an accepted interval can be revoked at

any time before it ends. A fair online algorithm must make decisions without knowledge of future inputs while adhering to fairness constraints. This setup allows us to isolate the efficiency loss caused by online uncertainty, over and above the inherent loss associated with enforcing EF1 fairness.

Our results in the online setting are closely related to the literature on *temporal fair division* (Cookson, Ebadian, and Shah, 2025; Elkind, Lam, Latifian, Neoh, and Teh, 2025). This line of work studies fairness guarantees that must be upheld at every time step, but typically assumes complete knowledge of inputs in advance. In contrast, we require EF1 to be maintained at all times while operating online, without access to future intervals. By establishing tight bounds for deterministic online algorithms, we clarify the extent of additional efficiency that must be sacrificed to uphold fairness when decisions are made irrevocably and under uncertainty.

Valuation Models and Results Overview. Previous research in online interval scheduling has shown that no constant-factor competitive guarantees are possible, even on a single machine and in the absence of fairness constraints, when intervals can have arbitrary weights (Woeginger, 1994). Therefore, we focus on identical machines primarily in the *unweighted* setting, where all intervals have the same weight. In the fair allocation literature, this is often referred to as *uniform valuations* and is closely related to the equitable coloring problem (Hajnal and Szemerédi, 1970).

Although uniform valuations may appear simple in the context of fair division, they represent a *central and well-studied* model in scheduling, particularly in the online setting. In this scenario, maximizing the total accepted weight reduces to maximizing the total number of accepted intervals. This is a classic objective that remains challenging under online arrivals and feasibility constraints. Furthermore, under uniform valuations, the EF1 requirement coincides with the stronger notion of envy-freeness up to *any* item (EFX), as all items have equal value. Within this setting, we obtain tight constant-factor guarantees in both the offline and online models; summarized in Table 7.1.

In addition to examining uniform valuations, we ask whether enforcing EF1 allows for a constant efficiency loss under more general valuations. We show that in the offline setting when weights take a constant number of distinct values, it is possible to achieve constant-factor EF1 approximations. Furthermore, we establish that the 3/2 lower bound continues to hold even in the highly structured *unit-length* setting with arbitrary weights. This shows that efficiency loss is due to the fairness constraint, not interval lengths or valuation complexity. In the deterministic online setting, an additional loss occurs due to sequential arrivals, which leads to a tight competitive ratio of $2 - \frac{1}{m}$. Together, these results provide a clear separation between the efficiency loss caused by fairness and the loss associated with online uncertainty.

Although we state our results for goods with nonnegative weights, the unweighted results also transfer to the chore setting: under uniform valuations, EF1 again reduces to the same load-balancing condition, namely that any two machines receive numbers of intervals differing by at most one.

We also experimentally evaluate our online algorithm on real-world benchmark instances. These experiments show that the algorithm consistently outperforms its theoretical guarantees in practice. This suggests that while the worst-case competitive ratio is tight, the algorithm performs significantly better on typical instances.

	Upper Bound	Lower Bound
Competitive Ratio (Online, No Fairness)	1 [FN95]	1 [FN95]
Price of Fairness (Offline)	$\frac{3}{2}$ [T. 7.1]	$\frac{3}{2}$ [T. 7.5]
Fair Competitive Ratio (Online + Fair)	$2 - \frac{1}{m}$ [T. 7.7]	$2 - \frac{1}{m}$ [T. 7.19]

Table 7.1: Summary of tight guarantees in the unweighted (uniform valuation) setting. The optimal competitive ratio of 1 for online interval scheduling without fairness is due to Faigle and Nawijn, 1995, denoted by [FN95] in the table. All ratios are measured against the offline optimum *without* fairness.

Technical Overview. Our algorithms and lower bounds are guided by a common structural constraint imposed by envy-based fairness: machines cannot differ too much in their assigned load without creating unavoidable envy. In the unweighted setting, EF1 admits a particularly simple characterization in terms of per-machine counts, which allows both our algorithms and lower bounds to reason about fairness using discrete load balance.

In the offline setting, this insight leads to a two-phase approach. We first separate feasibility from fairness by starting from an efficiency-optimal schedule that ignores fairness constraints. Fairness is then enforced on top of this schedule through a combination of *rebalancing* and carefully chosen *swaps* of intervals between machines. Each such swap loses one interval, and the matching 3/2 lower bound shows that this loss is unavoidable, tightly characterizing the offline price of fairness.

In the online setting, the same structural tension is amplified by limited foresight. Our online algorithm again separates efficiency from fairness by filtering intervals through an efficiency-optimal benchmark and enforcing EF1 through balanced assignments and carefully chosen revocations. The analysis identifies a block structure in the execution and compares the intervals lost by the fair algorithm against the benchmark within each block. To carry out this comparison, we associate rejected intervals with machines through a bipartite charging graph that captures how fairness constraints force rejections and revocations over time. The key structural insight is that, within each block, these losses can be charged injectively to machines while leaving at least one machine uncharged. This yields a tight bound of $m - 1$ losses against m accepted intervals per block, which in turn leads to the competitive ratio $2 - \frac{1}{m}$. The matching lower bound shows that this dependence on the number of machines is inherent.

7.1.2 Related Work

The online interval scheduling problem was introduced by Lipton and Tomkins (1994). In their model, the intervals arrive in the order of their start times, and must be accepted or rejected irrevocably upon their arrival, following a non-overlap constraint. Early work on this problem showed that no deterministic algorithm can achieve a constant competitive ratio for irrevocable scheduling on a single machine (Long and Thakur, 1993). Similarly,

for randomized algorithms, a lower bound of $\mathcal{O}(\log \Delta)$ was shown, where Δ is the ratio of the length of the longest and shortest intervals (Lipton and Tomkins, 1994).

Subsequent studies explored the idea of *revoking* previously accepted intervals, or *preemption*, to achieve improved competitive guarantees (Borodin and Karavasilis, 2023; Faigle and Nawijn, 1995; Garay, Gopal, Kutten, Mansour, and M. Yung, 1997; Tomkins, 1995). In particular, Faigle and Nawijn (1995) studied revocable online interval scheduling for multiple identical machines. They showed that a simple greedy "deadline replacement" strategy achieves a competitive ratio of 1 in the unweighted case.

In the *weighted* setting, arbitrary weights preclude finite competitive ratios in general, both for deterministic and randomized algorithms (Canetti and Irani, 1995; Woeginger, 1994). Positive results have been shown under structured settings, such as when the weight of each interval is correlated with its length (Woeginger, 1994). The work of Fung, Poon, and D. K. W. Yung (2012) is particularly relevant to our study. They investigate the problem of scheduling *unit-length* intervals with *arbitrary weights* on multiple identical machines. Their work introduces a greedy algorithm with a competitive ratio 2 for even values of m and $2+\frac{2}{2m-1}$ for odd values of m. Their algorithm matches the lower bound of 2 in the absence of fairness constraints (Fung, Poon, and Zheng, 2008). We refer the reader to the surveys by Kolen, Lenstra, Papadimitriou, and Spieksma (2007) and Kovalyov, Ng, and T. C. E. Cheng (2007) for a comprehensive overview of interval scheduling and related allocation problems.

Recent research has focused on online interval scheduling on a single machine within a *learning-augmented* framework, also referred to as algorithms with predictions (Antoniadis, Shahheidar, Shahkarami, and Soltani, 2026; Boyar, Favrholdt, Kamali, and Larsen, 2023; Karavasilis, 2025). This area of study aims to develop online algorithms whose competitive performance improves when the predictions are accurate and degrades gracefully when the predictions are less reliable.

We now turn our attention to the literature on fair division. In the *offline* setting, recent works have studied the fair allocation of discrete items under conflict constraints (Hummel and Hetland, 2022; Igarashi, Manurangsi, and Yoneda, 2025; Y. Kumar, Equbal, Gurjar, Nath, and Vaish, 2024). In this context, all items and their pairwise conflicts are provided to the algorithm in advance. Due to the offline nature of the problem, the focus shifts from competitive analysis to investigating the *existence* of a desirable, feasible allocation; specifically, one that is approximately envy-free and economically efficient (e.g., maximal or Pareto optimal).

The *online* fair division problem has been extensively studied in recent years. Several works have explored achieving either exact or approximate envy-freeness alongside Pareto optimality as indivisible items are presented over time (Aleksandrov, Aziz, Gaspers, and Walsh, 2015; Aleksandrov and Walsh, 2019; Aleksandrov and Walsh, 2020; Benade, Kazachkov, Procaccia, Psomas, and Zeng, 2024; He, Procaccia, Psomas, and Zeng, 2019; Hosseini, Huang, Igarashi, and Shah, 2024; Kulkarni, Mehta, Narayan, and Ponitka, 2026). With the exception of He, Procaccia, Psomas, and Zeng, 2019, these studies require that the items be assigned irrevocably. Additionally, some research focuses on the *repeated* fair division problem, where the same set of items is presented to the same set of agents at each time step (Caragiannis and Narang, 2024; Igarashi, Lackner, Nardi, and Novaro, 2024).

The studies by Elkind, Lam, Latifian, Neoh, and Teh, 2025 and Cookson, Ebadian, and Shah, 2025 focus on the problem of *temporal fair division*. In this context, the algorithm is aware of the order in which items will arrive and must ensure that the allocation remains "anytime fair", i.e., satisfies approximate envy-freeness cumulatively at every time step. Choi and M. Li, 2026 study a generalization of this model where an item arriving at time t can be assigned anytime in the next r steps, i.e., within the time steps $t, t+1, \ldots, t+r-1$. Notably, these works do not take conflict constraints into account.

7.2 Preliminaries

Interval Scheduling Model. We consider interval scheduling on m identical machines. The input consists of a set $\mathcal{I}$ of n intervals, where each interval $I_j \in \mathcal{I}$ is specified by a start time s_j, an end time e_j, and a nonnegative weight $w(I_j)$. We assume all start and end times are nonnegative integers in the range $[0, T]$ for some $T \in \mathbf{Z}_{\geq 0}$. The length of an interval is $l_j = e_j - s_j$. A schedule assigns accepted intervals to machines such that no two intervals assigned to the same machine overlap in time; intervals assigned to different machines may overlap.

Offline and Online Models. In the *offline* setting, the entire set $\mathcal{I}$ is known in advance. In the *online* setting, intervals arrive sequentially in nondecreasing order of their start times. When an interval I_j arrives at time s_j, the algorithm learns (s_j, e_j) and must immediately decide whether to accept or reject it, without knowledge of future intervals. Once an interval is rejected, this decision is irrevocable. We allow *revocation* in the online model: an accepted interval may be removed from the schedule at any time before its end time e_j.

Machines and Assignments. Scheduling takes place on m identical machines, which we denote by $\mathcal{A} = \{1, 2, \ldots, m\}$. If an interval is accepted, it must be assigned to a machine $i \in \mathcal{A}$. We assume without loss of generality that intervals are indexed in order of arrival, so that $s_1 \leq s_2 \leq \cdots \leq s_n$.For a given schedule, we denote by S_i the set of intervals assigned to machine i, and we refer to $|S_i|$ as the load of machine i.

Efficiency Objective. Each accepted interval I_j contributes weight $w(I_j)$ to the objective. The goal is to maximize the total weight of accepted intervals. In the *unweighted* setting, all intervals have unit weight, and the objective reduces to maximizing the number of accepted intervals. This setting is also referred to as uniform valuation.

We denote by OPT the optimal offline algorithm without fairness constraints. Let $|\mathsf{OPT}(\mathcal{I})|$ denote the total weight achieved by OPT on instance $\mathcal{I}$, and let $|\mathsf{ALG}(\mathcal{I})|$ denote the total weight achieved by an algorithm ALG.

Performance Measures. We will now define the measures that we use to measure the performance of our algorithms in the offline and online settings.

- *Competitive Ratio*: In the *online* setting, the algorithm is provided the intervals sequentially. The performance of a (possibly unfair) online algorithm $\mathsf{ALG}^{\texttt{online}}$ is

measured by the *competitive ratio*, defined as the smallest α such that for every input instance $\mathcal{I}$ we have $|\mathsf{OPT}(\mathcal{I})| \leq \alpha \cdot |\mathsf{ALG}^{\texttt{online}}(\mathcal{I})|$. An algorithm is *c-competitive* if its competitive ratio is at most c.

- *Price of Fairness*: In the *offline* setting, the algorithm is provided all intervals upfront. The performance of an offline fair algorithm $\mathsf{ALG}^{\texttt{offline-fair}}$ is measured by its price of fairness, defined as the smallest α such that for every input instance $\mathcal{I}$ we have $|\mathsf{OPT}(\mathcal{I})| \leq \alpha \cdot |\mathsf{ALG}^{\texttt{offline-fair}}(\mathcal{I})|$.
- *Fair Competitive Ratio*: This performance measure evaluates the efficiency loss in the presence of both *fairness* and *online* constraint. Let $\mathsf{ALG}^{\texttt{online-fair}}$ denote an online algorithm that satisfies EF1. Then, the *fair competitive ratio* of such an algorithm is defined as the smallest α such that for every input instance $\mathcal{I}$ we have $|\mathsf{OPT}(\mathcal{I})| \leq \alpha \cdot |\mathsf{ALG}^{\texttt{online-fair}}(\mathcal{I})|$.

Benchmark. Throughout the paper, all performance guarantees are stated with respect to the offline optimum $|\mathsf{OPT}(\mathcal{I})|$ computed without fairness constraints. Let $\mathcal{E}$ denote the efficiency-optimal online algorithm of Faigle and Nawijn (1995). In the unweighted setting, $\mathcal{E}$ attains the offline optimum, i.e. $|\mathcal{E}(\mathcal{I})| = |\mathsf{OPT}(\mathcal{I})|$.

Fairness Notions. In addition to efficiency, we study fairness across machines. We view each machine as an agent and the intervals assigned to it as the agent's bundle. For a machine $i \in \mathcal{A}$, let $S_i \subseteq \mathcal{I}$ denote the set of intervals assigned to i. The utility of machine i is additive, i.e.

$$u_i(S_i) = \sum_{I_j \in S_i} w(I_j).$$

We adopt envy-based fairness notions from the fair division literature. A schedule is *envy-free up to one interval (EF1)* if for every pair of machines $i, k \in \mathcal{A}$ with $i \neq k$ and $S_k \neq \emptyset$, there exists an interval $I \in S_k$ such that

$$u_i(S_i) \geq u_i(S_k) - w(I).$$

A schedule is *envy-free up to any interval (EFX)* if for every pair $i, k \in \mathcal{A}$ with $S_k \neq \emptyset$ and every interval $I \in S_k$,

$$u_i(S_i) \geq u_i(S_k) - w(I).$$

In the unweighted setting, EF1 and EFX coincide. These notions are inspired by the *envy-freeness up to one good* (Budish, 2011; Lipton, Markakis, Mossel, and Saberi, 2004) and *envy-freeness up to any good* (Caragiannis, Kurokawa, Moulin, Procaccia, Shah, and J. Wang, 2019) notions in the fair division literature.

7.3 Offline EF1 Scheduling

We begin by studying the offline version of interval scheduling under EF1 fairness, where the entire set of intervals is known in advance. As discussed earlier, this problem can be viewed both as a fair allocation problem with scheduling constraints and as a scheduling problem with a fairness constraint. A central question in this setting is how much efficiency

must be sacrificed in order to guarantee EF1. Our goal in this section is to characterize the loss of efficiency caused by enforcing fairness in the offline setting, by designing approximation algorithms for maximum EF1 allocation and by proving matching lower bounds that establish the intrinsic cost of EF1.

7.3.1 Maximum EF1 Allocation

Without fairness constraints, the offline interval scheduling problem on identical machines admits an efficient optimal allocation (Arkin and Silverberg, 1987; Bouzina and Emmons, 1996). This provides a natural benchmark for evaluating the cost of enforcing fairness. We first consider the *unweighted* setting, where all intervals have unit weight. Our focus is on computing an allocation that satisfies EF1 while retaining as much efficiency as possible. In particular, we show that EF1 can be enforced in the offline setting with only a 3/2 price of fairness.

Theorem 7.1 (Offline EF1 Upper Bound: Unweighted)**.** *In the* unweighted offline *setting with m identical machines, there exists a polynomial-time EF1 allocation algorithm* ALG *whose price of fairness is at most* 3/2*. That is, for every instance $\mathcal{I}$,*

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \leq \frac{3}{2},$$

where $\mathsf{OPT}(\mathcal{I})$ *denotes the optimal offline allocation without fairness constraints.*

We next describe the algorithm and prove Theorem 7.1.

***Rebalance-and-Swap* Algorithm.** Our algorithm starts from an optimal offline allocation without fairness constraints and incrementally transforms it into an EF1 allocation. This makes the efficiency loss due to fairness easier to analyze, as we can directly compare to the initial optimal allocation.

This is accomplished via a subroutine Rebalance, that for a given set of machines $\mathcal{A}$, produces a new allocation such that the load on each machine in $\mathcal{A}$ is either $\lfloor \mu \rfloor$ or $\lceil \mu \rceil$, where μ is the average load across machines in $\mathcal{A}$ after removing $(m-1)$ intervals. It does this by repeatedly considering the two machines with the largest and smallest load, and invoking a procedure Swap that exchanges intervals between these two machines such that the load on one of the machines becomes either $\lfloor \mu \rfloor$ or $\lceil \mu \rceil$. The machine with the balanced load is then removed from consideration, and the process continues until all machines in $\mathcal{A}$ have load either $\lfloor \mu \rfloor$ or $\lceil \mu \rceil$.

The key component in our algorithm is the Swap procedure that takes as input two machines i and $\bar{i}$, along with target loads t_i and $t_{\bar{i}}$. It works by taking a time t and swapping intervals that start after time t between the two machines. We choose the time t such that we only need to discard one interval from one of the machines to achieve the target loads. The full description of these procedures are given below.

We next show that the procedures Swap and Rebalance are well-defined and achieve their intended load adjustments.

Lemma 7.2. *For machines $i, \bar{i}$, $\ell'_i \geq 0$ and $\ell'_{\bar{i}} \geq 0$ with $\ell'_i + \ell'_{\bar{i}} = |S_i| + |S_{\bar{i}}| - 1$, Swap$(i, \bar{i}, \ell'_i, \ell'_{\bar{i}})$ is well-defined and results in $|S_i| = \ell'_i$ and $|S_{\bar{i}}| = \ell'_{\bar{i}}$.*

Input : Interval set $\mathcal{I}$ and m identical machines
Output : An EF1 allocation of intervals to machines

Compute an optimal offline allocation OPT without fairness constraints Initialize S_i to be the set of intervals assigned to machine i in OPT
while *there exist machines* $i, \bar{i}$ *with* $|S_i| \geq 2$ *and* $|S_{\bar{i}}| = 0$ **do**
  Reassign an interval from i to $\bar{i}$
end
Reorder the machines so that $|S_1| \geq |S_2| \geq \cdots \geq |S_m|$
if $|\mathsf{OPT}| \leq \frac{3}{2}m$ **then**
  For each machine i with $|S_i| \geq 2$, drop intervals until $|S_i| = 1$ **return** the current allocation
end
else
  if $|\mathsf{OPT}| \leq 3m - 1$ **then**
    Set $x \leftarrow \left\lceil \frac{2}{3}|\mathsf{OPT}| - m \right\rceil$ Set $t_i \leftarrow 2$ for all $i \leq x$ Invoke Rebalance on machine set $\{1, \ldots, x\}$ with targets $(t_i)_{i=1}^{x}$ For each $i = x+1, \ldots, m$, drop intervals from S_i until $|S_i| = 1$
  end
  else
    Set $L_- \leftarrow \left\lfloor \frac{|\mathsf{OPT}| - (m-1)}{m} \right\rfloor$ and $L_+ \leftarrow \left\lceil \frac{|\mathsf{OPT}| - (m-1)}{m} \right\rceil$ Choose k such that $kL_- + (m-k)L_+ = |\mathsf{OPT}| - (m-1)$ Set $t_i \leftarrow L_-$ for all $i \leq k$ and $t_i \leftarrow L_+$ for all $i > k$ Invoke Rebalance on machine set $\{1, \ldots, m\}$ with targets $(t_i)_{i=1}^{m}$
  end
end

Algorithm 7.1: *Rebalance-and-Swap* Algorithm

Input : Machine set $\mathcal{M}$, target loads $(t_i)_{i \in \mathcal{M}}$ with $\sum_{i \in \mathcal{M}} t_i \leq \sum_{i \in \mathcal{M}} |S_i| - (|\mathcal{M}| - 1)$
Output : Each machine $i \in \mathcal{M}$ has $|S_i| = t_i$

Order the machines as $i_1, \ldots, i_{|\mathcal{M}|}$ such that $t_{i_1} \leq t_{i_2} \leq \cdots \leq t_{i_{|\mathcal{M}|}}$ Permute the interval sets so that $|S_{i_1}| \geq |S_{i_2}| \geq \cdots \geq |S_{i_{|\mathcal{M}|}}|$ For each $k = 1, \ldots, |\mathcal{M}|$ set $\ell_{i_k} \leftarrow |S_{i_k}|$
for $k = 1, \ldots, |\mathcal{M}| - 1$ **do**
  Invoke Swap on machines i_k and i_{k+1} with targets t_{i_k} and $\sum_{j=1}^{k}(\ell_{i_j} - t_{i_j} - 1) + \ell_{i_{k+1}}$
end
if $|S_{i_{|\mathcal{M}|}}| > t_{i_{|\mathcal{M}|}}$ **then**
  Remove arbitrary intervals from $S_{i_{|\mathcal{M}|}}$ until $|S_{i_{|\mathcal{M}|}}| = t_{i_{|\mathcal{M}|}}$
end

Algorithm 7.2: Procedure Rebalance

Proof. To see that there exists a feasible time t in line 7.3 and that the procedure is well-defined, observe that the function $|A_i(t)| - |A_{\bar{i}}(t)|$ takes values in $\mathbf{Z}$ and has jumps of size at most 1. Furthermore, we have $|A_i(0)| - |A_{\bar{i}}(0)| = 0$ and $|A_i(T)| - |A_{\bar{i}}(T)| = \ell_i - \ell_{\bar{i}} > 0$. Since $\ell_i - \ell_{\bar{i}} \geq \ell_i - \ell'_i \geq 0$, there must be some time t where $|A_i(t)| - |A_{\bar{i}}(t)| = \ell_i - \ell'_i$.

Input : Machines $i, \bar{i}$ and target loads $\ell'_i, \ell'_{\bar{i}} \geq 0$ such that $\ell'_i + \ell'_{\bar{i}} = |S_i| + |S_{\bar{i}}| - 1$
Output : $|S_i| = \ell'_i$ and $|S_{\bar{i}}| = \ell'_{\bar{i}}$

For each $i' \in \{i, \bar{i}\}$ define $A_{i'}(t) := \{j \in S_{i'} : e_j \leq t\}$ and $B_{i'}(t) := \{j \in S_{i'} : s_j \geq t\}$
Find the smallest time t such that

$$|A_i(t)| - |A_{\bar{i}}(t)| = |S_i| - \ell'_i = \ell'_{\bar{i}} - |S_{\bar{i}}| + 1$$

Set $S'_i \leftarrow A_{\bar{i}}(t) \cup B_i(t)$ and $S'_{\bar{i}} \leftarrow A_i(t) \cup B_{\bar{i}}(t)$ **if** $|S'_{\bar{i}}| > \ell'_{\bar{i}}$ **then**
  Remove one interval from $S'_{\bar{i}}$
end
Assign $S_i \leftarrow S'_i$ and $S_{\bar{i}} \leftarrow S'_{\bar{i}}$

Algorithm 7.3: Procedure Swap

As t is the smallest such time, an interval finishes on machine i at time t. Hence, we have $A_i(t) \cup B_i(t) = S_i$. For machine $\bar{i}$, we have $|S_{\bar{i}} \setminus (A_{\bar{i}}(t) \cup B_{\bar{i}}(t))| \leq 1$, since at most one interval can be active on machine $\bar{i}$ at time t.

So, we have $|S'_i| = |A_{\bar{i}}(t)| + |B_i(t)| = |A_{\bar{i}}(t)| + |S_i| - |A_i(t)| = \ell'_i$, and $|S'_{\bar{i}}| \geq |A_i(t)| + |B_{\bar{i}}(t)| \geq |A_i(t)| + |S_{\bar{i}}| - |A_{\bar{i}}(t)| - 1 = \ell'_{\bar{i}}$. Then if $|S'_{\bar{i}}| > \ell'_{\bar{i}}$, we can remove an interval from $S'_{\bar{i}}$ so that $|S'_{\bar{i}}| = \ell'_{\bar{i}}$. This ensures that $|S_i| = \ell'_i$ and $|S_{\bar{i}}| = \ell'_{\bar{i}}$, as desired. □

Lemma 7.3. *Let $\mathcal{M}$ be a set of machines and let $(t_i)_{i \in \mathcal{M}}$ be target loads specifying the desired number of intervals on each machine $i \in \mathcal{M}$. If $\sum_{i \in \mathcal{M}} t_i \leq \sum_{i \in \mathcal{M}} |S_i| - (|\mathcal{M}| - 1)$, then the procedure Rebalance is well-defined and results in $|S_i| = t_i$ for all $i \in \mathcal{M}$.*

Proof. We verify that the procedure Rebalance is well-defined by checking that each call to Swap satisfies its preconditions, and then argue that the procedure results in $|S_i| = t_i$ for all $i \in \mathcal{M}$. For convenience, we will use $M := |\mathcal{M}|$ to denote the number of machines in $\mathcal{M}$.

We first check the preconditions for the call $\textsc{Swap}(k, k+1, \ell'_i, \ell'_{\bar{i}})$ for each $1 \leq k \leq M - 1$. Recall that Swap requires $\ell'_{i_k} \geq 0$, $\ell'_{i_{k+1}} \geq 0$, and $\ell'_{i_k} + \ell'_{i_{k+1}} = |S_{i_k}| + |S_{i_{k+1}}| - 1$. Note that we have $\ell'_{i_k} = t_{i_k} \geq 0$ and $\ell'_{i_{k+1}} = \sum_{j=1}^{k} (\ell_{i_j} - t_{i_j} - 1) + \ell_{i_{k+1}}$. Hence:

$$\begin{aligned} \ell'_{i_k} + \ell'_{i_{k+1}} &= t_{i_k} + \sum_{j=1}^{k} (\ell_{i_j} - t_{i_j} - 1) + \ell_{i_{k+1}} = \sum_{j=1}^{k-1} (\ell_{i_j} - t_{i_j} - 1) + \ell_{i_k} + \ell_{i_{k+1}} - 1 \\ &= |S_{i_k}| + |S_{i_{k+1}}| - 1. \end{aligned}$$

Now let us verify that $\ell'_{i_{k+1}} \geq 0$. Recall that $\ell_{i_1} \geq \ell_{i_2} \geq \ldots \geq \ell_{i_{|\mathcal{M}|}}$ and $t_{i_1} \leq t_{i_2} \leq \ldots \leq t_{i_{|\mathcal{M}|}}$. Hence, we have that $\sum_{j=1}^{k} \ell_{i_j} \geq \frac{k}{M} \sum_{j=1}^{M} \ell_{i_j}$ and $\sum_{j=1}^{k} t_{i_j} \leq \frac{k}{M} \sum_{j=1}^{M} t_{i_j}$. This implies that:

$$\begin{aligned}
\ell'_{i_{k+1}} &= \sum_{j=1}^{k} (\ell_{i_j} - t_{i_j} - 1) + \ell_{i_{k+1}} \geq \sum_{j=1}^{k+1} \ell_{i_j} - \sum_{j=1}^{k} t_{i_j} - k \\
&\geq \frac{k+1}{M} \sum_{j=1}^{M} \ell_{i_j} - \frac{k}{M} (\sum_{j=1}^{M} \ell_{i_j} - (M-1)) - k \\
&= \frac{1}{M} \sum_{j=1}^{M} \ell_{i_j} - k/M \geq (M-1)/M - k/M \geq 0.
\end{aligned}$$

Each call to $\textsc{Swap}(i_k, i_{k+1}, \ell'_i, \ell'_{\bar{i}})$ fixes the $|S_{i_k}| = t_{i_k}$. Moreover, subsequent swap operations do not modify S_{i_k}, since later calls only involve machines $i_{k+1}, i_{k+2}, \ldots, i_M$. After the loop terminates, the final machine i_M may have load larger than t_{i_M}, in which case the algorithm removes arbitrary intervals until $|S_{i_M}| = t_{i_M}$. Thus, every machine $i \in \mathcal{M}$ ends with exactly t_i intervals, completing the proof. □

Now we can prove Theorem 7.1, showing that the above algorithm achieves EF1 with price of fairness at most 3/2.

Proof of Theorem 7.1. We analyze the allocation returned by the *Rebalance-and-Swap* algorithm according to the value of $|\mathsf{OPT}(\mathcal{I})|$. Recall that in the unweighted setting, EF1 is equivalent to the condition that the number of intervals assigned to any two machines differs by at most 1.

We show that in all cases the algorithm outputs an EF1 allocation and satisfies

$$|\mathsf{ALG}(\mathcal{I})| \;\geq\; \tfrac{2}{3}|\mathsf{OPT}(\mathcal{I})|.$$

Case 1: $|\mathsf{OPT}(\mathcal{I})| \leq \frac{3}{2}m$**.** In this regime, the optimal allocation assigns at most one interval to each machine. If $|\mathsf{OPT}(\mathcal{I})| \leq m$, then each interval can be placed on a distinct machine and the algorithm returns the same allocation, so $|\mathsf{ALG}(\mathcal{I})| = |\mathsf{OPT}(\mathcal{I})|$.

If $m < |\mathsf{OPT}(\mathcal{I})| \leq \frac{3}{2}m$, then the algorithm assigns exactly one interval to each machine, yielding $|\mathsf{ALG}(\mathcal{I})| = m$. Since $m \geq \frac{2}{3}|\mathsf{OPT}(\mathcal{I})|$ in this range, the approximation guarantee holds. In both subcases, all machines have equal load, so the allocation is EF1.

Case 2: $\frac{3}{2}m < |\mathsf{OPT}(\mathcal{I})| \leq 3m - 1$**.** In this case, the algorithm outputs an allocation with

$$x \;=\; \left\lceil \tfrac{2}{3}|\mathsf{OPT}(\mathcal{I})| - m \right\rceil$$

machines receiving two intervals and the remaining $m - x$ machines receiving one interval. Thus, machine loads differ by at most one, and the allocation is EF1.

The total number of accepted intervals is

$$|\mathsf{ALG}(\mathcal{I})| \;=\; 2x + (m - x) \;=\; m + x \;=\; \left\lceil \tfrac{2}{3}|\mathsf{OPT}(\mathcal{I})| \right\rceil \;\geq\; \tfrac{2}{3}|\mathsf{OPT}(\mathcal{I})|,$$

as claimed.

Case 3: $|\mathsf{OPT}(\mathcal{I})| \geq 3m$**.** Here the algorithm produces an allocation in which machines receive either

$$L_- \;=\; \left\lfloor \tfrac{|\mathsf{OPT}(\mathcal{I})|-(m-1)}{m} \right\rfloor \quad \text{or} \quad L_+ \;=\; \left\lceil \tfrac{|\mathsf{OPT}(\mathcal{I})|-(m-1)}{m} \right\rceil$$

intervals, with k machines receiving L_- intervals and the remaining $m-k$ machines receiving L_+ intervals. Again, loads differ by at most one, so the allocation is EF1.

The total number of accepted intervals is

$$|\mathsf{ALG}(\mathcal{I})| \;=\; kL_- + (m-k)L_+ \;=\; |\mathsf{OPT}(\mathcal{I})| - (m-1).$$

Since $|\mathsf{OPT}(\mathcal{I})| \geq 3m$, we have $m-1 \leq \frac{1}{3}|\mathsf{OPT}(\mathcal{I})|$, and therefore

$$|\mathsf{ALG}(\mathcal{I})| \;\geq\; |\mathsf{OPT}(\mathcal{I})| - \tfrac{1}{3}|\mathsf{OPT}(\mathcal{I})| \;=\; \tfrac{2}{3}|\mathsf{OPT}(\mathcal{I})|.$$

In all cases, the algorithm returns an EF1 allocation with value at least $\frac{2}{3}|\mathsf{OPT}(\mathcal{I})|$, completing the proof. □

Extension to Multiple Weight Levels. We now show that the offline EF1 guarantee extends beyond uniform valuation to settings with a bounded number of distinct weights, at the cost of a proportional loss in efficiency.

Corollary 7.4 (Offline EF1 with Multiple Weight Levels)**.** *In the* offline *setting with m identical machines, suppose that interval weights take values from a set of k distinct positive weights. Then there exists a polynomial-time EF1 allocation algorithm whose price of fairness is at most $\frac{3}{2}\,k$. That is, for every instance $\mathcal{I}$,*

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \;\leq\; \tfrac{3}{2}\,k.$$

Proof. Let the set of distinct weights be $\{\omega_1, \ldots, \omega_k\}$, and for each $t \in [k]$ let

$$\mathcal{I}_t \;=\; \{\, I \in \mathcal{I} : w(I) = \omega_t \,\}$$

denote the sub-instance consisting of intervals of weight ω_t.

For each t, we apply the algorithm of Theorem 7.1 to the unweighted instance $\mathcal{I}_t$, obtaining an EF1 schedule ALG_t. Since all intervals in $\mathcal{I}_t$ have the same weight, Theorem 7.1 implies

$$|\mathsf{OPT}(\mathcal{I}_t)| \;\leq\; \tfrac{3}{2}\,|\mathsf{ALG}_t|.$$

Let $t^\star \in [k]$ be an index maximizing $|\mathsf{ALG}_t|$, and output $\mathsf{ALG}_{t^\star}$. The resulting schedule is EF1.

To relate this to the global offline optimum, observe that for each t, the set of intervals of weight ω_t selected by $\mathsf{OPT}(\mathcal{I})$ forms a feasible schedule for the sub-instance $\mathcal{I}_t$. Hence

$$|\mathsf{OPT}(\mathcal{I})| \;\leq\; \sum_{t=1}^{k} |\mathsf{OPT}(\mathcal{I}_t)|.$$

Therefore,

$$|\mathsf{OPT}(\mathcal{I})| \;\leq\; \sum_{t=1}^{k} |\mathsf{OPT}(\mathcal{I}_t)| \;\leq\; \sum_{t=1}^{k} \tfrac{3}{2}\,|\mathsf{ALG}_t| \;\leq\; \tfrac{3}{2}\,k \cdot |\mathsf{ALG}_{t^\star}|.$$

This proves that the price of fairness is at most $\frac{3}{2}k$. □

7.3.2 Lower Bounds

We now show that the 3/2 price of fairness established in the previous subsection is unavoidable. Specifically, we prove that no EF1 allocation algorithm can achieve a better approximation guarantee with respect to the offline optimum without fairness. Our lower bounds hold in two fundamental settings: the unweighted case, and the unit-length weighted case, where intervals may have arbitrary weights but identical lengths. Together, these results show that the efficiency loss arises from the fairness constraint itself, rather than from specific features of the valuation model or interval lengths.

Theorem 7.5 (Offline EF1 Lower Bound: Unweighted)**.** *In the* unweighted offline *setting with m identical machines, no EF1 allocation algorithm can achieve a price of fairness strictly smaller than $\frac{3}{2}$. That is, for every EF1 allocation algorithm* ALG*, there exists an instance $\mathcal{I}$ such that*

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \;\geq\; \tfrac{3}{2}.$$

Proof. Fix $m \geq 2$. Consider the instance $\mathcal{I}$ consisting of the following intervals, all of unit weight.

- $2m-1$ *short* intervals of unit length: for each $t \in \{0, 1, \ldots, 2m-2\}$ there is one interval
$$I_t \;=\; [t, t+1).$$
- $m-1$ *long* intervals of length $2m-1$: for each $q \in \{1, \ldots, m-1\}$ there is one interval
$$L_q \;=\; [0, 2m-1).$$

Benchmark (offline optimum without fairness). An offline optimum without fairness schedules all $3m-2$ intervals: place the $m-1$ long intervals on $m-1$ distinct machines, and place all $2m-1$ short intervals sequentially on the remaining machine. Hence

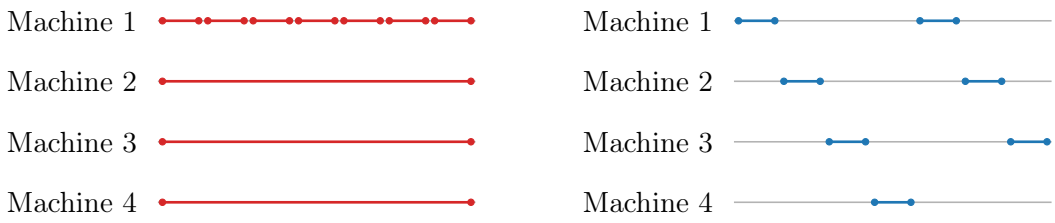
$$|\mathsf{OPT}(\mathcal{I})| \;=\; (m-1) + (2m-1) \;=\; 3m-2.$$

Any EF1 schedule accepts at most $2m-1$ intervals. In the unweighted setting, EF1 coincides with EFX and is equivalent to the following condition on the per-machine counts: if $C_i := |S_i|$ denotes the number of accepted intervals assigned to machine $i \in \mathcal{A}$, then EF1 holds if and only if

$$\max_{i \in \mathcal{A}} C_i \;-\; \min_{i \in \mathcal{A}} C_i \;\leq\; 1. \tag{7.1}$$

Indeed, since every interval has weight 1 and utilities are additive, machine i envies machine k exactly when $C_k > C_i$, and removing one interval from k reduces its value by 1.

We now argue that any feasible EF1 schedule for $\mathcal{I}$ can accept at most $2m-1$ intervals. First, note that among the $m-1$ long intervals, at most $m-1$ can be accepted (one per machine), and any machine that accepts a long interval cannot accept any short interval (since each long interval overlaps all short ones).



(a) Offline optimum without fairness, accepting all $3m-2$ intervals.

(b) An EF1 allocation, which can accept at most $2m-1$ intervals.

Figure 7.1: Lower-bound instance for Theorem 7.5 in the unweighted offline setting, illustrated for $m = 4$. The offline optimum without fairness schedules all $3m-2$ intervals, whereas any EF1 allocation can accept at most $2m-1$ intervals; one such EF1 allocation is shown.

If an EF1 schedule accepts at least one long interval, then some machine has $C_i = 1$ (it carries exactly that long interval). By (7.1), every machine must then satisfy $C_i \leq 2$, and hence the total number of accepted intervals is at most

$$\sum_{i \in \mathcal{A}} C_i \;\leq\; 2m.$$

However, in this instance one cannot reach $2m$ accepted intervals: achieving $2m$ would require $C_i = 2$ for all i, meaning every machine would need to accept two intervals. This is impossible because any machine that carries a long interval cannot carry a short one, while there are only $m-1$ long intervals. Thus at least one machine carries no long interval, and the maximum total becomes $2m-1$.

If, on the other hand, the EF1 schedule accepts no long intervals, then it can only accept short intervals. Since there are only $2m-1$ short intervals, in this case the total accepted is at most $2m-1$ as well.

Therefore every EF1 allocation ALG satisfies

$$|\mathsf{ALG}(\mathcal{I})| \;\leq\; 2m-1.$$

Price of fairness. Combining the two bounds,

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \;\geq\; \frac{3m-2}{2m-1}.$$

This quantity approaches $\frac{3}{2}$ as m grows. In particular, no EF1 allocation algorithm can guarantee a price of fairness strictly smaller than $\frac{3}{2}$. □

We next show that this lower bound persists even in a more structured setting, where all intervals have identical length but arbitrary weights.

Theorem 7.6 (Offline EF1 Lower Bound: Unit-Length)**.** *In the* unit-length offline *setting with $m \geq 2$ identical machines and nonnegative weights, no EF1 allocation algorithm*

can achieve a price of fairness strictly smaller than $\frac{3}{2}$. That is, for every EF1 allocation algorithm ALG*, there exists an instance $\mathcal{I}$ such that*

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \geq \frac{3}{2}.$$

Proof. Fix $\varepsilon \in (0,1)$ and assume first that m is even. Partition the machines into $m/2$ disjoint pairs. For each pair $p \in \{1, \ldots, m/2\}$, we create three unit-length intervals

$$I_1^p = [0,1), \qquad I_2^p = \left[\tfrac{1}{2}, \tfrac{3}{2}\right), \qquad I_3^p = [1,2),$$

with weights $w(I_1^p) = w(I_3^p) = 1$ and $w(I_2^p) = 1 - \varepsilon$. Let $\mathcal{I}$ be the union of all these intervals.

Benchmark (offline optimum without fairness). Fix a pair p and consider any feasible schedule that accepts all three intervals $\{I_1^p, I_2^p, I_3^p\}$ using the two machines of that pair. Since I_2^p overlaps both I_1^p and I_3^p, any machine that runs I_2^p cannot run either neighbor. Therefore, to accept all three intervals, one must schedule I_2^p on one machine and schedule both I_1^p and I_3^p (which do not overlap) on the other machine of the pair. Hence, within each pair p, the unconstrained optimum value is $w(I_1^p) + w(I_2^p) + w(I_3^p) = 3 - \varepsilon$. By scheduling each pair independently on its two machines, we obtain

$$|\mathsf{OPT}(\mathcal{I})| \geq \frac{m}{2}(3 - \varepsilon).$$

EF1 forces dropping I_2^p if I_1^p and I_3^p are both accepted. Fix a pair p. Suppose an EF1 schedule accepts all three intervals of this pair. As argued above, feasibility then forces one machine (call it A) to receive $\{I_1^p, I_3^p\}$ and the other machine (call it B) to receive $\{I_2^p\}$ (within this pair). Consider machine B's utility. It assigns value 2 to the bundle $\{I_1^p, I_3^p\}$ and value $1 - \varepsilon$ to the bundle $\{I_2^p\}$. Removing either single item from A leaves value 1, which is still strictly larger than $1 - \varepsilon$. Thus B continues to envy A even after removing one interval from A's bundle, contradicting EF1. Therefore, in any EF1 schedule, whenever both I_1^p and I_3^p are accepted, interval I_2^p must be rejected.

Per-pair upper bound under EF1. From the previous paragraph, an EF1 schedule can accept at most two intervals from each pair p. The maximum total weight obtainable from pair p under EF1 is therefore at most

$$\max\{\, w(I_1^p) + w(I_3^p),\ w(I_1^p) + w(I_2^p),\ w(I_2^p) + w(I_3^p) \,\} = 2.$$

Summing over all $m/2$ pairs, every EF1 allocation ALG satisfies

$$|\mathsf{ALG}(\mathcal{I})| \leq \frac{m}{2} \cdot 2 = m.$$

Price of fairness. Combining the bounds,

$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \geq \frac{\frac{m}{2}(3-\varepsilon)}{m} = \frac{3-\varepsilon}{2}.$$

Letting $\varepsilon \to 0$ yields the lower bound $\frac{3}{2}$.

(a) Offline optimum without fairness. **(b)** An EF1 allocation.

Figure 7.2: Lower-bound instance for Theorem 7.6 in the unit-length offline setting, illustrated for $m = 4$. The offline optimum without fairness accepts all three intervals in each pair, whereas any EF1 allocation can accept at most two intervals per pair; one such EF1 allocation is shown.

Odd m. For odd m, apply the same construction on $m - 1$ machines and place no intervals on the remaining machine; the bound remains $\geq (3-\varepsilon)/2$ and hence approaches $3/2$ as $\varepsilon \to 0$. □

The lower-bound instances for Theorems 7.5 and 7.6 are illustrated in Figures 7.1 and 7.2, respectively, for $m = 4$ machines.

Together, Theorems 7.5 and 7.6 show that the 3/2 price of fairness is intrinsic to EF1 and cannot be avoided even under strong structural restrictions on the intervals.

7.4 Online EF1 Scheduling

We now turn to the online setting, where intervals arrive sequentially and decisions must be made without knowledge of future inputs. Recall that when intervals can have arbitrary weights, no constant-factor competitive guarantees are possible even on a single machine without fairness constraints (Woeginger, 1994). This motivates the consideration of the *unweighted* setting where all intervals have unit weight.

In the online model, enforcing fairness introduces an additional source of difficulty beyond feasibility, since EF1 must be maintained at all times under irrevocable rejections and limited foresight. Our goal is to understand how much additional efficiency is lost when EF1 is imposed in the presence of online uncertainty. We show that this loss can be bounded by a constant factor, and we provide a matching lower bound establishing the tightness of our guarantees.

A useful term to consider is the *leader* of a machine. Given a partial schedule, the leader of a machine is the rightmost interval assigned to it. Note that, given the schedule, the leader is uniquely specified, and if there is no interval on a machine, we say that the leader is $\emptyset$. We will also find it useful to call a machine *available* at time t if an interval starting at time t can be assigned to the machine without creating any conflict. Otherwise, we will call the machine *busy*.

Greedy-Optimal Algorithm (Without Fairness). In the unweighted setting, the offline optimum can be achieved even in the online model. As shown by Faigle and Nawijn

(1995), there exists an online algorithm that maximizes the number of accepted intervals; we refer to this algorithm as *Greedy-Optimal* and denote it by $\mathcal{E}$.

The Greedy-Optimal algorithm is a simple greedy procedure with revocation. When an interval $I_i = (s_i, e_i)$ arrives, if it can be assigned to a machine without causing a conflict, it is accepted. Otherwise, the algorithm compares its end time e_i with the end time of all the leaders, i.e., the interval assigned to the machine with the largest end time. If e_i is smaller, the algorithm revokes the leader with the largest end time and assigns I_i to that machine; otherwise, I_i is rejected. This rule guarantees that the number of accepted intervals is maximized and thus achieves optimal efficiency.

7.4.1 Online EF1 Algorithm

We now present our fair online algorithm, called *Greedy-Balanced*. The algorithm is designed for the unweighted setting and builds directly on the Greedy-Optimal benchmark algorithm described above. At a high level, Greedy-Balanced follows the same acceptance decisions as Greedy-Optimal whenever possible, but modifies the assignment of intervals to machines in order to maintain EF1 at all times. This balancing step may require revoking previously accepted intervals, but only when necessary to preserve fairness.

We show that Greedy-Balanced maintains EF1 throughout the execution and achieves a competitive ratio of $2 - \frac{1}{m}$ with respect to the offline optimum without fairness. This guarantee is tight, as we establish in the subsequent lower bound subsection.

Greedy-Balanced Algorithm. Let $\mathcal{E}$ be the Greedy-Optimal online algorithm of Faigle and Nawijn (1995). Our Greedy-Balanced algorithm runs a parallel execution of $\mathcal{E}$ and only considers intervals that are accepted by $\mathcal{E}$ for scheduling.

For each arriving interval $I_i = (s_i, e_i)$, the algorithm proceeds as follows. At a high level, our algorithm proceeds in four steps. First, it filters intervals according to $\mathcal{E}$ and only considers those accepted by $\mathcal{E}$. Second, it checks if the interval can be assigned to a feasible machine while preserving EF1. Specifically, the algorithm considers the set of available machines at time s_i that are *light* (i.e., the machines with minimum load) and places the interval on an arbitrary available light machine whenever such a placement exists. Third, if no such assignment is possible, the algorithm attempts to repair the current schedule by replacing a previously accepted overlapping interval with the new one. In this case, it replaces I_i with the leader having the largest end time among the set of light machines. Fourth, if neither assignment nor replacement is possible, the interval is rejected. Formally, the algorithm proceeds as follows.

(1) **Filter by $\mathcal{E}$:** If $\mathcal{E}$ rejects I_i, then our algorithm also rejects I_i. If $\mathcal{E}$ revokes an interval J to accept I_i and our algorithm is also currently running J on some machine, then it also revokes J and schedules I_i on the same machine. Otherwise, it proceeds to the next step.

(2) **EF1-preserving placement:** Let $\mathcal{F}_i \subseteq \mathcal{A}$ be the set of machines on which I_i can be scheduled at time s_i without creating a conflict, and let C_ℓ denote the current number of accepted intervals assigned to machine ℓ. If $\mathcal{F}_i \neq \emptyset$, choose

$$f \in \arg\min_{\ell \in \mathcal{F}_i} C_\ell$$

where ties can be broken arbitrarily. If assigning I_i to f preserves EF1, then assign I_i to f and continue to the next arrival.

(3) **Repair via replacement on a light machine:** If assigning I_i to any machine in $\mathcal{F}_i$ violates EF1, let
$$L_i = \{k \in \mathcal{A} : C_k = \min_{\ell \in \mathcal{A}} C_\ell\}$$
be the set of light machines. If there exists a machine $k \in L_i$ and an interval $J \in S_k$ such that J overlaps I_i and $e_J > e_i$, then replace J with the interval I_i on the machine k. Among all such pairs (k, J), choose one with maximum e_J.

(4) **Otherwise, reject I_i:** If no such replacement is possible, reject I_i.

We now state the performance guarantee of the Greedy-Balanced algorithm.

Theorem 7.7 (Online EF1 Upper Bound: Unweighted)**.** *In the* unweighted online *setting with m identical machines, the* Greedy-Balanced *algorithm maintains EF1 at all times and has competitive ratio at most $2 - \frac{1}{m}$. That is, for every input instance $\mathcal{I}$,*
$$\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \leq 2 - \tfrac{1}{m}.$$

Proof. Recall that the Greedy-Optimal algorithm $\mathcal{E}$ is optimal in the unweighted online setting, and therefore
$$|\mathcal{E}(\mathcal{I})| = |\mathsf{OPT}(\mathcal{I})|$$
for every instance $\mathcal{I}$. It thus suffices to prove that
$$\frac{|\mathcal{E}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \leq 2 - \frac{1}{m}.$$

We assume that all s_j and e_j are distinct and strictly positive. This is without loss of generality since although the allocation can be different, the number of accepted intervals would be the same.

The next observation follows from the description of the algorithm.

Observation 7.8. *The* Greedy-Balanced *algorithm is EF1 at all times. Furthermore, the loads are non-decreasing with time.*

Note that, by the above observation, at any time point t, there exists an integer α such that each machine has a load of α or $\alpha+1$. Furthermore, a machine can have a load of $\alpha+2$ only if all other machines have load at least $\alpha+1$. Therefore, our *light* machines correspond to the machines with load α, and we call all the machines with load $\alpha+1$ *heavy*. This motivates a natural indexing of the timeline into *blocks* as follows.

For each $k \geq 1$, let t_k be the first time that all the machines have load exactly k, and let ω be the largest k such that t_k is finite. For convenience, let $t_0 := 0$. For each $k = 0, \ldots, \omega - 1$, we define the k-th *block* by
$$B_k := [t_k, t_{k+1}),$$

and we define the final block by $B_\omega := [t_\omega, \infty)$.

Note that the interval that brings a machine from load k to $k+1$ for the first time might be revoked later. Recall that revocation does not decrease the load of a machine.

Call the start time s_i an *acceptance event* if the interval is allocated in step (2), and a *rejection event* if it revokes an interval in step (3) or if it gets rejected in step (4). Let A_k be the number of acceptance events in block B_k for each $k = 0, \ldots, \omega$.

Observe that for every $0 \leq k < \omega$, we have $A_k = m$, since block B_k ends exactly when every machine has received one more accepted interval. Furthermore, there are no rejection events in block B_0, since at time $t_0 = 0$ there are no intervals assigned to any machine, so the first m intervals accepted by $\mathcal{E}$ are also accepted by ALG and assigned to distinct machines.

A central ingredient of the analysis is a mapping ψ that associates each interval accepted by $\mathcal{E}$ with a machine. Intuitively, $\psi(I)$ represents the machine on which interval (I) would naturally be placed if fairness were ignored. Importantly, ψ is not the allocation produced by ALG: in order to maintain EF1, ALG may assign an interval to a different machine even when $\psi(I)$ is available. The role of ψ is instead to provide a fixed reference schedule against which we compare the decisions of ALG. The following lemma shows that such a mapping can be chosen so that $\psi(I)$ is always available for I at its start time, while intervals mapped to the same machine form a conflict-free set.

Lemma 7.9. *Given an instance $\mathcal{I}$, there exists a mapping $\psi : \mathsf{OPT}(\mathcal{I}) \to \mathcal{A}$ from the set of intervals accepted by OPT to the machines such that*

*(1) (ψ **is conflict-free**) all intervals that are mapped to the same machine by ψ are mutually non-overlapping, i.e., for any machine $\ell \in \mathcal{A}$, the set of intervals $\{I_j \in \mathsf{OPT}(\mathcal{I}) : \psi(I_j) = \ell\}$ is conflict-free, and*

*(2) (ψ **is consistent with** ALG) any interval $I_i = (s_i, e_i) \in \mathsf{OPT}(\mathcal{I})$ does not conflict with the set of intervals assigned by ALG to the machine $\psi(I_i)$ until time s_i. Therefore, assigning I_i to $\psi(I_i)$ does not create any conflict.*

Proof. We proceed by induction on the intervals of $\mathcal{E}(\mathcal{I})$ in nondecreasing order of start time. For the first interval, choose any machine and define ψ accordingly; both properties hold trivially. Suppose that ψ has already been defined for all intervals starting before s_i, and that the two stated properties hold for those intervals. Let $I_i = (s_i, e_i)$ be the next interval to be mapped.

Let C_i be the set of intervals of $\mathcal{E}(\mathcal{I})$ that contain s_i and have already been processed. Since $\mathcal{E}(\mathcal{I})$ is feasible on m machines, at most $m-1$ other intervals can contain s_i, so $|C_i| \leq m-1$. By property (1), the intervals in C_i are mapped to distinct machines.

We now claim that every machine that is busy in ALG at time s_i is already mapped to by some interval in C_i. Indeed, let μ be a busy machine at time s_i, and let J be the earliest-start interval in C_i that ALG assigns to μ. We claim that $\psi(J) = \mu$. Suppose otherwise. Since J is assigned by ALG to machine μ but $\psi(J) \neq \mu$, the definition of ψ implies that when J arrived, machine μ was already occupied by an interval J' with an earlier start time and a later end time, and ALG assigned J to μ by executing step (3) and revoking J'. Since $e_{J'} > e_J$ and J contains s_i, the interval J' also contains s_i. Hence $J' \in C_i$, contradicting the choice of J as the earliest-start interval in C_i assigned to μ.

Therefore every busy machine is already used by an interval in C_i. Since the m machines are partitioned into busy machines and available machines, and C_i uses at most $m-1$ distinct machines, there exists a machine $\mu^\star$ that is neither busy in ALG at time s_i nor mapped to by any interval in C_i. We define

$$\psi(I_i) := \mu^\star.$$

By construction, $\mu^\star$ is available at time s_i, proving property (2). Moreover, no interval in C_i is mapped to $\mu^\star$, so I_i does not overlap any interval already mapped to $\mu^\star$, and property (1) is preserved. This completes the induction. □

The auxiliary graph G_k of block B_k. Fix a block $B_k = [t_k, t_{k+1})$. We associate with B_k a directed graph G_k, which we call the *auxiliary* graph, whose vertices are the machines, and whose edges represent "lost intervals", defined as follows: Consider a rejection event s_i in the block B_k caused by an arriving interval $I_i = (s_i, e_i)$. By part (2) of Lemma 7.9, the machine $\psi(I_i)$ is feasible for I_i with respect to the schedule constructed by ALG until time s_i. We call the machine $\psi(I_i)$ the *provider* of the event s_i.

If the event occurs in step (3), then ALG revokes some interval J from a machine r in order to accept I_i. In this case, we call machine r the *receiver* of the event, and we call interval J its *lost interval*. Otherwise, if the event occurs in step (4), then ALG rejects I_i. In this case, we define the receiver to be a dummy node $\perp$, and we take the lost interval to be I_i itself.

Let $\mathcal{E}[t_0 : t_{k+1})$ denote the set of intervals accepted and not yet revoked by the (possibly unfair) Greedy-Optimal algorithm $\mathcal{E}$ until the end of block B_k. Consider any interval $I_i = (s_i, e_i) \in \mathcal{E}[t_0 : t_{k+1})$ such that $s_i \in B_k$. If s_i is a rejection event (i.e., ALG either revokes an interval to accommodate I_i in step (3) or rejects I_i in step (4)), we add a directed edge in the graph G_k from the provider node $\psi(I_i)$ to the receiver node.

Finally, let

$$R_k := |E(G_k)|$$

denote the number of edges in the graph G_k.

Our goal is to bound the number of edges R_k. Towards this, we will first show that G_k is a *bipartite* graph such that all provider nodes are contained in the left part and all receiver nodes are contained in the right part. (Thus, within a block, a machine can either be a provider or a receiver or none.) Then, we will use a *charging* argument that charges each edge in G_k to a distinct vertex (or machine). Using this argument, we will show that the number of edges in G_k is at most $m-1$, where m is the number of machines. Subsequently, we will show how this bound on the number of edges provides the desired bound on the competitive ratio of ALG.

Lemma 7.10. *At the end of block B_k (i.e., just before time t_{k+1}), the sets of providers and receivers in the graph G_k are disjoint.*

Proof. Recall that block B_k lasts from $[t_k, t_{k+1})$. At time t_k, the sets of providers and receivers for the block B_k are empty. We will show that starting from time t_k and until the end of block B_k, once a machine becomes a provider (respectively, receiver) for block B_k, it cannot later become a receiver (respectively, provider) for the same block. This will suffice to prove the lemma.

Provider cannot become a receiver: Suppose that a machine p becomes a provider at some time during block B_k. Then, there exists a rejection event in B_k triggered by an interval $I_i = (s_i, e_i)$ such that

$$\psi(I_i) = p.$$

By Lemma 7.9, machine p is available at time s_i. We claim that p is heavy at time s_i. Indeed, if p were light, then step (2) would place I_i on p since it is available, contradicting that I_i gives rise to a rejection event. Thus, p is available and heavy at time s_i.

Recall from Observation 7.8 that the load of a machine is non-decreasing. Thus, machine p stays heavy until the end of block B_k. Since becoming a receiver requires a machine to be light, it follows that machine p cannot become a receiver during the remainder of the block B_k.

Receiver cannot become a provider: Now suppose a machine r becomes a receiver at some time s_i during block B_k. Then, ALG revokes an interval J from machine r in order to accept an interval $I_i = (s_i, e_i)$. Since r is chosen as a receiver in step (3), it must be light at time s_i.

We will assume that $\mathcal{E}$ will not revoke the interval J until the end of block B_k. Indeed, if J is revoked by $\mathcal{E}$, then, as discussed above, its corresponding edge does not exist in the auxiliary graph G_k since it only considers intervals $I_i = (s_i, e_i) \in \mathcal{E}[t_0 : t_{k+1})$.

Since step (3) chooses, among all light receivers (which are all busy), one whose leader interval has maximum end time, we have

$$e_{J_\ell} \leq e_J$$

for every light machine ℓ at time s_i, where J_ℓ denotes the leader interval currently assigned to ℓ.

Moreover, while a machine remains light, the end time of its leader interval can only decrease over time. This is because step (1) replaces intervals only with earlier-ending ones, and step (3) also replaces an interval with one that has a smaller end time.

Now suppose for contradiction that later in the same block, machine r becomes the provider of some interval $I_{i'} = (s_{i'}, e_{i'})$ with $s_{i'} > s_i$. Then

$$\psi(I_{i'}) = r.$$

If $\psi(J) \neq r$, then by a argument similar to that in the proof of Lemma 7.9, there exists an interval J' such that $J \subseteq J'$ (i.e., the interval J starts after and ends before the interval J') and $\psi(J') = r$. By Lemma 7.9, since we have $\psi(I_{i'}) = \psi(J') = r$, $I_{i'}$ should not overlap with J' and therefore J, hence we have

$$s_{i'} > e_J.$$

Now we show that all light machines are available at time $s_{i'}$. Let ℓ be any machine that is light at time $s_{i'}$. By observation 7.8, loads are nondecreasing, and therefore a light machine at time $s_{i'}$ is also light at any earlier time point in this block; specifically, it is light at time s_i. Hence, the end time of the leader interval of machine ℓ should be at most e_J. As long as a machine remains light, the end time of its leader cannot increase. Therefore, at time $s_{i'}$, the leader interval on machine ℓ has end time at most e_J. Since $s_{i'} > e_J$, ℓ is available at time $s_{i'}$ for the interval $I_{i'}$.

At time $s_{i'}$, there must exist a light machine since we are still inside the block B_k. By the above reasoning, any such machine must be available. Therefore, $I_{i'}$ cannot give rise to a rejection event, which is a contradiction. Thus, a receiver in G_k cannot later become a provider in the same block.

We conclude that the sets of providers and receivers in G_k are disjoint. □

Charging scheme. Recall that each edge of the auxiliary graph G_k is directed from its provider to its receiver (possibly the dummy node $\perp$). We charge each edge of G_k either to its receiver or to its provider, according to the following rule.

> Each receiver is charged the *last* incoming edge it receives in G_k. Every other edge (an edge entering the dummy node $\perp$, or an edge entering a receiver that later receives another incoming edge) is charged to its provider.

By construction, every edge is charged to exactly one node, and every receiver is charged at most once. The main difficulty is to show that each provider is charged at most once. To this end, we establish three structural properties of G_k (Claims 7.11, 7.12 and 7.14), which together imply the key fact behind the charging scheme: every edge charged to a provider is the *last* edge leaving that provider (Corollary 7.13 and Claim 7.14).

Each edge of G_k is associated with a rejection event, and hence with the start time of some interval; we call this time the *creation time* of the edge, and we order the edges of G_k by increasing creation time. Fix a provider p and consider the edges leaving p in this order. We first show that whenever p creates a new outgoing edge, its previous receiver must already be heavy; since step (3) selects only light machines as receivers, that machine can never receive an edge again.

Claim 7.11 (A new outgoing edge kills the previous receiver)**.** *Fix a provider p in G_k, and let*

$$p \to r_1,\ p \to r_2,\ \ldots,\ p \to r_{q-1},\ p \to r_q$$

be the edges leaving p, listed in order of creation time, where each r_j is a receiver machine and r_q is either a receiver machine or the dummy node $\perp$. Then, for every $j < q$, the machine r_j is heavy when the edge $p \to r_{j+1}$ is created, and no edge created afterwards can enter r_j.

Proof. Fix $j < q$. Let the edge $p \to r_j$ be created by a step (3) event triggered by an interval I_j, which ALG places on machine r_j, and let the edge $p \to r_{j+1}$ be created by a rejection event triggered by an interval I_{j+1}. Since both edges leave p, we have $\psi(I_j) = \psi(I_{j+1}) = p$, so I_j and I_{j+1} are non-overlapping by property (1) of Lemma 7.9; hence $s_{j+1} > e_{I_j}$. Suppose that r_j is light at time s_{j+1}. While a machine remains light, the end time of its leader can only decrease, and I_j is the leader of r_j at time s_j; hence the leader of r_j at time s_{j+1} ends before s_{j+1}, so r_j is available. But then step (2) would place I_{j+1} on r_j, contradicting that s_{j+1} is a rejection event. Thus r_j is heavy at time s_{j+1}. Since loads never decrease within a block, r_j remains heavy for the rest of the block, and since step (3) selects only light machines as receivers, no edge created after $p \to r_{j+1}$ can enter r_j. □

The next claim shows that when a receiver of p acquires a second incoming edge, which by Claim 7.11, can only happen before the next edge leaving p is created, the provider p is *killed*: it cannot create any further edge in the block.

Claim 7.12 (A new incoming edge kills the previous provider)**.** *Let r be a receiver in G_k, and let*

$$p_1 \to r \qquad \text{and} \qquad p_2 \to r$$

be two edges in G_k such that the edge from p_1 is created before the edge from p_2, and let s_2 be the creation time of $p_2 \to r$. Then p_1 cannot be the provider of any edge created after time s_2.

Proof. Let the edge $p_1 \to r$ be created at time s_1 by a step (3) event triggered by an interval I_1, so that $\psi(I_1) = p_1$ and ALG places I_1 on machine r, and let the edge $p_2 \to r$ be created at time s_2 by a step (3) event triggered by an interval I_2.

Since r is chosen as a receiver at both times, it is light at times s_1 and s_2, and hence, by Observation 7.8, throughout $[s_1, s_2]$. Interval I_1 is the leader of r at time s_1, and while a machine remains light, the end time of its leader can only decrease; hence the leader J of r at time s_2 satisfies $e_J \le e_{I_1}$. Moreover, since s_2 is a rejection event, every light machine is busy at time s_2 (otherwise step (2) would place I_2 on an available light machine), and step (3) selects a light machine whose leader has maximum end time; since it selects r, we obtain

$$e_{J_\ell} \;\le\; e_J \;\le\; e_{I_1}$$

for every machine ℓ that is light at time s_2, where J_ℓ denotes the leader of ℓ at time s_2.

Suppose, towards a contradiction, that p_1 is the provider of an edge of G_k created at some time $s_I > s_2$, triggered by an interval I with $\psi(I) = p_1$. Since $\psi(I) = \psi(I_1)$, property (1) of Lemma 7.9 implies that I and I_1 are non-overlapping, and hence

$$s_I > e_{I_1}.$$

By the same argument as in the second part of the proof of Lemma 7.10, every machine that is light at time s_I was already light at time s_2, and its leader at time s_I has end time at most $e_{I_1} < s_I$; hence every light machine is available for I at time s_I. Since $s_I < t_{k+1}$, at least one machine is light at time s_I, so s_I cannot be a rejection event, a contradiction. Therefore p_1 cannot be the provider of any edge created after time s_2. □

Combining the two claims yields the key structural property behind the charging scheme.

Corollary 7.13. *If an edge $p \to r$ of G_k is not the last incoming edge of its receiver r, then it is the last outgoing edge of its provider p.*

Proof. Let e' be an incoming edge of r created after $p \to r$, say at time s'. Suppose, towards a contradiction, that p creates an edge after $p \to r$, and let $p \to r''$ be the first such edge. By Claim 7.11, no edge can enter r after $p \to r''$ is created; hence e' is created before $p \to r''$. But then Claim 7.12 implies that p cannot be the provider of any edge created after time s', contradicting the existence of $p \to r''$. Hence $p \to r$ is the last edge leaving p. □

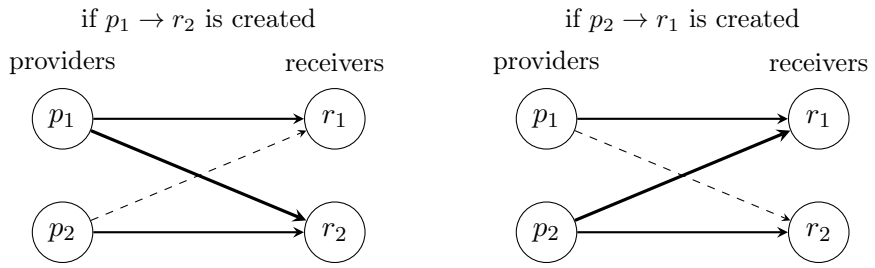


Figure 7.3: Visual illustration of Claims 7.11 and 7.12. Starting from edges $p_1 \to r_1$ and $p_2 \to r_2$, once one crossed edge is created, the other crossed edge cannot occur later. Indeed, a new outgoing edge kills the previous receiver, while a new incoming edge kills the previous provider. Dashed edges indicate configurations that cannot arise.

The previous claim handles the case in which a provider is killed by a later incoming edge at one of its receivers; see Figure 7.3 for an illustration of this interaction together with Claim 7.11. We now consider the remaining case, in which a provider creates an edge to the dummy node $\perp$, and show that this event also kills the provider; in particular, an edge entering $\perp$ is likewise the last edge leaving its provider.

Claim 7.14 (An incoming edge to $\perp$ kills its provider)**.** *Let $p \to \perp$ be an edge in G_k, and let s be its creation time. Then p cannot be the provider of any edge created after time s.*

Proof. Let the edge $p \to \perp$ be created by a step (4) event triggered by an interval $I_i = (s_i, e_i)$, so that $\psi(I_i) = p$ and $s_i = s$. Since step (4) is executed, steps (2) and (3) fail. In particular, every machine ℓ that is light at time s_i is busy, and its leader J_ℓ satisfies

$$e_{J_\ell} \leq e_i.$$

Suppose, towards a contradiction, that p is the provider of an edge of G_k created at some time $s_I > s_i$, triggered by an interval I with $\psi(I) = p$. Since the edge $p \to \perp$ belongs to G_k, we have $I_i \in \mathcal{E}[t_0 : t_{k+1})$; as $\psi(I) = \psi(I_i)$, property (1) of Lemma 7.9 yields

$$s_I > e_i.$$

The contradiction now follows exactly as in the proof of Claim 7.12, with e_i in place of e_{I_1}: every machine that is light at time s_I was already light at time s_i, and its leader at time s_I has end time at most $e_i < s_I$, so every light machine is available for I; since $s_I < t_{k+1}$, at least one light machine exists, and s_I cannot be a rejection event. □

The previous claims establish the structural properties of G_k needed for the charging argument; see Figure 7.4 for an illustration. In particular, every edge charged to a provider—an edge entering $\perp$, or an edge that is not the last incoming edge of its receiver—is the last edge leaving that provider, by Claim 7.14 and Corollary 7.13, respectively. We can now show that each machine is charged at most once.

Lemma 7.15. *Each machine is charged at most once.*

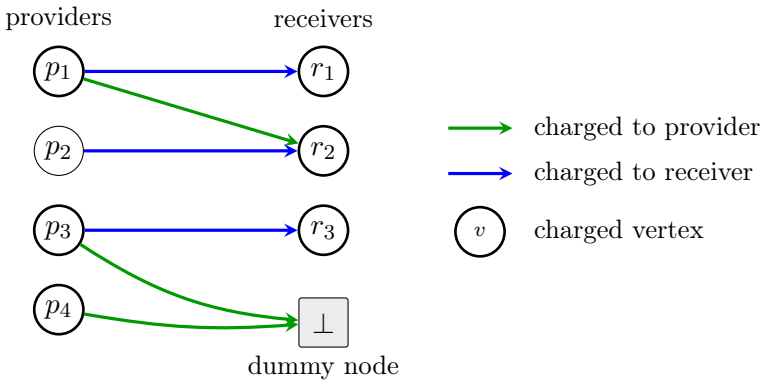


Figure 7.4: Illustration of the charging scheme on the bipartite graph G_k for a block B_k. Providers appear on the left, receivers on the right, and the dummy node $\perp$ represents step (4) edges. The last incoming edge of each receiver is charged to that receiver (blue). All remaining edges (earlier incoming edges and edges entering $\perp$) are charged to their providers (green); each such edge is the last edge leaving its provider. Here, $p_1 \to r_2$ is created before $p_2 \to r_2$; note that machine p_2 is uncharged.

Proof. By the charging rule, a receiver is charged only its last incoming edge, and is therefore charged at most once. Now consider a provider p. An edge leaving p is charged to p in exactly two cases: it enters the dummy node $\perp$, or it enters a receiver r and is not the last incoming edge of r. In the first case, the edge is the last edge leaving p by Claim 7.14; in the second case, it is the last edge leaving p by Corollary 7.13. Since at most one edge leaving p can be its last, at most one edge is charged to p. Finally, by Lemma 7.10, no machine is both a provider and a receiver in G_k, so no machine can be charged once in each role. Therefore each machine is charged at most once. □

Lemma 7.16. *At least one machine is uncharged.*

Proof. If some machine is neither a provider nor a receiver in G_k, then it is never charged, and we are done. We may therefore assume that every machine is a provider or a receiver; by Lemma 7.10, these two sets partition the machines. Suppose, towards a contradiction, that every machine is charged. Let s be the creation time of the last edge of G_k, and note that $s < t_{k+1}$, since all edges of G_k are created inside the block $B_k = [t_k, t_{k+1})$.

We show that every machine is heavy at time s, which yields the desired contradiction. As shown in the proof of Lemma 7.10, every provider is heavy from the moment it first becomes a provider; since loads never decrease within a block, all providers are heavy at time s. Now consider any receiver r, and let $q \to r$ be its last incoming edge, i.e., the edge charged to r. We claim that $q \to r$ is not the last edge leaving q. Indeed, only the last edge leaving a provider can be charged to it (by Claim 7.14 and Corollary 7.13); since $q \to r$ is charged to r, if it were the last edge leaving q, then no edge would be charged to q, contradicting our assumption. Hence q creates a further edge after $q \to r$, and no later than time s. By Claim 7.11, machine r is heavy from the creation of that edge onward, in particular, at time s.

Thus every machine is heavy at time s, i.e., every machine has load $k+1$, so $t_{k+1} \le s$, contradicting $s < t_{k+1}$. For the final block B_ω, the same argument shows that every

machine would have load $\omega + 1$ at time s, so $t_{\omega+1}$ would be finite, contradicting the maximality of ω. □

Claim 7.17. *For every $0 \le k \le \omega$, we have $R_k \le m - 1$.*

Proof. By the charging scheme, every edge of G_k is charged to some machine. By Lemma 7.15, each machine is charged at most once, and by Lemma 7.16, at least one machine is uncharged. Since there are m machines,

$$R_k = |E(G_k)| \le m - 1.$$ □

Recall that $A_k = m$ for every $0 \le k < \omega$, and that $R_0 = 0$ since block B_0 contains no rejection events. Moreover, since the total load of ALG increases only at acceptance events, we have $|\mathsf{ALG}(\mathcal{I})| = \sum_{k=0}^{\omega} A_k$. The next lemma relates $|\mathcal{E}(\mathcal{I})|$ to the acceptance events and the edges of the auxiliary graphs. The key observation is that if ALG loses an interval that $\mathcal{E}$ keeps until the end, then the trigger of the corresponding rejection event is also kept by $\mathcal{E}$ until the end; hence every such loss is witnessed by an edge.

Lemma 7.18. $|\mathcal{E}(\mathcal{I})| \le \sum_{k=0}^{\omega} A_k + \sum_{k=0}^{\omega} R_k.$

Proof. We exhibit an injection from $\mathcal{E}(\mathcal{I})$ into the union of the acceptance events and the edges of the graphs $G_0, \dots, G_\omega$; since the acceptance events number $\sum_k A_k$ and the edges number $\sum_k R_k$, this proves the lemma. Let $I \in \mathcal{E}(\mathcal{I})$, so that $\mathcal{E}$ accepts I upon arrival and never revokes it; in particular, ALG neither rejects nor revokes I in step (1).

If $I \in \mathsf{ALG}(\mathcal{I})$, we map I to the acceptance event that opened the *slot* that I finally occupies: each step (2) event opens a new slot, steps (1) and (3) replace the interval currently occupying a slot, and distinct intervals of $\mathsf{ALG}(\mathcal{I})$ occupy distinct slots.

If ALG rejects I upon arrival in step (4), then s_I is a rejection event whose trigger is I itself. Since $I \in \mathcal{E}(\mathcal{I}) \subseteq \mathcal{E}[t_0 : t_{k+1})$ for the block B_k containing s_I, this event creates an edge, and we map I to it.

Otherwise, ALG accepts I and later revokes it in a step (3) event triggered by an interval $T = (s_T, e_T)$ with $e_T < e_I$. We claim that $\mathcal{E}$ never revokes T; then $T \in \mathcal{E}(\mathcal{I}) \subseteq \mathcal{E}[t_0 : t_{k+1})$ for the block containing s_T, the event creates an edge, and we map I to it. Suppose, towards a contradiction, that $\mathcal{E}$ revokes T at the arrival of some interval at time s'. Since $\mathcal{E}$ revokes only a leader that conflicts with the arriving interval, we have $s' < e_T < e_I$. At time s', interval I is scheduled by $\mathcal{E}$ on some machine (note that $s_I < s_T < s'$). Every other interval on that machine starts before $s' < e_I$ and is disjoint from I, and hence ends before s_I; thus I is the leader of that machine, with end time e_I. Since $\mathcal{E}$ revokes the leader with the *largest* end time and $e_I > e_T$, the revoked interval is not T, a contradiction.

The map is injective: distinct intervals of $\mathsf{ALG}(\mathcal{I})$ correspond to distinct acceptance events, and each rejection event has a single lost interval, so distinct intervals of $\mathcal{E}(\mathcal{I}) \setminus \mathsf{ALG}(\mathcal{I})$ are mapped to distinct edges. □

By Claim 7.17 and Lemma 7.18, we can bound the competitive ratio as follows:

$$\begin{aligned}\frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} = \frac{|\mathcal{E}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} &\le \frac{\sum_{k=0}^{\omega} A_k + \sum_{k=0}^{\omega} R_k}{\sum_{k=0}^{\omega} A_k} \\ &= 1 + \frac{R_0 + \sum_{k=1}^{\omega} R_k}{\sum_{k=0}^{\omega-1} A_k + A_\omega} \\ &\le 1 + \frac{\omega(m-1)}{\omega m + A_\omega} \;\le\; 1 + \frac{\omega(m-1)}{\omega m} = 2 - \frac{1}{m}.\end{aligned}$$

This completes the proof. □

We next show that this competitive ratio is optimal by proving a matching lower bound.

7.4.2 Lower Bounds

We now show that the competitive ratio achieved by Greedy-Balanced is optimal among deterministic online algorithms that satisfy EF1. Specifically, we prove that no deterministic EF1 online algorithm can achieve a competitive ratio strictly smaller than $2 - \frac{1}{m}$ with respect to the offline optimum without fairness. This lower bound shows that the additional efficiency loss observed in the online setting is unavoidable and arises from the interplay between online uncertainty and the fairness constraint. It specifically applies to deterministic algorithms, and does not rule out the possibility of improved guarantees using randomization. Designing randomized online algorithms with better competitive ratios is an interesting direction for future work.

Theorem 7.19 (Online EF1 Lower Bound)**.** *In the* unweighted online *setting with m identical machines, every deterministic online algorithm that satisfies EF1 has competitive ratio at least $2 - \frac{1}{m}$.*

Proof. Fix any deterministic online EF1 algorithm ALG. We construct an adversarial input instance $\mathcal{I}$ in phases. In this instance, all intervals have unit weight.

EF1 as a count-balance invariant. In the unweighted setting, EF1 is equivalent to the condition that the number of accepted intervals per machine differs by at most 1 at all times. Formally, letting $C_i(t)$ be the number of intervals currently assigned to machine i right after time t, EF1 implies

$$\max_{i \in \mathcal{A}} C_i(t) - \min_{i \in \mathcal{A}} C_i(t) \le 1 \qquad \text{for all times } t. \tag{7.2}$$

Phase 0 (initial imbalance). We first feed ALG a sequence of disjoint unit-length intervals

$$J_q = [q, q+1) \qquad (q = 0, 1, 2, \ldots),$$

and stop as soon as ALG has accepted $m - 1$ of them. (If ALG accepts fewer than $m - 1$ of these disjoint intervals forever, then $|\mathsf{ALG}(\mathcal{I})|$ is finite while $|\mathsf{OPT}(\mathcal{I})|$ can be made arbitrarily large by continuing to release disjoint intervals, so the competitive ratio is unbounded. Hence we may assume we eventually stop.)

At the stopping time, ALG has accepted exactly $m-1$ intervals in total, so by (7.2) there must be a machine with count 0 and the remaining $m-1$ machines have count 1. Relabel machines so that machine 1 is *light* (count 0), and machines $2, \dots, m$ are *heavy* (count 1).

Recurring blocks. We now append an arbitrary number B of *recurring blocks*. Each block starts at some time T strictly after the end of all previously released intervals (so blocks do not overlap in time). Within a block, we release two batches of intervals.

Batch I (adaptive nested intervals). Initialize $b_1 = T+1$ and $c_1 = T+2$. For $i = 1, 2, \dots, m$, release the interval

$$P_i = \Big[T,\ \frac{b_i + c_i}{2}\Big).$$

After observing ALG's (deterministic) placement decision for P_i, update

$$(b_{i+1}, c_{i+1}) \ = \ \begin{cases} (b_i,\ \frac{b_i+c_i}{2}) & \text{if ALG assigns } P_i \text{ to machine 1,} \\ (\frac{b_i+c_i}{2},\ c_i) & \text{otherwise.} \end{cases}$$

Let

$$x \ := \ \frac{b_{m+1} + c_{m+1}}{2}.$$

By construction of the updates, we have the following separation property:

$$\begin{aligned} &\text{if } P_i \text{ is assigned to machine 1, then} \quad e_{P_i} \ \geq \ x, \\ &\text{if } P_i \text{ is assigned to some machine } k \neq 1, \text{ then} \quad e_{P_i} \ \leq \ x. \end{aligned} \tag{7.3}$$

In particular, letting $I^{(1)}$ denote the (unique) interval on machine 1 at the end of Batch I,

$$E_1 \ := \ e_{I^{(1)}} \ \geq \ x, \tag{7.4}$$

whereas for each $k \in \{2, \dots, m\}$, if machine k received an interval $I^{(k)}$ in Batch I then

$$E_k \ := \ e_{I^{(k)}} \ \leq \ x, \tag{7.5}$$

and if machine k received no interval in Batch I we set $E_k := T$.

Batch II (intervals feasible on machines $2, \dots, m$ but conflicting with machine 1). For each machine $k \in \{2, \dots, m\}$, release an interval

$$Q_k \ = \ (E_k, E_1),$$

Release the Q_k's in nondecreasing order of their start times (i.e., sorted by E_k), so the overall arrival order remains nondecreasing in start times. By definition, each Q_k is feasible on machine k (it starts at E_k, immediately after the Batch I interval on that machine, if any), and it is *infeasible* on machine 1 because Q_k overlaps $I^{(1)} = [T, E_1)$ (since $E_k < E_1$ and Q_k starts at E_k).

Claim 7.20. *In any recurring block,* ALG *accepts at most m intervals.*

Proof. All intervals in Batch I share the same start time T, hence they mutually overlap, so feasibility allows ALG to accept at most one P_i per machine. Therefore ALG accepts at most m intervals from Batch I.

We now argue that ALG cannot accept any interval from Batch II. Every Q_k conflicts with machine 1, hence any accepted Q_k must be placed on some machine in $\{2, \ldots, m\}$, increasing that machine's count while leaving machine 1's count unchanged. At the start of every block, machine 1 is light and machines $2, \ldots, m$ are heavy (from Phase 0 and previous blocks). By (7.2), ALG must first "catch up" machine 1 before it can increase any heavy machine further. Thus, if ALG accepts m intervals in Batch I (filling all machines once), thereafter, accepting any Q_k would immediately violate (7.2) because it would make some machine in $\{2, \ldots, m\}$ exceed machine 1 by 2 again while machine 1 cannot accept any Q_k. Consequently, in this case ALG accepts zero intervals from Batch II.

If ALG rejects some Batch I interval, then it has accepted fewer than m intervals in the block already, which can only decrease $|\mathsf{ALG}(\mathcal{I})|$ and therefore can only worsen the eventual competitive ratio lower bound. Hence, in all cases, ALG accepts at most m intervals per block. □

Claim 7.21. *For each recurring block, the offline optimum without fairness accepts* $2m-1$ *intervals.*

Proof. In Batch I, OPT can accept m intervals by placing one P_i on each machine (they overlap in time but machines are distinct). In Batch II, OPT can accept all $m-1$ intervals $Q_2, \ldots, Q_m$ by placing Q_k on machine k for each $k \in \{2, \ldots, m\}$: by construction, Q_k starts at E_k, i.e., after the end of the Batch I interval on machine k (or at time T if no Batch I interval was placed there), so there is no conflict on machine k. Thus OPT accepts $m + (m-1) = 2m-1$ intervals in the block. □

Competitive ratio lower bound. Let the number of recurring blocks B grow. By Claim 7.20, ALG accepts at most m intervals per block, while by Claim 7.21, OPT accepts $2m-1$ per block. The additive contribution of Phase 0 is $m-1$ for both ALG and OPT and becomes negligible as $B \to \infty$. Therefore,

$$\sup_{\mathcal{I}} \frac{|\mathsf{OPT}(\mathcal{I})|}{|\mathsf{ALG}(\mathcal{I})|} \geq \lim_{B\to\infty} \frac{(m-1) + B(2m-1)}{(m-1) + Bm} = \frac{2m-1}{m} = 2 - \frac{1}{m}.$$

This completes the proof. □

.

7.5 Experimental Evaluation

We complement our theoretical analysis with an empirical evaluation of the Greedy-Balanced algorithm using real-world benchmark instances. Our theoretical guarantees focus on worst-case scenarios and are adversarial in nature, which means they may not necessarily reflect typical performance. In this section, we will examine how the algorithm behaves on realistic inputs and compare its empirical performance against the worst-case bounds we established earlier.

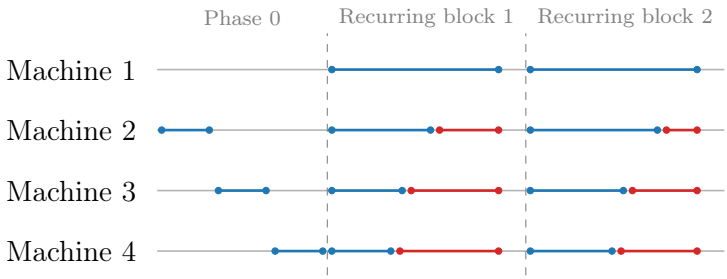
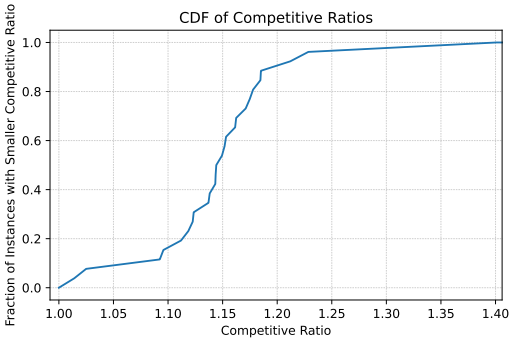


Figure 7.5: Illustration of the adversarial construction for Theorem 7.19 for $m = 4$. Phase 0 creates an initial EF1 imbalance with one light machine. In each recurring block, Batch I (blue) places one interval on each machine, with the interval on machine 1 ending latest. Batch II then releases $m-1$ intervals (red) that fit on machines 2–4 but overlap the interval on machine 1; EF1 prevents ALG from assigning them to the heavy machines while machine 1 receives none.

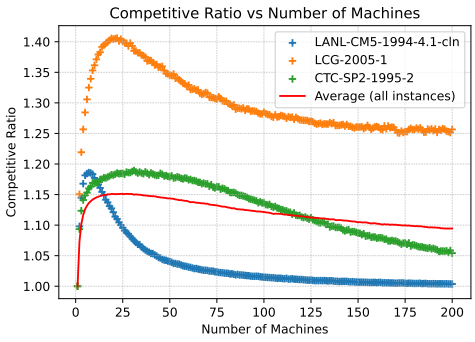
Dataset. We evaluate Greedy-Balanced using benchmark instances derived from scheduling traces of computing clusters from the Parallel Workloads Archive (Chapin, Cirne, Feitelson, J. P. Jones, Leutenegger, Schwiegelshohn, Smith, and Talby, 1999; Feitelson, 2005; Feitelson, Tsafrir, and Krakov, 2014). This dataset has been used in previous experimental studies on interval scheduling algorithms (Antoniadis, Shahheidar, Shahkarami, and Soltani, 2026; Boyar, Favrholdt, Kamali, and Larsen, 2023). The traces record jobs scheduled on parallel processors, with each job specified by a start time, an end time, and a number of requested processors. For our analysis, we treat each job as requiring a single machine and disregard the multiplicity of processors mentioned in the original logs. We vary the number of machines, running each instance for $m = 1, 2, \ldots, 300$. Instances with more than 10^6 jobs are excluded from consideration. This results in a total of 33 instances, ranging from 18238 to 728871 jobs. All experiments were conducted on standard CPU-based workstation. Since our evaluation prioritizes solution quality instead of runtime, hardware specifications do not affect the results.

Results. As shown in fig. 7.6(a), Greedy-Balanced consistently outperforms its worst-case theoretical guarantees in practice. The maximum competitive ratio observed across all instances and machine counts is 1.306, which occurs for the instance `UniLu-Gaia-2014-2` with $m = 49$ machines. The average of the maximum competitive ratio for each instance across all machine counts is 1.235. Overall, these results suggest that while fairness constraints impose an inherent efficiency loss in the worst case, their practical impact on typical instances is significantly smaller.

An interesting observation is that as the number of machines increases, the competitive ratio initially grows and then starts to decline, as shown in Figure 7.6(b). Intuitively, adding more machines introduces additional fairness constraints, which can temporarily worsen the performance. However, having more machines also provides greater flexibility in scheduling, as more machines are idle at any given time, ultimately leading to improved performance.



(a) The distribution of competitive ratios across all instances for $m = 25$, shown as the fraction of instances with competitive ratio at most a given value. In total 33 instances were evaluated.



(b) Average competitive ratio of *Greedy-Balanced* over all instances as well as the ratio for three instances chosen to illustrate the variability across the dataset, as a function of the number of machines.

7.6 Conclusion

Our work advocates a particular way of incorporating fairness into optimization problems, namely by treating fairness notions as explicit constraints rather than as objectives to be optimized. This viewpoint leads to a different class of algorithmic questions: rather than asking whether a fair solution exists, we ask how much efficiency must be sacrificed to guarantee fairness, and whether this loss can be bounded by a constant. Our results show that, at least for interval scheduling, fairness can be enforced at a controlled and well-characterized cost, and that this cost persists even in structured settings. More broadly, we believe that this perspective provides a systematic way of integrating fairness considerations into scheduling, resource allocation, and other optimization problems.

Another insight emerging from our study is the importance of separating different sources of inefficiency. In the offline setting, we isolate the efficiency loss intrinsic to enforcing fairness itself, while in the online setting additional loss arises from uncertainty and irrevocable decisions. By comparing tight bounds across these settings, we distinguish which limitations stem from fairness and which are consequences of operating online. At the same time, our work leaves several natural directions open. We treat fairness as a hard constraint, whereas some applications may call for softer notions that allow controlled fairness violations in exchange for improved efficiency. We also focus on EF1, which coincides with EFX in our unweighted setting, while other fairness notions may exhibit qualitatively different algorithmic behavior.

Finally, our online results focus on the unweighted setting, since the weighted version of the general problem does not admit a constant competitive ratio. Nevertheless, weighted variants under additional structure, such as unit-length intervals with arbitrary weights, remain an interesting direction for future work. More generally, extending the fairness-as-constraint paradigm to other scheduling and online optimization problems may lead to a deeper understanding of the relationship between fairness, efficiency, and uncertainty. Moreover, our experimental evaluation suggests that the extremal instances driving the

lower bounds may be rare in practice, indicating that fair algorithms could perform substantially better than their worst-case guarantees suggest.

Bibliography

Abousamra, A., D. P. Bunde, and K. Pruhs (2013). "An Experimental Comparison of Speed Scaling Algorithms with Deadline Feasibility Constraints." In: *arXiv preprint arXiv:1307.0531*.

Agrawal, P., E. Balkanski, V. Gkatzelis, T. Ou, and X. Tan (2022). "Learning-Augmented Mechanism Design: Leveraging Predictions for Facility Location." In: *EC*.

Albers, S. (2010). "Energy-Efficient Algorithms." In: *Commun. ACM* 53.5, pp. 86–96.

Albers, S., A. Antoniadis, and G. Greiner (2015). "On Multi-Processor Speed Scaling with Migration." In: *J. Comput. Syst. Sci.* 81.7, pp. 1194–1209.

Albers, S. and H. Fujiwara (2007). "Energy-Efficient Algorithms for Flow Time Minimization." In: *ACM Trans. Algorithms* 3.4, p. 49.

Aleksandrov, M., H. Aziz, S. Gaspers, and T. Walsh (2015). "Online Fair Division: Analysing a Food Bank Problem." In: *IJCAI*, pp. 2540–2546.

Aleksandrov, M. and T. Walsh (2019). "Strategy-Proofness, Envy-Freeness and Pareto Efficiency in Online Fair Division with Additive Utilities." In: *PRICAI*. Springer, pp. 527–541.

Aleksandrov, M. and T. Walsh (2020). "Online Fair Division: A Survey." In: *AAAI*, pp. 13557–13562.

Almanza, M., F. Chierichetti, S. Lattanzi, A. Panconesi, and G. Re (2021). "Online Facility Location with Multiple Advice." In: *NeurIPS*.

Alon, N., M. Feldman, A. D. Procaccia, and M. Tennenholtz (2010). "Strategyproof Approximation of the Minimax on Networks." In: *Math. Oper. Res.* 35.3, pp. 513–526.

Alon, N., F. Fischer, A. D. Procaccia, and M. Tennenholtz (2011). "Sum of Us: Strategyproof Selection from the Selectors." In: *TARK*, pp. 101–110.

Angel, E., E. Bampis, F. Kacem, and D. Letsios (2019). "Speed Scaling on Parallel Processors with Migration." In: *J. Comb. Optim.* 37.4, pp. 1266–1282.

Anshelevich, E., O. Bhardwaj, E. Elkind, J. Postl, and P. Skowron (2018). "Approximating Optimal Social Choice Under Metric Preferences." In: *Artif. Intell.* 264, pp. 27–51.

Anshelevich, E., A. Filos-Ratsikas, N. Shah, and A. A. Voudouris (2021). "Distortion in Social Choice Problems: The First 15 Years and Beyond." In: *IJCAI*.

Anshelevich, E. and J. Postl (2017). "Randomized Social Choice Functions Under Metric Preferences." In: *J. Artif. Intell. Res.* 58, pp. 797–827.

Anshelevich, E. and W. Zhu (2021). "Ordinal Approximation for Social Choice, Matching, and Facility Location Problems Given Candidate Positions." In: *ACM Trans. Economics and Comput.* 9.2.

Antoniadis, A., C. Coester, M. Eliáš, A. Polak, and B. Simon (2023). "Online Metric Algorithms with Untrusted Predictions." In: *ACM Trans. Algorithms* 19.2, 19:1–19:34.

Antoniadis, A., T. Gouleakis, P. Kleer, and P. Kolev (2020). "Secretary and Online Matching Problems with Machine Learned Advice." In: *NeurIPS*, pp. 7933–7944.

Antoniadis, A., P. Jabbarzade, and G. Shahkarami (2024). *A Novel Prediction Setup for Online Speed-Scaling*. Available for download at `https://github.com/INFORMSJoC/2022.0387`.

Arkin, E. M. and E. B. Silverberg (1987). "Scheduling Jobs with Fixed Start and End Times." In: *Discret. Appl. Math.* 18.1, pp. 1–8.

Arrow, K. J. (1963). *Social Choice and Individual Values*. 2nd. Cowles Foundation.

Azar, Y., D. Panigrahi, and N. Touitou (2022). "Online Graph Algorithms with Predictions." In: *SODA*.

Aziz, H. (2020). "Justifications of Welfare Guarantees Under Normalized Utilities." In: *SIGecom Exch.* 17.2, pp. 71–75.

Balcan, M.-F., S. Prasad, and T. Sandholm (2023). "Bicriteria Multidimensional Mechanism Design with Side Information." In: *NeurIPS*.

Balkanski, E., V. Gkatzelis, and X. Tan (2023). "Strategyproof Scheduling with Predictions." In: *ITCS*.

Balkanski, E., V. Gkatzelis, X. Tan, and C. Zhu (2024). "Online Mechanism Design with Predictions." In: *EC*.

Balkanski, E., N. Perivier, C. Stein, and H.-T. Wei (2023). "Energy-Efficient Scheduling with Predictions." In: *NeurIPS*.

Bamas, É., A. Maggiori, L. Rohwedder, and O. Svensson (2020). "Learning-Augmented Energy Minimization via Speed Scaling." In: *NeurIPS*.

Bamas, É., A. Maggiori, and O. Svensson (2020). "The Primal-Dual Method for Learning-Augmented Algorithms." In: *NeurIPS*, pp. 20083–20094.

Banerjee, S., V. Gkatzelis, A. Gorokh, and B. Jin (2022). "Online Nash Social Welfare Maximization with Predictions." In: *SODA*.

Bansal, N., D. P. Bunde, H.-L. Chan, and K. Pruhs (2011). "Average Rate Speed Scaling." In: *Algorithmica* 60.4, pp. 877–889.

Bansal, N., H.-L. Chan, D. Katz, and K. Pruhs (2012). "Improved Bounds for Speed Scaling in Devices Obeying the Cube-Root Rule." In: *Theory Comput.* 8.1, pp. 209–229.

Bansal, N., C. Coester, R. Kumar, M. Purohit, and E. Vee (2020). "Scale-Free Allocation, Amortized Convexity, and Myopic Weighted Paging." In: *arXiv preprint arXiv:2011.09076*.

Bansal, N., T. Kimbrel, and K. Pruhs (2007). "Speed Scaling to Manage Energy and Temperature." In: *J. ACM* 54.1, 3:1–3:39.

Bansal, N., K. Pruhs, and C. Stein (2009). "Speed Scaling for Weighted Flow Time." In: *SIAM J. Comput.* 39.4, pp. 1294–1308.

Bar-Noy, A., R. Canetti, S. Kutten, Y. Mansour, and B. Schieber (1999). "Bandwidth Allocation with Preemption." In: *SIAM J. Comput.* 28.5, pp. 1806–1828.

Barak, Z., A. Gupta, and I. Talgam-Cohen (2024). "MAC Advice for Facility Location Mechanism Design." In: *NeurIPS*.

Barberà, S., F. Gul, and E. Stacchetti (1993). "Generalized Median Voter Schemes and Committees." In: *J. Econ. Theory* 61.2, pp. 262–289.

Beasley, J. E. and B. Cao (1998). "A Dynamic Programming Based Algorithm for the Crew Scheduling Problem." In: *Comput. Oper. Res.* 25.7-8, pp. 567–582.

Benade, G., A. M. Kazachkov, A. D. Procaccia, A. Psomas, and D. Zeng (2024). "Fair and Efficient Online Allocations." In: *Oper. Res.* 72.4, pp. 1438–1452.

Berg, M. de, O. Cheong, M. van Kreveld, and M. Overmars (2008). *Computational Geometry: Algorithms and Applications*. 3rd. Springer.

Berger, B., M. Feldman, V. Gkatzelis, and X. Tan (2024). "Learning-Augmented Metric Distortion via (p, q)-Veto Core." In: *EC*.

Bertsimas, D., V. F. Farias, and N. Trichakis (2011). "The Price of Fairness." In: *Oper. Res.* 59.1, pp. 17–31.

Black, D. (1948). "On the Rationale of Group Decision-Making." In: *J. Polit. Econ.* 56.1.

Blum, A. and C. Burch (2000). "On-Line Learning and the Metrical Task System Problem." In: *Mach. Learn.* 39.1, pp. 35–58.

Border, K. C. and J. S. Jordan (1983). "Straightforward Elections, Unanimity and Phantom Voters." In: *Rev. Econ. Stud.* 50.1, pp. 153–170.

Borodin, A., D. Halpern, M. Latifian, and N. Shah (2022). "Distortion in Voting with Top-t Preferences." In: *IJCAI*.

Borodin, A. and C. Karavasilis (2023). "Any-Order Online Interval Selection." In: *WAOA*, pp. 175–189.

Bousquet, N., S. Norin, and A. Vetta (2014). "A Near-Optimal Mechanism for Impartial Selection." In: *WINE*, pp. 133–146.

Bouzina, K. I. and H. Emmons (1996). "Interval Scheduling on Identical Machines." In: *J. Glob. Optim.* 9.3-4, pp. 379–393.

Boyar, J., L. M. Favrholdt, S. Kamali, and K. S. Larsen (2023). "Online Interval Scheduling with Predictions." In: *WADS*, pp. 193–207.

Brams, S. J. and A. D. Taylor (1996). *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.

Brandt, F., V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, eds. (2016). *Handbook of Computational Social Choice*. Cambridge University Press.

Budish, E. (2011). "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes." In: *J. Polit. Econ.* 119.6, pp. 1061–1103.

Canetti, R. and S. Irani (1995). "Bounding the Power of Preemption in Randomized Scheduling." In: *STOC*, pp. 606–615.

Caragiannis, I., G. Christodoulou, and N. Protopapas (2022). "Impartial Selection with Additive Approximation Guarantees." In: *Theory Comput. Syst.* 66, pp. 721–742.

Caragiannis, I., C. Kaklamanis, P. Kanellopoulos, and M. Kyropoulou (2012). "The Efficiency of Fair Division." In: *Theory Comput. Syst.* 50.4, pp. 589–610.

Caragiannis, I. and G. Kalantzis (2024). "Randomized Learning-Augmented Auctions with Revenue Guarantees." In: *IJCAI*.

Caragiannis, I., D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang (2019). "The Unreasonable Fairness of Maximum Nash Welfare." In: *ACM Trans. Economics and Comput.* 7.3, p. 12.

Caragiannis, I. and S. Narang (2024). "Repeatedly Matching Items to Agents Fairly and Efficiently." In: *Theor. Comput. Sci.* 981, p. 114246.

Caragiannis, I., S. Nath, A. D. Procaccia, and N. Shah (2017). "Subset Selection via Implicit Utilitarian Voting." In: *J. Artif. Intell. Res.* 58, pp. 123–152.

Caragiannis, I. and A. D. Procaccia (2011). "Voting Almost Maximizes Social Welfare Despite Limited Communication." In: *Artif. Intell.* 175, pp. 1655–1671.

Caragiannis, I., N. Shah, and A. A. Voudouris (2022). "The Metric Distortion of Multiwinner Voting." In: *Artif. Intell.* 313, p. 103802.

Cembrano, J., F. Fischer, D. Hannon, and M. Klimm (2024). "Impartial Selection with Additive Guarantees via Iterated Deletion." In: *Games Econ. Behav.* 144, pp. 203–224.

Cembrano, J., F. Fischer, and M. Klimm (2023). "Improved Bounds for Single-Nomination Impartial Selection." In: *EC*, p. 449.

Chan, H., A. Filos-Ratsikas, B. Li, M. Li, and C. Wang (2021). "Mechanism Design for Facility Location Problems: A Survey." In: *IJCAI*, pp. 4356–4365.

Chapin, S. J., W. Cirne, D. G. Feitelson, J. P. Jones, S. T. Leutenegger, U. Schwiegelshohn, W. Smith, and D. Talby (1999). "Benchmarks and Standards for the Evaluation of Parallel Job Schedulers." In: *JSSPP*. Vol. 1659. Lecture Notes in Computer Science. Springer, pp. 67–90.

Charikar, M. and P. Ramakrishnan (2022). "Metric Distortion Bounds for Randomized Social Choice." In: *SODA*.

Charikar, M., P. Ramakrishnan, K. Wang, and H. Wu (2024). "Breaking the Metric Voting Distortion Barrier." In: *SODA*.

Chawla, S. and D. Christou (2024). "Online Time-Windows TSP with Predictions." In: *APPROX/RANDOM*.

Chen, Q., N. Gravin, and S. Im (2024). "Strategic Facility Location via Predictions." In: *WINE*.

Chen, X., M. Li, and C. Wang (2020). "Favorite-Candidate Voting for Eliminating the Least Popular Candidate in a Metric Space." In: *AAAI*.

Chen, Z.-Z., G. Lin, R. Rizzi, J. Wen, D. Xu, Y. Xu, and T. Jiang (2005). "More Reliable Protein NMR Peak Assignment via Improved 2-Interval Scheduling." In: *J. Comput. Biol.* 12.2, pp. 129–146.

Cheng, Y., Z. Jiang, K. Munagala, and K. Wang (2020). "Group Fairness in Committee Selection." In: *ACM Trans. Economics and Comput.* 8.4, pp. 1–18.

Chledowski, J., A. Polak, B. Szabucki, and K. T. Żołna (2021). "Robust Learning-Augmented Caching: An Experimental Study." In: *ICML*, pp. 1920–1930.

Choi, K. W. and M. Li (2026). "Temporal Fair Division of Indivisible Goods with Scheduling." In: *arXiv preprint arXiv:2601.12835*.

Choo, D., T. Gouleakis, C. K. Ling, and A. Bhattacharyya (2024). "Online Bipartite Matching with Imperfect Advice." In: *arXiv preprint arXiv:2405.09784*.

Christodoulou, G., A. Sgouritsa, and I. Vlachos (2024). "Mechanism Design Augmented with Output Advice." In: *NeurIPS*.

Colini-Baldeschi, R., S. Klumper, G. Schäfer, and A. Tsikiridis (2024). "To Trust or Not to Trust: Assignment Mechanisms with Predictions in the Private Graph Model." In: *EC*.

Condorcet, N. de (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Reprint 2016. Cambridge University Press.

Cookson, B., S. Ebadian, and N. Shah (2025). "Temporal Fair Division." In: *AAAI*, pp. 13727–13734.

Critchlow, D. L., R. H. Dennard, and S. Schuster (2000). "Design and Characteristics of n-Channel Insulated-Gate Field-Effect Transistors." In: *IBM J. Res. Dev.* 44.1, pp. 70–83.

Dantzig, G. B. and D. R. Fulkerson (1954). "Minimizing the Number of Tankers to Meet a Fixed Schedule." In: *Nav. Res. Logist. Q.* 1.3, pp. 217–222.

Dütting, P., S. Lattanzi, R. P. Leme, and S. Vassilvitskii (2021). "Secretaries with Advice." In: *EC*, pp. 409–429.

Elkind, E. and P. Faliszewski (2014). "Recognizing 1-Euclidean Preferences: An Alternative Approach." In: *SAGT*.

Elkind, E., P. Faliszewski, P. Skowron, and A. Slinko (2017). "Properties of Multiwinner Voting Rules." In: *Soc. Choice Welf.* 48, pp. 599–632.

Elkind, E., M. Lackner, and D. Peters (2022). "Preference Restrictions in Computational Social Choice: A Survey." In: *arXiv preprint arXiv:2205.09092*.

Elkind, E., A. Lam, M. Latifian, T. Y. Neoh, and N. Teh (2025). "Temporal Fair Division of Indivisible Items." In: *AAMAS*, pp. 676–685.

Enelow, J. M. and M. J. Hinich (1984). *The Spatial Theory of Voting: An Introduction*. Cambridge University Press.

Even, S., A. Itai, and A. Shamir (1976). "On the Complexity of Timetable and Multicommodity Flow Problems." In: *SIAM J. Comput.* 5.4, pp. 691–703.

Faigle, U. and W. M. Nawijn (1995). "Note on Scheduling Intervals On-Line." In: *Discret. Appl. Math.* 58.1, pp. 13–17.

Fain, B., A. Goel, K. Munagala, and N. Prabhu (2019). "Random Dictators with a Random Referee: Constant Sample Complexity Mechanisms for Social Choice." In: *AAAI*.

Faliszewski, P., P. Skowron, A. Slinko, and N. Talmon (2017). "Multiwinner Voting: A New Challenge for Social Choice Theory." In: *Trends in Computational Social Choice*. AI Access, pp. 27–47.

Fang, J., Q. Fang, W. Liu, and Q. Nong (2024). "Mechanism Design with Predictions for Facility Location Games with Candidate Locations." In: *TAMC*, pp. 38–49.

Feitelson, D. G. (2005). *Parallel Workloads Archive*. `https://www.cs.huji.ac.il/labs/parallel/workload/`.

Feitelson, D. G., D. Tsafrir, and D. Krakov (2014). "Experience with Using the Parallel Workloads Archive." In: *J. Parallel Distributed Comput.* 74.10, pp. 2967–2982.

Feldman, M. and Y. Wilf (2013). "Strategyproof Facility Location and the Least Squares Objective." In: *EC*, pp. 873–890.

Fiat, A., Y. Rabani, and Y. Ravid (1994). "Competitive k-Server Algorithms." In: *J. Comput. Syst. Sci.* 48.3, pp. 410–428.

Fischer, F. and M. Klimm (2015). "Optimal Impartial Selection." In: *SIAM J. Comput.* 44.5, pp. 1263–1285.

Foley, D. (1967). "Resource Allocation and the Public Sector." In: *Yale Economic Essays*, pp. 45–98.

Ford, L. R. and D. R. Fulkerson (1962). *Flows in Networks*. Reprint 2024. Princeton University Press.

Fotakis, D., E. Gergatsouli, T. Gouleakis, and N. Patris (2021). "Learning Augmented Online Facility Location." In: *arXiv preprint arXiv:2107.08277*.

Fotakis, D. and L. Gourvès (2022). "On the Distortion of Single Winner Elections with Aligned Candidates." In: *Auton. Agents Multi Agent Syst.* 36.2, p. 37.

Fotakis, D., L. Gourvès, and J. Monnot (2016). "Conference Program Design with Single-Peaked and Single-Crossing Preferences." In: *WINE*.

Fotakis, D., L. Gourvès, and P. Patsilinakos (2025). "On the Distortion of Committee Election with 1-Euclidean Preferences and Few Distance Queries." In: *AAAI*.

Fung, S. P. Y., C. K. Poon, and D. K. W. Yung (2012). "On-Line Scheduling of Equal-Length Intervals on Parallel Machines." In: *Inf. Process. Lett.* 112.10, pp. 376–379.

Fung, S. P. Y., C. K. Poon, and F. Zheng (2008). "Online Interval Scheduling: Randomized and Multiprocessor Cases." In: *J. Comb. Optim.* 16.3, pp. 248–262.

Gabrel, V. (1995). "Scheduling Jobs Within Time Windows on Identical Parallel Machines: New Model and Algorithms." In: *Eur. J. Oper. Res.* 83.2, pp. 320–329.

Garay, J. A., I. S. Gopal, S. Kutten, Y. Mansour, and M. Yung (1997). "Efficient On-Line Call Control Algorithms." In: *J. Algorithms* 23.1, pp. 180–194.

Garg, N., V. V. Vazirani, and M. Yannakakis (1997). "Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees." In: *Algorithmica* 18.1, pp. 3–20.

Ghodsi, M., M. Latifian, and M. Seddighin (2019). "On the Distortion Value of the Elections with Abstention." In: *AAAI*.

Gibbard, A. (1973). "Manipulation of Voting Schemes: A General Result." In: *Econometrica* 41.4, pp. 587–601.

Gkatzelis, V., D. Halpern, and N. Shah (2020). "Resolving the Optimal Metric Distortion Conjecture." In: *FOCS*.

Gkatzelis, V., K. Kollias, A. Sgouritsa, and X. Tan (2022). "Improved Price of Anarchy via Predictions." In: *EC*, pp. 529–557.

Gkatzelis, V., D. Schoepflin, and X. Tan (2025). "Clock Auctions Augmented with Unreliable Advice." In: *SODA*.

Goel, A., R. Hulett, and A. K. Krishnaswamy (2018). "Relating Metric Distortion and Fairness of Social Choice Rules." In: *NetEcon*.

Goel, A., A. K. Krishnaswamy, and K. Munagala (2017). "Metric Distortion of Social Choice Rules: Lower Bounds and Fairness Properties." In: *EC*, pp. 287–304.

Goel, S. and W. Hann-Caruthers (2020). "Optimality of the Coordinate-Wise Median Mechanism for Strategyproof Facility Location in Two Dimensions." In: *Soc. Choice Welf.*

Gollapudi, S. and D. Panigrahi (2019). "Online Algorithms for Rent-or-Buy with Expert Advice." In: *ICML*, pp. 2319–2327.

Gupta, U. I., D. T. Lee, and J. Y.-T. Leung (1979). "An Optimal Solution for the Channel-Assignment Problem." In: *IEEE Trans. Computers* 28.11, pp. 807–810.

Hajnal, A. and E. Szemerédi (1970). "Proof of a Conjecture of P. Erdős." In: *Combinatorial Theory and Its Applications*. Vol. 4. Colloq. Math. Soc. János Bolyai. North-Holland, pp. 601–623.

He, J., A. D. Procaccia, A. Psomas, and D. Zeng (2019). "Achieving a Fairer Future by Changing the Past." In: *IJCAI*, pp. 343–349.

Herbster, M. and M. K. Warmuth (1998). "Tracking the Best Expert." In: *Mach. Learn.* 32.2, pp. 151–178.

Holzman, R. and H. Moulin (2013). "Impartial Nominations for a Prize." In: *Econometrica* 81.1, pp. 173–196.

Hosseini, H., Z. Huang, A. Igarashi, and N. Shah (2024). "Class Fairness in Online Matching." In: *Artif. Intell.* 335, p. 104177.

Hummel, H. and M. L. Hetland (2022). "Fair Allocation of Conflicting Items." In: *Auton. Agents Multi Agent Syst.* 36.1, p. 8.

Igarashi, A., M. Lackner, O. Nardi, and A. Novaro (2024). "Repeated Fair Allocation of Indivisible Items." In: *AAAI*, pp. 9781–9789.

Igarashi, A., P. Manurangsi, and H. Yoneda (2025). "Dividing Conflicting Items Fairly." In: *IJCAI*, pp. 3908–3915.

Im, S., R. Kumar, M. M. Qaem, and M. Purohit (2021). "Online Knapsack with Frequency Predictions." In: *NeurIPS*.

Irani, S. and K. Pruhs (2005). "Algorithmic Problems in Power Management." In: *SIGACT News* 36.2, pp. 63–76.

Istrate, G. and C. Bonchiş (2022). "Mechanism Design with Predictions for Obnoxious Facility Location." In: *arXiv preprint arXiv:2212.09521*.

Istrate, G., C. Bonchiş, and V. Bogdan (2024). "Equilibria in Multiagent Online Problems with Predictions." In: *arXiv preprint arXiv:2405.11873*.

Jessee, S. A. (2012). *Ideology and Spatial Voting in American Elections*. Cambridge University Press.

Jiang, S. H.-C., E. Liu, Y. Lyu, Z. G. Tang, and Y. Zhang (2022). "Online Facility Location with Predictions." In: *ICLR*.

Jiang, Z., K. Munagala, and K. Wang (2020). "Approximately Stable Committee Selection." In: *STOC*.

Jones, N. (2018). "How to Stop Data Centres from Gobbling up the World's Electricity." In: *Nature* 561, pp. 163–166.

Kalayci, Y., D. Kempe, and V. Kher (2024). "Proportional Representation in Metric Spaces and Low-Distortion Committee Selection." In: *AAAI*.

Karavasilis, C. (2025). "Interval Selection with Binary Predictions." In: *IJCAI*, pp. 8527–8535.

Kizilkaya, F. E. and D. Kempe (2022). "Plurality Veto: A Simple Voting Rule Achieving Optimal Metric Distortion." In: *IJCAI*.

Kolen, A. W. J., J. K. Lenstra, C. H. Papadimitriou, and F. C. R. Spieksma (2007). "Interval Scheduling: A Survey." In: *Nav. Res. Logist.* 54.5, pp. 530–543.

Kovalyov, M. Y., C. T. Ng, and T. C. E. Cheng (2007). "Fixed Interval Scheduling: Models, Applications, Computational Complexity and Algorithms." In: *Eur. J. Oper. Res.* 178.2, pp. 331–342.

Kraska, T., A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis (2018). "The Case for Learned Index Structures." In: *SIGMOD*, pp. 489–504.

Kulkarni, P. R., R. Mehta, V. V. Narayan, and T. Ponitka (2026). "Online Fair Division with Subsidy: When Do Envy-Free Allocations Exist, and at What Cost?" In: *AAMAS*.

Kumar, Y., S. Equbal, R. Gurjar, S. Nath, and R. Vaish (2024). "Fair Scheduling of Indivisible Chores." In: *AAMAS*, pp. 2345–2347.

Lattanzi, S., T. Lavastida, B. Moseley, and S. Vassilvitskii (2020). "Online Scheduling via Learned Weights." In: *SODA*.

Lindermayr, A. and N. Megow (n.d.). *ALPS: Algorithms with Predictions*. `https://algorithms-with-predictions.github.io/`.

Lipton, R. J., E. Markakis, E. Mossel, and A. Saberi (2004). "On Approximately Fair Allocations of Indivisible Goods." In: *EC*, pp. 125–131.

Lipton, R. J. and A. Tomkins (1994). "Online Interval Scheduling." In: *SODA*, pp. 302–311.

Littlestone, N. and M. K. Warmuth (1994). "The Weighted Majority Algorithm." In: *Inf. Comput.* 108.2, pp. 212–261.

Long, D. D. E. and M. N. Thakur (1993). "Scheduling Real-Time Disk Transfers for Continuous Media Applications." In: *IEEE Symposium on Mass Storage Systems*, pp. 227–232.

Lu, P., Z. Wan, and J. Zhang (2024). "Competitive Auctions with Imperfect Predictions." In: *EC*.

Lykouris, T. and S. Vassilvitskii (2021). "Competitive Caching with Machine Learned Advice." In: *J. ACM* 68.4, 24:1–24:25.

Mackenzie, A. (2015). "Symmetry and Impartial Lotteries." In: *Games Econ. Behav.* 94, pp. 15–28.

Meir, R. (2019). "Strategyproof Facility Location for Three Agents on a Circle." In: *SAGT*, pp. 18–33.

Merrill, S. and B. Grofman (1999). *A Unified Theory of Voting: Directional and Proximity Spatial Models*. Cambridge University Press.

Meyerson, A. (2001). "Online Facility Location." In: *FOCS*, pp. 426–431.

El-Mhamdi, E.-M., S. Farhadkhani, R. Guerraoui, and L.-N. Hoang (2023). "On the Strategyproofness of the Geometric Median." In: *AISTATS*, pp. 2603–2640.

Mirrlees, J. A. (1971). "An Exploration in the Theory of Optimum Income Taxation." In: *Rev. Econ. Stud.* 38.2, pp. 175–208.

Mitzenmacher, M. (2018). "A Model for Learned Bloom Filters, and Optimizing by Sandwiching." In: *NeurIPS*, pp. 462–471.

Mitzenmacher, M. (2020). "Scheduling with Predictions and the Price of Misprediction." In: *ITCS*. Vol. 151. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 14:1–14:18.

Mitzenmacher, M. and S. Vassilvitskii (2022). "Algorithms with Predictions." In: *Commun. ACM* 65.7, pp. 33–35.

Miyagawa, E. (2001). "Locating Libraries on a Street." In: *Soc. Choice Welf.* 18.3, pp. 527–541.

Moulin, H. (1980). "On Strategy-Proofness and Single Peakedness." In: *Public Choice* 35.4, pp. 437–455.

Moulin, H. (2004). *Fair Division and Collective Welfare*. MIT Press.

Munagala, K. and K. Wang (2019). "Improved Metric Distortion for Deterministic Social Choice Rules." In: *EC*, pp. 245–262.

Nishizeki, T., J. Vygen, and X. Zhou (2001). "The Edge-Disjoint Paths Problem Is NP-Complete for Series-Parallel Graphs." In: *Discret. Appl. Math.* 115.1, pp. 177–186.

Peters, H., H. van der Stel, and T. Storcken (1993). "Range Convexity, Continuity, and Strategy-Proofness of Voting Schemes." In: *Zeitschrift für Operations Research* 38.2, pp. 213–229.

Procaccia, A. D. and J. S. Rosenschein (2006). "The Distortion of Cardinal Preferences in Voting." In: *CIA*.

Procaccia, A. D. and M. Tennenholtz (2013). "Approximate Mechanism Design Without Money." In: *ACM Trans. Economics and Comput.* 1.4, 18:1–18:26.

Pulyassary, H. (2022). "Algorithm Design for Ordinal Settings." MA thesis. University of Waterloo.

Pulyassary, H. and C. Swamy (2021). "On the Randomized Metric Distortion Conjecture." In: *arXiv preprint arXiv:2111.08698*.

Purohit, M., Z. Svitkina, and R. Kumar (2018). "Improving Online Algorithms via ML Predictions." In: *NeurIPS*.

Rédei, L. (1934). "Ein Kombinatorischer Satz." In: *Acta Litterarum ac Scientiarum Regiae Universitatis Hungaricae Francisco-Josephinae, Sectio Scientiarum Mathematicarum* 7, pp. 39–43.

Rohatgi, D. (2020). "Near-Optimal Bounds for Online Caching with Machine Learned Advice." In: *SODA*, pp. 1834–1845.

Roughgarden, T. (2021). *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press.

Satterthwaite, M. A. (1975). "Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions." In: *J. Econ. Theory* 10.2, pp. 187–217.

Seiden, S. S. (2000). "Randomized Algorithms for That Ancient Scheduling Problem." In: *J. Algorithms* 36.1, pp. 51–77.

Sleator, D. D. and R. E. Tarjan (1985). "Amortized Efficiency of List Update and Paging Rules." In: *Commun. ACM* 28.2, pp. 202–208.

Suksompong, W. (2021). "Constraints in Fair Division." In: *SIGecom Exch.* 19.2, pp. 46–61.

Tamura, S. (2016). "Characterizing Minimal Impartial Rules for Awarding Prizes." In: *Games Econ. Behav.* 95, pp. 41–46.

Tang, P., D. Yu, and S. Zhao (2020). "Characterization of Group-Strategyproof Mechanisms for Facility Location in Strictly Convex Space." In: *EC*, pp. 133–157.

Tomkins, A. (1995). "Lower Bounds for Two Call Control Problems." In: *Inf. Process. Lett.* 56.3, pp. 173–178.

Varian, H. R. (1974). "Equity, Envy, and Efficiency." In: *J. Econ. Theory* 9.1, pp. 63–91.

Voudouris, A. A. (2023). "Tight Distortion Bounds for Distributed Metric Voting on a Line." In: *Oper. Res. Lett.* 51.3, pp. 266–269.

Wagner, D. and K. Weihe (1995). "A Linear-Time Algorithm for Edge-Disjoint Paths in Planar Graphs." In: *Combinatorica* 15.1, pp. 135–150.

Walsh, T. (2020). "Strategy-Proof Mechanisms for Facility Location in Euclidean and Manhattan Space." In: *arXiv preprint arXiv:2009.07983*.

Wang, S., J. Li, and S. Wang (2020). "Online Algorithms for Multi-Shop Ski Rental with Machine Learned Advice." In: *NeurIPS*.

Wei, A. (2020). "Better and Simpler Learning-Augmented Online Caching." In: *APPROX/RANDOM*.

Wierman, A., L. L. H. Andrew, and A. Tang (2012). "Power-Aware Speed Scaling in Processor Sharing Systems: Optimality and Robustness." In: *Perform. Evaluation* 69.12, pp. 601–622.

Woeginger, G. J. (1994). "On-Line Scheduling of Jobs with Fixed Start and End Times." In: *Theor. Comput. Sci.* 130.1, pp. 5–16.

Xu, C. and P. Lu (2022). "Mechanism Design with Predictions." In: *IJCAI*, pp. 571–577.

Yao, F. F., A. J. Demers, and S. Shenker (1995). "A Scheduling Model for Reduced CPU Energy." In: *FOCS*, pp. 374–382.