The Complexity of Nash Equilibria, Local Optima, and Pareto-Optimal Solutions

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker

Heiko Röglin

aus Dortmund

Berichter: Universitätsprofessor Dr. Berthold Vöcking
Universitätsprofessor Dr. Dr.h.c. Wolfgang Thomas

Tag der mündlichen Prüfung: 10. April 2008

Abstract

An instance of a combinatorial optimization problem is usually described by an objective function that is to be optimized over a set of feasible solutions. The decisions that people, companies, and other economic entities face every day are more complex for various reasons: In many situations, there is more than one objective and one is rather interested in finding the most appropriate trade-off than in optimizing a single criterion. Further complications arise when decisions are made by selfish agents instead of being centrally coordinated, and even decisions that can be modeled as combinatorial optimization problems are often intractable under standard complexity-theoretic assumptions. These difficulties gave rise to a variety of solution concepts, including Pareto-optimal solutions, Nash equilibria, and local optima, which are the topic of this thesis.

When it comes to the allocation of scarce resources, decisions are often made by selfish agents. The predominant solution concept in such situations is that of a Nash equilibrium, which is a state in which no agent can benefit from unilaterally changing her strategy. We consider congestion games and two-sided markets, two extensively studied models for resource allocation among selfish agents. In both of these models, a set of players competes for a set of resources, the difference being that in two-sided markets each resource can be allocated by at most one player, whereas in congestion games a resource can be allocated by several players simultaneously at the cost of a decreased payoff. Two-sided markets have been introduced to model markets on which different kinds of agents are matched to one another, for example, students and colleges or interns and hospitals, and they have found applications in many different areas. Congestion games can, for instance, be used to model routing in large networks like the Internet.

For congestion games, we analyze the complexity of computing a pure Nash equilibrium. This problem can be phrased as a local search problem that belongs to the complexity class PLS. We present a new approach that enables us to prove PLS-hardness results for different classes of congestion games like market sharing games and overlay network games. Our approach also yields a short proof for the PLS-completeness of network congestion games for directed and undirected networks. In two-sided markets, pure Nash equilibria can be computed efficiently, but many real-life markets lack a central authority to match agents. This motivates the study of the processes that take place when players consecutively improve their strategies. We demonstrate that coordination is necessary by constructing examples on which uncoordinated agents need in expectation exponentially many steps to reach an equilibrium if they improve their strategies in a random order. On the positive side, we identify a special class of two-sided markets with real-life applications for which we prove that uncoordinated agents reach an equilibrium in

expected polynomial time. We conclude the part about resource allocation among selfish agents by introducing a natural class of resource sharing games that both extends congestion games and two-sided markets. We prove several results for this model that unify the theory of these two special cases.

In the second part of this dissertation, we consider optimization problems with two criteria. Since, in general, it is not possible to optimize both criteria simultaneously, one has to find an appropriate trade-off. Solutions that are optimal in the sense that no criterion can be improved without deteriorating the other one are called Pareto-optimal, and the set of Pareto-optimal solutions is an important solution concept for bicriteria optimization problems as it helps to filter out unreasonable trade-offs. Even though in practice for most bicriteria problems the number of Pareto-optimal solutions is typically small, for almost all bicriteria problems, instances exist with an exponentially large Pareto set. The discrepancy between theory and practice arises because worst-case results are overly pessimistic as inputs occurring in practice are often very different from worst-case instances. We study the number of Pareto-optimal solutions in the framework of smoothed analysis, which is a hybrid of worst-case and average-case analysis, in which an adversary specifies an instance that is subsequently slightly perturbed at random.

We prove an almost tight polynomial bound on the expected number of Pareto-optimal solutions for general bicriteria integer optimization problems. Our results directly imply a tight polynomial bound on the expected running time of a heuristic for the binary knapsack problem and they significantly improve the known results for heuristics for the bounded knapsack problem and for the bicriteria shortest path problem. Our results also enable us to improve and simplify the previously known analysis of the smoothed complexity of integer programming. For certain problems such as the bicriteria spanning tree problem, there are no algorithms known for enumerating the set of Pareto-optimal solutions efficiently in its size. We present a method that allows us to enumerate the set of Pareto-optimal solutions for semi-random inputs in expected polynomial time with a small failure probability for all problems for which this set can be enumerated in pseudopolynomial time.

Local search is not only important because computing pure Nash equilibria can be phrased as a local search problem, but it also plays a crucial role in the design of heuristics for various NP-hard optimization problems. In particular, for the famous traveling salesperson problem, local search heuristics like 2-Opt achieve amazingly good results on real-world instances both with respect to running time and approximation ratio. There are numerous experimental studies on the performance of 2-Opt, but the theoretical knowledge about this heuristic is still very limited. Not even its worst-case running time on Euclidean instances was known so far. We clarify this issue by presenting, for every p, a family of two-dimensional L_p instances on which 2-Opt can take an exponential number of steps. In order to explain the discrepancy between this worst-case result and the observations in practice, we analyze 2-Opt in the framework of smoothed analysis. We show that the expected number of local improvements on semi-random Manhattan and Euclidean instances is polynomial, improving previous average-case results significantly. In addition, we prove an upper bound on the expected approximation factor with respect to all L_p metrics that depends polynomially on the magnitude of the random perturbation.

Zusammenfassung

Eine Instanz eines kombinatorischen Optimierungsproblems kann durch eine Menge gültiger Lösungen und eine Zielfunktion beschrieben werden, und das Ziel ist es, eine gültige Lösung auszuwählen, die die Zielfunktion optimiert. Viele alltägliche Entscheidungen, mit denen sich Personen, Unternehmen und andere Wirtschaftseinheiten konfrontiert sehen, können nicht als kombinatorische Optimierungsprobleme im klassischen Sinne modelliert werden, weil es beispielsweise mehrere gleichwertige Zielfunktionen gibt, die nicht zugleich optimiert werden können. Weitere Komplikationen entstehen in Situationen, in denen viele eigennützig handelnde Agenten involviert sind und es nicht möglich ist, eine zentral koordinierte Entscheidung zu treffen. Außerdem erweisen sich selbst Entscheidungen, die als kombinatorisches Optimierungsproblem formuliert werden können, unter Standardannahmen der Komplexitätstheorie oftmals als praktisch unlösbar. Diese Schwierigkeiten haben zur Entwicklung einer Vielzahl unterschiedlicher Lösungskonzepte beigetragen. In dieser Arbeit beschäftigen wir uns insbesondere mit Nash-Gleichgewichten, Pareto-optimalen Lösungen und lokalen Optima.

Situationen, in denen Entscheidung von eigennützig handelnden Agenten getroffen werden, treten häufig im Zusammenhang mit der Allokation knapper Ressourcen auf und Nash-Gleichgewichte sind das vorherrschende Lösungskonzept in solchen Situationen. Bei diesen Gleichgewichten handelt es sich um Zustände, die in der Hinsicht stabil sind, dass kein Agent davon profitieren kann, seine momentane Entscheidung zu ändern. Wir beschäftigen uns mit den beiden in der Ökonomik ausführlich untersuchten Modellen der Congestion-Spiele und zweiseitigen Märkte. In beiden Modellen konkurrieren Agenten um eine Menge von Ressourcen mit dem Unterschied, dass in einem zweiseitigen Markt jede Ressource nur von einem Agenten belegt werden kann, wohingegen sich in einem Congestion-Spiel mehrere Agenten eine Ressource auf Kosten einer geringeren Auszahlung teilen können. Zweiseitige Märkte dienen als Modell für Märkte, auf denen verschiedene Arten von Agenten z. B. Studenten und Universitäten einander zugeordnet werden. Congestion-Spiele werden beispielsweise benutzt, um Routing in großen Netzwerken wie dem Internet zu modellieren.

Das Problem, ein Nash-Gleichgewicht in einem Congestion-Spiel zu berechnen, kann als lokales Suchproblem aus der Komplexitätsklasse PLS formuliert werden. Wir präsentieren einen neuen Ansatz, der die PLS-Vollständigkeit für diverse Klassen von Congestion-Spielen zeigt und zu einer signifikanten Vereinfachung bereits bekannter Reduktionen für Netzwerk-Congestion-Spiele führt. In zweiseitigen Märkten können Nash-Gleichgewichte effizient berechnet werden, es gibt jedoch in vielen Situationen, in denen dieses Modell Anwendung findet, keine zentrale Stelle, die Agenten einander zuordnet. Wir konstruieren Instanzen, die zeigen, dass

Koordination notwendig ist, weil unkoordinierte Agenten im Erwartungswert exponentiell viele Schritte benötigen, um ein Gleichgewicht zu erreichen, wenn sie ihre Strategien in einer zufälligen Reihenfolge verbessern. Wir identifizieren jedoch eine interessante eingeschränkte Klasse von zweiseitigen Märkten, in denen unkoordinierte Agenten schnell ein Gleichgewicht finden. Wir schließen das Kapitel mit der Einführung eines neuen Modells, das sowohl Congestion-Spiele als auch zweiseitige Märkte verallgemeinert. Wir beweisen einige Resultate für dieses neue Modell, die die Theorie der beiden Spezialfälle vereinheitlichen.

Der zweite Teil dieser Dissertation beschäftigt sich mit bikriteriellen Optimierungsproblemen. Da es im Allgemeinen nicht möglich ist, beide Zielfunktionen gleichzeitig zu optimieren, besteht die Schwierigkeit in solchen Situationen darin, einen guten Kompromiss zu finden. Ein Kompromiss, in dem kein Kriterium verbessert werden kann, ohne das andere zu verschlechtern, heißt Pareto-optimal, und die Menge der Pareto-optimalen Lösungen ist ein wichtiges Lösungskonzept für bikriterielle Probleme. Obwohl man in Anwendungen oft nur wenige Pareto-optimale Lösungen beobachtet, existieren für fast alle bikriteriellen Probleme Instanzen mit exponentiell vielen Pareto-optimalen Lösungen. Der Grund für diese Diskrepanz ist, dass theoretische Ergebnisse auf worst-case Instanzen beruhen, die sich stark von typischen Instanzen, die in der Praxis auftreten, unterscheiden. Diesem Problem entgegnen wir, indem wir unseren Analysen das semi-zufällige Eingabemodell der geglätteten Analyse zu Grunde legen, in dem ein Gegner eine Eingabe vorgeben kann, die anschließend einer leichten zufälligen Perturbation unterworfen wird.

Wir zeigen eine nahezu scharfe polynomielle Schranke für die erwartete Anzahl Pareto-optimaler Lösungen für bikriterielle Optimierungsprobleme. Dieses Ergebnis liefert scharfe polynomielle Schranken für die erwartete Laufzeit einer Heuristik für das Rucksackproblem und signifikant verbesserte Schranken für Heuristiken für das beschränkte Rucksackproblem und das bikriterielle kürzeste Wege Problem. Das Ergebnis erlaubt uns ebenfalls, die bereits bekannte Analyse der geglätteten Komplexität ganzzahliger Optimierungsprobleme zu vereinfachen und zu verbessern. Als weiteres Ergebnis zeigen wir, wie man auf semi-zufälligen Eingaben die Menge der Pareto-optimalen Lösungen effizient erzeugen kann für Probleme, für die diese Menge im Worst-Case in pseudopolynomieller Zeit erzeugt werden kann.

Lokale Suche ist nicht nur interessant wegen des Zusammenhangs mit Nash-Gleichgewichten, sondern spielt auch bei zahlreichen Heuristiken für NP-harte Optimierungsprobleme wie z. B. das Problem des Handlungsreisenden (TSP) eine wichtige Rolle. 2-Opt ist eine sehr einfache lokale Suchheuristik für das TSP, die bemerkenswert gute Ergebnisse in Bezug auf Laufzeit und Approximationsgüte erzielt. Es gibt zahlreiche experimentelle Studien von 2-Opt, das theoretische Wissen ist jedoch sehr begrenzt, so war bisher nicht einmal die worst-case Laufzeit auf euklidischen Eingaben bekannt. Wir beantworten diese Frage, indem wir für jede L_p-Metrik eine Familie von zweidimensionalen Instanzen konstruieren, auf denen 2-Opt exponentiell viele Schritte machen kann, bevor es ein lokales Optimum erreicht. Um dieses Resultat mit den Beobachtungen in der Praxis in Einklang zu bringen, untersuchen wir auch 2-Opt in einem semi-zufällige Eingabemodell. Wir verbessern bekannte average-case Resultate deutlich und zeigen, dass die erwartete Anzahl lokaler Verbesserungen auf semi-zufälligen Eingaben polynomiell ist. Des Weiteren zeigen wir, dass die erwartete Approximationsgüte auf semi-zufälligen Instanzen polynomiell von der Stärke der Perturbation abhängt.

Acknowledgments

It would not have been possible to write this thesis without the support and encouragement of several people, to whom I wish to express my gratitude.

Most of all, I thank my advisor, Berthold Vöcking, for his invaluable guidance and support. I am especially grateful for his enthusiasm and for always leading my research into the right direction while leaving me a lot of freedom at the same time. I also thank Wolfgang Thomas for his effort as co-referee of this thesis.

I am indebted to my collaborators Heiner Ackermann, René Beier, Matthias Englert, Paul Goldberg, Vahab Mirrokni, Alantha Newman, and Matthias Westermann for many inspiring discussions (not only about research). In particular, I thank Alantha, Paul, and Vahab for inviting me to the Max-Planck-Institut für Informatik, the University of Liverpool, and Microsoft Research, respectively.

I thank the whole team of the Algorithms and Complexity group at RWTH Aachen for the great (working) atmosphere. Special thanks go to Matthias Englert for proofreading this thesis and for many valuable suggestions.

Last but not least, I thank my family for their continuous encouragement and support.

This work was supported by DFG grant Vo889/2.

Contents

Chapter 1

Introduction

Economic entities face decisions every day. These decisions are usually quite complex and involve the optimization of multiple objectives. Often there is either a single decision-maker interested in simultaneously optimizing several, possibly conflicting, criteria or there are several selfish agents, each of which interested in optimizing one of the objectives. This led to an increased interest in different *solution concepts* that extend the classical theory of combinatorial optimization. Rather than finding a solution that optimizes a single objective function, one has to trade off different criteria against each other and find an appropriate compromise. Often, one also has to cope with limits of tractability imposed by the computational complexity of the involved problems. These issues gave rise to a variety of solution concepts, including, in particular, *Pareto-optimal solutions*, *Nash equilibria*, and *local optima*, whose computational complexity is the main topic of this dissertation.

We begin our investigations with situations in which decisions are made by selfish agents. This line of research is motivated by the Internet, which is built, operated, and used by countless different economic entities with varying interests and without a central authority. It is natural to model these entities as selfish agents interested in maximizing their own benefit. In order to understand the agents' behavior and the consequences arising from the lack of coordination, it is necessary to combine classical methods from computer science with techniques from mathematical economics such as game theory and mechanism design. The predominant solution concept in such situations is that of a *Nash equilibrium*, which is a state in which no agent can benefit from unilaterally changing her strategy. Nash equilibria are a natural solution concept because they are stable states if agents are rational and non-cooperative. Since in economics and computer science non-cooperative situations usually arise when it comes to the allocation of scarce resources, we study different models for *resource allocation* among selfish agents. Our results are about the complexity of computing Nash equilibria and about the question whether uncoordinated agents reach an equilibrium without central coordination.

In the second part of this dissertation, control is given to a decision maker who is solely responsible for choosing a solution from a given set of options. We assume that with each of these options two objective values are associated and that the decision maker is interested in optimizing both of them. Since, in general, it is not possible to optimize both criteria simultaneously, the decision maker has to find an appropriate *trade-off*. This is a natural assumption because economic entities often face decisions that involve the optimization of more than one criterion. For

example, when making an investment one tries to optimize the expected return as well as the risk, when buying products one has to make a compromise between quality and price, and when compressing data one has to trade off information loss against size. Those trade-offs in which no criterion can be improved without deteriorating the other one are called *efficient* or *Pareto-optimal*. The set of Pareto-optimal solutions, or *Pareto set* for short, is an important solution concept for multi-criteria optimization problems as it helps the decision maker to filter out unreasonable trade-offs. For algorithmic applications, the Pareto set is only useful if the number of Pareto-optimal solutions is not too large. We analyze the number of Pareto-optimal solutions for bicriteria integer optimization problems in a very general framework and obtain an almost tight polynomial bound, showing that the Pareto set for many bicriteria integer optimization problems can be computed efficiently.

Our analysis of the number of Pareto-optimal solutions is not based on the usual worst-case perspective of theoretical computer science, in which an optimization problem is judged by its worst instance. The reason is that, even though in practice for most bicriteria problems the number of Pareto-optimal solutions is typically small, for almost all these problems, instances exist with an exponentially large Pareto set. The discrepancy between theory and practice arises because worst-case results are overly pessimistic as inputs occurring in practice are often very different from artificial worst-case instances. We study the number of Pareto-optimal solutions in the framework of *smoothed analysis*. Smoothed analysis is a hybrid of worst-case and average-case analysis, in which an adversary specifies an instance that is subsequently slightly perturbed at random. This semi-random input model often yields more realistic results than a worst-case or average-case analysis alone.

Motivated by the intractability of many combinatorial optimization problems, the third part of this dissertation is devoted to the study of *local search*. If, for example, a company has to solve a logistics problem that can be formulated as a version of the traveling salesperson problem, then finding an optimal solution is usually hard even if there is only a single decision maker and only one objective. In such situations, it becomes necessary to relax the requirement of finding an optimal solution and to apply a different solution concept. For many hard optimization problems, good *approximations* are obtained by local search algorithms, which start with an arbitrary feasible solution and improve this solution consecutively until a solution is reached that cannot be improved further by a local modification. The definition of a local modification is problem-specific, but if it is defined in the right way, then local search turns out to be a powerful tool for obtaining good approximations quickly in practice. Another motivation for studying the solution concept of *local optima* is that there is a close connection between Nash equilibria and local optima as the problem of computing a Nash equilibrium for the resource allocation games studied in the first part can be phrased as a local search problem.

Since local search is particularly successful for the famous and well-studied traveling salesperson problem, we study the 2-Opt heuristic, the probably most basic local search heuristic for the traveling salesperson problem. This heuristic achieves amazingly good results on real-world Euclidean instances both with respect to running time and approximation ratio. There are numerous experimental studies on its performance, but the theoretical knowledge is still very limited. We

show that this heuristics performs very poorly in the worst case, and in order to explain its success in practice, we show that it performs well in the framework of smoothed analysis both with respect to running time and approximation ratio.

In the remainder of this introduction, we describe the considered solution concepts and our results in more detail and we discuss related work. The introduction is supposed to give an overview of our results. Hence, in order to improve the readability, some results are not stated in their most precise and most general form. Readers who are interested in more details are referred to the following chapters.

1.1 Nash Equilibria

Due to its distributed and uncoordinated nature, the Internet created a new paradigm in computer science. Since there is no central authority, the Internet is built, operated, and used by various, often competing economic entities, and hence, an in-depth understanding cannot be gained by classical methods from computer science alone. It seems rather necessary to combine these methods with economic techniques like *game theory*, which studies how economic entities interact and make decisions. From this combination, the active research area of *algorithmic game theory* has emerged.

A *game* is an abstract model for situations in which decisions are made by non-cooperative agents. A finite game in normal form consists of a finite set $\mathcal{N} = \{1, \dots, n\}$ of players, and each player $i \in \mathcal{N}$ is equipped with a finite set Σ_i of *pure strategies* and a *payoff function* $p_i \colon \prod_{i \in \mathcal{N}} \Sigma_i \to \mathbb{N}$. Every player $i \in \mathcal{N}$ can choose one of her strategies from the set Σ_i, and we assume that every player is interested in choosing a strategy that maximizes her own payoff. A player $i \in \mathcal{N}$ receives a payoff of $p_i(S)$ if the choices of the players result in the strategy combination $S \in \Sigma := \Sigma_1 \times \cdots \times \Sigma_n$. Instead of letting each player pick a pure strategy, game theory also studies *mixed strategies*, that is, each player $i \in \mathcal{N}$ can choose a probability distribution over her strategy space Σ_i. If players play mixed strategies instead of pure ones, we assume that every player is interested in maximizing her *expected payoff*.

Given a game, the most intriguing question is how the players behave. In particular, it is interesting whether the players end up in some kind of stable state, and if so, how long it takes to reach this state and which properties it has. To answer these questions, John Nash introduced in his Nobel Prize-winning dissertation [Nas50] the solution concept of *non-cooperative equilibrium*, nowadays known as *Nash equilibrium*. The definition of a Nash equilibrium is based on the simple observation that if agents are rational and non-cooperative, a strategy combination $S = (S_1, \dots, S_n)$ cannot be stable if a player $i \in \mathcal{N}$ can increase her payoff by defecting from her chosen strategy S_i. Strategy combinations in which no player can benefit from unilaterally defecting from her strategy are exactly the non-cooperative equilibria defined by Nash. For special classes of games, this solution concept has already been used by Cournot [Cou38] and by von Neumann and Morgenstern [vNM47]. The latter consider non-cooperative equilibria for two-player *zero-sum games*, in which the payoffs of the players add up to the same value for every possible strategy combination $S \in \Sigma$. Nash's ingenious contribution was to show that every finite normal-form game possesses a Nash equilibrium when

players are allowed to play mixed strategies. This appealing property made Nash equilibria one of the most popular solution concept in game theory.

Let us mention that there are, of course, also other solution concepts in game theory and further refinements of Nash equilibria that we do not discuss in detail in this thesis. For example, Reinhard Selten, who shared the Nobel Prize in economics in 1994 with John Harsanyi and John Nash, introduced the notion of *trembling-hand perfect equilibrium*, which takes into account that players are not perfect and make mistakes with a small probability. John Harsanyi suggested the notion of *Bayesian Nash equilibrium*, which is an appropriate solution concept if players have incomplete information about the other players' payoffs and try to predict their actions. Another example is the solution concept of *correlated equilibrium* which considers the case that the probability distributions chosen by the players can be correlated. For more details about these and other solution concepts like *subgame perfect equilibria* and *evolutionary stable equilibria*, we refer the reader to the books about game theory by Owen [Owe95] and by Fudenberg and Tirole [FT91].

In this thesis, we study Nash equilibria because they are one of the most appealing and best-studied solution concepts in game theory. One objection to Nash equilibria is that the strategies in a Nash equilibrium can be quite complex, that is, players might have to choose strategies with a large support and carefully chosen probabilities to obtain an equilibrium. We believe that it is often more realistic to assume that players choose pure strategies rather than the much more complex randomized ones. The games that we study in this thesis have the very appealing property that they always possess *pure Nash equilibria*, that is, in order to obtain a Nash equilibrium players do not have to randomize over their strategy set. Hence, we will only be concerned with pure equilibria in this thesis, and the term *Nash equilibrium* always denotes a pure Nash equilibrium if not explicitly mentioned otherwise.

Since non-cooperative situations arise often in the context of the allocation of scarce resources, we investigate two extensively studied models for resource allocation among selfish agents, namely *congestion games* and *two-sided matching markets*. In both of these models, a set of players competes for a set of resources, the difference being that in two-sided markets each resource can be allocated by at most one player, whereas in congestion games a resource can be allocated by several players simultaneously at the cost of a decreased payoff. Two-sided markets have been introduced to model markets on which different kinds of agents are matched to one another, for example, students and colleges or interns and hospitals, and they have found applications in many different areas. Congestion games can, for instance, be used to model load balancing among servers or routing in large networks like the Internet.

A lot of research has focused on properties of Nash equilibria. Typical properties of interest are, for example, the *price of anarchy*, which is the ratio of the social cost in the worst Nash equilibrium to the minimum social cost possible, and the *price of stability*, which is the ratio of the social cost in the best Nash equilibrium to the minimum social cost possible. Our research is not concerned with properties of equilibria, but we are rather interested in the questions whether selfish agents reach an equilibrium quickly without coordination and if equilibria can be computed efficiently by a central authority. The former question is

motivated by the observation that in many game-theoretic scenarios, there is no central authority and agents have to make decisions without being centrally coordinated in any way. In such scenarios, there are usually many different equilibria and it is unrealistic to assume that the agents start out in one of them. Instead they need to coordinate themselves somehow to reach an equilibrium. The second question arises in scenarios in which a central authority is present. For example, in most networking applications, agents interact based on an underlying protocol that proposes a solution to all participants, who have to decide whether to follow the proposed solution or to defect from it. If the proposed solution is an equilibrium, then all agents have an incentive to follow the protocol, and hence, it is an important question for the protocol designer whether equilibria can be computed efficiently.

In the following, we introduce congestion games and two-sided markets formally and we state our results.

1.1.1 Congestion Games

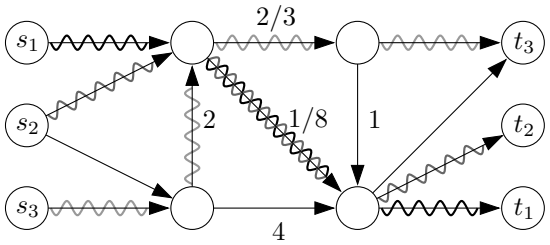
A *congestion game* Γ is a tuple $(\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$ where $\mathcal{N} = \{1, \dots, n\}$ denotes a set of players, $\mathcal{R}$ with $m := |\mathcal{R}|$ a set of resources, $\Sigma_i \subseteq 2^{\mathcal{R}}$ the strategy space of player $i \in \mathcal{N}$, and $d_r \colon \mathbb{N} \to \mathbb{N}$ a delay function associated with resource $r \in \mathcal{R}$. We denote by $S = (S_1, \dots, S_n)$ the *state* of the game in which player i plays strategy $S_i \in \Sigma_i$. Furthermore, we denote by $S \oplus S_i'$ the state $S' = (S_1, \dots, S_{i-1}, S_i', S_{i+1}, \dots, S_n)$, i.e., the state S except that player i plays strategy S_i' instead of S_i. For a state S, we define the *congestion* $n_r(S)$ on resource r by $n_r(S) = |\{i \mid r \in S_i\}|$, that is, $n_r(S)$ is the number of players sharing resource r in state S. We assume that every player $i \in \mathcal{N}$ wants to minimize her delay $\delta_i(S)$, which is defined as $\sum_{r\in S_i} d_r(n_r(S))$.

Given a state S, we call a strategy $S_i^* \in \Sigma_i$ a *best response* of player i to S if, for all $S_i' \in \Sigma_i$, $\delta_i(S \oplus S_i^*) \leq \delta_i(S \oplus S_i')$, and we call it a *better response* if $\delta_i(S \oplus S_i^*) < \delta_i(S)$. In the following, we use the term *best response sequence* to denote a sequence of consecutive strategy changes in which each step is a best response that strictly decreases the delay of the corresponding player. Analogously, a *better response sequence* is a sequence of consecutive better responses. A state S is a *Nash equilibrium* if no player can decrease her delay by changing her strategy, i.e., for all $i \in \mathcal{N}$ and for all $S_i' \in \Sigma_i$, $\delta_i(S) \leq \delta_i(S \oplus S_i')$. We call a congestion game *symmetric* if all players share the same set of strategies, otherwise we call it *asymmetric*.

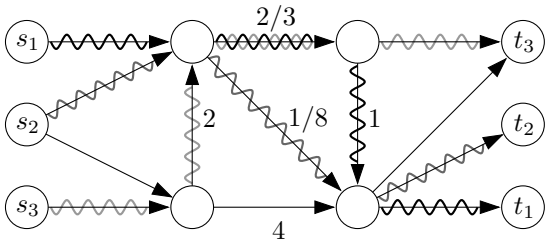
Congestion games can, for example, be used to model routing in large networks like the Internet. In such *network congestion games*, the resources are the edges of a graph $G = (V, E)$ that represents the network and every player $i \in \mathcal{N}$ has to allocate a path from her source node $s_i \in V$ to her target node $t_i \in V$. Figure 1.1.1 shows a network congestion game with three players.

Related Work

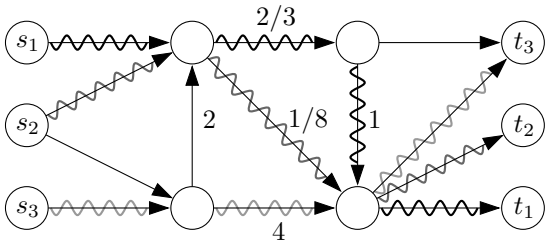
Congestion games are a well-studied model for resource allocation among uncoordinated selfish agents. They are widely used to model routing [FPT04], network design [ADK+04], and load balancing [EDKM03, GLMT04]. Rosenthal [Ros73]



(a) Given the strategies shown in the figure, the delays of players 1, 2, and 3 are 8, 8, and 4, respectively. Both the first and the second player can decrease her delay by changing her path.



(b) This state is obtained by letting player 1 play a best response.



(c) A Nash equilibrium obtained by letting player 3 play a best response.

Figure 1.1.1: In this example, the delay functions d_e are shown in the form a/b, meaning that $d_e(1) = a$ and $d_e(x) = b$ for $x > 1$. If b is missing, the delay function takes value a for all congestions, and edges without without delay functions are assumed to have delay 0.

shows with a potential function argument that every congestion game possesses a pure Nash equilibrium. This argument does not only prove the existence of pure Nash equilibria, but it also shows that if players iteratively play better responses, they eventually reach an equilibrium after a finite number of steps, regardless of the initial state in which they start out. Fabrikant, Papadimitriou, and Talwar [FPT04] show, however, that there are congestion games with initial states from which every better response sequence to an equilibrium has exponential length. In particular, they show that asymmetric network congestion games with this property exist. On the positive side, Ieong et al. [IMN+05] show that in *singleton congestion games*, in which all of the players' strategies consist only of single resources, every best response sequence has polynomial length.

Ackermann, Röglin, and Vöcking [ARV06a] extend this result from singleton to *matroid congestion games*, that is, if the set of strategies of each player consists of the bases of a *matroid* over the set of resources, then the lengths of all best response sequences are polynomially bounded in the number of players and resources. This result holds regardless of the global structure of the game and for

any kind of delay functions. They show that the result is tight in the following sense: any condition on the players' strategy spaces that yields a subexponential bound on the lengths of all best response sequences implies that the strategy spaces are essentially the bases of matroids. In other words, the matroid property is a necessary and sufficient condition on the players' strategy spaces for guaranteeing polynomial time convergence to a Nash equilibrium.

Fabrikant, Papadimitriou, and Talwar [FPT04] observe that the problem of computing a Nash equilibrium in a congestion game can be phrased as a *local search problem*. An instance of a local search problem is given by a set of feasible solutions, an objective function that assigns a value to each solution, and a neighborhood for each solution. The goal is to find a *local optimum*, that is, a solution in whose neighborhood there is no solution with a better objective value. Given a congestion game, we consider the set $\Sigma := \Sigma_1 \times \cdots \times \Sigma_n$ of strategy combinations as the set of feasible solutions, and we define that the neighborhood of a strategy combination $S \in \Sigma$ consists of all strategy combinations $S' \in \Sigma$ that can be obtained from S by letting one player play a better response. If we take Rosenthal's [Ros73] potential function as objective function, this yields a local search problem whose local optima coincide with the Nash equilibria of the congestion game.

The local search problem obtained this way belongs to the complexity class PLS (polynomial local search). This class, defined by Johnson, Papadimitriou, and Yannakakis [JPY88], contains all local search problems with polynomial time-searchable neighborhoods. To be more precise, a local search problem belongs to PLS if it is possible to compute in polynomial time an initial feasible solution, if the objective function can be evaluated in polynomial time, and if it is possible to decide in polynomial time whether a given solution is locally optimal or not and to find a better solution in its neighborhood in the latter case. Fabrikant, Papadimitriou, and Talwar [FPT04] show that the problem of computing Nash equilibria of asymmetric network congestion games is PLS-complete. That is, if it was possible to compute Nash equilibria in polynomial time, then local optima for all problems in PLS could be found efficiently as well, which is not known to be the case.

In contrast to the PLS-completeness result, Fabrikant, Papadimitriou, and Talwar [FPT04] show that for symmetric network congestion games, Nash equilibria can be computed in polynomial time by solving a min-cost flow problem. However, this positive result leaves open the question whether the better or the best response dynamics reaches an equilibrium quickly in symmetric network congestion games.

Our Results

Unfortunately, the reduction in [FPT04] that shows the PLS-completeness for network congestion games is not very instructive. It is based on a PLS-completeness result due to Schäffer and Yannakakis [SY91] for *positive not-all-equal 3-SAT (PosNAE3SAT)*, which is a local search version of the satisfiability problem for a restricted class of formulas. The reduction by Schäffer and Yannakakis is, however, quite involved and, according to Fabrikant, Papadimitriou, and Talwar, possibly the most complex reduction in the literature if one excludes the PCP theorem. Their proof for network congestion games completely reworks the PLS-completeness proof of PosNAE3SAT and even adds some further complications.

We present an alternative approach for proving PLS-hardness of structured congestion games that more directly reveals which kind of substructures cause the trouble. Our approach is based on a very restricted class of congestion games, which we call *threshold congestion games*. In a threshold congestion game, the set of resources is partitioned into two classes, and every player has a unique resource in the first class that only she can allocate and an arbitrarily fixed subset of resources in the second class. Every player has only two possible strategies, namely she can either allocate her private resource in the first class or she can interact with the other players by allocating the fixed subset of resources in the second class. Taking the PLS-completeness of PosNAE3SAT for granted, we present an easy reduction showing that computing Nash equilibria in threshold congestion games is PLS-complete.

Despite their simple structure, threshold games are a natural and interesting class of games. Our main interest, however, stems from the fact that threshold games are a good starting point for PLS-reductions because of their simple structure. We demonstrate the applicability of our approach by showing that threshold games can easily be reduced to network congestion games with directed or undirected edges and linear delay functions, to *market sharing games*, which were introduced by Goemans et al. [GLMT04] to model non-cooperative content distribution in wireless networks, and to *overlay network games*. In the last-mentioned class of congestion games, players have to build a spanning tree on a given subset of nodes that are (virtually) completely connected on the basis of fixed routing paths in an underlying communication network. Formal definitions of market sharing games and overlay network games can be found in Section 2.1.3.

Theorem 1.1.1. *Computing a pure Nash equilibrium in congestion games of the following sorts is* PLS*-complete:*

a) network congestion games with (un)directed edges and non-decreasing linear delay functions

b) market sharing games with polynomially bounded costs

c) overlay network games with linear non-decreasing delay functions

Let us remark that all considered PLS-reductions are so-called *tight* PLS*-reductions*, which implies that they do not only prove the PLS-hardness of the considered classes of games but, in addition, they show that these classes contain instances of games with initial states from which every better response sequence to an equilibrium has exponential length. Furthermore, this kind of reduction implies that it is PSPACE-hard to compute a reachable Nash equilibrium for a given initial state.

Additionally, we show that, in contrast to the PLS-hardness results, the negative results for the convergence time of asymmetric network congestion games directly carry over to the symmetric case.

Theorem 1.1.2. *There exists a family of symmetric network congestion games with initial states from which every better response sequence to a Nash equilibrium has exponential length.*

To the best of our knowledge, this is the first result that separates the class of congestion games with polynomial convergence time from the class of congestion games for which equilibria can be computed efficiently.

1.1.2 Two-Sided Matching Markets

Two-sided matching markets are the second model for resource allocation among selfish agents that we consider. One main function of many real-life markets is to match agents of different kinds to one another, for example men and women, students and colleges [GS62], interns and hospitals [Rot84, Rot96], and firms and workers. Gale and Shapley [GS62] introduced the abstract model of two-sided markets to study these problems. A two-sided market consists of two disjoint groups of agents $\mathcal{X}$ and $\mathcal{Y}$, and each agent has preferences about the agents of the other side and can be matched to one of them. A matching is *stable* if it does not contain any *blocking pair*, that is, a pair of agents from different sides who are not matched to each other but prefer each other to their current matches. Many real-life markets lack a central authority to match agents, and hence, we think of a two-sided market $(\mathcal{X}, \mathcal{Y})$ as a game $G(\mathcal{X}, \mathcal{Y})$ among the agents in $\mathcal{X}$. Each of these agents can propose to one agent from $\mathcal{Y}$, and every agent $y \in \mathcal{Y}$ is matched to the most preferred agent from $\mathcal{X}$ that proposes to him. Stable matchings in a two-sided market $(\mathcal{X}, \mathcal{Y})$ correspond to *pure Nash equilibria* of the corresponding game $G(\mathcal{X}, \mathcal{Y})$, and vice versa. Following the common terminology in two-sided markets, we call agents from $\mathcal{X}$ and $\mathcal{Y}$ *women* and *men*, respectively. If it is more appropriate, we also use the terms *players* and *resources* to denote agents from $\mathcal{X}$ and $\mathcal{Y}$, respectively. Intuitively, as in congestion games, players compete for the resources, the main difference being that in two-sided markets every resource can only be matched to a single player.

Given a matching M and a blocking pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ in M, we say that the matching M' is obtained from M by *resolving* the blocking pair (x, y) if the following holds: $\{x, y\} \in M'$, any partners with whom x and y are matched in M are unmatched in M', and all other edges in M and M' coincide. Consider two agents $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. If a blocking pair (x, y) is resolved, we say that x plays a *better response*. If there does not exist a blocking pair (x, y') such that x prefers y' to y, then we say that x plays a *best response* when the blocking pair (x, y) is resolved.

Related Work and Our Results

Besides their theoretical appeal, two-sided matching models have proven useful in the empirical study of many labor markets such as the National Resident Matching Program (NRMP) [Rot84]. Gale and Shapley [GS62] show that regardless of the preferences of the agents, a stable matching exists in every two-sided market. Furthermore, they give an algorithm that computes such a matching in polynomial time. Since the seminal work of Gale and Shapley, there has been a significant amount of work studying two-sided markets, especially on various extensions of the basic model like *many-to-one markets*, in which every player can be matched to several resources. See for example, the book by Knuth [Knu76], the book by Gusfield and Irving [GI89], or the book by Roth and Sotomayor [RS90].

In many real-life markets, there is no central authority to match agents, and agents are self-interested entities. This motivates the study of *uncoordinated two-sided markets*, first proposed by Knuth [Knu76]. As discussed above, a two-sided market can be seen as a game among agents of one side, and the Nash equilibria in this game correspond to stable matchings, and vice versa. In such uncoordinated

markets, it is an interesting question whether players reach a stable matching. In order to answer this question, Knuth [Knu76] studies the *better response dynamics*, in which players consecutively play better responses. He shows that this dynamics can cycle. Hence, in contrast to congestion games, two-sided markets are not potential games and it might happen that players never reach a stable matching if they play certain better responses in a certain order. Since there is no central coordination that tells the players in which order they should change their strategies, Knuth proposed to study the following *random better response dynamics*: at each step, a blocking pair is picked uniformly at random from the set of all blocking pairs and resolved. Answering a question posed by Knuth, Roth and Vande Vate [RV90] prove that from every matching, there exists a better response sequence to a stable matching. This implies that the random better response dynamics eventually reaches a stable matching with probability 1.

Roth and Vande Vate, however, do not address the question of *convergence time*. We demonstrate that coordination is necessary by constructing examples on which the random better response dynamics needs with high probability exponentially many steps to reach an equilibrium.

Theorem 1.1.3. *There exists a family of two-sided markets with initial matchings from which the random better response dynamics needs in expectation and with high probability exponentially many steps to reach a stable matching.*

Both Knuth's cycle [Knu76] and Roth and Vande Vate's proof [RV90] are only valid for the better response dynamics but not for the *best response dynamics*. We strengthen both results to best responses.

Theorem 1.1.4. *There exists a two-sided market in which the best response dynamics can cycle, but for every two-sided market with n women and m men and every matching M, there exists a sequence of at most $2nm$ best responses starting in M and leading to a stable matching.*

Theorem 1.1.4 implies that the *random best response dynamics*, in which at each point in time a player is picked uniformly at random and allowed to play a best response if she can improve, converges to a stable matching with probability 1. Again, we are interested in the convergence time and show that also for the best response dynamics coordination is necessary.

Theorem 1.1.5. *There exists a family of two-sided markets with initial matchings from which the random best response dynamics needs in expectation and with high probability exponentially many steps to reach a stable matching.*

This lower bound raises the question whether one can find interesting classes of two-sided markets for which the (expected) convergence time is polynomial. In this regard, we consider the natural class of *correlated two-sided markets*, which are inspired from real-life one-sided market games in which players have preferences about a set of resources, and the preferences of resources are correlated with the preferences of players. In a correlated two-sided market, there is a payoff associated with every possible pair from $\mathcal{X} \times \mathcal{Y}$. Both players and resources are interested in maximizing their payoff, that is, an agent i prefers an agent j to an agent j' if the payoff associated with pair (i, j) is larger than the payoff associated with pair (i, j').

This special class of two-sided markets is also studied by Abraham et al. [ALMO07] who consider complexity related questions for the *roommates problem*, in which one looks for a stable matching in a graph that is not necessarily bipartite. We obtain the following result for correlated two-sided markets.

Theorem 1.1.6. *Correlated two-sided markets are potential games and the random best response dynamics reaches in expected polynomial time a stable matching.*

We extend this result to *many-to-one two-sided markets* in which every player can be matched to a subset of resources [RS90]. These markets are extensively studied in labor markets [KC82, RS90, Fle03, EO06] in which an employer is interested in hiring a set of employees. Each employer has some preferences about subsets of employees, and each employee has preferences about the employers. We consider *matroid two-sided markets* in which the feasible subsets of employees for an employer are independent sets of a matroid and essentially extend Theorem 1.1.6 to that case. Two-sided matroid markets arise naturally if, for example, every employer is interested in hiring a fixed number of workers or if the workers can be partitioned into different classes and a certain number of workers from each class is to be hired. Appendix A contains a brief introduction to matroids.

1.1.3 Congestion Games with Priorities

We conclude the part about resource allocation among selfish agents by introducing a natural class of resource sharing games that extends both congestion games and two-sided markets. We prove several results for this model that unify the theory of these two special cases.

One drawback of the standard model of congestion games is that resources do not have any preferences about the players. In typical load balancing applications, however, different jobs have different priorities, and depending on the policy, jobs with a low priority are stopped or slowed down when jobs with higher priorities are present. We introduce *congestion games with priorities* to model the scenario in which a job can prevent jobs with lower priorities from being processed. In our model, each resource can partition the set of players into classes of different priorities. As long as a resource is only allocated by players with the same priority, these players incur a delay depending on the congestion, as in standard congestion games. But if players with different priorities allocate a resource, only players with the highest priority incur a delay depending on the number of players with this priority, and players with lower priorities incur an infinite delay. Intuitively, they are displaced by the players with the highest priority. This model is applicable if every player controls a stream of jobs rather than only a single one. In the latter case, it might be more reasonable to assume that jobs with lower priorities incur a large but finite delay.

Formally, the games that we consider are congestion games as defined in Section 1.1.1 in which each resource $r \in \mathcal{R}$ assigns a *priority* or *rank* $\mathrm{rk}_r(i)$ to every player $i \in \mathcal{N}$. For a state S, let $\mathrm{rk}_r(S) = \max_{i:r\in S_i} \mathrm{rk}_r(i)$ denote the *rank of the resource* r. We say that player i *allocates* resource r if $r \in S_i$, and we say that player i is *assigned* to resource r if $r \in S_i$ and $\mathrm{rk}_r(i) = \mathrm{rk}_r(S)$. We define $n_r^*(S)$ to be the number of players that are assigned to resource r, that is, the number of players i with $r \in S_i$ and $\mathrm{rk}_r(i) = \mathrm{rk}_r(S)$. The delay that an assigned player i

incurs on r is $d_r(n_r^*(S))$. Players who allocate a resource but are not assigned to it incur an infinite delay. We say that the priorities are *consistent* if the priorities assigned to the players coincide for all resources.

A well-studied extension of standard congestion games are *player-specific congestion games*, in which different players can have different delay functions for the same resource. To be precise, every player $i \in \mathcal{N}$ has her own delay function $d_r^i \colon \mathbb{N} \to \mathbb{N}$ for each resource $r \in \mathcal{R}$, and the delay that player $i \in \mathcal{N}$ incurs in state S is defined to be $\sum_{r \in S_i} d_r^i(n_r(S))$. In its most general form, our model also covers player-specific congestion games with priorities.

Interestingly, our model of player-specific congestion games with priorities does not only extend congestion games but also the model of two-sided matching markets. In the same way as it is in many situations not realistic to assume that in congestion games the resources have no preferences about the players, it is in two-sided markets often unrealistic to assume that the preference lists of the resources are strict. Our model of player-specific congestion games with priorities can also be seen as a model of *two-sided markets with ties* in which several players can be assigned to one resource. If different players propose to a resource, only the most preferred ones are assigned to it. If the most preferred player is not unique, several players share the *payoff* of the resource. Such two-sided markets correspond to our model of congestion games with priorities, except that players are now interested in maximizing their payoffs instead of minimizing their delays, which does not affect our results.

One application of our model are markets into which different companies can invest. As long as the investing companies are of comparable size, they share the payoff of the market, but large companies can utilize their market power to eliminate smaller companies completely from the market. Player-specific congestion games and two-sided markets are the special cases of our model in which all players are assigned the same priority or distinct priorities, respectively. In the following, we use the terms *two-sided market with ties* and *player-specific congestion game with priorities* interchangeably.

Previous Work and Our Results

Milchtaich [Mil96] introduces player-specific congestion games. He shows that, like two-sided markets, player-specific singleton congestion games are not potential games but that they possess pure Nash equilibria that can be computed in polynomial time. His proof is constructive in the sense that it also shows that from every state there is a polynomially long sequence of better responses to a Nash equilibrium. Ackermann, Röglin, and Vöcking [ARV06b] extend these results to player-specific matroid congestion games, in which the strategy space of each player consists of the bases of a matroid.

Two-sided markets with ties are extensively studied in the literature (see, for example, the aforementioned books [Knu76, GI89, RS90] or [IMMM99]). In the models studied in the literature, ties are usually somehow broken, that is, despite ties in the preference lists, every resource can be assigned to at most one player. Hence, these models differ significantly from our model.

Motivated by the application of congestion games to load balancing, we mainly consider congestion games in which each player has to choose exactly one resource

to allocate, namely one server on which her job is to be processed. We show that singleton congestion games with priorities are potential games, implying that uncoordinated players who iteratively play better responses eventually reach a pure Nash equilibrium. If the priorities of the resources are consistent, then we even obtain polynomial-time convergence to a Nash equilibrium.

Theorem 1.1.7. *Singleton congestion games with priorities are potential games. If the priorities are consistent, the random better response dynamics reaches a Nash equilibrium in polynomial time in expectation.*

For player-specific congestion games with priorities, we show that pure Nash equilibria exist that can be computed in polynomial time. The proof of this result unifies arguments for player-specific congestion games [ARV06b] and for two-sided markets [GS62].

Theorem 1.1.8. *Every player-specific singleton congestion game with priorities possesses a pure Nash equilibrium that can be computed in polynomial time.*

Additionally, we show that this result can also be extended to player-specific *matroid* congestion games with priorities.

1.2 Pareto-Optimal Solutions

People, companies, and other economic entities face decisions every day. Many of these decisions cannot be formulated as combinatorial optimization problems in the usual sense because there are several conflicting objectives to be optimized and one has to find an appropriate *trade-off* rather than a solution that optimizes a single criterion. Of course, for most problems the notion of an "appropriate trade-off" is hard to formalize and often there is no consensus on how different criteria should be traded off against each other. However, there is little disagreement that a trade-off can only be reasonable if no criterion can be improved without deteriorating at least one of the other criteria. Trade-offs with this property are called *efficient* or *Pareto-optimal* and they play a crucial role in multi-criteria decision making as they help to filter out unreasonable solutions.

We focus on the study of *integer optimization problems* in which there are two criteria, say cost and weight, that are to be minimized. An instance of such a problem is described by a set $\mathcal{S} \subseteq \mathcal{D}^n$ of feasible solutions and by the two objective functions cost $c\colon \mathcal{S} \to \mathbb{R}$ and weight $w\colon \mathcal{S} \to \mathbb{R}$, where n denotes the number of integer variables and $\mathcal{D} \subset \mathbb{Z}$ denotes a finite set of integers. The assumption that both criteria are to be minimized is without loss of generality as our results are not affected by a change of the optimization direction of one of the objective functions. In this setting, a solution is Pareto-optimal if there does not exist another solution with at most the same cost and at most the same weight and with one of these inequalities being strict. We call the set of Pareto-optimal solutions the *Pareto set* and we assume that it is reduced, that is, if there are two Pareto-optimal solutions with the same cost and the same weight, then the Pareto set contains only one of them. Figure 1.2.1 shows an example for a Pareto set.

This framework covers a very broad class of bicriteria optimization problems. Already in the binary case (i.e., $\mathcal{D} = \{0, 1\}$), it covers naturally many standard

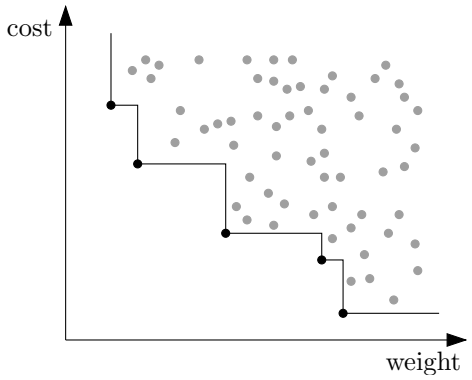


Figure 1.2.1: The solutions from $\mathcal{S}$ are represented by dots. Pareto-optimal solutions are black, and dominated solutions are gray.

problems like the *bicriteria shortest path problem* and the *bicriteria spanning tree problem* by introducing a decision variable for each edge and by letting $\mathcal{S}$ encode the set of all paths or spanning trees, respectively. Another important problem that can be formulated in this framework is the *knapsack problem*, in which a capacity t and set of n items with profits $p_1, \ldots, p_n$ and weights $w_1, \ldots, w_n$ are given. Usually, the goal is to find the most profitable subset of the items whose total weight is at most t, but if one ignores the capacity and tries to maximize the profit and to minimize the weight, the knapsack problem becomes a bicriteria problem. The general case, in which $\mathcal{D}$ is an arbitrary finite subset of $\mathbb{Z}$, covers, for example, the *bicriteria network flow problem* and the *bounded knapsack problem*, in which there are several identical copies of each item.

A well-studied way to formalize the notion of "most appropriate trade-off" is to specify a function that combines the objectives into a single criterion. For an instance of a bicriteria integer optimization problem, we can, for example, specify a function $f\colon \mathbb{R}^2 \to \mathbb{R}$ that assigns to each solution $x \in \mathcal{S}$ a value $f(c(x), w(x))$, which is without loss of generality to be minimized. Under the mild assumption that the function f is non-decreasing in both components, the function f is minimized by (at least) one of the Pareto-optimal solutions. Hence, a possible and indeed very common approach to find a solution that optimizes such a function f is to generate the set of Pareto-optimal solutions and to select the one with minimum value. For a bicriteria problem Π and a function f, we call the resulting single-criterion problem the f-Π *problem*.

1.2.1 Smoothed Analysis

Even though generating the Pareto set is of great interest in many scenarios and widely used in practice, this approach fails to yield reasonable results in the worst case because even integer optimization problems with a simple combinatorial structure (including all aforementioned examples) can have exponentially many Pareto-optimal solutions. In practice, however, generating the Pareto set is often feasible because typically the number of Pareto-optimal solutions does not attain its worst-case bound. The discrepancy between practical experience and worst-case results motivates the study of the number of Pareto-optimal solutions in a more realistic input model. One possible approach is to study the average number of

Pareto-optimal solutions rather than the worst-case number. In order to analyze the average, one has to define a probability distribution on the set of instances with respect to which the average is taken. In most situations, however, it is not clear how to choose a probability distribution that reflects typical inputs. In order to bypass these problems, Spielman and Teng [ST04] introduced the notion of *smoothed analysis*. They consider a *semi-random input model* in which an adversary specifies an input that is subsequently slightly *perturbed* at random. Often semi-random input models are more realistic than worst-case or average-case input models because the adversary can specify an arbitrary input with a certain structure and the subsequent *perturbation* generates an instance that is still close to the adversarial one but additionally possesses a small amount of randomness.

Inspired by Spielman and Teng's model of smoothed analysis, we consider a semi-random input model for bicriteria integer optimization problems. In this model, an adversary can choose an arbitrary set $\mathcal{S} \subseteq \mathcal{D}^n$ of feasible solutions, an arbitrary weight function $w\colon \mathcal{S} \to \mathbb{R}$ and an arbitrary linear cost function $c\colon \mathcal{S} \to \mathbb{R}$ of the form $c(x) = c_1x_1 + \cdots + c_nx_n$ with $c_1, \ldots, c_n \in [-1, 1]$. In the following, we refer to the coefficients $c_1, \ldots, c_n$ as *costs*. The restriction to linear cost functions is not unrealistic as in most applications (including all aforementioned examples) at least one objective function is linear. Furthermore, the restriction to the interval $[-1, 1]$ can simply be achieved by scaling the coefficients appropriately. The weight function and the set of feasible solutions are not perturbed, which is important as the set $\mathcal{S}$ might encode the combinatorial structure of the problem (e.g., $\mathcal{S}$ can encode the set of spanning trees of some graph). Only the cost function is randomly perturbed by adding an independent *Gaussian random variable* with mean 0 and standard deviation σ to each cost c_i.

The standard deviation σ can be seen as a parameter measuring how close the analysis is to a worst-case analysis: the smaller σ is chosen, the smaller is the influence of the perturbation and hence, the closer is the analysis to a worst-case analysis. Our probabilistic analysis is not restricted to Gaussian perturbations but is much more general. In fact, it covers arbitrary probability distributions with bounded density functions and finite absolute mean values. In particular, if one is interested in obtaining a positive domain for the costs, one can restrict the adversary to costs $c_i \in [0, 1]$ and perturb them by adding independent random variables that are uniformly distributed over some interval $[0, a]$. For such uniform distributions, the standard deviation σ is proportional to the interval length a, and choosing $a = \sqrt{12} \cdot \sigma$ yields a standard deviation of σ. For the sake of a simple presentation, we restrict ourselves to Gaussian and uniform random variables in this introduction.

We analyze the expected number of Pareto-optimal solutions and algorithms for enumerating the Pareto set in the semi-random input model described above. One crucial parameter that appears in the bound on the number of Pareto-optimal solutions and in the bounds on the running times of the algorithms is the standard deviation σ. We have argued above that one can interpolate between worst-case and average-case analysis by adjusting this parameter, and hence, it is not surprising that the expected number of Pareto-optimal solutions and the expected running times of the algorithms increase when σ becomes smaller. However, the bounds we prove grow only polynomially in the input size and in σ^{-1}. This implies that the instances with a large number of Pareto-optimal solutions are isolated peaks

and very fragile to random influences. Hence, it is unlikely that such instances occur in practice. Before we present our results in more detail, we discuss related work in the field of multi-objective optimization and smoothed analysis.

1.2.2 Related Work

Multi-objective optimization is a very active and well-studied research area, and hence, it is by far not possible to present a complete overview of the existing literature. For this, we refer the reader to the book by Ehrgott [Ehr00] and the survey collection by Ehrgott and Gandibleux [EG02]. In the following, we first give an overview of the literature on multi-objective optimization that is most relevant to our research, and then we discuss related work in the area of smoothed analysis.

Multi-Objective Optimization

There exists a vast body of literature that focuses on multi-objective optimization. In particular, many algorithms for generating the Pareto set of various optimization problems such as the (bounded) knapsack problem [NU69, KW00], the bicriteria shortest path problem [CM85, Han80, SA00], and the bicriteria network flow problem [Ehr99, MG98] have been proposed. Since for all these problems the number of Pareto-optimal solutions can be exponential (see, e.g., [Ehr00]), none of these algorithms runs in polynomial time in the worst case. In practice, however, generating the Pareto set is tractable in many situations. For instance, Müller-Hannemann and Weihe [MHW01] study experimentally the number of Pareto-optimal solutions in multi-criteria shortest path problems. They consider examples that arise from computing the set of best train connections (in view of travel time, fare, and number of train changes) and conclude that in this application scenario generating the complete Pareto set is tractable even for large instances.

One way of coping with the bad worst-case behavior is to relax the requirement of finding the complete Pareto set. A solution x is *ε-dominated* by another solution x' if $c(x')/c(x) \leq 1+\varepsilon$ and $w(x')/w(x) \leq 1+\varepsilon$. We say that $\mathcal{P}_\varepsilon$ is an *ε-approximation of a Pareto set $\mathcal{P}$* if for any solution $x \in \mathcal{P}$, there is a solution $x' \in \mathcal{P}_\varepsilon$ that ε-dominates it. In his pioneering work, Hansen [Han80] presents an approximation scheme for computing ε-approximate Pareto sets of the bicriteria shortest path problem. Papadimitriou and Yannakakis [PY00] show that for any Pareto set $\mathcal{P}$, there is an ε-approximation of $\mathcal{P}$ with polynomially (in the input size and $1/\varepsilon$) many points. Furthermore, they define the *gap version* of a multi-criteria optimization problem with d objectives as follows: given an instance and a vector $b \in \mathbb{R}^d$, either return a solution whose objective vector dominates b or report (correctly) that there does not exist any solution whose objective vector is better than b by more than a $(1+\varepsilon)$ factor in all coordinates. They show that an FPTAS for approximating the Pareto set of a multi-criteria optimization problem exists if and only if the gap version of the problem can be solved in polynomial time. In particular, this implies that if the exact single-criterion version of a problem (i.e., the question "Is there a solution with weight exactly x?") can be solved in pseudopolynomial time, then its multi-criteria version admits an FPTAS for approximating the Pareto set. There are pseudopolynomial time algorithms for the exact versions of the spanning tree problem [BP87], the shortest path

problem [PY82], and the perfect matching problem [MVV87], and hence, for the multi-criteria versions of these problems FPTAS for approximating the Pareto set exist. Vassilvitskii and Yannakakis [VY05] show how to compute ε-approximate Pareto sets whose size is at most three times as large as the smallest such set for bicriteria problems whose gap versions can be solved in polynomial time. Diakonikolas and Yannakakis [DY07] improve this factor to two and show that this is the best possible that can be achieved in polynomial time.

A lot of research has been conducted on f-Π problems. For instance, it is well known that if f is a concave function, an optimal solution of the f-Π problem can be found on the border of the convex hull of the solutions [HT90]. For some problems, there are algorithms generating this set of solutions. In particular, for the spanning tree problem it is known that there are only polynomially many solutions on the border of the convex hull [Dey97], and efficient algorithms for enumerating them exist [AEGH98]. Thus, there is a polynomial-time algorithm for solving the f-spanning tree problem for any concave function f. Katoh [Kat92] describes how one can use f-spanning tree problems with concave objective functions to solve many other problems in combinatorial optimization. For instance, a well-studied application is the minimum cost reliability spanning tree problem, where one is interested in finding a spanning tree minimizing the ratio of cost to reliability. It is also known how to solve the f-shortest path problem for functions f being both pseudoconcave and pseudoconvex in polynomial time [Hen86]. Tsaggouris and Zaroliagis [TZ04] investigate the non-additive shortest path problem, which is to find a path P minimizing $f_c(c(P)) + f_w(w(P))$ for some convex functions f_c and f_w. This problem arises as core problem in different applications, e.g., in the context of computing traffic equilibria. They develop exact algorithms with exponential running time using a Lagrangian relaxation. Ackermann et al. [ANRV07] show that the f-shortest path, the f-spanning tree, and the f-perfect matching problem are NP-hard even for simple polynomial functions f. They show that an FPTAS for approximating the Pareto set can be transformed into an FPTAS for the f-Π problem if f has at most polylogarithmic elasticity, which includes, for instance, polynomials and functions of the form $f(c, w) = c^{\text{polylog}(c)} + w^{\text{polylog}(w)}$. Furthermore, they show that the restriction to polylogarithmic elasticity is necessary.

Smoothed Analysis

In a seminal paper, Spielman and Teng [ST04] introduce the notion of smoothed analysis and analyze the running time of the simplex algorithm for solving linear programs in this framework. For most deterministic pivot rules that have been suggested, examples are known showing that in the worst case, the simplex algorithm can take an exponential number of steps, but the simplex algorithm is still one of the most competitive algorithms for solving linear programs in practice. It is fast and reliable even for large-scale instances and for the pivot rules that have been shown to require an exponential number of iterations in the worst case. Examples on which the simplex algorithm needs many iterations occur only very rarely in practice. In Spielman and Teng's semi-random model, first an adversary specifies an arbitrary linear program in which the coefficients in the constraint matrix and the thresholds are chosen from the interval $[-1, 1]$. In the second step of

their input model, the coefficients in the constraint matrix and the thresholds are perturbed by adding an independent Gaussian random variable with mean 0 and standard deviation σ to each of them. Spielman and Teng show that the expected running time of the simplex algorithm is polynomial in the input size and σ^{-1} for the shadow-vertex pivot rule.

Since the invention of smoothed analysis in 2001, many different results on the smoothed analysis of algorithms have been obtained, including improved analyses of the simplex algorithm and results on different algorithms for solving linear programs, local search algorithms, various discrete optimizations problems, and the competitive ratio of online algorithms. An overview of these results can be found on the smoothed analysis homepage [Spi].

Most relevant to our research are two results by Beier and Vöcking. In the first article [BV04], they analyze the expected number of Pareto-optimal solutions for binary optimization problems. They consider the aforementioned semi-random model with $\mathcal{D} = \{0, 1\}$ and show that the expected number of Pareto-optimal solutions is bounded from above by $O(n^4/\sigma)$, no matter how $\mathcal{S}$ and w are chosen. This result implies that the Nemhauser/Ullmann algorithm [NU69], which solves the knapsack problem by first enumerating the Pareto set and then selecting the most profitable Pareto-optimal solution satisfying the capacity constraint, has a polynomial expected running time of $O(n^5/\sigma)$. Furthermore, they present a lower bound of $\Omega(n^2)$ on the expected number of Pareto-optimal solutions for profits that are chosen uniformly at random from the interval $[0, 1]$.

In the second article [BV06], Beier and Vöcking analyze the *smoothed complexity of binary optimization problems*. They consider optimization problems with one objective function $c\colon \mathcal{S} \to \mathbb{R}$ in which the set of feasible solutions is given as $\mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_m$, where $\mathcal{S} \subseteq \{0, 1\}^n$ denotes a fixed ground set and $\mathcal{B}_i$ denotes a halfspace induced by a linear constraint of the form $w_{i,1}x_1 + \cdots + w_{i,n}x_n \leq t_i$. Similar to the aforementioned model, it is assumed that the coefficients $w_{i,j}$ are perturbed by adding independent random variables to them. This model covers, for example, the constrained shortest path problem, in which one looks for the cheapest path whose weight lies below a given threshold. In this example, $\mathcal{S}$ encodes the set of all paths, which is not affected by the perturbation, and $\mathcal{B}_1$ encodes all subsets of edges whose total weight lies below the threshold. Based on the probabilistic analysis of certain structural properties, Beier and Vöcking show that a binary optimization problem in this form has *polynomial smoothed complexity* if and only if there exists a pseudopolynomial time algorithm (w.r.t. the coefficients $w_{i,j}$) for solving the problem. This characterization is extended to the case of integer optimization problems, where $\mathcal{D} \subset \mathbb{Z}$ is a finite set of integers, by Röglin and Vöcking [RV07]. The term polynomial smoothed complexity is defined analogously to the way polynomial complexity is defined in average-case complexity theory [Wan97], adding the requirement that the running time should be polynomially bounded not only in the input size but also in σ^{-1}.

1.2.3 Our Results

We present a new approach for bounding the expected number of Pareto-optimal solutions for bicriteria integer optimization problems. This approach yields the first bounds for integer optimization problems and improves the known bound

for the binary case significantly. In the following, we assume that the perturbed coefficients are chosen from the interval $[-1, 1]$ by the adversary and that they are perturbed by adding independent Gaussian or uniform random variables with constant expected absolute value and standard deviation at least σ. For the sake of simplicity, we do not present our results in their most general form in this introduction. In fact, our analysis covers arbitrary probability distributions with bounded density functions and finite expected absolute values and arbitrary finite sets $\mathcal{D} \subset \mathbb{Z}$.

Theorem 1.2.1. *Let $\mathcal{D} = \{0, \ldots, k-1\}$ and let $\mathcal{S} \subseteq \mathcal{D}^n$ and $w\colon \mathcal{S} \to \mathbb{R}$ be chosen arbitrarily. If the costs are randomly perturbed, the expected number of Pareto-optimal solutions is bounded from above by $O(n^2k^2 \log(k)/\sigma)$.*

Additionally, we prove two lower bounds on the expected number of Pareto-optimal solutions.

Theorem 1.2.2. *For $\mathcal{D} = \{0, \ldots, k-1\}$, there exist instances of bicriteria integer optimization problems in which the expected number of Pareto-optimal solutions is $\Omega(n^2k^2)$ if the costs are chosen uniformly at random from the interval $[-1, 1]$.*

The proof of this lower bound uses crucially the fact that the adversary can choose arbitrary weight functions $w\colon \mathcal{S} \to \mathbb{R}$. If we restrict the adversary to linear weight functions, the lower bound becomes weaker.

Theorem 1.2.3. *For $\mathcal{D} = \{0, \ldots, k-1\}$, there exist instances of bicriteria integer optimization problems with linear weight functions in which the expected number of Pareto-optimal solutions is $\Omega(n^2k \log k)$ if the costs are chosen uniformly at random from the interval $[-1, 1]$.*

For the binary case $\mathcal{D} = \{0, 1\}$, the upper bound in Theorem 1.2.1 simplifies to $O(n^2/\sigma)$, which improves the previously known bound by a factor of $\Theta(n^2)$ and matches the lower bound due to Beier and Vöcking [BV04] in terms of n. Hence, our method yields tight bounds in terms of n and almost tight bounds in terms of k for the expected number of Pareto-optimal solutions, and thereby even simplifies the proof.

Applications of the Results

Before we describe the algorithmic applications of our results, we have to specify how the perturbed numbers in the input are encoded. Since the distributions are continuous, these numbers are irrational with probability 1. In the following, we assume that the bits after the binary point can be accessed by asking an oracle in time $O(1)$ per bit. The bits are revealed one by one from left to right. Instead of considering this oracle model, one can also quantize the distributions because our results are not significantly affected if every random number is rounded to the next number that can be represented by a polynomial number of bits.

The *Nemhauser/Ullmann algorithm* solves the knapsack problem by enumerating all Pareto-optimal solutions [NU69]. Its running time on an instance with n items is $\Theta(\sum_{i=1}^{n} q_i)$, where q_i denotes the number of Pareto-optimal solutions of the knapsack instance that consists only of the first i items. Using Beier and

Vöcking's [BV04] upper bound on the expected number of Pareto-optimal solutions yields a bound of $O(n^5/\sigma)$ for the expected running time of this algorithm on semi-random instances in which the profits or weights are randomly perturbed. Based on our improved bound on the expected number of Pareto-optimal solutions, we conclude the following corollary.

Corollary 1.2.4. *For semi-random knapsack instances in which the profits or weights are perturbed, the expected running time of the Nemhauser/Ullmann algorithm is $O(n^3/\sigma)$.*

For uniformly distributed profits, Beier and Vöcking present a lower bound on the expected running time of $\Omega(n^3)$. Hence, our upper bound on the expected running time of the Nemhauser/Ullmann algorithm is tight in terms of the number of items n.

In the bounded knapsack problem, a number $k \in \mathbb{N}$ and a set of n items with profits and weights are given. It is assumed that k instances of each of the n items are given. An instance of the bounded knapsack problem with n items can be transformed into an instance of the (binary) knapsack problem with $\Theta(n \log(k+1))$ items [KPP04]. Using this transformation, the bounded knapsack problem can be solved by the Nemhauser/Ullmann algorithm in running time $\Theta(\sum_{i=1}^{n \log(k+1)} q_i)$, where q_i denotes the number of Pareto-optimal solutions of the binary knapsack instance that consists only of the first i items. Based on our results on the expected number of Pareto-optimal solutions, we obtain the following corollary.

Corollary 1.2.5. *The expected running time of the Nemhauser/Ullmann algorithm on semi-random bounded knapsack instances in which the profits or weights are perturbed is bounded from above by $O(n^3k^2(\log^2(k+1))/\sigma)$ and bounded from below by $\Omega(n^3k\log^2(k+1))$.*

Different algorithms have been proposed for enumerating the Pareto set in bicriteria shortest path problems [CM85, SA00]. Beier [Bei04] shows that the expected running time of a modified version of the Bellman/Ford algorithm [CM85] is $O(nm^5/\sigma)$ for graphs with n nodes and m edges. Based on our improved analysis of the expected number of Pareto-optimal solutions, his proof yields the following corollary.

Corollary 1.2.6. *For semi-random bicriteria shortest path problems in which one objective function is linear and its coefficients are perturbed, the expected running time of the modified Bellman/Ford algorithm is $O(nm^3/\sigma)$*

Enumeration of the Pareto Set

For certain problems such as the bicriteria spanning tree problem, there are no algorithms known for enumerating the Pareto set efficiently with respect to its size. We present a method that allows us to enumerate the set of Pareto-optimal solutions for semi-random inputs with a small failure probability for all problems for which this set can be enumerated in pseudopolynomial time. Together with the bound on the expected number of Pareto-optimal solutions, this result yields

heuristics with polynomial running time and small failure probability for enumerating the set of Pareto-optimal solutions for semi-random instances of these problems. Our approach works if the domain is binary (i.e., $\mathcal{D} = \{0, 1\}$), if both the cost function and the weight function are linear with perturbed coefficients, and if all weights are non-negative.

Theorem 1.2.7. *Let* Π *be a bicriteria binary optimization problem and assume that both the weight and the cost function are linear. If there exists a pseudopolynomial time (with respect to weights and costs) algorithm for generating the reduced Pareto set of* Π*, then there exists an algorithm for generating the Pareto set of* Π *on semi-random inputs with failure probability at most* p *and running time* $\mathrm{poly}(N, \sigma^{-1}, p^{-1})$*, where* N *denotes the input size.*

It can easily be seen that for any bicriteria problem Π, a pseudopolynomial algorithm for the exact and single objective version of Π can be turned into an algorithm with pseudopolynomial worst-case complexity for generating the Pareto set. Therefore, in the smoothed model, there exists a polynomial-time algorithm for enumerating the Pareto set of Π with small failure probability if there exists a pseudopolynomial algorithm for the exact and single objective version of Π. Furthermore, given the exact Pareto set for a problem Π, one can solve the f-Π problem for any non-decreasing function f exactly. Thus, in our semi-random model, we can, for example, find spanning trees that minimize functions that are hard to approximate within any factor in the worst case.

Smoothed Complexity of Integer Programming

In [RV07] we show that an integer optimization problem has *polynomial smoothed complexity* if and only if there exists a pseudopolynomial time algorithm for solving the problem. To be more precise, we consider integer optimization problems in which an objective function $c\colon \mathcal{S} \to \mathbb{R}$ is to be minimized over a feasible region that is defined as the intersection of a fixed ground set $\mathcal{S} \subseteq \mathcal{D}^n$ with halfspaces $\mathcal{B}_1, \ldots, \mathcal{B}_m$ that are induced by m linear constraints of the form $w_{i,1}x_1 + \cdots + w_{i,n}x_n \leq t_i$, where the $w_{i,j}$ are independently perturbed by adding Gaussian or uniformly distributed random variables with standard deviation σ to them. It is shown that an algorithm whose worst-case running time is pseudopolynomial with respect to the absolute values $|w_{i,j}|$ of the coefficients can be turned into an algorithm with polynomial smoothed complexity. This analysis is based on the probabilistic analysis of certain structural properties of semi-random integer optimization problems. Based on the upper bound on the expected number of Pareto-optimal solutions, this analysis can be significantly improved and simplified.

The term *polynomial smoothed complexity* is defined such that it is robust under different machine models analogously to the way polynomial average-case complexity is defined [BV06, Wan97]. One disadvantage of this definition is that polynomial smoothed and average-case complexity do not imply expected polynomial running time. For the binary case, it is shown [BV06] that problems that admit a pseudolinear algorithm, that is, an algorithm whose running time is bounded by $O(\mathrm{poly}(N) \cdot W)$, where N denotes the input size and W the largest coefficient $|w_{i,j}|$ in the input, can be solved in expected polynomial time in the

smoothed model. Based on the improved analysis of the aforementioned structural properties, we generalize this result to the integer case.

Theorem 1.2.8. *Every integer optimization problem that can be solved in time $O(\text{poly}(N) \cdot W)$, where N denotes the input size and $W = \max_{i,j} |w_{i,j}|$, admits an algorithm with expected polynomial (in N and σ^{-1}) running time for perturbed instances, in which an independent Gaussian or uniformly distributed random variable with standard deviation σ is added to each coefficient $w_{i,j}$.*

1.3 Local Optima

In the last part of this thesis, we turn our attention back to classical combinatorial optimization problems in which decisions are made by a single decision maker and in which there is only one objective function to be optimized. Even without the complications of selfish agents and multiple objectives, many combinatorial optimization problems that are relevant for practical applications cannot be solved efficiently on large-scale instances, unless $\mathsf{P} = \mathsf{NP}$. The notion of NP-hardness is a worst-case measure, and hence, it sometimes yields overly pessimistic results, as already discussed in Section 1.2 for the number of Pareto-optimal solution. Many NP-hard combinatorial optimization problems are, however, not only hard in the worst case but also on typical instances. For example, for the important and well-studied *traveling salesperson problem (TSP)*, no algorithms are known that are efficient on large-scale instances that occur in practical applications. For such problems, one has to relax the requirement of finding an optimal solution and one has to apply different solution concepts.

A very common way to deal with NP-hard optimization problems is to compute solutions that are only approximately optimal. An algorithm is said to guarantee an *approximation ratio* of α if it computes on any instance a solution whose value differs from the optimal value by a factor of at most α. Approximation algorithms have been studied extensively in the theory of combinatorial optimization (see, e.g., [Vaz04]) and many breakthroughs have been achieved. However, for various problems like the TSP it turned out that the approximation algorithms that yield the best performance in theory are clearly outperformed in practice by algorithms with worse approximation and running time guarantees. The reason for this discrepancy is that also the theory of approximation algorithms is based on a worst-case notion, which is too pessimistic for many algorithms.

In practice, *local search* is a very important technique for designing approximation algorithms. Local search was introduced in the late 1950s for the TSP and is based on a very simple idea: start with an arbitrary solution and improve this solution consecutively until a solution is reached that cannot be improved further by a *local modification*. To be more precise, given an instance I of a combinatorial optimization problem Π, we associate with every feasible solution $x \in \mathcal{S}(I)$ a set $\mathcal{N}(I, x) \subseteq \mathcal{S}(I)$ of solutions that can be reached from x by a local modification. By this, we turn Π into a local search problem as defined in Section 1.1, and the solutions that cannot be improved further by local modifications are exactly the local optima. Of course, the definition of local modification is problem-specific, and even for a single problem, there are usually several reasonable ways to define local modifications. Many experimental studies for different problems show that local

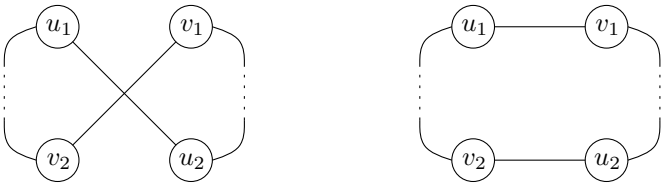


Figure 1.3.1: An improving 2-change: the edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$ are replaced by the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$.

search is a robust and powerful tool for obtaining good approximations quickly if the notion of local modification is defined in the right way. For these problems, *local optima* are a solution concept that forms a good compromise between quality and efficiency.

We study 2-Opt, a local search algorithm for the traveling salesperson problem, because the TSP is one of the best-studied combinatorial optimization problems and local search is particularly successful for this problem. In an instance of the TSP, we are given a set of *vertices* and for each pair of distinct vertices a distance. The goal is to find a *tour* of minimum length that visits every vertex exactly once and returns to the initial vertex at the end. The *2-Opt* algorithm is probably the most basic local search heuristic for the TSP. It starts with an arbitrary initial tour and incrementally improves this tour by making successive improvements that exchange two of the edges in the tour with two other edges. More precisely, in each *improving step* the 2-Opt algorithm selects two edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$ from the tour such that u_1, u_2, v_1, v_2 are distinct and appear in this order in the tour, and it replaces these edges by the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$, provided that this change decreases the length of the tour (see Figure 1.3.1 for an illustration). The algorithm terminates in a local optimum in which no further improving step is possible. We use the term *2-change* to denote a local improvement made by 2-Opt.

1.3.1 Related Work

The idea of using local search for the TSP dates back to the late 1950s. In 1958 Croes [Cro58] and Bock [Boc58] suggested variants of 2-Opt and 3-Opt, respectively. After that, a lot of research has been conducted in this area and different local search algorithms for the TSP have been suggested and experimentally evaluated. For more details, we refer the reader to a survey article by Johnson and McGeoch [JM97]. Nowadays, local search is applied to a wide range of problems like, e.g., scheduling [AGP97], graph partitioning [KL70], and pattern matching [RL01].

Despite many theoretical analyses and experimental evaluations of the TSP, there is still a considerable gap between the theoretical results and the experimental observations. One important special case is the *Euclidean TSP* in which the vertices are points in $\mathbb{R}^d$ for some $d \in \mathbb{N}$ and the distances are measured according to the Euclidean metric. This special case is known to be NP-hard in the strong sense [Pap77], but it admits a polynomial time approximation scheme (PTAS), shown independently in 1996 by Arora [Aro98] and Mitchell [Mit99]. These approximation schemes are based on dynamic programming. The most successful

algorithms on practical instances, however, rely on the principle of local search. The 2-Opt heuristic performs amazingly well on real-world Euclidean instances like, e.g., the ones in the well-known TSPLIB [Rei91]. Usually the 2-Opt heuristic needs a clearly subquadratic number of improving steps until it reaches a local optimum and the computed solution is within a few percentage points of the global optimum [JM97].

There are numerous experimental studies on the performance of 2-Opt. However, the theoretical knowledge about this heuristic is still very limited. Let us first discuss the number of local improvement steps made by 2-Opt before it finds a locally optimal solution. When talking about the number of local improvements, it is convenient to consider the *transition graph*. The vertices in this graph correspond to the possible tours and an arc from a vertex v to a vertex u is contained if u is obtained from v by performing an improving 2-Opt step. On the positive side, van Leeuwen and Schoone consider a 2-Opt variant for the Euclidean plane in which only steps are allowed that remove a crossing from the tour. Such steps can introduce new crossings, but van Leeuwen and Schoone [vLS81] show that after $O(n^3)$ steps, 2-Opt has found a tour without any crossing. On the negative side, Lueker [Lue75] constructs TSP instances whose transition graphs contain exponentially long paths. Hence, 2-Opt can take an exponential number of steps before it finds a locally optimal solution. This result is generalized to k-Opt, for arbitrary $k \geq 2$, by Chandra, Karloff, and Tovey [CKT99].

In order to explain the success of 2-Opt in practice, Kern [Ker89] initiated the study of 2-Opt on random instances. For Euclidean instances in which n points are placed uniformly at random in the unit square, he shows that the length of the longest path in the transition graph is bounded by $O(n^{16})$ with probability $1 - c/n$ for some constant c. Chandra, Karloff, and Tovey [CKT99] improve this result by bounding the expected length of the longest path in the transition graph by $O(n^{10} \log n)$. That is, independent of the initial tour and the choice of the local improvements, the expected number of 2-changes is bounded by $O(n^{10} \log n)$. For instances in which n points are placed uniformly at random in the unit square and the distances are measured according to the Manhattan metric, Chandra, Karloff, and Tovey show that the expected length of the longest path in the transition graph is bounded by $O(n^6 \log n)$.

Similar to the running time, the good approximation ratios obtained by 2-Opt on practical instances cannot be explained by a worst-case analysis. In fact, there are quite negative results on the worst-case behavior of 2-Opt. For example, Chandra, Karloff, and Tovey [CKT99] show that there are Euclidean instances in the plane for which 2-Opt has local optima whose costs are $\Omega\left(\frac{\log n}{\log \log n}\right)$ times larger than the optimal costs. However, the same authors also show that the expected approximation ratio of the worst local optimum for instances with n points drawn uniformly at random from the unit square is bounded from above by a constant.

We are aware of one article about the smoothed complexity of local search algorithms. Arthur and Vassilvitskii [AV06b] consider the *Iterative Closest Point (ICP)* and the *k-means algorithm*, two well-known algorithms for pattern matching and clustering, respectively. The input to the ICP algorithm consists of two point sets and, roughly speaking, the objective is to compute a translation of the first point set such that its distance to the second point set becomes minimal.

This problem is known to be NP-hard and the ICP algorithm, which is widely used in computational geometry and known for its simplicity and its efficiency in practice, computes a locally optimal solution. Arthur and Vassilvitskii show that the number of iterations of this algorithm can be $\Omega(n/d)^{d+1}$ for d-dimensional point sets of cardinality n. To reconcile this bound with the observations made in practice, they analyze the ICP algorithm in a semi-random input model in which an adversary can specify positions for the points that are subsequently perturbed by adding a d-dimensional Gaussian random vector with standard deviation σ to each point. They show that the expected running time is bounded from above by a polynomial (in n and σ^{-1}) whose degree is independent of the dimension d. The k-means algorithm partitions a given point set into k clusters with the objective to minimize the total distance of the points to their respective cluster center. Since finding optimal clusters is NP-hard, the k-means algorithm computes only clusters that are locally optimal. Using similar techniques as for the ICP algorithm, Arthur and Vassilvitskii show that the expected running time of the k-means algorithm is polynomially bounded in n^k and σ^{-1} for semi-random sets with n points, which is significantly better than the known worst-case bound of $2^{\Omega(\sqrt{n})}$ [AV06a].

1.3.2 Our Results

The negative results on the worst-case number of local improvements due to Lueker and Chandra, Karloff, and Tovey use arbitrary graphs whose edge lengths do not satisfy the triangle inequality. Hence, they leave open the question about the worst-case complexity of 2-Opt on metric TSP instances. In particular, Chandra, Karloff, and Tovey ask whether it is possible to construct Euclidean TSP instances on which 2-Opt can take an exponential number of steps [CKT99]. We resolve this question by constructing such instances in the Euclidean plane. In chip design applications, often TSP instances arise in which the distances are measured according to the Manhattan metric. Also for this metric and for every other L_p *metric* (see Section 4.1 for a formal definition of L_p metric), we construct instances with exponentially long paths in the 2-Opt transition graph.

Theorem 1.3.1. *For every $p \in \mathbb{N} \cup \{\infty\}$ and $n \in \mathbb{N}$, there is a two-dimensional TSP instance with $16n$ vertices in which the distances are measured according to the L_p metric and whose transition graph contains a path of length $2^{n+4} - 22$.*

Additionally, we consider a more general probabilistic input model than Kern and Chandra, Karloff, and Tovey and improve the previously known bounds. This probabilistic model is inspired by the model of smoothed analysis and is similar to our semi-random input model for bicriteria optimization problems. The considered model allows that different vertices are placed according to different continuous probability distributions in the unit hypercube $[0,1]^d$ for some constant *dimension* $d \geq 2$. The distribution of a vertex v_i is defined by a density function $f_i \colon [0,1]^d \to [0,\phi]$ for some given $\phi \geq 1$. (Readers who are unfamiliar with continuous random variables are referred to Appendix B.1 for a brief introduction.) Our upper bounds depend on the number n of vertices and the upper bound ϕ on the density. We denote instances created by this input model as *ϕ-perturbed Euclidean* or *Manhattan instances*, depending on the underlying metric. Similar to the reciprocal of the standard deviation σ^{-1} in Section 1.2, the maximal

density ϕ can be seen as a parameter specifying how close the analysis is to a worst case analysis since the larger ϕ is, the more concentrated the probability mass can be and hence, the better worst case instances can be approximated by the distributions. For $\phi = 1$ and $d = 2$, every point has a uniform distribution over the unit square, and hence the input model equals the uniform model analyzed before. Our results narrow the gap between the subquadratic number of improving steps observed in experiments and the upper bounds from the probabilistic analysis. With slight modifications, this model also covers a smoothed analysis, in which first an adversary specifies the positions of the points and after that each position is slightly perturbed by adding a Gaussian random vector with small standard deviation σ. In this case, the maximal density ϕ is proportional to σ^{-d}.

We also consider a model in which an arbitrary graph $G = (V, E)$ is given and for each edge $e \in E$, a probability distribution according to which the edge length $\mathsf{d}(e)$ is chosen independently of the other edge lengths. Again, we restrict the choice of distributions to distributions that can be represented by density functions $f_e \colon [0, 1] \to [0, \phi]$ with maximal density at most ϕ for a given $\phi \geq 1$. We denote inputs created by this input model as *ϕ-perturbed graphs*. Observe that in this input model only the distances are perturbed whereas the graph structure is not changed by the randomization. This can be useful if one wants to explicitely prohibit certain edges. However, if the graph G is not complete, one has to initialize 2-Opt with a Hamiltonian cycle to start with.

We prove the following theorem about the expected length of the longest path in the 2-Opt transition graph for the three probabilistic input models discussed above. It is assumed that the dimension $d \geq 2$ is an arbitrary constant.

Theorem 1.3.2. *The expected length of the longest path in the 2-Opt transition graph*

a) is $O(n^4 \cdot \phi)$ for ϕ-perturbed Manhattan instances with n points.

b) is $O(n^{4+1/3} \cdot \log(n\phi) \cdot \phi^{8/3})$ for ϕ-perturbed Euclidean instances with n points.

c) is $O(m \cdot n^{1+o(1)} \cdot \phi)$ for ϕ-perturbed graphs with n vertices and m edges.

Usually, 2-Opt is initialized with a tour computed by some tour construction heuristic. One particular class are *insertion heuristics*, which insert the vertices one after another into the tour. We show that also from a theoretical point of view, using such an insertion heuristic yields a significant improvement for metric TSP instances because the initial tour 2-Opt starts with is much shorter than the longest possible tour. In the following theorem, we summarize our results on the expected number of local improvements.

Theorem 1.3.3. *The expected number of steps performed by 2-Opt*

a) is $O(n^{4-1/d} \cdot \log n \cdot \phi)$ on ϕ-perturbed Manhattan instances with n points when 2-Opt is initialized with a tour obtained by an arbitrary insertion heuristic.

b) is $O(n^{4+1/3-1/d} \cdot \log^2(n\phi) \cdot \phi^{8/3})$ on ϕ-perturbed Euclidean instances with n points when 2-Opt is initialized with a tour obtained by an arbitrary insertion heuristic.

In fact, our analysis shows not only that the expected number of local improvements is polynomially bounded but it also shows that the second moment

and hence the variance is bounded polynomially for ϕ-perturbed Manhattan and graph instances. For the Euclidean metric, we cannot bound the variance but the 3/2-th moment polynomially.

Finally, we consider the expected approximation ratio of 2-Opt. We generalize the result due to Chandra, Karloff, and Tovey to our input model in which different points can have different distributions with bounded density ϕ and to all L_p metrics.

Theorem 1.3.4. *Let $p \in \mathbb{N} \cup \{\infty\}$. For ϕ-perturbed L_p instances, the expected approximation ratio of the worst tour that is locally optimal for 2-Opt is bounded by $O(\sqrt[d]{\phi})$.*

1.4 Outline and Bibliographical Notes

In the following three chapters, we prove the results about Nash equilibria (Chapter 2), Pareto-optimal solutions (Chapter 3), and local optima (Chapter 4) stated in the introduction. In Chapter 5, we draw conclusions and discuss future research directions.

The results presented in Chapter 2 have been presented in part in preliminary form at two conferences and as technical report:

- [ARV06a] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the Impact of Combinatorial Structure on Congestion Games. In *Proc. of the 47th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 613–622, 2006.
- [AGM+07a] Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. Uncoordinated Two-Sided Markets. Technical Report AIB-2007-22, RWTH Aachen, 2007.
- [AGM+07b] Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. A Unified Approach to Congestion Games and Two-Sided Markets. In *Proc. of the 3rd Int. Workshop on Internet & Network Economics (WINE)*, pages 30–41, 2007.

The results presented in Chapter 3 have been presented in part in preliminary form at the following conferences (the second publication has also appeared as invited contribution to a special issue of *Theoretical Computer Science* with selected papers from *ISAAC 2005*):

- [BRV07] René Beier, Heiko Röglin, and Berthold Vöcking. The Smoothed Number of Pareto Optimal Solutions in Bicriteria Integer Optimization. In *Proc. of the 12th Int. Conf. on Integer Programming and Combinatorial Optimization (IPCO)*, pages 53–67, 2007.
- [ANRV07] Heiner Ackermann, Alantha Newman, Heiko Röglin, and Berthold Vöcking. Decision-Making Based on Approximate and Smoothed Pareto Curves. In *Proc. of the 16th Int. Symp. on Algorithms and Computation (ISAAC)*, pages 675–684, 2005. *Theoretical Computer Science*, 378(3):253–270, 2007.

The results presented in Chapter 4 have been presented in part in preliminary form at the following conference:

- [ERV07] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst-Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. In *Proc. of the 18th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1295–1304, 2007.

Chapter 2

Pure Nash Equilibria

In this chapter, we consider different models for resource allocation among selfish agents. The first two models, congestion games and two-sided matching markets, are well-known in economics. We study the complexity of computing a pure Nash equilibrium and whether uncoordinated agents reach an equilibrium quickly. We conclude the chapter by introducing a new model for resource allocation that combines features of congestion games and two-sided markets and contains both these models as special cases.

2.1 Complexity of Equilibria in Congestion Games

In this section, we analyze the complexity of pure Nash equilibria in various kinds of structured congestion games. Since the problem of computing a pure Nash equilibrium in a congestion game can be phrased as a local search problem, we start this section by introducing the complexity class PLS (polynomial local search). We summarize some known facts about this class and illustrate the connection to congestion games. After that, we introduce a very restricted class of congestion games, which we call *threshold congestion games*, and we show that even for this restricted class, the problem of computing a pure Nash equilibrium is PLS-complete. We believe that, due to their simple structure, threshold congestion games are a good starting point for PLS-reductions. To demonstrate this, we present several reductions from threshold congestion games to other classes of congestion games. Some of these reductions simplify previously known proofs significantly and improve known results.

2.1.1 Local Search and the Complexity Class PLS

As already discussed in the introduction, an instance $I \in \mathcal{I}_\Pi$ of a *local search problem* Π consists of a finite set of feasible solutions $\mathcal{S}(I)$, an objective function $c_I \colon \mathcal{S}(I) \to \mathbb{N}$, and for every feasible solution $x \in \mathcal{S}(I)$, a neighborhood $\mathcal{N}(I, x) \subseteq \mathcal{S}(I)$. Given such an instance I, the goal is to find a *locally optimal solution* x^*, that is, a solution that does not have a strictly better neighbor. A neighbor x' of a solution x is strictly better if the objective value $c(x')$ is larger or smaller than $c(x)$ in the case of a maximization or minimization problem, respectively. The class PLS has been defined by Johnson, Papadimitriou, and Yannakakis [JPY88] and it contains all local search problems with polynomial time searchable neighborhoods. Formally it is defined as follows.

Definition 2.1.1. *A local search problem* Π *belongs to* PLS *if there exist three polynomial-time algorithms* A_Π*,* B_Π*, and* C_Π *with the following properties:*

1. *Given an instance* I *of* Π*, algorithm* A_Π *computes some feasible solution* $x_0 \in \mathcal{S}(I)$*.*
2. *Given an instance* I *of* Π *and a feasible solution* $x \in \mathcal{S}(I)$*, algorithm* B_Π *computes the objective value* $c_I(x)$*.*
3. *Given an instance* I *of* Π *and a feasible solution* $x \in \mathcal{S}(I)$*, algorithm* C_Π *determines whether* x *is locally optimal or not and finds a better solution in the neighborhood of* x *in the latter case.*

Johnson, Papadimitriou, and Yannakakis [JPY88] present a reduction concept for PLS and identify problems that are complete with respect to this reduction concept. In the following definitions, we denote by Π_1 and Π_2 problems in PLS.

Definition 2.1.2. *A problem* Π_1 *is* PLS*-reducible to a problem* Π_2 *if there exist polynomial-time computable functions* f *and* g *with the following properties:*

1. *The function* f *maps instances of* Π_1 *to instances of* Π_2*.*
2. *The function* g *maps pairs* (I, y)*, where* I *denotes an instance of* Π_1 *and* y *denotes a solution of* $f(I)$*, to solutions of* I*.*
3. *For all instances* I *of* Π_1 *and all locally optimal solutions* y *of* $f(I)$*, the solution* $g(I, y)$ *is a local optimum of instance* I*.*

A local search problem Π *in* PLS *is* PLS-complete *if every problem in* PLS *is* PLS*-reducible to* Π*.*

Given an instance I of a local search problem Π, we denote by $\mathrm{TG}(I)$ the *transition graph* that contains a node $v(x)$ for every feasible solution $x \in \mathcal{S}(I)$ and a directed edge from a node $v(x_1)$ to a node $v(x_2)$ if x_2 is in the neighborhood of x_1 and if the objective value $c(x_2)$ is strictly better than the objective value $c(x_1)$. Schäffer and Yannakakis [SY91] introduce the notion of a *tight* PLS*-reduction*, which preserves several properties of the transition graph.

Definition 2.1.3. *A* PLS*-reduction* (f, g) *from a problem* Π_1 *to a problem* Π_2 *is* tight *if for any instance* $I \in \mathcal{I}_{\Pi_1}$*, one can choose a subset* $\mathcal{Q}_I \subseteq \mathcal{S}(f(I))$ *of feasible solutions for the image instance* $f(I) \in \mathcal{I}_{\Pi_2}$ *such that the following properties are satisfied:*

1. *The set* $\mathcal{Q}_I$ *contains all local optima of the instance* $f(I)$*.*
2. *There exists a polynomial-time algorithm that, given an instance* $I \in \mathcal{I}_{\Pi_1}$ *and a feasible solution* $x \in \mathcal{S}(I)$*, constructs a feasible solution* $y \in \mathcal{Q}_I$ *of* $f(I)$ *such that* $g(I, y) = x$*.*
3. *Suppose that the transition graph* $TG(f(I))$ *of* $f(I)$ *contains a directed path from* $y_1 \in \mathcal{Q}_I$ *to* $y_2 \in \mathcal{Q}_I$ *such that all internal path vertices are outside* $\mathcal{Q}_I$*, and let* $x_1 = g(I, y_1)$ *and* $x_2 = g(I, y_2)$ *be the corresponding feasible solutions of* I*. Then either* $x_1 = x_2$ *or* $TG(I)$ *contains an edge from* x_1 *to* x_2*.*

An important property of tight PLS-reductions is that they do not shorten paths in the transition graph. That is, if the graph $TG(I)$ of an instance I of Π_1 contains a node whose shortest distance to a local optimum is $a \in \mathbb{N}$, then $TG(f(I))$ contains a node whose shortest distance to a local optimum is at least a. Moreover, consider the following problem: given an instance I of Π_1 and a solution x of I, find a local optimum that is reachable from x in the transition graph. If this problem is PSPACE-complete for Π_1 and there exists a tight PLS-reduction from Π_1 to Π_2, then the problem is also PSPACE-complete for Π_2 [SY91]. In the following definition, we introduce a special case of tight PLS-reductions, which we call *embedding PLS-reduction*. These reductions are less general than tight PLS-reductions, but for a given reduction, their properties can usually be verified more easily.

Definition 2.1.4. *We call a* PLS*-reduction (f, g) from a problem Π_1 to a problem Π_2 an* embedding PLS-reduction *if it satisfies the following two properties:*

1. *For every instance I of Π_1, the transition graph $TG(I)$ is isomorphic to a subgraph $TG^*(f(I))$ of $TG(f(I))$ that contains all local optima of the instance $f(I)$ and has no outgoing edges.*
2. *When restricted to this subgraph, the function $g(I, \cdot)$ must be an isomorphism between $TG^*(f(I))$ and $TG(I)$ and its inverse must be computable in polynomial time.*

We call (f, g) an isomorphic PLS-reduction *if it is an embedding* PLS*-reduction and if for every instance I of Π_1 the transition graphs $TG(I)$ and $TG(f(I))$ are isomorphic.*

It is easy to verify that embedding PLS-reductions are tight PLS-reductions. In order to see this, observe that the set of feasible solutions that corresponds to the nodes of $TG^*(f(I))$ defines a set $\mathcal{Q}_I$ with the properties in Definition 2.1.3. The first and the second condition are directly satisfied by the definition of an embedding PLS-reduction. The third condition is satisfied because $TG^*(f(I))$ has no outgoing edges and therefore paths with the properties as in condition 3 cannot have internal path vertices and must therefore be single edges with both endpoints from $TG^*(f(I))$. Since $TG^*(f(I))$ is isomorphic to $TG(I)$, the third condition must be satisfied for these edges.

Relationship between Congestion Games and the Complexity Class PLS

As described in Section 1.1, the four components of a congestion game are the set of players $\mathcal{N}$, the set of resources $\mathcal{R}$, the strategy spaces Σ_i for $i \in \mathcal{N}$, and the delay functions $d_r \colon \mathbb{N} \to \mathbb{N}$ for $r \in \mathcal{R}$. Rosenthal [Ros73] shows that every congestion game possesses a pure Nash equilibrium by considering the potential function $\Phi \colon \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{N}$ with

$$\Phi(S) = \sum_{r \in \mathcal{R}} \sum_{j=1}^{n_r(S)} d_r(j) \ .$$

This potential function associates with every state a value, and if a state S' is obtained from a state S by letting one player play a better response that decreases

her delay by some value a, then also the potential decreases by exactly a. Games that admit such a potential function are called *(exact) potential games*. Since the potential is bounded from below by 0 and from above by some finite value, this shows not only that every congestion game possesses a pure Nash equilibrium, but it also shows that players reach such an equilibrium after a finite number of better responses.

A congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$ can be transformed into an instance I of a local search problem as follows: The set $\mathcal{S}(I)$ of feasible solutions corresponds to the set of states of Γ, that is, $\mathcal{S}(I) = \Sigma = \Sigma_1 \times \cdots \times \Sigma_n$, and a state S' is in the neighborhood of a state S if S' can be obtained from S by letting one player change her strategy. Furthermore, we use Rosenthal's potential function Φ as cost function $c\colon \mathcal{S}(I) \to \mathbb{N}$. This way, we obtain a local search problem and it is easy to see that the aforementioned properties of the function Φ ensure that the local optima in this local search problem coincide with the pure Nash equilibria in the congestion game. If the strategy spaces are explicitly given, then, given a state, better and best responses of the players can be computed in polynomial time and it can be checked in polynomial time whether a given state is locally optimal. This yields the following observation.

Observation 2.1.5. *The problem of computing a pure Nash equilibrium in congestion games belongs to the complexity class* PLS.

For succinctly represented classes of congestion games, in which the strategy spaces are only given implicitly, the problem of computing a pure Nash equilibrium might not belong to PLS because it might not be possible to compute better responses in polynomial time. In network congestion games we can, however, find better responses and check whether a given state is a pure Nash equilibrium by solving shortest path problems, and hence, network congestion games belong to PLS.

The relationship between congestion games and PLS has first been observed by Fabrikant, Papadimitriou, and Talwar [FPT04]. They considered different classes of congestion games and obtained the following PLS-completeness results.

Theorem 2.1.6. *([FPT04]) It is* PLS*-complete to find a pure Nash equilibrium in congestion games of the following sorts:*

a) general congestion games

b) symmetric congestion games

c) asymmetric network congestion games with directed edges

The PLS-completeness for general and symmetric congestion games is obtained by short reductions from the problem *positive not-all-equal 3-SAT (PosNAE3SAT)*, which is a local search version of the satisfiability problem for a restricted class of formulas. An instance of PosNAE3SAT is given by a set of weighted clauses each of which consisting of three positive literals. An assignment of truth values to the variables is said to satisfy a clause if not all its three literals are assigned the same value. An assignment is locally optimal if the total weight of the satisfied clauses cannot be increased by flipping the value of one of the variables. Schäffer and Yannakakis [SY91] prove that computing such a

local optimum is PLS-complete. Their reduction is, however, quite involved and, according to Fabrikant, Papadimitriou, and Talwar [FPT04], possibly the most complex reduction in the literature if one excludes the PCP theorem. The proof for the PLS-completeness of network congestion games is not very instructive as it completely reworks the PLS-completeness proof of PosNAE3SAT and even adds some further complications. In the following, we present an alternative approach for proving hardness of structured congestion games that more directly reveals which kind of substructures cause the trouble and that also shows the hardness of asymmetric network congestion games with undirected edges and linear delay functions.

Additionally, Fabrikant, Papadimitriou, and Talwar [FPT04] show how the problem of computing a pure Nash equilibrium in a symmetric network congestion game, in which all players have the same source and the same target node, can be reduced to solving a min-cost flow problem. This reduction implies that pure Nash equilibria can be computed efficiently in symmetric network congestion games.

2.1.2 Threshold Congestion Games

We define *threshold congestion games* to be a class of congestion games in which the set of resources $\mathcal{R}$ is divided into two disjoint subsets $\mathcal{R}_{\text{out}}$ and $\mathcal{R}_{\text{in}}$. The set $\mathcal{R}_{\text{out}}$ contains a resource r_i for every player $i \in \mathcal{N}$. This resource has a fixed delay T_i, called the *threshold* of player i. Each player i has only two strategies, namely the strategy $S_i^{\text{Out}} = \{r_i\}$ and a strategy $S_i^{\text{in}} \subseteq \mathcal{R}_{\text{in}}$, where S_i^{in} can be an arbitrary subset of $\mathcal{R}_{\text{in}}$. The preferences of player i can be described in a simple and intuitive way: player i prefers strategy S_i^{in} to strategy S_i^{Out} if the delay of S_i^{in} is smaller than the threshold T_i. *Quadratic threshold congestion games* are a subclass of threshold games in which the set $\mathcal{R}_{\text{in}}$ contains one resource r_{ij} for every unordered pair $\{i, j\} \subseteq \mathcal{N}$ of players and in which for every player $i \in \mathcal{N}$, the strategy S_i^{in} is defined as $\{r_{ij} \mid j \in \mathcal{N}, j \neq i\}$.

We show that computing Nash equilibria in quadratic threshold congestion games is PLS-complete despite the simple structure of these games. Our proof is by a reduction from a local search version of *Max-Cut*, which can be described as follows: an instance of Max-Cut consists of a graph $G = (V, E, w)$ with edge weights $w\colon E \to \mathbb{N}$, and the goal is to find a partition of the vertices V into two classes, say left and right, such that the total weight of the edges that cross the cut becomes maximal. In the local search version, we are looking for a partition that is locally optimal in the sense that the total weight of the edges that cross the cut cannot be increased by flipping one vertex from the left to the right side, or vice versa. This local search problem can also be described as a game, the so-called *party affiliation game*, in which players correspond to nodes that can choose whether they want to belong to the left or the right side. Edges reflect some kind of symmetric antipathy, that is, a player wants to choose a side such that the total weight of her edges to the other side is maximized. Schäffer and Yannakakis [SY91] show that computing a locally optimal cut is PLS-complete and that there are instances of Max-Cut with states whose distance to sinks in the transition graph is exponential. Based on this result, we prove the following theorem.

Theorem 2.1.7. *Computing a pure Nash equilibrium in quadratic threshold congestion games with non-decreasing delay functions is* PLS*-complete. Furthermore, there exist quadratic threshold congestion games with states from which every sequence of better responses to a pure Nash equilibrium has exponential length.*

Proof. Assume that a party affiliation game $G = (V, E, w)$ is given. For a player $i \in V$, let W_i denote the sum of the weights of all of her incident edges. If a state, that is, a partition of the players into two classes A and B, is given, we denote by $W_i^{(B)}$ the sum of the weights of the edges that connect player i with nodes in class B. The preferences of a player i in the party affiliation game can be described in the following way: player i prefers strategy A to strategy B if $W_i^{(B)} > \frac{1}{2}W_i$, she prefers strategy B to strategy A if $W_i^{(B)} < \frac{1}{2}W_i$, and she is indifferent if $W_i^{(B)} = \frac{1}{2}W_i$.

We show how to represent the party affiliation game in form of a quadratic threshold congestion game. We set $\mathcal{N} = V$, that is, for each player in the party affiliation game, we introduce a player in the threshold congestion game. We introduce a resource r_{ij} for every unordered pair $\{i, j\} \subseteq \mathcal{N}$ of players. If the edge $\{i, j\}$ does not exist in G, then the delay of r_{ij} is 0. Otherwise, the delay of r_{ij} is 0 if the resource is used by only one player and its delay is $w(\{i, j\})$ if it is used by two players. We set the threshold T_i of player i to $\frac{1}{2}W_i$.

Now let us describe the function g, which maps states of the congestion game to states of the party affiliation game. We identify the strategy S_i^{in} of player i in the congestion game with the strategy B in the party affiliation game, and we identify the strategy S_i^{Out} with strategy A. Due to the choice of the threshold T_i, this implies that the preferences of the players in the congestion game coincide with the preferences in the party affiliation game. Hence, there is a one-to-one correspondence between the transition graphs of both games so that our construction yields an isomorphic PLS-reduction. Since isomorphic PLS-reductions are tight PLS-reduction, they preserve distances in the transition graph, which implies the second part of the theorem. □

In the following, we use quadratic threshold games as the starting point for further PLS-reductions. For some of these reductions, it is helpful to make some additional assumptions on the delay functions.

Remark 2.1.8. *Without loss of generality, each resource $r \in \mathcal{R}_{in}$ in a quadratic threshold congestion game with non-decreasing delay functions has a linear delay function of the form $d_r(x) = a_r x$ with $a_r \geq 0$. Furthermore, all thresholds can be assumed to be positive.*

In the proof of Theorem 2.1.7, the delay function of a resource $r_{ij} \in \mathcal{R}_{\text{in}}$ has the form $d_{r_{ij}}(k) = w_{ij}k - w_{ij}$. The preferences of the players are not affected by adding w_{ij} to each delay function $d_{r_{ij}}$, which then becomes $d_{r_{ij}}(k) = w_{ij}k$, if one simultaneously increases each threshold T_i by $\sum_{j \neq i} w_{ij} = W_i$. After this transformation, every resource r has a delay function of the form $d_r(k) = a_r k$.

2.1.3 Reductions from Threshold Congestion Games

In this section, we use quadratic threshold congestion games as starting point for proving the PLS-completeness of several classes of congestion games.

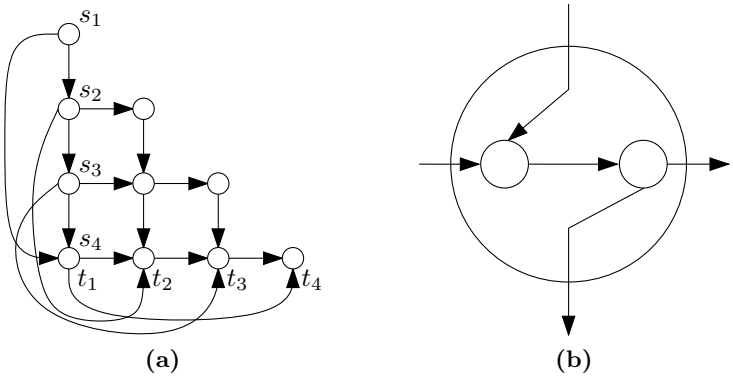


Figure 2.1.1: Illustration of the proof of Theorem 2.1.9.

Network Congestion Games

In a network congestion game, the resources are the edges of a graph and every player has a designated source and a designated target node. Every player has to choose a path between her source and target node and wants to minimize the delay of the chosen path. As summarized in Theorem 2.1.6, computing pure Nash equilibria of network congestion games is PLS-complete. In the following, we present a much simpler proof for this fact. In contrast to the previous proof, our proof also works for linear delay functions and it can be extended to undirected networks.

Theorem 2.1.9. *Computing a pure Nash equilibrium in network congestion games with non-decreasing linear delay functions is* PLS*-complete.*

Proof. Let Γ be a quadratic threshold congestion game. We map Γ to a network congestion game as follows. The network consists of the lower-left triangle of an $n \times n$ grid (including the vertices on the diagonal) in which the vertical edges are directed downwards and the horizontal edges are directed from left to right. For every player i in Γ, we introduce a player i in the network congestion game whose source node s_i is the i-th node (from top to bottom) in the first column and whose target node t_i is the i-th node (from left to right) in the last row. For every player $i \in \mathcal{N}$, we add an edge from s_i to t_i, called *threshold edge*. Note that, due to the directions of the grid edges, the threshold edge of player i can only be used by player i. Figure 2.1.1 (a) illustrates our construction in the case of 4 players.

Our first goal is to define delay functions in such a way that there are only two relevant strategies for player i: the threshold edge (s_i, t_i) and the *row-column path* from s_i to t_i, i.e., the path from s_i along the edges of row i until column i and then along the edges of column i to t_i. All other paths shall have such high delays that they are dominated by these two paths, regardless of the other players' choices. We achieve this goal by assigning the constant delay function 0 to all vertical edges and the constant delay function $D \cdot i$ to all horizontal edges in row i, where D denotes a large integer. Furthermore, for the time being, we assume that the threshold edge (s_i, t_i) has the constant delay $D \cdot i \cdot (i-1)$. This way, each player i has only two undominated strategies: its threshold edge and its row-column path. The delays of these two alternative routes are so far identical.

Now we define additional delay functions for the nodes, that is, we view also the nodes as resources. (Figure 2.1.1 (b) shows how the nodes can be replaced by gadgets such that all resources are edges.) For $1 \leq i < j \leq n$, the node v_{ij} in column i and row j is identified with the resource $r_{ij} \in \mathcal{R}_{\text{in}}$ in the quadratic threshold game. In particular, we assume that the node has the same delay function as the corresponding resource in the threshold game. This way, the row-column path of player i corresponds to the strategy S_i^{in} of the threshold game. Furthermore, we increase the delay on the threshold edge of player i from $D \cdot i \cdot (i-1)$ to $D \cdot i \cdot (i-1) + T_i$, where T_i is the delay of resource $r_i \in \mathcal{R}_{\text{out}}$ in the threshold game. This way, the threshold edge of player i corresponds to the strategy S_i^{Out} of the threshold game.

If we choose D larger than the sum of all delays in the threshold game, then for every player all strategies except for her row-column path and her threshold edge are dominated and hence, they can be ignored. The remaining strategy spaces of the players and the corresponding delay functions are isomorphic to the strategies and delay functions of the threshold game. In particular, also the Nash equilibria of the two games coincide. Thus, our construction is a PLS-reduction. Moreover, the transition graph of the threshold game and the subgraph of the transition graph of the network congestion game in which players are restricted to either the threshold edge or the row-column path are isomorphic. It is easy to see that all properties in Definition 2.1.4 are satisfied, and hence, the reduction is an embedding PLS-reduction. □

It is not difficult to modify the reduction above so that the linear delay functions have offset 0. In fact, one only needs to replace the constant delay functions of the form $d_e(x) = a$ by linear delay functions $d_e(x) = a \cdot x$. Thus, even congestion games in networks with *link speeds* are PLS-complete. Next we consider network congestion games with undirected edges and linear delay functions and prove that computing a pure Nash equilibrium remains PLS-complete for these games.

Theorem 2.1.10. *Computing a pure Nash equilibrium in network congestion games with undirected edges and non-decreasing linear delay functions is* PLS*-complete.*

Proof. We give a PLS-reduction from quadratic threshold games to network congestion games with undirected edges and linear delay functions. The reduction is similar to the one in Theorem 2.1.9, except that we slightly change the structure of the network and that we adapt the delay functions of the edges accordingly.

Let Γ be a quadratic threshold congestion game. The undirected graph that we construct has the same structure as in the case of networks with directed edges, except that we remove the directions of the edges. Moreover we split every threshold edge $\{s_i, t_i\}$ into two edges by introducing a node s_i', i.e., we introduce the edges $e_i^s = \{s_i, s_i'\}$ and $e_i^t = \{s_i', t_i\}$ and we remove the edge $\{s_i, t_i\}$. Again, for every player i in Γ, we introduce a player in the network congestion game. In this reduction the source node of player i is s_i' and her target node is t_i.

In the previous reduction we could force a player to decide between the threshold edge and the row-column path by considering directed edges and carefully designed delay functions. Now we have to achieve the same effect with the delay functions only. We do not change the delay functions of the nodes v_{ij} and of

the horizontal edges. Thus, a horizontal edge in the i-th row has constant delay $i \cdot D$, and the delay function of node v_{ij} coincides with the delay function of the resource $r_{ij} \in \mathcal{R}_{\text{in}}$. We change the delay function of every vertical edge from 0 to n^3D. Additionally, we set the delay of the edge $e_i^s = \{s_i, s_i'\}$ to $3n^4D$.

We claim that for every player i, if one excludes the direct edge between s_i' and t_i, the only path connecting s_i' and t_i which can be a best response is the row-column path. Let $D_i^{\max}$ denote the maximal delay that can occur on this path without taking into account the edge $\{s_i, s_i'\}$ but including the delay caused by the nodes on the path. We can bound this delay by

$$D_i^{\max} < (i-1)iD + (n-i)n^3D + D \ .$$

If player i chooses any other path connecting s_i' and t_i, then she either passes a node s_j' with $j \neq i$, or she allocates more than $n-i$ vertical edges, or she allocates a horizontal edge in a row $j > i$.

In the first case, the delay caused by the edge $\{s_j, s_j'\}$ is $3n^4D$. This delay is larger than $D_i^{\max}$ and hence choosing such a path cannot be a best response. In the second case, the delay is at least $(n-i+1)n^3D$ which is also larger than $D_i^{\max}$. Finally, consider the third case and assume that player i allocates $n-i$ vertical edges but at least one edge from a row $j > i$. Then her delay is at least

$$(n-i)n^3D + (i-2)iD + jD \geq (n-i)n^3D + (i-1)iD + D \ ,$$

which is also larger than $D_i^{\max}$.

Finally, we choose the delay of the edge e_i^t to be

$$T_i + 3n^4D + (n-i)n^3D + (i-1)iD \ ,$$

where T_i denotes the threshold of player i in the given quadratic threshold game.

Now assume that every player plays a best response and hence, either uses the direct edge between s_i' and t_i or the row-column path. Observe that under this assumption, the delay of edge e_i^t equals the threshold of player i plus the delay caused by the grid edges of the aforementioned path and the edge $\{s_i, s_i'\}$. Hence by the same arguments as in Theorem 2.1.9, a Nash equilibrium of the constructed network congestion game corresponds to a Nash equilibrium of the given quadratic threshold game. Moreover, analogously to Theorem 2.1.9, this reduction is an embedding PLS-reduction. □

In symmetric network congestion games, a Nash equilibrium can be computed in polynomial time [FPT04]. Nonetheless, selfish players do not necessarily find an equilibrium in polynomial time.

Theorem 2.1.11. *There exists a family of symmetric network congestion games (with directed or undirected edges) with linear delay functions and corresponding initial states from which every better response sequence to a pure Nash equilibrium is exponentially long.*

Proof. We prove the theorem by simulating asymmetric network congestion games by symmetric ones. In the case of asymmetric network congestion games, the existence of instances with the claimed properties follows from the fact that such

instances exist for Max-Cut as shown by Schäffer and Yannakakis [SY91] and the fact that the reductions presented in the proofs of Theorems 2.1.9 and 2.1.10 are tight. Let Γ be an asymmetric network congestion game on a graph G and let $S = (P_1, \ldots, P_n)$ be a state of Γ. Let $S(G)$ be the set of source and $T(G)$ the set of target nodes of the network G and assume without loss of generality that no two players have the same source or the same target node. In order to obtain a symmetric network congestion game Γ', we introduce a common source s and a common target t such that s is connected to every source $s_i \in S(G)$ and such that every target $t_i \in T(G)$ is connected to t. For every new edge $e = (s, \cdot)$ and $e = (\cdot, t)$, we define the delay function d_e by $d_e(x) = x \cdot D$ with D being a number larger than the maximum total delay of every path in G.

Assume that player i initially chooses path P_i with the additional edges (s, s_i) and (t_i, t) and denote this state by S'. Obviously, the players behave in the same way as they do in the asymmetric game since no player has an incentive to share an edge $(s, \cdot)$ or $(\cdot, t)$ with another player. Thus, if in Γ every better response sequence starting in S is exponentially long, then every better response sequence in Γ' starting in S' is exponentially long as well. □

The simulation of asymmetric networks by symmetric ones also implies the following theorem.

Theorem 2.1.12. *In network congestion games (with directed or undirected edges) with a common source and possibly different targets (or vice versa), and with non-decreasing delay functions, a pure Nash equilibrium can be found in polynomial time.*

Proof. We use the same simulation as in the proof of Theorem 2.1.11. Assume that a network with a common source node is given and assume without loss of generality that no two players have the same target node. We make the network symmetric by introducing a new common target node and connecting this new target to each original target node by an edge with delay function d_e such that $d_e(1) = 0$ and $d_e(x) = D$ for $x > 1$. Again, D is assumed to be larger than the sum of all delays in the original game. Due to Fabrikant, Papadimitriou, and Talwar [FPT04], a Nash equilibrium in this symmetric network congestion game can be computed in polynomial time. Observe that each of the new edges connecting one of the original target nodes with the new target node is used by exactly one player in every Nash equilibrium. Hence, every equilibrium of the symmetric game can be transformed into a Nash equilibrium of the original game in polynomial time. □

Market Sharing Games

Market Sharing games have been introduced by Goemans et al. [GLMT04] to model non-cooperative content distribution in wireless networks. An instance of a market sharing game consists of a set $\mathcal{N} = \{1, \ldots, n\}$ of players, a set $\mathcal{M}$ of $l := |\mathcal{M}|$ markets, and a bipartite graph $G = (\mathcal{N} \cup \mathcal{M}, E)$. An edge between player i and market m indicates that player i is interested in market m. Furthermore, for each market m, a cost c_m and a so-called query rate $q_m \in \mathbb{N}$ are given, and, for each player i, a budget B_i is specified. The query rate q_m determines the payoff

of market m, which is equally distributed among the players who allocate that market, i.e., the payoff function of market m is given by $p_m(x) = q_m/x$. In terms of congestion games, the markets are the resources and the costs and budgets implicitly define the sets of feasible strategies. To be more precise, Σ_i consists of all sets $\mathcal{M}' \subseteq \mathcal{M}$ such that for all $m \in \mathcal{M}'$, $(i, m) \in E$ and $\sum_{m \in \mathcal{M}'} c_m \leq B_i$. Observe that the set of strategies has a knapsack-like structure. The players are interested in allocating a set $\mathcal{M}'$ of markets with maximum payoff. Thus, we define the *delay of a market* to be equal to its negative payoff.

If the costs of all markets are equal to 1, a market sharing game is called *uniform*. Goemans et al. [GLMT04] give an algorithm for computing a Nash equilibrium of a uniform market sharing game in polynomial time. Ackermann, Röglin, and Vöcking [ARV06a] show that in every uniform market sharing game, players reach a Nash equilibrium after at most a polynomial number of best responses.

If we allow arbitrary costs, then it becomes NP-hard to determine a best response since computing a best response corresponds to solving an instance of the knapsack problem. As a consequence, the problem of finding a Nash equilibrium is not contained in PLS, unless P = NP. However, if the costs are polynomially bounded, then the problem of finding a Nash equilibrium is in PLS because it is well-known that the knapsack problem can be solved in pseudopolynomial time. In this case, we can easily enforce that a player $i \in \mathcal{N}$ decides between either allocating one market m_i or a set of markets $\{m_i^{(1)}, \ldots, m_i^{(k)}\}$ by setting the cost of market m_i to k, the cost of each market $m_i^{(j)}$ to 1, and the budget of player i to k. This way, the only possible best responses of player i are the strategies $\{m_i\}$ and $\{m_i^{(1)}, \ldots, m_i^{(k)}\}$, regardless of the strategies of the other players. This is the main observation needed for reducing threshold congestion games to market sharing games with polynomially bounded costs.

Theorem 2.1.13. *Computing a pure Nash equilibrium in market sharing games with polynomially bounded costs is* PLS*-complete.*

Proof. We give a PLS-reduction from quadratic threshold congestion games. Let Γ be a quadratic threshold game. Due to Remark 2.1.8, we can assume w.l.o.g. that each resource $r \in \mathcal{R}_{\text{in}}$ has a linear delay function of the form $d_r(x) = a_r x$ with $a_r \geq 0$.

We construct a market sharing game Γ_M as follows. For every resource $r_{ij} \in \mathcal{R}_{\text{in}}$ with $a_{r_{ij}} > 0$, we introduce a market m_{ij} with cost 1 and query rate $q_{m_{ij}} = 2a_{r_{ij}}$. Furthermore, for every resource $r_i \in \mathcal{R}_{\text{out}}$, we introduce a market m_i with cost $|S_i^{\text{in}}|$ and query rate $3P_i - T_i$ where P_i denotes the sum of the coefficients $a_{r_{ij}}$ of the resources $r_{ij} \in S_i^{\text{in}}$. Observe that we can assume w.l.o.g. that $3P_i - T_i > 0$ since otherwise S_i^{in} is always the only best response for player i in Γ and hence, player i can be removed from the game. For every player i of Γ, there is also a player i in the market sharing game. This player has the budget $B_i = |S_i^{\text{in}}|$ and is interested in all markets that correspond to the resources in $S_i^{\text{Out}} \cup S_i^{\text{in}}$. Observe that this construction yields a market sharing game with polynomially bounded costs.

Let S be an arbitrary state of Γ_M. From S we construct a state $\tilde{S}$ of the quadratic threshold game as follows. If player i participates in market m_i, then we set the corresponding threshold game player i to its strategy S_i^{Out}, otherwise to strategy S_i^{in}.

Fix a player i in Γ, let $\mathcal{R}_i^1$ denote the resources in S_i^{in} that she allocates alone in state $\tilde{S} \oplus S_i^{\text{in}}$, and let $\mathcal{R}_i^2$ denote the resources in S_i^{in} that she shares with another player in that state. Then her delay in state $\tilde{S} \oplus S_i^{\text{in}}$ can be written as $\sum_{r \in \mathcal{R}_i^1} a_r + 2\sum_{r \in \mathcal{R}_i^2} a_r = P_i + \sum_{r \in \mathcal{R}_i^2} a_r$. The strategy S_i^{in} is a best response in state $\tilde{S}$ if and only if $P_i + \sum_{r \in \mathcal{R}_i^2} a_r \leq T_i$. The payoff player i receives in state S when choosing all markets m_{ij} she is interested in can be written as $2\sum_{r \in \mathcal{R}_i^1} a_r + \sum_{r \in \mathcal{R}_i^2} a_r = P_i + \sum_{r \in \mathcal{R}_i^1} a_r$. This is a best response if and only if $P_i + \sum_{r \in \mathcal{R}_i^1} a_r \geq 3P_i - T_i$ which is equivalent to $P_i + \sum_{r \in \mathcal{R}_i^2} a_r \leq T_i$. Thus, S is a Nash equilibrium if and only if $\tilde{S}$ is a Nash equilibrium. Moreover, the reduction is an embedding reduction and the transition graph of the threshold game is isomorphic to the transition graph of the market sharing game if we disregard states in which there is a player i who allocates a proper subset of S_i^{in}. □

Overlay Network Games

An overlay network is a network built on top of another network with fixed routing paths between all pairs of nodes. For example, Stoica et al. [SAZ+04] suggest to generalize the Internet point to point communication to provide services like multicast, anycast, and mobility on the basis of overlay networks. In the case of multicast and anycast, the overlay network is an arborescence connecting the source with the receivers. We simplify the scenario in many aspects and introduce the following overlay network congestion game: in an *overlay network game*, we are given an undirected graph $G = (V, E)$ with a delay function $d_e \colon \mathbb{N} \to \mathbb{N}$ for every edge $e \in E$ and a fixed routing path P_{uv} between any pair of nodes u and v. For simplicity, we assume that the path P_{uv} from u to v corresponds to the path P_{vu} from v to u. Every player $i \in \mathcal{N}$ wants to allocate a multicast tree T_i on a subset $V_i \subseteq V$ of the nodes. From the point of view of a player $i \in \mathcal{N}$, there is an edge between every pair of nodes $u, v \in V_i$, and every player i wants to allocate a spanning tree on the resulting complete graph with node set V_i. In the underlying network G, the edge between two nodes $u, v \in V_i$ is simulated by the corresponding path P_{uv}, that is, if the chosen spanning tree contains an edge between two nodes $u \in V_i$ and $v \in V_i$, the traffic is sent along the routing path P_{uv} in the underlying network G. In particular, the delay of the edge (u, v) in the overlay network equals the delay on the path P_{uv} in the network G. We show that finding a Nash equilibrium in an overlay network game is PLS-complete, although, from a local point of view, every player solves a matroid optimization problem. This is remarkable because in matroid congestion games, a pure Nash equilibrium can be computed efficiently by simulating the best response dynamics [ARV06a].

Theorem 2.1.14. *Computing a pure Nash equilibrium in overlay network games with linear delay functions is* PLS*-complete.*

Proof. We give a PLS-reduction from quadratic threshold congestion games to overlay network games. As in the proof of Theorem 2.1.9, we use the lower-left triangle of an $n \times n$-grid as basis of our construction, but now with undirected edges, and we use the identifiers $s_1, \ldots, s_n$, $t_1, \ldots, t_n$, and v_{ij} to denote the same nodes as before. The edges in the grid all have delay 0, the delay function of node

v_{ij} still equals the delay function of resource r_{ij}. Additionally, for each player $i \in \mathcal{N}$, we add a node t'_i and an edge $\{t_i, t'_i\}$ with delay 0. Instead of having an edge $\{s_i, t_i\}$, we add an edge $\{s_i, t'_i\}$ with delay function $d_{\{s_i,t'_i\}}(x) = T_i$. In the network, the routing path between s_i and t_i is defined to be the row-colum path as in the proof of Theorem 2.1.9. The routing paths between s_i and t'_i and between t_i and t'_i in the overlay network are defined to be the direct edges contained in the network G. For every player i in the quadratic threshold game, we define a player in the overlay network game with $V_i = \{s_i, t_i, t'_i\}$. Using the assumptions from Remark 2.1.8, our construction yields an overlay network game with linear delay functions.

Every best response of player i must contain the edge between t_i and t'_i since it has delay 0. Hence, every player decides between either taking the virtual edge between s_i and t_i in the overlay network or the edge between s_i and t'_i. In the former case, her message is routed along the path through the grid. Analogously to the proof of Theorem 2.1.9, this shows that it is PLS-complete to find a Nash equilibrium in an overlay network game. Moreover, observe that the reduction is embedding since the subgraph of the transition graph of the network game that contains exactly those states in which every player i uses the edge $\{t_i, t'_i\}$ is isomorphic to the transition graph of the quadratic threshold game, contains all local optima, and has no outgoing edges. □

2.2 Uncoordinated Two-Sided Markets

In this section, we consider two-sided markets without a central authority to match agents. We start by repeating some of the definitions from Section 1.1.2 and make them more formal. We assume that two disjoint groups of agents $\mathcal{X}$ and $\mathcal{Y}$ are given and that each agent has preferences about the agents of the other side. An agent $i \in \mathcal{X} \cup \mathcal{Y}$ can be matched to one agent j on the other side. Then she gets a *payoff* of $p_i(j)$. If the preference list of agent i is $(a_1, a_2, \ldots, a_n)$, we say that agent i gets payoff $k \in \{0, \ldots, n-1\}$ if she is matched to agent a_{n-k}. Also, we say that an agent has payoff -1 if she is unmatched. Given a matching M, we denote the payoff of an agent i in matching M by $p_i(M)$. Throughout this section, we call agents from $\mathcal{X}$ women or players and we call agents from $\mathcal{Y}$ men or resources.

We model the uncoordinated two-sided market $(\mathcal{X}, \mathcal{Y})$ as a game $G(\mathcal{X}, \mathcal{Y})$ among agents in $\mathcal{X}$. A strategy of an agent $x \in \mathcal{X}$ is to choose an agent $y \in \mathcal{Y}$, and the goal of each agent $x \in \mathcal{X}$ is to maximize her payoff $p_x(y)$. Given a strategy vector, agent x obtains payoff $p_x(y)$ if she proposes to y, and if she is the *winner* of y. Agent x is the winner of y if y ranks x highest among all agents who currently propose to him. Additionally, agent y obtains payoff $p_y(x)$ if x is the winner of y.

In general, there are no dependencies between the preference lists of agents. We introduce *correlated* two-sided markets to model correlated preference lists. Assume that there is a payoff $p_{x,y} \in \mathbb{N}$ associated with every pair (x, y) of agents $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ such that $p_x(y) = p_y(x) = p_{x,y}$. The preference lists of the agents from $\mathcal{X} \cup \mathcal{Y}$ are then defined according to these payoffs, e.g., an agent $y \in \mathcal{Y}$ prefers an agent $x \in \mathcal{X}$ to an agent $x' \in \mathcal{X}$ if $p_{x,y} > p_{x',y}$. We assume that for every agent i, the payoffs associated with all pairs including agent i are pairwise distinct. Then the preference lists are uniquely determined by the ordering of the

	m_1	m_2	m_3	...	m_{n-2}	m_{n-1}	m_n
w_1	1	2	3	...	$n-2$	$n-1$	n
w_2	n	1	2	...	$n-3$	$n-2$	$n-1$
w_3	$n-1$	n	1	...	$n-4$	$n-3$	$n-2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
w_{n-1}	3	4	5	...	n	1	2
w_n	2	3	4	...	$n-1$	n	1

Figure 2.2.1: The weights of the edges in our construction.

payoffs.

In a *many-to-one two-sided market*, the strategy space $\mathcal{F}_x \subseteq 2^{\mathcal{Y}}$ of every player $x \in \mathcal{X}$ is a collection of subsets of resources, that is, every player $x \in \mathcal{X}$ can propose to a subset $S_x \in \mathcal{F}_x$ of resources. Each resource $y \in \mathcal{Y}$ has a strict preference list over the set of players in $\mathcal{X}$. Given a vector of strategies $S = (S_1, \ldots, S_n)$ for the players from $\mathcal{X} = \{1, \ldots, n\}$, a resource y is matched to the winner x of y, that is, the most preferred player who proposes to y. The goal of each player $x \in \mathcal{X}$ is to maximize the total payoff of the resources that she wins. More formally, given a strategy vector S, let $T_x(S) \subseteq S_x$ be the set of resources that agent x wins. The goal of each player x is to maximize $\sum_{y \in T_x(S)} p_x(y)$.

A *matroid two-sided market* is a many-to-one two-sided market in which for each player x, the family $\mathcal{F}_x$ of subsets of resources corresponds to the independent sets of a matroid. In other words, in a matroid two-sided market for every player $x \in \mathcal{X}$, the set system $(\mathcal{Y}, \mathcal{F}_x)$ is a matroid. (Appendix A contains a brief introduction to matroids.) Such matroid two-sided markets arise naturally if, for example, every employer is interested in hiring a fixed number of workers or if the workers can be partitioned into different classes and a certain number of workers from each class is to be hired. We define *correlated matroid two-sided markets* analogously to the singleton case, that is, there is a payoff $p_{x,y} \in \mathbb{N}$ associated with every pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $p_x(y) = p_y(x) = p_{x,y}$.

2.2.1 Better Response Dynamics

In this section, we consider the random better response dynamics and present instances for which it takes with high probability an exponential number of steps to reach a stable matching. This is in contrast to the result due to Roth and Vande Vate [RV90] that from every matching there exists a polynomial sequence of better responses leading to a stable matching. In the following, we assume that $\mathcal{X} = \{w_1, \ldots, w_n\}$ and $\mathcal{Y} = \{m_1, \ldots, m_n\}$ for some $n \in \mathbb{N}$. We present our instances using an edge-weighted bipartite graph with an edge for each pair of woman and man. A woman w prefers a man m to a man m' if the weight of the edge $\{w, m\}$ is *smaller* than the weight of $\{w, m'\}$. On the other hand, a man m prefers a woman w to a woman w' if the weight of the edge $\{m, w\}$ is *larger* than the weight of the edge $\{m, w'\}$. The weights of the edges in the bipartite graph are depicted in Figure 2.2.1. Before we analyze the number of better responses needed to reach a stable matching, we prove a structural property of the instances shown in Figure 2.2.1.

Lemma 2.2.1. *For the family of two-sided markets that is depicted in Figure 2.2.1, a matching M is stable if and only if it is perfect and every woman has the same payoff in M.*

Proof. First we show that every perfect matching M in which every woman has the same payoff is stable. One crucial property of our construction is that whenever a woman w and a man m are married, the sum $p_w(m) + p_m(w)$ of their payoffs is $n-1$. In order to see this, assume that the edge between w and m has weight $l+1$. Then there are l men whom woman w prefers to m, i.e., $p_w(m) = n-1-l$. Furthermore, there are $n-1-l$ women whom man m prefers to w, i.e., $p_m(w) = l$. This implies $p_w(m) + p_m(w) = n-1$, regardless of l. We consider the case that every woman has payoff k and hence every man has a payoff of $n-1-k$ in M. Assume that there exists a blocking pair (w, m). Currently w has payoff k, m has payoff $n-1-k$, and w and m are not married to each other. Since (w, m) is a blocking pair, $p_w(m) > k$ and hence $p_m(w) = n-1-p_w(m) < n-1-k = p_m(M)$, contradicting the assumption that (w, m) is a blocking pair.

Now we have to show that a state M in which not every woman has the same payoff cannot be a stable matching. We can assume that M is a perfect matching as otherwise it obviously cannot be stable. Let M be a perfect matching and define $l(M)$ to be the lowest payoff that one of the women receives, i.e., $l(M) = \min\{p_w(M) \mid w \in \mathcal{X}\}$. Furthermore, by $L(M)$ we denote the set of women receiving payoff $l(M)$, i.e., $L(M) = \{w \in \mathcal{X} \mid p_w(M) = l(M)\}$. We claim that there exists at least one woman in $L(M)$ who forms a blocking pair with one of the men.

First we consider the case that the lowest payoff is unique, i.e., $L(M) = \{w\}$. Let m be the man with $p_w(m) = l(M)+1$. We claim that (w, m) is a blocking pair. To see this, let M' denote the matching obtained from M by resolving (w, m). We have to show that the payoff $p_m(M)$ of man m in matching M is smaller than his payoff $p_m(M')$ in M'. Due to our construction $p_m(M') = n-1-p_w(m)$ and $p_m(M) = n-1-p_{w'}(m)$, where w' denotes m's partner in M. Due to our assumption, w is the unique woman with the lowest payoff in M. Hence, $p_{w'}(m) = p_{w'}(M) > p_w(M) = p_w(m) - 1$. This implies $p_m(M') \geq p_m(M)$, which in turn implies $p_m(M') > p_m(M)$ since $w \neq w'$, and hence, (w, m) is a blocking pair.

It remains to consider the case that the woman with the lowest payoff is not unique. We claim that also in this case we can identify one woman in $L(M)$ who forms a blocking pair. Let $w^{(1)} \in L(M)$ be chosen arbitrarily and let $m^{(1)}$ denote her partner in M. Let $m^{(2)}$ denote the man with $p_{w^{(1)}}(m^{(2)}) = p_{w^{(1)}}(m^{(1)}) + 1$ and let $w^{(2)}$ denote the woman married to $m^{(2)}$ in M. If the payoff of $w^{(2)}$ in M is larger than the payoff of $w^{(1)}$ in M, then by the same arguments as for the case $|L(M)| = 1$ it follows that $(w^{(1)}, m^{(2)})$ is a blocking pair. Otherwise, if $p_{w^{(1)}}(M) = p_{w^{(2)}}(M)$, we continue our construction with $w^{(2)}$. To be more precise, we choose the man $m^{(3)}$ with $p_{w^{(2)}}(m^{(3)}) = p_{w^{(2)}}(m^{(2)}) + 1$ and denote by $w^{(3)}$ his partner in M. Again either $w^{(3)} \in L(M)$ or $(w^{(2)}, m^{(3)})$ is a blocking pair. In the former case, we continue the process analogously, yielding a sequence $m^{(1)}, m^{(2)}, m^{(3)}, \ldots$ of men. If the sequence is finite, a blocking pair exists. Now we consider the case that the sequence is not finite. Let $j \in \{1, \ldots, n\}$ be chosen such that $m^{(1)} = m_j$. Due to the weights shown in Figure 2.2.1, it holds $m^{(i)} = m_{(j-i \bmod n)+1}$ for $i \in \mathbb{N}$.

Hence, in this case, every man appears in the sequence, and hence, every woman has the same payoff $l(M)$. □

Now we can prove that with high probability the number of better responses needed to reach a stable matching is exponential.

Theorem 2.2.2. *There exists an infinite family of two-sided markets $I_1, I_2, I_3, \ldots$ and corresponding matchings $M_1, M_2, M_3, \ldots$ such that, for $n \in \mathbb{N}$, I_n consists of n women and n men and a sequence of random better responses starting in M_n needs $2^{\Omega(n)}$ steps to reach a stable matching with probability $1 - 2^{-\Omega(n)}$.*

Proof. We consider the instances shown in Figure 2.2.1. In Lemma 2.2.1, we have shown that in any stable matching all women have the same payoff. For a given matching M, we are interested in the most common payoff among the women and denote by $\chi(M)$ the number of women receiving this payoff, i.e.,

$$\chi(M) = \max_{i \in \{0,\ldots,n-1\}} |\{w \in \mathcal{X} \mid p_w(M) = i\}| \ .$$

In the following, we show that whenever $\chi(M)$ is at least $15n/16$, then $\chi(M)$ is more likely to decrease than to increase. This yields a biased random walk that takes with high probability exponentially many steps to reach $\chi(M) = n$. If the most common payoff is unique, which is always the case if $\chi(M) > n/2$, then we denote by $\mathcal{X}'(M)$ the set of women receiving this payoff and by $\mathcal{Y}'(M)$ the set of men matched to women from $\mathcal{X}'(M)$.

Let $\delta = 15/16$ and assume that $\chi(M) \geq \delta n$. First, we consider the case that the current matching M is not perfect, i.e., there exists at least one unmatched woman w and at least one unmatched man m. We call a blocking pair *good* if for the matching M' obtained from resolving it, $\chi(M') \leq \chi(M) - 1$. On the other hand, we call a blocking pair *bad* if $\chi(M') = \chi(M)+1$ or if M' is a perfect matching. Let us count the number of good and of bad blocking pairs. Let k denote the most common payoff. Both the unmarried woman w and the unmarried man m form a blocking pair with each person who prefers her/him to his/her current partner. Since the current payoff of the women in $\mathcal{X}'(M)$ is k, at most k of these women do not improve their payoff by marrying the unmarried man m. Analogously, since the payoff of the men in $\mathcal{Y}'(M)$ is $n-1-k$, at most $n-1-k$ of these men do not improve their payoff by marrying the unmarried woman w. This implies that the number of good blocking pairs is at least $\max\{\delta n - k, \delta n - n + 1 + k\} \geq (\delta - 1/2)n$. On the other hand, there can be at most $(1-\delta)n + 1$ bad blocking pairs. This follows easily because only women from $\mathcal{X} \setminus \mathcal{X}'(M)$ can form bad blocking pairs and each of these women forms at most one bad blocking pair as there is only one man who is at position $n - k$ in her preference list. Furthermore, there exists at most one blocking pair that makes the matching perfect.

The aforementioned arguments show that for a matching M with $\chi(M) \geq \delta n$ and sufficiently large n, the ratio of good blocking pairs to bad blocking pairs is bounded from below by

$$\frac{(\delta - 1/2)n}{(1-\delta)n + 1} \geq \frac{7}{2} \ .$$

This implies that the conditional probability of choosing a good blocking pair under the condition that either a good or a bad blocking pair is chosen is bounded from below by $7/9$.

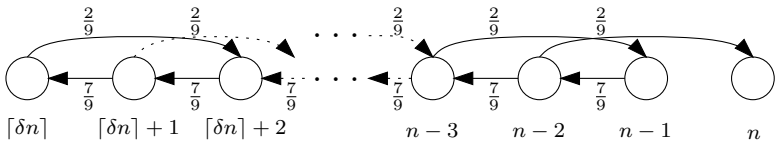


Figure 2.2.2: Transition probabilities of the random walk.

If a good blocking pair is chosen, χ decreases by at least 1. If a bad blocking pair is chosen, χ increases by 1 or the matching obtained is perfect. In the latter case, after the next step again a matching M'' is obtained that is not perfect. For this matching M'', we have $\chi(M'') \leq \chi(M) + 2$. Since we are interested in proving a lower bound, we can pessimistically assume that the current matching is not perfect and that whenever a bad blocking pair is chosen, χ increases by 2. Hence, we can obtain a lower bound on the number of better responses needed to reach a stable state, i.e., a state M with $\chi(M) = n$, by considering a random walk on the set $\{\lceil \delta n \rceil, \lceil \delta n \rceil + 1, \ldots, n\}$ that starts at $\lceil \delta n \rceil$, terminates when it reaches n, and has the transition probabilities as shown in Figure 2.2.2. This is a biased random walk. If we start with an arbitrary matching M satisfying $\chi(M) \leq \delta n$, then one can show by applying standard arguments from the theory of random walks (see, e.g., [Fel68]) that the biased random walk takes $2^{\Omega(n)}$ steps with probability $1 - 2^{-\Omega(n)}$ to reach state n. □

2.2.2 Best Response Dynamics

In this section, we study the best response dynamics in two-sided markets. We start by presenting an example that shows that this dynamics can cycle.

Theorem 2.2.3. *There exists a two-sided market with three women and three men in which the best response dynamics can cycle.*

Proof. Let w_1, w_2, w_3 denote the women and let m_1, m_2, m_3 denote the men. We choose the following preference lists for women and men:

w_1	m_2	m_3	m_1
w_2	m_1	m_2	m_3
w_3	m_3	m_1	m_2

m_1	w_1	w_3	w_2
m_2	w_2	w_1	w_3
m_3	w_1	w_2	w_3

We describe a state by a triple (x, y, z), meaning that the first woman is married to man m_x, the second woman to man m_y, and the third woman to man m_z. A value of -1 indicates that the corresponding woman is unmarried. The following sequence of states constitutes a cycle in the best response dynamics:

$$(-1, 2, 3) \to (3, 2, -1) \to (3, 1, -1) \to (3, -1, 1)$$
$$\to (2, -1, 1) \to (-1, 2, 1) \to (-1, 2, 3) \ . \qquad \square$$

Roth and Vande Vate [RV90] show that from every matching there exists a polynomial sequence of better responses leading to a stable matching. We show that this is also true for the best response dynamics.

Theorem 2.2.4. *For every two-sided market with n women and m men and every matching M, there exists a sequence of at most $2nm$ best responses starting in M and leading to a stable matching.*

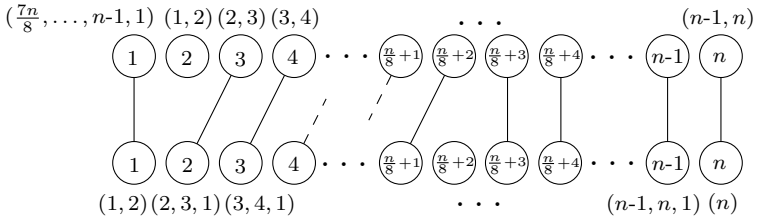


Figure 2.2.3: Nodes in the upper and lower row correspond to women and men, respectively. The figure also shows the initial state and the preference lists. The lists are only partially defined, but they can be completed arbitrarily.

Proof. We divide the sequence of best responses into two phases. In the first phase, only married women are allowed to change their marriages. If no married woman can improve her payoff anymore, then the second phase starts. In the second phase, all women are allowed to play best responses in an arbitrary order. In the first phase, we use the potential function

$$\Phi(M) = \sum_{x \in X} (m - p_x(M)) \ ,$$

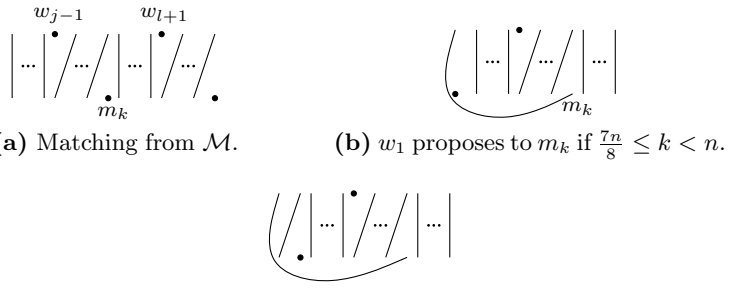
where X denotes the set of married women. This potential function decreases with every best response of a married woman by at least 1 because this woman increases her payoff and the set X can only become smaller. Since Φ is bounded from above by nm, the first phase terminates after at most nm best responses in a state in which no married woman can improve her payoff.

Now consider the second phase. We claim that if we start in a state M' in which no married woman can improve her marriage, then every sequence of best responses terminates after at most nm steps in a stable matching. Assume that we start in a state M' in which no married woman can improve her marriage and that an unmarried woman plays a best response and marries a man x, leading to state M''. Then the payoff of x can only increase. Hence, man x does not accept proposals in state M'' that he did not accept in M'. This implies that also in M'' no married woman can improve her marriage. Since no married woman becomes unhappy with her marriage, men are never left and therefore they can only improve their payoffs. With every best response one man increases his payoff by at least 1. This concludes the proof of the theorem as each of the m men can increase his payoff at most n times. □

Finally, we also extend Theorem 2.2.2 to the best response dynamics.

Theorem 2.2.5. *There exists an infinite family of two-sided markets $I_1, I_2, I_3, \ldots$ and corresponding matchings $M_1, M_2, M_3, \ldots$ such that, for $n \in \mathbb{N}$, I_n consists of n women and n men and a sequence of random best responses starting in M_n needs $2^{\Omega(n)}$ steps to reach a stable matching with probability $1 - 2^{-\Omega(n)}$.*

Proof. For every large enough $n \in \mathbb{N}$, we construct an instance I_n with n women and n men in which the preference lists and the initial state M_n are chosen as shown in Figure 2.2.3. That is, every woman w_i with $i \in \{2, \ldots, n\}$ prefers man m_{i-1} to man m_i whom she prefers to every other man. Woman w_1 prefers the men $m_{7n/8}, \ldots, m_{n-1}$ to man m_1 whom she prefers to every other man. Man m_1



(a) Matching from $\mathcal{M}$. **(b)** w_1 proposes to m_k if $\frac{7n}{8} \leq k < n$.

(c) A new diagonal is introduced.

Figure 2.2.4: One phase of the best response dynamics.

prefers woman w_1 to woman w_2 whom he prefers to every other woman. Every man m_i with $i \in \{2, \ldots, n-1\}$ prefers woman w_i to woman w_{i+1} whom he prefers to woman w_1 whom he prefers to every other woman. Man m_n prefers woman w_n to all other women.

Let $\mathcal{M}$ denote the set of matchings that contain the edges

$$(w_1, m_1), \ldots, (w_{j-2}, m_{j-2}), (w_j, m_{j-1}), \ldots, (w_k, m_{k-1}),$$
$$(w_{k+1}, m_{k+1}), \ldots, (w_l, m_l), (w_{l+2}, m_{l+1}), \ldots, (w_n, m_{n-1})$$

for some $j < k < l$ with $n/16 \leq k - j \leq n/4$, $k < n/4$, and $l \geq 5n/8$ (cf. Figure 2.2.4 (a)). We claim that if one starts in a matching that belongs to $\mathcal{M}$, then with probability $1-2^{-cn}$, for an appropriate constant $c > 0$, another matching from $\mathcal{M}$ is reached after $\Theta(n)$ many steps. Since no matching from $\mathcal{M}$ is stable, this implies the theorem.

If the current matching belongs to $\mathcal{M}$, then there are at most three women who have an incentive to change their marriage. Woman w_{j-1} can propose to man m_{j-1}, woman w_{k+1} can propose to man m_k, and, if $l < n$, woman w_{l+1} can propose to man m_{l+1}. Intuitively, as long as we are in a state that belongs to $\mathcal{M}$, there exists one block of diagonal marriages in the first half, and possibly a second block at the right end of the gadget. In every step the left end of the first block, the right end of the first block, and the left end of the second block move with the same probability one position to the right. Since the length of the first block is $\Omega(n)$, one can show by a standard application of a Chernoff bound that the probability that the first block vanishes, i.e., its left end catches up with its right end, before its right end reaches man m_n is exponentially small. Furthermore, since the distance between the first and the second block is $\Omega(n)$, the probability that the right end of the first block catches up with the left end of the second block before the second block has vanished is also exponentially small.

When the right end of the first block has reached man $m_{7n/8}$, i.e., $m_{7n/8}$ is unmarried, then with probability exponentially close to 1, the second block has already vanished (see Figure 2.2.4 (b)) because the initial distance between the two blocks is at least $3n/8$ and only with probability $2^{-\Omega(n)}$ it decreases to $n/8$ before the second block vanishes. Now consider the case that the second block has vanished and the right end of the first block lies in the interval $\{7n/8, \ldots, n-1\}$, woman w_1 has an incentive to change her marriage since she prefers m_k with $k \in \{7n/8, \ldots, n-1\}$ to m_1. Once she has changed her strategy, a new block of

diagonals can be created on the left end of the gadget (see Figure 2.2.4 (c)). In particular, woman w_1 will only return to m_1 if no man m_k with $k \in \{7n/8, \ldots, n-1\}$ is unmarried, that is, she will only return to m_1 if the right end of the first block has reached man m_n. Since it is as likely that a new diagonal at the beginning is inserted as it is that the right end of the block moves one position further to the right, the expected length of the newly created block is $n/8$. By Lemma 2.2.6 it follows that the length of the new block lies with high probability in the interval $[n/16, n/4]$. Only with exponentially small probability the left end of the block has not passed man $m_{5n/8}$ when the right end has reached man m_n because this would imply that the length of the block has increased from at most $n/4$ to $3n/8$. If none of these exponentially unlikely failures events occurs, we are again in a matching from $\mathcal{M}$. □

In the following lemma, we use the notion of a *geometric random variable with parameter* $1/2$. Such a random variable X describes in a sequence of Bernoulli trials with success probability $1/2$, the number of failures before the first success is obtained, that is, for $i \in \{0, 1, 2, \ldots\}$, $\mathbf{Pr}[X = i] = (1/2)^{i+1}$.

Lemma 2.2.6. *Let X be the sum of $n/8$ geometric random variables with parameter $p = 1/2$. There exists a constant $c > 0$ such that*

$$\mathbf{Pr}[X \notin [n/16, n/4]] \leq 2e^{-cn} \ .$$

Proof. The random variable X is negative binomially distributed with parameters $n/8$ and $1/2$. For a series of independent Bernoulli trials with success probability $1/2$, the random variable X describes the number of failures before the $(n/8)$-th success is obtained. For $a \in \mathbb{N}$, let Y_a be a binomially distributed random variable with parameters a and $1/2$. Then

$$\mathbf{Pr}[X > n/4] = \mathbf{Pr}\left[Y_{3n/8} < n/8\right] = \mathbf{Pr}\left[Y_{3n/8} < \frac{2}{3}\mathbf{E}\left[Y_{3n/8}\right]\right] \leq e^{-cn} \ ,$$

where the last inequality follows, for an appropriate constant $c > 0$, from a Chernoff bound (see, e.g., Theorem B.2.1). Furthermore,

$$\mathbf{Pr}[X < n/16] = \mathbf{Pr}\left[Y_{3n/16} > n/8\right] = \mathbf{Pr}\left[Y_{3n/16} > \frac{4}{3}\mathbf{E}\left[Y_{3n/16}\right]\right] \leq e^{-cn} \ . \quad \square$$

2.2.3 Correlated Two-Sided Markets

In this section, we show that in contrast to general two-sided markets, correlated two-sided markets are potential games in which the random better and best response dynamics converge in expected polynomial time.

Theorem 2.2.7. *Correlated two-sided markets are potential games, that is, the better response dynamics cannot cycle.*

Proof. We prove the theorem using a potential function $\Phi \colon \mathcal{M} \to \mathbb{N}^m$, where $\mathcal{M}$ denotes the set of all matchings and m is the number of resources. The potential function Φ maps a matching M to the vector $(\Phi_1(M), \ldots, \Phi_m(M))$, where $\Phi_i(M)$ denotes the i-th highest payoff received by one of the resources in M. We use the

convention that a resource has a payoff of -1 if it is unmatched. Let M denote the current matching and assume that a player $x \in \mathcal{X}$ plays a better response, which leads to a matching M'. We show that $\Phi(M')$ is lexicographically larger than $\Phi(M)$. Let y' denote the resource to which player x is matched in M'. There are two possible cases: either player x is unmatched in matching M or she is matched to a resource y in matching M. In the first case, the only change in the potential is that the payoff of resource y' increases. Hence, also the potential increases in that case. In the second case, player x improves and hence, $p_x(y') > p_x(y)$. Since we deal with correlated markets, we obtain $p_{y'}(x) = p_x(y') > p_x(y) = p_y(x)$. In the potential function, the element $p_y(x)$ is removed and the element $p_{y'}(x)$ is added and hence, the potential function increases lexicographically. □

It is easy to prove that correlated two-sided markets are not only potential games, but that also the random better and best response dynamics converge in expected polynomial time.

Theorem 2.2.8. *In correlated two-sided markets, the random better and best response dynamics reach a stable matching in expected polynomial time.*

Proof. Let n denote the number of players and let m denote the number of resources. We first consider the best response dynamics. Let p denote the highest possible payoff that can be achieved. As long as no pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $p_y(x) = p$ is contained in the matching, there exists one player whose best response would result in such a pair. Since this player is allowed to play a best response with probability at least $1/n$ in each step, it takes $O(n)$ best responses until a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $p_y(x) = p$ is contained in the matching in expectation. After that, player x never leaves resource y anymore. Furthermore, x cannot be displaced from y since no player is strictly preferred to x by resource y. Hence, the assignment of x to y can be fixed and we can remove x and y from the game. By this, we obtain another two-sided market with one player and resource less, and we can inductively apply the same argument to this game. Hence, the random best response dynamics terminates after $O(n^2)$ steps in expectation.

Similar arguments can also be applied to the random better response dynamics. As long as no pair with the highest possible payoff p is formed, there is at least one blocking pair (x, y) with $p_y(x) = p$. Since there can be at most nm blocking pairs, it takes $O(nm)$ steps in expectation until an assignment with profit p is obtained. Then we can remove player x and resource y and apply the same argument to the remaining two-sided market. □

Correlated Matroid Two-Sided Markets

In matroid two-sided markets, we consider a restricted class of better responses, so-called *lazy better* responses, introduced in [ARV06b]. Given a state S, we call a better response of a player $x \in \mathcal{X}$ from S_x to S'_x *lazy* if it can be decomposed into a sequence of strategies $S_x = S_x^0, S_x^1, \ldots, S_x^k = S'_x$ such that $|S_x^{i+1} \setminus S_x^i| = 1$ and the payoff of player x in state $S \oplus S_x^{i+1}$ is strictly larger than her payoff in state $S \oplus S_x^i$ for all $i \in \{0, \ldots, k-1\}$. That is, a lazy better response can be decomposed into a sequence of additions and exchanges of single resources such that each step strictly increases the payoff of the corresponding player. In Lemma A.1, we show

that for matroid strategy spaces, there does always exist a best response that is lazy. In particular, the best response that exchanges the least number of resources is lazy, and in singleton games, every better response is lazy.

Theorem 2.2.9. *Correlated matroid two-sided markets are potential games with respect to the lazy better response dynamics.*

Proof. The theorem follows by the same potential function as Theorem 2.2.7. Let S denote a state and assume that a player $x \in \mathcal{X}$ plays a lazy better response and changes her strategy from S_x to S'_x. Since the better response is lazy, it can be decomposed into a sequence of strategies $S_x = S_x^0, S_x^1, \ldots, S_x^k = S'_x$ such that $|S_x^{i+1} \setminus S_x^i| = 1$ and the payoff of player x in state $S \oplus S_x^{i+1}$ is strictly larger than her payoff in state $S \oplus S_x^i$ for all $i \in \{0, \ldots, k-1\}$. By the same arguments as in Theorem 2.2.7, it follows that the potential of $S \oplus S_x^{i+1}$ is lexicographically larger than the potential of $S \oplus S_x^i$ for every $i \in \{0, \ldots, k-1\}$, yielding the theorem. □

We show that the restriction to lazy better responses in Theorem 2.2.9 is necessary because even the best response dynamics can cycle in correlated matroid two-sided markets.

Theorem 2.2.10. *The best response dynamics in correlated matroid two-sided markets can cycle.*

Proof. Let $\mathcal{X} = \{1, 2\}$ denote the players and let $\mathcal{Y} = \{a, b, c, d\}$ denote the markets. The set of strategies of player 1 is

$$\mathcal{F}_1 = \{\emptyset, \{a\}, \{d\}\}$$

and the set of strategies of player 2 is

$$\mathcal{F}_2 = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, c\}, \{c, d\}, \{a, d\}\} \ .$$

The payoffs associated with the possible edges in $\mathcal{X} \times \mathcal{Y}$ are defined as follows:

$$p_{1,a} = 5, p_{1,d} = 3, p_{2,a} = 7, p_{2,b} = 1, p_{2,c} = 7, p_{2,d} = 2 \ .$$

Given these payoffs, the following sequence of states is a cycle in the best response dynamics:

$$(\{d\}, \{a, d\}) \to (\{d\}, \{b, c\}) \to (\{a\}, \{b, c\}) \to (\{a\}, \{a, d\}) \to (\{d\}, \{a, d\}) \ . \quad \square$$

Correlated matroid two-sided markets are not only potential games with respect to the lazy better response dynamics, but they also converge in expected polynomial time if at each point in time a player is chosen uniformly at random and allowed to play a lazy best response.

Theorem 2.2.11. *In correlated matroid two-sided markets, the random lazy best response dynamics converges to a stable matching in expected polynomial time.*

Proof. The proof follows the arguments in Theorem 2.2.8. Let p denote the highest possible payoff that can be achieved. It takes $O(n)$ best responses in expectation until a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $p_y(x) = p$ is contained in the matching. This

follows since players allocate optimal bases and an optimal basis of a matroid must contain the most valuable element. After an edge $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $p_y(x) = p$ is contained in the matching, player x will never leave resource y again because she only plays lazy best responses. Furthermore, x cannot be displaced from y since no player is strictly preferred to x by resource y. Hence, the assignment of x to y can be fixed and we can modify the strategy space of x by contracting its matroid by removing y (see Appendix A for a formal definition of this operation). By this contraction, we obtain another matroid two-sided market in which the rank of x's matroid is decreased by 1. Now we can inductively apply the same argument to this game. □

2.3 Player-Specific Congestion Games with Priorities

In Section 1.1.3, we introduced player-specific congestion games with priorities, in which each resource $r \in \mathcal{R}$ assigns a *priority* or *rank* $\mathrm{rk}_r(i)$ to every player $i \in \mathcal{N}$. For a state S, let $\mathrm{rk}_r(S) = \max_{i:r\in S_i} \mathrm{rk}_r(i)$ denote the rank of the resource r. We say that player i *allocates* resource r if $r \in S_i$, and we say that player i is *assigned* to resource r if $r \in S_i$ and $\mathrm{rk}_r(i) = \mathrm{rk}_r(S)$. We define $n_r^*(S)$ to be the number of players that are assigned to resource r, that is, the number of players i with $r \in S_i$ and $\mathrm{rk}_r(i) = \mathrm{rk}_r(S)$. The delay that an assigned player i incurs on r is $d_r^i(n_r^*(S))$, where d_r^i denotes a non-decreasing delay function. Players who allocate a resource but are not assigned to it incur an infinite delay. We say that the priorities are *consistent* if the priorities assigned to the players coincide for all resources. Since in singleton congestion games, each player can allocate only one resource at each point in time, we can specify the *state* of a singleton congestion game by a vector $(r_1, \ldots, r_n) \in \mathcal{R}^n$, where r_i denotes the resource allocated by player $i \in \mathcal{N}$.

This model can also be seen as a model of *two-sided markets with ties*. We define a two-sided market with ties to be a two-sided market in which the preference lists of the resources can have ties. Assume that every player $i \in \mathcal{N}$ proposes to a resource r_i. We say that a player $i \in \mathcal{N}$ is matched to resource $r \in \mathcal{R}$ if $r = r_i$ and if there is no player $j \in \mathcal{N}$ such that $r = r_j$ and j is strictly preferred to i by r. For a resource r, we denote by $n_r(S)$ the number of players proposing to r and by $n_r^*(S)$ the number of players that are matched to r. We assume that every player i has a non-increasing payoff function $p_r^i \colon \mathbb{N} \to \mathbb{N}$ for every resource r. A player i who is matched to resource r receives a payoff of $p_r^i(n_r^*(S))$. Also for two-sided markets with ties, we call a state S a stable matching if none of the players can increase her payoff given the proposals of the other players.

In the remainder of this section, we use the terminology of congestion games. Our results do not only hold for singleton congestion games, in which the strategy spaces of the players consist of singleton sets, but also for *matroid congestion games*, in which the strategies of each player are the bases of some matroid over the resources. To be more precise, for $i \in \mathcal{N}$, we denote by $\mathcal{I}_i$ the set $\{X \mid X \subseteq Y \in \Sigma_i\}$, and we assume that the set system $(\mathcal{R}, \mathcal{I}_i)$ is a matroid whose set of bases is Σ_i. For a matroid congestion game Γ, we denote by $\mathrm{rk}(\Gamma)$ the maximal rank of one of the strategy spaces of the players. Examples of matroid congestion games are singleton games and games in which the resources are the edges of a graph and every player has to allocate a spanning tree. Again, these games can also be seen

as an extension of two-sided markets in which each player can propose to a subset of resources instead of only one, so-called *many-to-one markets*, and in which the preference lists of the resources can have ties. We first state our results for the singleton case and present the extension to the matroid case in Section 2.3.3.

2.3.1 Congestion Games with Priorities

In this section, we consider singleton congestion games with priorities but without player-specific delay functions. For games with consistent priorities, we show that the better response dynamics reaches a Nash equilibrium after a polynomial number of *rounds*. We use the term round to denote a sequence of activations of players in which every player gets at least once the chance to improve. For example, our result implies that a polynomial (expected) number of better responses suffices if players are activated in a round-robin fashion or uniformly at random. We also prove that games in which different resources can assign different priorities to the players are potential games. We leave open the question whether they converge in a polynomial number of rounds.

Theorem 2.3.1. *In singleton congestion games with consistent priorities, the better response dynamics reaches a Nash equilibrium after a polynomial number of rounds.*

Proof. Ieong et al. [IMN$^+$05] prove that in singleton congestion games every sequence of better responses terminates in a Nash equilibrium after a polynomial number of steps. Since the players with the highest priority are not affected by the other players, the result by Ieong et al. shows that after a polynomial number of rounds, none of them has an incentive to change her strategy anymore. From that point on, the strategies of these players are fixed and we can again apply the result by Ieong et al. to the players with the second highest priority. After a polynomially number of rounds, also none of them has an incentive to change her strategy anymore. After that, the argument can be applied to the players with the third highest priority and so on. □

Next we consider congestion games in which different resources can assign different priorities to the players.

Theorem 2.3.2. *Singleton congestion games with priorities are potential games.*

Proof. We set $\mathcal{D} = (\mathbb{N} \cup \{\infty\}) \times \mathbb{N}$ and for elements $x = (x_1, x_2) \in \mathcal{D}$ and $y = (y_1, y_2) \in \mathcal{D}$ we denote by "$<$" the lexicographic order on $\mathcal{D}$ in which the first component is to be minimized and the second component is to be maximized, i.e., we define $x < y$ if and only if $x_1 < y_1$ or if $x_1 = y_1$ and $x_2 > y_2$. We construct a potential function $\Phi\colon \Sigma_1 \times \cdots \times \Sigma_n \to \mathcal{D}^n$ that maps every state $S = (r_1, \ldots, r_n)$ to a vector of values from $\mathcal{D}$. In state S, every resource $r \in \mathcal{R}$ contributes $n_r(S)$ values to the vector $\Phi(S)$ and $\Phi(S)$ is obtained by sorting all values contributed by the resources in non-decreasing order according to the lexicographic order defined above. Resource r contributes the values $(d_r(1), \mathrm{rk}_r(S)), \ldots, (d_r(n_r^*(S)), \mathrm{rk}_r(S))$ to the vector $\Phi(S)$ and $n_r(S) - n_r^*(S)$ times the value $(\infty, 0)$. We claim that if state S' is obtained from S by letting one player play a better response, then $\Phi(S')$ is

lexicographically smaller than $\Phi(S)$, i.e., there is a k with $\Phi_j(S) = \Phi_j(S')$ for all $j < k$ and $\Phi_k(S') < \Phi_k(S)$.

Assume that in state S player i plays a better response by changing her allocation from resource r_i to resource r_i'. We compare the two vectors $\Phi(S)$ and $\Phi(S')$, and we show that the smallest element added to the potential vector is smaller than the smallest element removed from the potential vector, showing that the potential decreases lexicographically. Due to the strategy change of player i, either the value $(d_{r_i}(n^*_{r_i}(S)), \mathrm{rk}_{r_i}(S))$ or the value $(\infty, 0)$ is replaced by the value $(d_{r_i'}(n^*_{r_i'}(S')), \mathrm{rk}_{r_i'}(S'))$. Since player i plays a better response, $d_{r_i'}(n^*_{r_i'}(S')) < d_{r_i}(n^*_{r_i}(S))$ or $d_{r_i'}(n^*_{r_i'}(S')) < \infty$, respectively, and hence, the term added to the potential is smaller than the term removed from the potential. In the following we show that all values that are contained in $\Phi(S)$ but not in $\Phi(S')$ are larger than $(d_{r_i'}(n^*_{r_i'}(S')), \mathrm{rk}_{r_i'}(S'))$. Clearly, only terms for the resources r_i and r_i' change and we can restrict our considerations to these two resources.

Let us consider resource r_i first. If the rank of r_i does not decrease by the strategy change of player i or if no player allocates resource r_i in state S', then only the term $(d_{r_i}(n^*_{r_i}(S)), \mathrm{rk}_{r_i}(S))$ or $(\infty, 0)$ is not contained in the vector $\Phi(S')$ anymore. All other terms contributed by resource r_i do not change. If the rank of resource r_i is decreased by the strategy change of player i, then additionally some terms $(\infty, 0)$ in the potential are replaced by other terms. Obviously, the removed terms $(\infty, 0)$ are larger than $(d_{r_i'}(n^*_{r_i'}(S')), \mathrm{rk}_{r_i'}(S'))$.

Now we consider resource r_i'. If the rank of r_i' does not increase by the strategy change of player i or if no player allocates r_i' in state S, then only the term $(d_{r_i'}(n^*_{r_i'}(S')), \mathrm{rk}_{r_i'}(S'))$ is added to the potential. All other terms contributed by r_i' do not change. If the rank of r_i' is increased by the strategy change of player i, then additionally the terms $(d_{r_i'}(1), \mathrm{rk}_{r_i'}(S)), \ldots, (d_{r_i'}(n^*_{r_i'}(S)), \mathrm{rk}_{r_i'}(S))$ are replaced by $n^*_{r_i'}(S)$ terms $(\infty, 0)$. In this case, $n^*_{r_i'}(S') = 1$ and the smallest removed term, $(d_{r_i'}(1), \mathrm{rk}_{r_i'}(S))$, is larger than $(d_{r_i'}(1), \mathrm{rk}_{r_i'}(S')) = (d_{r_i'}(n^*_{r_i'}(S')), \mathrm{rk}_{r_i'}(S'))$ because $\mathrm{rk}_{r_i'}(S') > \mathrm{rk}_{r_i'}(S)$. □

2.3.2 Player-Specific Congestion Games with Priorities

In this section, we consider singleton congestion games with priorities and player-specific delay functions and we show that these games always possess Nash equilibria. Our proof also yields an efficient algorithm for finding an equilibrium.

Theorem 2.3.3. *Every player-specific singleton congestion game with priorities possesses a pure Nash equilibrium that can be computed in polynomial time by $O(m \cdot n^3)$ strategy changes.*

Proof. In order to compute an equilibrium, we compute a sequence of states $S^0, \ldots, S^k$ such that S^0 is the state in which no player allocates a resource and S^k is a state in which every player allocates a resource. Remember that we distinguish between allocating a resource and being assigned to it. Our construction ensures the invariant that in each state S^a in this sequence, every player who allocates a resource has no incentive to change her strategy. Clearly, this invariant is true for S^0 and it implies that S^k is a pure Nash equilibrium.

In state S^a we pick an arbitrary player i who is allocating no resource and we let her play her best response. If in state S^a there is no resource to which i can be assigned, then i can allocate an arbitrary resource without affecting the players who are already allocating a resource and hence without affecting the invariant. It remains to consider the case that after her best response, player i is assigned to a resource r. If we leave the strategies of the other players unchanged, then the invariant may not be true anymore after the strategy change of player i. The invariant can, however, only be false for players who are assigned to resource r in state S^a. We distinguish between two cases in order to describe how the strategies of these players are modified in order to maintain the invariant.

First we consider the case that the rank of resource r does not change by the strategy change of player i. If there is a player j who is assigned to resource r in S^a and who can improve her strategy after i is also assigned to r, then we change the strategy of j to the empty set, i.e., in state S^{a+1} player j belongs to the set of players who do not allocate any resource. Besides this, no further modifications of the strategies are necessary because all other players are not affected by the replacement of j by i on resource r. In the case that the rank of resource r increases by the strategy change of player i, all players who are assigned to resource r in state S^a are set to their empty strategy in S^{a+1}.

It only remains to show that the described process terminates after a polynomial number of strategy changes in a stable state. We prove this by a potential function that is the lexicographic order of two components. The most important component is the sum of the ranks of the resources, i.e., $\sum_{r\in\mathcal{R}} \mathrm{rk}_r(S^a)$, which is to be maximized. Observe that this sum does not decrease in any of the two aforementioned cases, and that it increases strictly in the second case. Thus we need to show that after a polynomial number of consecutive occurrences of the first case, the second case must occur. Therefore, we need a second and less important component in our potential function. In order to define this component, we associate with every pair $(i,r) \in \mathcal{N}\times\mathcal{R}$ for which i is assigned to r in state S^a a *tolerance* $\mathrm{tol}_a(i,r)$ that describes how many players (including i) can be assigned to r without changing the property that r is an optimal strategy for i, i.e.,

$$\min\{\max\{b \mid \text{in } S^a,\ r \text{ is best resp. for } i \text{ if } i \text{ shares } r \text{ with } b-1 \text{ players}\}, n\}\ .$$

The second component of the potential function is the sum of the tolerances of the assigned pairs in S^a, which is to be maximized. We denote the set of assignments in state S^a by $E^a \subseteq \mathcal{N}\times\mathcal{R}$ and define the potential function as

$$\Phi(S^a) = \left(\sum_{r\in\mathcal{R}} \mathrm{rk}_r(S^a), \sum_{(i,r)\in E^a} \mathrm{tol}_a(i,r)\right)\ .$$

In every occurrence of the first case, the second component increases by at least 1. Since the values of the components are bounded from above by $m\cdot n$ and n^2 and bounded below from 0, the potential function implies that there can be at most $m\cdot n^3$ strategy changes before an equilibrium is reached. This does not include the last strategy change of players who are not assigned to any resource in the final state. In their last strategy change, these players allocate an arbitrary resource, which does not affect the potential. However, there are less than n such strategy changes. □

Let us remark that the potential function does not imply that the considered games are potential games because it increases only if the strategy changes are made according to the above described policy.

2.3.3 Extensions to Matroid Strategy Spaces

In this section, we study player-specific congestion games with priorities in which each strategy space Σ_i consists of the bases of a matroid over the resources. For this setting, we generalize the results that we obtained for the singleton case.

Theorem 2.3.4. *In matroid congestion games with consistent priorities, the best response dynamics reaches a Nash equilibrium after a polynomial number of rounds.*

For matroid congestion games, it is known that every sequence of best responses reaches a Nash equilibrium after a polynomial number of steps [ARV06a]. Using this result yields the theorem analogously to the proof of Theorem 2.3.1.

Theorem 2.3.5. *Matroid congestion games with priorities are potential games with respect to lazy better responses.*

Since lazy best responses (cf. Section 2.2.3) can be decomposed into exchanges of single resources, the same potential function as in the proof of Theorem 2.3.2 also works for the matroid case. The restriction to lazy better responses in Theorem 2.3.5 is necessary, as shown by the following result.

Theorem 2.3.6. *The best response dynamics in matroid congestion games with priorities can cycle.*

Proof. Let $\mathcal{N} = \{1, 2\}$ denote the players and let $\mathcal{R} = \{a, b, c, d\}$ denote the resources. The set of strategies of player 1 is

$$\Sigma_1 = \{\{a\}, \{d\}\}$$

and the set of strategies of player 2 is

$$\Sigma_2 = \{\{a, b\}, \{b, c\}, \{c, d\}, \{a, d\}\} \ .$$

Resource a assigns a higher priority to player 2, and resource d assigns the same priority to both players. The delay functions are chosen as follows:

$$d_a(1) = 1, d_b(1) = 3, d_c(1) = 1, d_d(1) = 2, d_d(2) = 4 \ .$$

Given these delays, the following sequence of states is a cycle in the best response dynamics:

$$(\{d\}, \{a, d\}) \to (\{d\}, \{b, c\}) \to (\{a\}, \{b, c\}) \to (\{a\}, \{a, d\}) \to (\{d\}, \{a, d\}) \ . \quad \square$$

Similar arguments as for Theorem 2.3.3 yield the following generalization.

Theorem 2.3.7. *Every player-specific matroid congestion game Γ with priorities possesses a pure Nash equilibrium that can be computed in polynomial time by $O(m \cdot n^3 \cdot \mathrm{rk}(\Gamma))$ strategy changes.*

Proof. For $i \in \mathcal{N}$, we denote by $\mathcal{I}_i$ the set $\{X \mid X \subseteq Y \in \Sigma_i\}$, and we assume that the set system $(\mathcal{R}, \mathcal{I}_i)$ is a matroid. We use the same arguments as in the proof of Theorem 2.3.3, that is, we compute a sequence of states $S^0, \dots, S^k$ such that S^0 is the state in which every player allocates the empty set and S^k is a Nash equilibrium. In contrast to the definition of matroid congestion games, where each player i is required to allocate a basis from Σ_i, we also allow *partial strategies* from $\mathcal{I}_i$ in states S^a with $a < k$. To be precise, in states S^a with $a < k$ it can happen that the set of resources that a player allocates is a strict subset of a basis. For a player $i \in \mathcal{N}$, let $\mathcal{R}_i^a \subseteq \mathcal{R}$ denote the set of resources she can be assigned to in state S^a, i.e., $\mathcal{R}_i^a$ contains exactly those resources that are in state S^a not assigned to a player that they strictly prefer to i. Let $\mathcal{M}_i^a = (\mathcal{R}_i^a, \Sigma_i^a)$ denote the matroid that is obtained from $\mathcal{M}_i$ by deleting all resources in $\mathcal{R} \setminus \mathcal{R}_i^a$ (see Appendix A for a formal definition of this operation). The following invariant will be true for all states S^a.

Invariant 2.3.8. *For every player $i \in \mathcal{N}$, there exists a basis $B_i^a \in \Sigma_i^a$ of the matroid $\mathcal{M}_i^a$ with $S_i^a \subseteq B_i^a$ that has minimum delay given the partial strategies of the other players in S^a.*

That is, if the other players do not change their strategies, no player is forced to leave resources that she currently allocates in order to obtain a basis with minimum delay. If the basis B_i^a of the matroid $\mathcal{M}_i^a$ is not a basis of the matroid $\mathcal{M}_i$, then player i has no strategy with finite delay given the partial strategies of the other players in S^a.

Now we describe how state S^{a+1} is obtained from state S^a. If in state S^a every player i allocates a basis of the matroid $\mathcal{M}_i^a$, then due to the invariant, an equilibrium S^{a+1} is obtained from S^a by letting each player i allocate an arbitrary basis B_i^{a+1} of $\mathcal{M}_i$ with $S_i^a \subseteq B_i^{a+1}$. Assume that there exists a player $i \in \mathcal{N}$ who is not allocating a basis of $\mathcal{M}_i^a$. In order to obtain S^{a+1}, we choose an arbitrary resource $r \in B_i^a \setminus S_i^a$ and let player i allocate r, i.e., we set $S_i^{a+1} = S_i^a \cup \{r\}$. If we leave all other strategies unchanged, then the invariant may not be true anymore.

We distinguish between three different cases in order to determine the strategies of the other players in state S^{a+1}.

1. If no player allocates r in S^a, then $S_j^{a+1} = S_j^a$ for all $j \in \mathcal{N} \setminus \{i\}$.

2. If i is ranked higher in r's preference list than the players assigned to r in S^a, then resource r is removed from the strategies of all players assigned to r in S^a, i.e., for all these players j we set $S_j^{a+1} = S_j^a \setminus \{r\}$. The strategies of all other players remain as in S^a.

3. If i is tied in r's preference list with the players assigned to r in state S^a, then we check whether the invariant stays true if additionally i is assigned to r. If this is not the case, then we remove one player k from r for whom the invariant becomes false, i.e., we set $S_k^{a+1} = S_k^a \setminus \{r\}$ and $S_j^{a+1} = S_j^a$ for all $j \in \mathcal{N} \setminus \{i, k\}$.

First we show that the invariant stays true in all three cases. This is based on the following property of matroids, which is proven in Appendix A.

Lemma 2.3.9. *Let $(\mathcal{R}, \mathcal{I})$ be a matroid with weights $w\colon \mathcal{R} \to \mathbb{N}$ and let B be a basis of minimum weight. If the weight of a single resource $r \in B$ is increased such that B is no longer of minimum weight, then, in order to obtain a basis of minimum weight, it suffices to exchange r with a resource $r' \in \mathcal{R}$ of minimum weight such that $B \cup \{r'\} \setminus \{r\}$ is a basis.*

Consider the first case and assume that the invariant is true in state S^a. Since no player is assigned to resource r in state S^a, there is no player whose current delay is increased by assigning i to r, but there can be players $j \in \mathcal{N}$ with $r \in B_j^a$. For these players, either B_j^a is still a basis of minimum delay or, due to Lemma 2.3.9, they can choose a basis B_j^{a+1} with $S_j^a \subseteq B_j^{a+1}$ of minimum delay given that i is assigned to r. Since players $j \in \mathcal{N}$ with $r \notin B_j^a$ are not affected by the strategy change of player i, the invariant is also true in state S^{a+1}. In the second case, the invariant stays true for all players who are assigned to r in state S^a because they just need to exchange r with another resource to obtain a basis with minimum delay again due to Lemma 2.3.9. It stays true for all other players j with $r \in B_j^a$ due to Lemma 2.3.9, and again players $j \in \mathcal{N}$ with $r \notin B_j^a$ are not affected by the strategy change of player i. In the third case, for all players $j \in \mathcal{N} \setminus \{i, k\}$ the effects of the strategy changes of i and k cancel each other out, and hence, these players are not affected by the strategy changes of i and k. The invariant stays true for k due to Lemma 2.3.9.

It only remains to show that the described process terminates after a polynomial number of strategy changes in an equilibrium. This follows by the same potential function as in the proof of Theorem 2.3.3. The upper bound on the second component of the potential function increases by a $\mathrm{rk}(\Gamma)$ factor, which accounts for the increased number of strategy changes. □

Chapter 3

Pareto-Optimal Solutions

This chapter is devoted to the study of bicriteria integer optimization problems. We prove the results discussed in Section 1.2 and state them in their full generality. In the preliminaries, we introduce the considered semi-random input model and define some notations used throughout the chapter. After that, we prove the upper and lower bounds on the expected number of Pareto-optimal solutions. Then we demonstrate that the upper bound can be used to simplify and to improve the analysis of certain structural properties of semi-random integer optimization problems. We conclude the chapter by showing how these structural properties can be applied to analyze the smoothed complexity of integer programming and to enumerating the Pareto set.

3.1 Preliminaries

In the model described in the introduction, an adversary specifies a set $\mathcal{S} \subseteq \mathcal{D}^n$ of feasible solutions, an arbitrary weight function $w\colon \mathcal{S} \to \mathbb{R}$ and a linear cost function $c\colon \mathcal{S} \to \mathbb{R}$ of the form $c(x) = c_1x_1 + \cdots + c_nx_n$ with coefficients $c_1, \ldots, c_n \in [-1, 1]$. These coefficients are then randomly perturbed by adding an independent Gaussian random variable with mean 0 and standard deviation σ to each of them. In other words, the costs are independent Gaussian random variables with standard deviation σ and the adversary can specify their mean values in $[-1, 1]$. Our analysis does not make use of the fact that the costs are Gaussian random variables. In fact, the adversary can specify arbitrary probability distributions with bounded density and finite absolute mean values.

In the following, we assume that c_i is a random variable with density f_i and that $f_i(x) \leq \phi_i$ for all $x \in \mathbb{R}$. Furthermore, we denote by μ_i the expected absolute value of c_i, i.e., $\mu_i = \mathbf{E}[|c_i|] = \int_{x\in\mathbb{R}} |x| f_i(x)\,dx$. Let $\phi = \max_{i\in[n]} \phi_i$ and $\mu = \max_{i\in[n]} \mu_i$, and by slight abuse of notation, let c not only denote the cost function, but also the vector $(c_1, \ldots, c_n)$. Then the cost $c(x)$ of a solution $x \in \mathcal{S}$ can be expressed as $c \cdot x$. We denote by $[n]$ the set $\{1, \ldots, n\}$, we use the notations $d = |\mathcal{D}|$ and $D = \max\{a - b \mid a, b \in \mathcal{D}\}$, and we denote by H_i the i-th harmonic number, i.e., $H_i = \sum_{j=1}^{i} j^{-1}$. Readers who are unfamiliar with continuous random variables are referred to Appendix B.1 for a brief introduction.

Similar to the standard deviation σ for Gaussian random variables, the parameter ϕ can be seen as a measure how close the analysis is to a worst-case analysis. The larger ϕ is chosen, the more concentrated the probability mass can be. For Gaussian and uniformly distributed random variables, ϕ is proportional to σ^{-1}.

3.2 Expected Number of Pareto-Optimal Solutions

In this section, we analyze the expected number of Pareto-optimal solutions in semi-random integer optimization problems.

3.2.1 Upper Bound

Since the costs are continuous random variables, the probability that there exist two solutions with exactly the same costs is zero. Hence, we can ignore this event and assume that no two solutions with the same costs exist. Furthermore, we assume without loss of generality that there are no two solutions with the same weight. If the adversary specifies a weight function in which two solutions have the same weight, we apply an arbitrary tie-breaking, which can only increase the expected number of Pareto-optimal solutions. The following theorem is a generalization of Theorem 1.2.1.

Theorem 3.2.1. *Let $\mathcal{S} \subseteq \mathcal{D}^n$ be a set of solutions with an arbitrary weight function $w\colon \mathcal{S} \to \mathbb{R}$ with a finite domain $\mathcal{D} \subset \mathbb{Z}$. Let q denote the number of Pareto-optimal solutions. Then, using the notations defined in Section 3.1,*

$$\mathbf{E}[q] \leq 2DdH_d \left(\sum_{i=1}^{n} \phi_i\right)\left(\sum_{i=1}^{n} \mu_i\right) + d(n+1) + 1 \ .$$

For $\mathcal{D} = \{0, \ldots, k-1\}$, the bound simplifies to

$$\mathbf{E}[q] = O(\mu\phi n^2 k^2 \log k) \ .$$

Notice that the number of Pareto-optimal solutions is not affected when all weights are scaled by some positive constant $a > 0$. Our bound is invariant under this scaling too. The random variable ac_i has maximal density ϕ_i/a and the expected absolute value is $a\mu_i$. Hence, the factor a cancels out in our bound.

Proof of Theorem 3.2.1. We start the proof by defining d subsets of the Pareto set. We say that a Pareto-optimal solution x belongs to class $a \in \mathcal{D}$ if there exists an index $i \in [n]$ with $x_i \neq a$ such that the succeeding Pareto-optimal solution y satisfies $y_i = a$, where succeeding Pareto-optimal solution refers to the Pareto-optimal solution with the smallest weight among all solutions with lower cost than x (cf. Fig. 3.2.1). The Pareto-optimal solution with the lowest cost, which does not have a succeeding Pareto-optimal solution, is not contained in any of the classes, but every other Pareto-optimal solution belongs to at least one class. Let q_a denote the number of Pareto-optimal solutions in class a. Since $q \leq 1 + \sum_{a\in\mathcal{D}} q_a$, linearity of expectation implies

$$\mathbf{E}[q] \leq 1 + \sum_{a\in\mathcal{D}} \mathbf{E}[q_a] \ . \tag{3.2.1}$$

In Lemma 3.2.2, we present a method for bounding the expected number of Pareto-optimal solutions in class 0. We conclude the proof of the theorem by showing that counting the expected number of Pareto-optimal solutions in class a for $a \in \mathcal{D}$ with $a \neq 0$ can be reduced to counting the expected number of Pareto-optimal solutions in class 0. Starting from the original set $\mathcal{S}$, we obtain

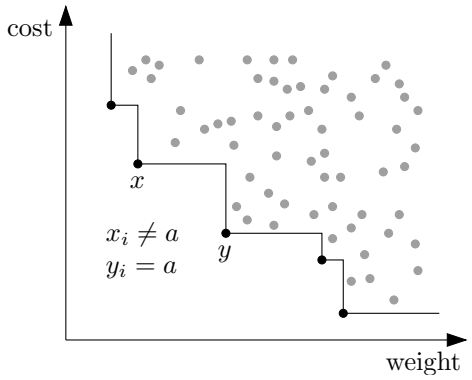


Figure 3.2.1: Solution y succeeds solution x, which is class-a Pareto-optimal.

a modified set $\mathcal{S}^a$ by subtracting $(a, \ldots, a)$ from each solution vector $x \in \mathcal{S}$, i.e., $\mathcal{S}^a = \{x - (a, \ldots, a) \mid x \in \mathcal{S}\}$. This way, the cost of each solution is reduced by $a \sum c_i$. Observe that this operation does not affect the set of Pareto-optimal solutions if the weights of the solutions remain as in $\mathcal{S}$. A solution x is class-a Pareto-optimal in $\mathcal{S}$ if and only if the corresponding solution $x - (a, \ldots, a)$ is class-0 Pareto-optimal in $\mathcal{S}^a$. Hence, the number q_a of class-a Pareto-optimal solutions in $\mathcal{S}$ corresponds to the number $q_0(S^a)$ of class-0 Pareto-optimal solutions in $\mathcal{S}^a$. We apply Lemma 3.2.2 for the solution set $\mathcal{S}^a$ with a corresponding domain $\mathcal{D}^a = \{z - a \mid z \in \mathcal{D}\}$. Since the difference between the largest and the smallest element of the domain does not change, Lemma 3.2.2 and Equation (3.2.1) yield that $\mathbf{E}[q]$ is bounded from above by

$$1 + \sum_{a \in \mathcal{D}} \mathbf{E}[q_0(S^a)] \leq 1 + \sum_{a \in \mathcal{D}} \left(2DH_d \left(\sum_{i=1}^{n} \phi_i \right) \left(\sum_{i=1}^{n} \mu_i \right) + n + 1 \right) \ ,$$

yielding the theorem. □

In the next lemma, we present an upper bound on the expected number of Pareto-optimal solutions in class 0.

Lemma 3.2.2. *Let $\mathcal{S} \subseteq \mathcal{D}^n$ be a set of solutions with an arbitrary weight function $w\colon \mathcal{S} \to \mathbb{R}$, where $\mathcal{D} \subset \mathbb{Z}$ denotes a finite domain with $0 \in \mathcal{D}$. Let q_0 denote the number of class-0 Pareto-optimal solutions. Then*

$$\mathbf{E}[q_0] \leq 2DH_d \left(\sum_{i=1}^{n} \phi_i \right) \left(\sum_{i=1}^{n} \mu_i \right) + n + 1 \ .$$

Proof. If the vector 0^n is a feasible solution, then we remove it from the set $\mathcal{S}$. Observe that if 0^n is a Pareto-optimal solution, then its preceding Pareto-optimal solution is class-0 Pareto-optimal, but it might not be class-0 Pareto-optimal with respect to the modified solution set $\mathcal{S} \setminus \{0^n\}$ anymore. Hence, removing 0^n from the set of feasible solutions can decrease the number of class-0 Pareto-optimal solutions by one. We take this into account and assume in the following that 0^n does not belong to $\mathcal{S}$.

The main part of the proof is an upper bound on the probability that there exists a class-0 Pareto-optimal solution whose cost lies in a small interval $(t, t + \varepsilon]$,

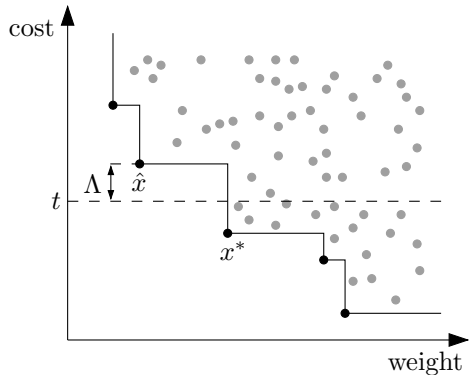


Figure 3.2.2: If $\hat{x}$ is a class-0 Pareto-optimal solution, then there must be an index i with $x_i^* = 0$ and $\hat{x}_i \neq 0$.

for some given $t \in \mathbb{R}$ and $\varepsilon > 0$. Roughly speaking, if ε is smaller than the smallest cost difference of two Pareto-optimal solutions, then this probability equals the expected number of class-0 Pareto-optimal solutions in the interval $(t, t+\varepsilon]$. Then we can divide $\mathbb{R}$ into intervals of length ε and sum these expectations to obtain the desired bound on the expected number of Pareto-optimal solutions.

Let $t \in \mathbb{R}$ be chosen arbitrarily. We define x^* to be the solution from $\mathcal{S}$ with the lowest weight among all solutions satisfying the constraint $c \cdot x \leq t$, that is,

$$x^* = \operatorname*{argmin}_{x \in \mathcal{S} : c \cdot x \leq t} w(x) \ .$$

If x^* exists, then it is Pareto-optimal. Let $\hat{x}$ denote the Pareto-optimal solution that precedes x^*, that is,

$$\hat{x} = \operatorname*{argmin}_{x \in \mathcal{S} : w(x) < w(x^*)} c \cdot x \ .$$

See Fig. 3.2.2 for an illustration of these definitions. We aim at bounding the probability that $\hat{x}$ is a class-0 Pareto-optimal solution whose cost falls into the interval $(t, t+\varepsilon]$.

In order to upper bound the probability that there exists a class-0 Pareto-optimal solution whose cost lies in the interval $(t, t+\varepsilon]$, we classify class-0 Pareto-optimal solutions to be *ordinary* or *extraordinary*. Considering only ordinary solutions allows us to prove a bound that depends not only on the length ε of the interval but also on $|t|$, the distance to zero. This captures the intuition that it becomes increasingly unlikely to observe solutions whose costs are much larger than the expected cost of the most expensive solution. The final bound is obtained by observing that there can be at most n extraordinary class-0 Pareto-optimal solutions.

We classify solutions to be ordinary or extraordinary as follows. Let x be a class-0 Pareto-optimal solution and let y be the succeeding Pareto-optimal solution, which must exist as the Pareto-optimal solution with the lowest cost is not class-0 Pareto-optimal. We say that x is extraordinary if for all indices $i \in [n]$ with $x_i \neq 0$ and $y_i = 0$, all Pareto-optimal solutions z that precede x satisfy $z_i \neq 0$. In other words, for those indices i that make x class-0 Pareto-optimal, y is the Pareto-optimal solution with the largest cost that is independent of c_i (see

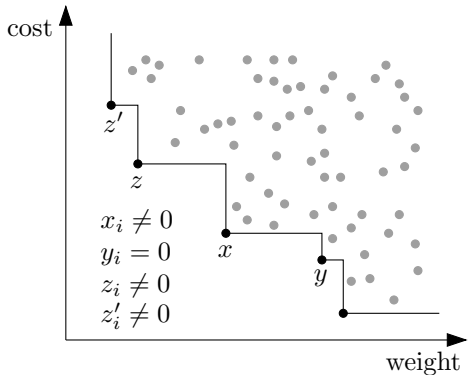


Figure 3.2.3: The solution x is an extraordinary class-0 Pareto-optimal solution if $z_i \neq 0$ and $z'_i \neq 0$ holds for all indices $i \in [n]$ for which $x_i \neq 0$ and $y_i = 0$.

Fig. 3.2.3). For every index $i \in [n]$, there can be at most one extraordinary class-0 Pareto-optimal solution. In the following, we restrict ourselves to solutions $\hat{x}$ that are ordinary, and we denote by $\mathcal{P}^0$ the set of ordinary class-0 Pareto-optimal solutions. We define the *loser gap* to be the slack of the solution $\hat{x}$ from the threshold t, that is,

$$\Lambda(t) = \begin{cases} c \cdot \hat{x} - t & \text{if } x^* \text{ and } \hat{x} \text{ exist and } \hat{x} \in \mathcal{P}^0, \\ \perp & \text{otherwise.} \end{cases}$$

If $\Lambda(t) \leq \varepsilon$, then there exists a solution $x \in \mathcal{P}^0$ with $c \cdot x \in (t, t+\varepsilon]$, namely $\hat{x}$. The reverse is not true because it might be the case that $\hat{x} \notin \mathcal{P}^0$ and that there exists another solution $x \in \mathcal{P}^0$ with $c \cdot x \in (t, t+\varepsilon]$. If, however, ε is smaller than the minimum cost difference of two Pareto-optimal solutions, then the existence of a solution $x \in \mathcal{P}^0$ with $c \cdot x \in (t, t+\varepsilon]$ implies $\hat{x} = x$ and hence $\Lambda(t) \leq \varepsilon$. Let $\mathcal{F}(\varepsilon)$ denote the event that there are two Pareto-optimal solutions whose costs differ by at most ε, then

$$\mathbf{Pr}\left[\exists x \in \mathcal{P}^0 \colon c \cdot x \in (t, t+\varepsilon] \,\middle|\, \neg\mathcal{F}(\varepsilon)\right] = \mathbf{Pr}\left[\Lambda(t) \leq \varepsilon \mid \neg\mathcal{F}(\varepsilon)\right] \ . \tag{3.2.2}$$

In the following, we estimate, for a given $a > 0$, the expected number of Pareto-optimal solutions whose costs lie in the interval $(-a, a]$. For this, we partition the interval $(-a, a]$ into $2am$ subintervals of length m^{-1} each, and we let the number $2am$ of subintervals tend infinity. For $m \in \mathbb{N}$ and $i \in \{0, \ldots, 2am-1\}$, we set $I_i^m = (b_i, b_{i+1}]$ with $b_i = -a + im^{-1}$. Since the number of Pareto-optimal solutions is always bounded by d^n, we obtain

$$\mathbf{E}\left[|\mathcal{P}^0|\right] \leq \lim_{m\to\infty} \left(\mathbf{Pr}\left[\neg\mathcal{F}\left(m^{-1}\right)\right] \cdot \mathbf{E}\left[|\mathcal{P}^0| \,\middle|\, \neg\mathcal{F}\left(m^{-1}\right)\right] + \mathbf{Pr}\left[\mathcal{F}(m^{-1})\right] \cdot d^n\right) \ .$$

The probability that two given solutions have a cost difference of at most ε can be bounded from above by $\varepsilon\phi$. Hence, a union bound over all pairs of solutions yields

$$\mathbf{Pr}\left[\mathcal{F}(m^{-1})\right] \leq d^{2n}\phi m^{-1} \ ,$$

which tends to 0 when m tends to infinity. Hence, it holds

$$\mathbf{E}\left[|\mathcal{P}^0|\right] \leq \lim_{m\to\infty} \left(\mathbf{Pr}\left[\neg\mathcal{F}\left(m^{-1}\right)\right] \cdot \mathbf{E}\left[|\mathcal{P}^0| \,\middle|\, \neg\mathcal{F}\left(m^{-1}\right)\right]\right) \ . \tag{3.2.3}$$

Under the condition $\neg\mathcal{F}\left(m^{-1}\right)$, every interval I_i^m can contain at most one Pareto-optimal solution, and hence, under this condition, the probability that I_i^m contains a Pareto-optimal solution from $\mathcal{P}^0$ equals the expected number of Pareto-optimal solutions from $\mathcal{P}^0$ in I_i^m, yielding together with (3.2.2) and (3.2.3) that the expected number of ordinary class-0 Pareto-optimal solutions with costs in $(-a, a]$ is bounded from above by

$$\begin{aligned}
&\lim_{m\to\infty}\left(\mathbf{Pr}\left[\neg\mathcal{F}\left(m^{-1}\right)\right]\cdot\sum_{i=0}^{2am-1}\mathbf{Pr}\left[\exists x\in\mathcal{P}^0\colon c\cdot x\in I_i^m\;\middle|\;\neg\mathcal{F}\left(m^{-1}\right)\right]\right)\\
&=\lim_{m\to\infty}\left(\mathbf{Pr}\left[\neg\mathcal{F}\left(m^{-1}\right)\right]\cdot\sum_{i=0}^{2am-1}\mathbf{Pr}\left[\Lambda(b_i)\le m^{-1}\;\middle|\;\neg\mathcal{F}\left(m^{-1}\right)\right]\right)\\
&\le\lim_{m\to\infty}\sum_{i=0}^{2am-1}\mathbf{Pr}\left[\Lambda(b_i)\le m^{-1}\right]\ . \qquad (3.2.4)
\end{aligned}$$

The only missing part is to analyze the probability of the event $\Lambda(t)\le\varepsilon$ for given $t\in\mathbb{R}$ and $\varepsilon>0$, which is done in the following lemma.

Lemma 3.2.3. *Let*

$$C^+=\sum_{j:c_j>0}c_j\quad\text{and}\quad C^-=\sum_{j:c_j<0}c_j\ ,$$

and let d^+ and d^- denote the largest and the smallest element in $\mathcal{D}$, respectively. For all $t\in\mathbb{R}$ and $\varepsilon>0$,

$$\mathbf{Pr}[\Lambda(t)\le\varepsilon]\le 2\varepsilon H_d\left(\sum_{i=1}^{n}\phi_i\right)\cdot\begin{cases}\mathbf{Pr}[d^+C^+ + d^-C^-\ge t] & \text{for } t\ge 0,\\ \mathbf{Pr}[d^+C^- + d^-C^+\le t] & \text{for } t\le 0.\end{cases}$$

By using Lemma 3.2.3, we can upper bound (3.2.4) by

$$\begin{aligned}
&\lim_{m\to\infty}\sum_{i=am}^{2am-1}\left(2H_d\left(\sum_{j=1}^{n}\phi_j\right)\cdot\mathbf{Pr}\left[d^+C^+ + d^-C^-\ge b_i\right]\cdot m^{-1}\right)\\
&+\lim_{m\to\infty}\sum_{i=0}^{am-1}\left(2H_d\left(\sum_{j=1}^{n}\phi_j\right)\cdot\mathbf{Pr}\left[-d^+C^- - d^-C^+\ge -b_i\right]\cdot m^{-1}\right)\ .
\end{aligned}$$

By the definition of the Riemann integral, we can rewrite the previous limit as

$$\begin{aligned}
&2H_d\left(\sum_{i=1}^{n}\phi_i\right)\int_0^a\mathbf{Pr}\left[d^+C^+ + d^-C^-\ge t\right]\,dt\\
&+2H_d\left(\sum_{i=1}^{n}\phi_i\right)\int_0^a\mathbf{Pr}\left[-d^+C^- - d^-C^+\ge t\right]\,dt\ .
\end{aligned}$$

This term is an upper bound on the expected number of ordinary class-0 Pareto-optimal solutions in the interval $(-a, a]$. As $0\in\mathcal{D}$, it holds $d^+\ge 0$ and $d^-\le 0$,

and hence we have $d^+C^+ + d^-C^- \geq 0$ and $-d^+C^- - d^-C^+ \geq 0$. Using this and letting a tend to infinity yield that the expected number of ordinary class-0 Pareto-optimal solutions can be bounded from above by

$$\begin{aligned}
&2H_d \left(\sum_{i=1}^{n} \phi_i\right) \left(\mathbf{E}\left[d^+C^+ + d^-C^-\right] + \mathbf{E}\left[-d^+C^- - d^-C^+\right]\right) \\
&= 2H_d \left(\sum_{i=1}^{n} \phi_i\right) (d^+ - d^-) \cdot \mathbf{E}\left[C^+ - C^-\right] \\
&= 2H_d \left(\sum_{i=1}^{n} \phi_i\right) (d^+ - d^-) \cdot \mathbf{E}\left[\sum_{i=1}^{n} |c_i|\right] = 2DH_d \left(\sum_{i=1}^{n} \phi_i\right) \left(\sum_{i=1}^{n} \mu_i\right) \quad .
\end{aligned}$$

Taking into account that at most n class-0 Pareto-optimal solutions can be extraordinary and that by removing 0^n from $\mathcal{S}$ we might have decreased the number of class-0 Pareto-optimal solutions by one yields the lemma. □

We conclude the proof of Theorem 3.2.1 by proving Lemma 3.2.3.

Proof of Lemma 3.2.3. In order to analyze the probability of the event $\Lambda(t) \leq \varepsilon$, we define a set of auxiliary random variables such that $\Lambda(t)$ is guaranteed to always take a value also taken by one of the auxiliary random variables. Then we analyze the auxiliary random variables and use a union bound to conclude the desired bound for $\Lambda(t)$.

Define $\mathcal{D}' = \mathcal{D} \setminus \{0\}$ and $\mathcal{S}^{x_i=v} = \{x \in \mathcal{S} \mid x_i = v\}$ for all $i \in [n]$ and $v \in \mathcal{D}$. We denote by $x^{*(i)}$ the solution from $\mathcal{S}^{x_i=0}$ with lowest weight with cost at most t, that is,

$$x^{*(i)} = \operatorname*{argmin}_{x \in \mathcal{S}^{x_i=0} : c \cdot x \leq t} w(x) \quad .$$

For each $i \in [n]$ and $v \in \mathcal{D}'$, we define the set $\mathcal{L}^{(i,v)}$ as follows. If $x^{*(i)}$ does not exist or if no solution in $\mathcal{S}^{x_i=0}$ with cost larger than t exists, then we define $\mathcal{L}^{(i,v)} = \emptyset$. Otherwise $\mathcal{L}^{(i,v)}$ consists of all solutions from $\mathcal{S}^{x_i=v}$ that have smaller weight than $x^{*(i)}$. Let $\hat{x}^{(i,v)}$ denote the Pareto-optimal solution from the set $\mathcal{L}^{(i,v)}$ with the lowest cost, that is,

$$\hat{x}^{(i,v)} = \operatorname*{argmin}_{x \in \mathcal{L}^{(i,v)}} c \cdot x \quad .$$

Finally, we define for each $i \in [n]$ and $v \in \mathcal{D}'$, the auxiliary random variable

$$\Lambda_i^v(t) = \begin{cases} c \cdot \hat{x}^{(i,v)} - t & \text{if } \hat{x}^{(i,v)} \text{ exists,} \\ \perp & \text{otherwise.} \end{cases}$$

If $\Lambda_i^v(t) \in (0, \varepsilon]$, which excludes $\Lambda_i^v(t) = \perp$, then the following three events must co-occur:

$\mathcal{E}_1$**:** There exists a solution $x \in \mathcal{S}^{x_i=0}$ with $c \cdot x \leq t$ (namely $x^{*(i)}$).

$\mathcal{E}_2$**:** There exists a solution $x \in \mathcal{S}^{x_i=0}$ with $c \cdot x > t$ (due to the definition of $\mathcal{L}^{(i,v)}$).

$\mathcal{E}_3$**:** The solution $\hat{x}^{(i,v)}$ exists and its cost lies in the interval $(t, t + \varepsilon]$.

The events $\mathcal{E}_1$ and $\mathcal{E}_2$ depend only on the costs c_j, $j \neq i$. The existence and identity of $\hat{x}^{(i,v)}$ is completely determined by those costs as well. Hence, if we fix all costs except for c_i, then $\hat{x}^{(i,v)}$ is fixed and its cost is $\kappa + vc_i$ for some constant κ that depends on the costs already fixed. Observe that the random variable $\kappa + vc_i$ has density at most $\phi_i/|v|$. Hence, we obtain

$$\mathbf{Pr}\left[c \cdot \hat{x}^{(i,v)} \in (t, t+\varepsilon] \,\middle|\, \hat{x}^{(i,v)} \text{ exists} \right] \leq \varepsilon \frac{\phi_i}{|v|} \ .$$

For $t \geq 0$, the event $\mathcal{E}_2$ implies $d^+C^+ + d^-C^- > t$, and hence,

$$\mathbf{Pr}\left[\hat{x}^{(i,v)} \text{ exists}\right] \leq \mathbf{Pr}[\mathcal{E}_2] \leq \mathbf{Pr}\left[d^+C^+ + d^-C^- > t\right] \ .$$

For $t \leq 0$, the event $\mathcal{E}_1$ implies $d^+C^- + d^-C^+ \leq t$, and hence,

$$\mathbf{Pr}\left[\hat{x}^{(i,v)} \text{ exists}\right] \leq \mathbf{Pr}[\mathcal{E}_1] \leq \mathbf{Pr}\left[d^+C^- + d^-C^+ \leq t\right] \ .$$

By combining these results, we get

$$\mathbf{Pr}[\Lambda_i^v(t) \in (0, \varepsilon]] \leq \varepsilon \frac{\phi_i}{|v|} \cdot \begin{cases} \mathbf{Pr}[d^+C^+ + d^-C^- > t] & \text{for } t \geq 0, \\ \mathbf{Pr}[d^+C^- + d^-C^+ \leq t] & \text{for } t \leq 0. \end{cases}$$

Next we argue that $\Lambda(t) \leq \varepsilon$ implies $\Lambda_i^v(t) \in (0, \varepsilon]$ for at least one pair $(i, v) \in [n] \times \mathcal{D}'$. So assume that $\Lambda(t) \leq \varepsilon$. By definition, x^* and $\hat{x}$ exist and $\hat{x}$ is an ordinary class-0 Pareto-optimal solution. Since $\hat{x}$ is class-0 Pareto-optimal and x^* is the succeeding Pareto-optimal solution, there exists an index $i \in [n]$ such that

(a) $x_i^* = 0$ and $\hat{x}_i = v \neq 0$ for some $v \in \mathcal{D}'$, and

(b) there exists a solution $x \in \mathcal{S}$ with cost larger than t.

The second condition is a consequence of the assumption that $\hat{x}$ is not extraordinary, i.e., there exists a Pareto-optimal solution z with $z_i = 0$ that has larger cost than $\hat{x}$ and hence larger cost than t. Recall that $x^{*(i)}$ is defined to be the solution with the smallest weight in $\mathcal{S}^{x_i=0}$ with $c \cdot x \leq t$. As $x^* \in \mathcal{S}^{x_i=0}$, $x^* = x^{*(i)}$. Moreover, $\mathcal{L}^{(i,v)}$ consists of all solutions from $\mathcal{S}^{x_i=v}$ that have smaller weight than x^*. Thus, $\hat{x} \in \mathcal{L}^{(i,v)}$. By construction, $\hat{x}$ has the smallest weight among the solutions in $\mathcal{L}^{(i,v)}$ and therefore, $\hat{x}^{(i,v)} = \hat{x}$ and $\Lambda_i^v(t) = \Lambda(t)$. Applying a union bound yields, for all $t \geq 0$,

$$\begin{aligned} \mathbf{Pr}[\Lambda(t) \leq \varepsilon] &\leq \sum_{i=1}^{n} \sum_{v \in \mathcal{D}'} \mathbf{Pr}[\Lambda_i^v(t) \in (0, \varepsilon]] \\ &\leq \sum_{i=1}^{n} \sum_{v \in \mathcal{D}'} \mathbf{Pr}\left[d^+C^+ + d^-C^- \geq t\right] \varepsilon \frac{\phi_i}{|v|} \\ &\leq 2\varepsilon H_d \left(\sum_{i=1}^{n} \phi_i \right) \cdot \mathbf{Pr}\left[d^+C^+ + d^-C^- \geq t\right] \ . \end{aligned}$$

For $t \leq 0$, we get analogously

$$\mathbf{Pr}[\Lambda(t) \leq \varepsilon] \leq 2\varepsilon H_d \left(\sum_{i=1}^{n} \phi_i \right) \cdot \mathbf{Pr}\left[d^+C^- + d^-C^+ \leq t\right] \ . \qquad \square$$

3.2.2 Lower Bound for Linear Weight Functions

In this section, we present a lower bound of $\Omega(n^2 k \log(k+1))$ on the expected number of Pareto-optimal solutions for $\mathcal{D} = \{0, \dots, k\}$, generalizing a bound of $\Omega(n^2)$ for the binary case presented in [BV04]. In the following section, we prove the stronger bound $\Omega(n^2k^2)$ under slightly stronger assumptions. In the bound presented in this section, the weight function is linear, that is, we specify a vector of weights $w = (w_1, \dots, w_n)$ such that $w(x) = w_1x_1 + \cdots + w_nx_n$. For the stronger bound we can only prove that there is some weight function $w\colon \mathcal{S} \to \mathbb{R}$ for which the bound holds, but this function might not be linear.

For the sake of an intuitive presentation, we describe the lower bounds in terms of the bounded knapsack problem. That is, we replace the cost function $c\colon \mathcal{S} \to \mathbb{R}$ by a profit function $p\colon \mathcal{S} \to \mathbb{R}$ which is to be maximized. For linear weight functions, we prove the following lower bound on the expected number of Pareto-optimal solutions.

Theorem 3.2.4. *Let $\mathcal{D} = \{0, \dots, k\}$ and suppose that the profits are drawn independently at random according to a continuous probability distribution with non-increasing density function $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Let q denote the number of Pareto-optimal solutions over $\mathcal{S} = \mathcal{D}^n$. Then there is a vector of weights $w_1, \dots, w_n \in \mathbb{R}_{>0}$ for which*

$$\mathbf{E}[q] \geq \frac{H_k}{4} k(n^2 - n) + kn + 1 \ .$$

If the profits are drawn according to the uniform distribution over some interval $[0, a]$ with $a > 0$, then the above inequality holds with equality.

Similarly, a lower bound of $\Omega(n^2 k \log(k+1))$ can be obtained for the case that f is the density of a Gaussian random variable with mean 0. Since all weights w_i are larger than 0, an item with a negative profit cannot be contained in any Pareto-optimal solution. Hence, we can ignore those items. Restricted to the interval $[0, \infty)$, the density of a Gaussian random variable with mean 0 is non-increasing and hence we can apply Theorem 3.2.4 to the items with positive profit. With high probability, there are $\Omega(n)$ such items. By similar reasoning, also Theorem 1.2.3 is implied by Theorem 3.2.4. By the same arguments as for Gaussian random variables, Theorem 3.2.4 can also be applied if the profits are chosen uniformly at random from the interval $[-1, 1]$. Choosing profits, which are to be maximized, uniformly at random from the interval $[-1, 1]$ is equivalent to choosing costs, which are to be minimized, uniformly at random from $[-1, 1]$, which implies Theorem 1.2.3.

Proof of Theorem 3.2.4. The set $\mathcal{S} = \mathcal{D}^n$ corresponds to the solution set of the bounded knapsack problem in which up to k identical instances of each item can be put into the knapsack. For the sake of a simple presentation, we describe our construction in terms of this knapsack problem. We fix the weights of all items by setting $w_i = (k+1)^i$ for all $i \in [n]$. This way, the lexicographic order of the solutions in $\mathcal{S}$ is the same as the order defined by the weight $w \cdot x$ of solutions. Since the density function of the profits is assumed to be non-increasing, the distribution function $F\colon \mathbb{R}_{\geq 0} \to [0, 1]$ is concave as $F' = f$. Furthermore, $F(0) = 0$. Observe that such a function is sub-additive, that is, $F(a + b) \leq F(a) + F(b)$ for every $a, b \geq 0$.

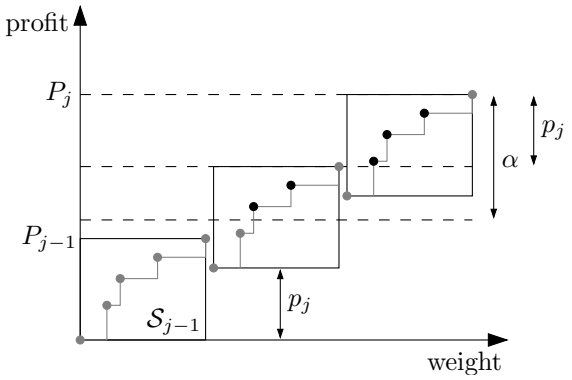


Figure 3.2.4: In this example, event A_1^j occurs. We have $k = 2$ and $X_\alpha^j = X_{p_j}^{j-1} + X_{\alpha-p_j}^{j-1} + 1$ with $X_{p_j}^{j-1} = 3$ and $X_{\alpha-p_j}^{j-1} = 2$.

Let $\mathcal{S}_j$ denote the set of the first $(k+1)^j$ solutions in the lexicographic order, which are exactly those solutions that contain only items $1, \ldots, j$. We define $P_j = k\sum_{i=1}^{j} p_i$ and we denote by $\mathcal{P}_j$ the set of Pareto-optimal solutions over $\mathcal{S}_j$. Observe that the last solution in $\mathcal{S}_j$ has profit P_j and it is Pareto-optimal with probability 1.

For any given $\alpha > 0$, let X_α^j denote the number of Pareto-optimal solutions in $\mathcal{P}_j$ with profit at least $P_j - \alpha$, not counting the last solution in this sequence, which is $(k, \ldots, k, 0, \ldots, 0)$. By induction we show $\mathbf{E}\left[X_\alpha^j\right] \geq j\sum_{i=1}^{k} F(\frac{\alpha}{i})$, where F denotes the distribution function of the profits. We partition the interval $[0, \infty)$ into disjoint intervals $I_0 = (\alpha, \infty)$, $I_l = (\alpha/(l+1), \alpha/l]$ for $l \in [k-1]$, and $I_k = [0, \alpha/k]$. For every $i \in [n]$ and for $l \in \{0, \ldots, k\}$, we denote by A_l^i the event that p_i lies in the interval I_l. For all $l \in [k-1]$ it holds $\mathbf{Pr}\left[A_l^i\right] = F(\alpha/l) - F(\alpha/(l+1))$, and it holds $\mathbf{Pr}\left[A_k^i\right] = F(\alpha/k)$.

For $j = 1$, the base case of the induction, observe that $X_\alpha^1 = \min\{\lfloor \alpha/p_1 \rfloor, k\}$ and hence $X_\alpha^1 = l$ holds if and only if the event A_l^1 occurs. Hence,

$$\mathbf{E}\left[X_\alpha^1\right] = \sum_{l=0}^{k} l \cdot \mathbf{Pr}\left[A_l^1\right] = k \cdot \mathbf{Pr}\left[A_k^1\right] + \sum_{l=1}^{k-1} l\big(F(\alpha/l) - F(\alpha/(l+1))\big) = \sum_{l=1}^{k} F\left(\frac{\alpha}{l}\right) \ .$$

Now we consider the case $j > 1$. We group the solutions in $\mathcal{S}_j$ into $k+1$ blocks, with block $l \in \{0, \ldots, k\}$ containing all solutions with $x_j = l$. Block 0 corresponds to $\mathcal{S}_{j-1}$. Each Pareto-optimal solution in $\mathcal{S}_{j-1}$ with profit in the interval $(P_{j-1} - p_j, P_{j-1}]$ gives rise to one new Pareto-optimal solution in each of the k following blocks. In the event A_0^j we have $X_\alpha^j = X_\alpha^{j-1}$ because all solutions that contribute to X_α^j are in block k. In the event A_1^j we have $X_{p_j}^{j-1} + 1$ Pareto-optimal solutions in block k and $X_{\alpha-p_j}^{j-1} + 1$ Pareto-optimal solutions in block $k-1$. Since the last solution is not counted in X_α^j, we have $X_\alpha^j = X_{p_j}^{j-1} + X_{\alpha-p_j}^{j-1} + 1$ (cf. Figure 3.2.4). By similar reasoning, event A_l^j implies $X_\alpha^j = lX_{p_j}^{j-1} + X_{\alpha-lp_j}^{j-1} + l$. Hence, it follows

that we can lower bound the expected value of X_α^j by

$$\begin{aligned}
&\sum_{l=0}^{k} \mathbf{Pr}\left[A_l^j\right] \left(l \cdot \mathbf{E}\left[X_{p_j}^{j-1} \;\middle|\; A_l^j\right] + \mathbf{E}\left[X_{\alpha-lp_j}^{j-1} \;\middle|\; A_l^j\right] + l\right) \\
&= \sum_{l=0}^{k} \int_{x \in I_l} f(x) \cdot \left(l \cdot \mathbf{E}\left[X_x^{j-1}\right] + \mathbf{E}\left[X_{\alpha-lx}^{j-1}\right] + l\right) dx \\
&\geq \sum_{l=0}^{k} \int_{x \in I_l} f(x) \cdot \left(l \cdot (j-1) \sum_{i=1}^{k} F\left(\frac{x}{i}\right) + (j-1) \sum_{i=1}^{k} F\left(\frac{\alpha - lx}{i}\right) + l\right) dx \;\;,
\end{aligned}$$

where the last inequality follows from the induction hypothesis. We can further rewrite this term as

$$\begin{aligned}
&\sum_{l=0}^{k} \int_{x \in I_l} f(x) \cdot \left((j-1) \sum_{i=1}^{k} \left[l \cdot F\left(\frac{x}{i}\right) + F\left(\frac{\alpha - lx}{i}\right)\right] + l\right) dx \\
&\geq \sum_{l=0}^{k} \int_{x \in I_l} f(x) \cdot \left((j-1) \sum_{i=1}^{k} F\left(\frac{\alpha}{i}\right) + l\right) dx \\
&= (j-1) \sum_{i=1}^{k} F\left(\frac{\alpha}{i}\right) + \sum_{l=0}^{k} l \cdot \mathbf{Pr}\left[A_l^j\right] = j \sum_{i=1}^{k} F\left(\frac{\alpha}{i}\right) \;\;,
\end{aligned}$$

where the inequality is due to the fact that the function F is sub-additive. If every profit is chosen uniformly at random from some interval $[0, a]$ with $a > 0$, then this term equals exactly the expected number of Pareto-optimal solutions.

Now let $Y_j = |\mathcal{P}_j| - |\mathcal{P}_{j-1}|$ denote the number of new Pareto-optimal solutions in $\mathcal{P}_j$. Observe that $Y_j = kX_{p_j}^{j-1} + k$. The additive k is due to the fact that the last solution in $\mathcal{P}_{j-1}$ is not counted in $X_{p_j}^{j-1}$ but yields k new solutions in $\mathcal{P}_j$. Since p_j and X_α^{j-1} are independent, the induction hypothesis implies

$$\mathbf{E}[Y_j] = \mathbf{E}\left[kX_{p_j}^{j-1} + k\right] \geq \mathbf{E}\left[k(j-1) \sum_{i=1}^{k} F\left(\frac{p_j}{i}\right) + k\right] \;\;.$$

Furthermore, the number of Pareto-optimal solutions in $\mathcal{P}_n$ is $q = 1 + \sum_{j=1}^{n} Y_j$. The additional 1 is due to the first solution $(0, \ldots, 0)$, which is always Pareto-optimal. Therefore,

$$\begin{aligned}
\mathbf{E}[q] &= 1 + \sum_{j=1}^{n} \mathbf{E}[Y_j] = 1 + \sum_{j=1}^{n} \mathbf{E}\left[kX_{p_j}^{j-1} + k\right] \\
&\geq 1 + \sum_{j=1}^{n} \mathbf{E}\left[k(j-1) \sum_{i=1}^{k} F\left(\frac{p_j}{i}\right) + k\right] \;\;.
\end{aligned}$$

The random variable $F(p_j)$ is uniformly distributed over the interval $[0, 1]$, thus $\mathbf{E}[F(p_j)] = 1/2$. As F is sub-additive, $i \cdot F(p_j/i) \geq F(p_j)$ holds, which implies $\mathbf{E}[F(p_j/i)] \geq \mathbf{E}[F(p_j)/i] = 1/(2i)$. Using $\mathbf{E}\left[\sum_{i=1}^{k} F(\frac{p_j}{i})\right] = \frac{1}{2} H_k$ yields

$$\mathbf{E}[q] \geq \frac{H_k}{4} k(n^2 - n) + kn + 1 \;\;.$$

If the profits are drawn according to the uniform distribution over some interval $[0, a]$ with $a > 0$, then the above inequality holds with equality. □

3.2.3 Lower Bound for General Weight Functions

Now we consider general weight functions and show a lower bound of $\Omega(n^2k^2)$ on the expected number of Pareto-optimal solutions for $\mathcal{D} = \{0, \dots, k\}$ and $\mathcal{S} = \mathcal{D}^n$. Every weight function induces a *ranking* on the set of solutions, and in the following, we use the terms weight function and ranking synonymously. We assume that k is a function of n with $(5(c+1)+1)\log n \leq k \leq n^c$ for some constant c. We use the *probabilistic method* to show that, for each sufficiently large $n \in \mathbb{N}$, a ranking exists for which the expected number of Pareto-optimal solutions is lower bounded by κn^2k^2 for some constant κ depending only on c. That is, we create a ranking at random (but independently of the profits) and show that the expected number of Pareto-optimal solutions (where the expectation is taken over both the random ranking and the random profits) satisfies the desired lower bound. This implies that, for each sufficiently large $n \in \mathbb{N}$, there must exist a deterministic ranking on $\{0, \dots, k\}^n$ for which the expected number of Pareto-optimal solutions (where the expectation is now taken only over the random profits) is at least κn^2k^2. The following theorem is a more precise version of Theorem 1.2.2.

Theorem 3.2.5. *Let $(5(c+1)+1)\log n \leq k \leq n^c$ for some $c \geq 2$ and assume that n is a multiple of $c+2$. There exists a ranking on $\{0, \dots, k\}^n$ and a constant κ depending only on c such that the expected number of Pareto-optimal solutions is lower bounded by κn^2k^2 if each profit p_i is chosen independently uniformly at random from the interval $[-1, 1]$.*

In order to describe the way of how the ranking is created, we introduce the notion of *virtual items*. Let $[n]$ be the set of *original items* and assume that we have k instances of each of these n items. A virtual item is a vector $x \in \mathcal{D}^n$. Intuitively, adding the virtual item x to the knapsack corresponds to inserting x_i instances of the i-th original item into the knapsack for every $i \in [n]$.

Assume that a sequence $x^{(1)}, \dots, x^{(l)}$ of virtual items is given. Based on this sequence, we create a ranking on the set of solutions $\mathcal{D}^n$ similar to the ranking used in Theorem 3.2.4 but for the binary case in which every virtual item can be "contained" at most once in every solution. That is, we create a ranking such that solutions that "contain" the i-th virtual item cannot dominate solutions that "consist" only of a subset of the first $i-1$ virtual items. Let $\mathcal{S}_0 = \{(0, \dots, 0)\}$ and assume that the solution $(0, \dots, 0)$ has the highest rank, i.e., that it cannot be dominated by any other solution. Let $\mathcal{S}_i$ denote the set of solutions that can be obtained by adding a subset of the first i virtual items, that is,

$$\mathcal{S}_i = \mathcal{S}_{i-1} \cup \left\{ x + x^{(i)} \mid x \in \mathcal{S}_{i-1} \right\} .$$

Let $\mathcal{S}_i^* = \mathcal{S}_i \setminus \mathcal{S}_{i-1}$. In the ranking we define, each solution from $\mathcal{S}_i^*$ is ranked lower than every solution from $\mathcal{S}_{i-1}$. It remains to define the ranking among two solutions $x, y \in \mathcal{S}_i^*$. The solutions x and y can uniquely be written as $x = x' + x^{(i)}$ and $y = y' + x^{(i)}$ for some $x', y' \in \mathcal{S}_{i-1}$. Based on this observation, we define the ranking between x and y to be the same as the one between x' and y'. Furthermore,

we define the ranking in such a way that all solutions in $\mathcal{S} \setminus \mathcal{S}_l$ are ranked lower than all solutions in $\mathcal{S}_l$. Hence, we do not need to consider the solutions in $\mathcal{S} \setminus \mathcal{S}_l$ anymore. For a given sequence of virtual items, this yields a fixed ranking among the solutions in $\mathcal{S}_l$.

Example. In order to illustrate the way of how the ranking is created, let us give an example. Let us assume that $n = 3$ and that three virtual items are chosen, namely $x^{(1)} = (1,0,1)$, $x^{(2)} = (1,1,0)$, and $x^{(3)} = (0,0,1)$. Then $\mathcal{S}_0 = \{(0,0,0)\}$, $\mathcal{S}_1 = \{(0,0,0),(1,0,1)\}$, $\mathcal{S}_2 = \{(0,0,0),(1,0,1),(1,1,0),(2,1,1)\}$, and $\mathcal{S}_3 = \{(0,0,0),(1,0,1),(1,1,0),(2,1,1),(0,0,1),(1,0,2),(1,1,1),(2,1,2)\}$. The solutions in $\mathcal{S}_3$ are listed according to the ranking, that is, $(0,0,0)$ is the highest ranked solution and $(2,1,2)$ is the lowest ranked solution.

Now we describe how the sequence of virtual items is chosen. We set $l = nk/(2e(c+2))$. Since we assumed that n is a multiple of $c+2$, we can partition the set of original items into $c+2$ groups with $n' = n/(c+2)$ items each. Let V denote the set of virtual items that contain one instance from each group, that is,

$$V = \left\{ x \in \{0,1\}^n \;\middle|\; \forall j \in \{0,\dots,c+1\} \colon \sum_{i=1}^{n'} x_{j\cdot n'+i} = 1 \right\} \quad .$$

Every virtual item $x^{(i)}$ is drawn independently and uniformly from the set V. It can happen that there exists an original item that occurs in more than k virtual items. In this case, the sequence of virtual items is not valid because we have only k instances of each item. Then the ranking is replaced by an arbitrary ranking on $\mathcal{D}^n$. The following lemma shows that this failure event is unlikely to occur.

Lemma 3.2.6. *The probability that the sequence of virtual items is not valid because more than k instances of one original item are contained in the virtual items is bounded from above by $1/(nk)^5$.*

Proof. For $i \in [n]$, let L_i denote the number of instances of item i that are contained in the virtual items. We can bound the probability that L_i exceeds k by

$$\begin{aligned}\mathbf{Pr}[L_i \geq k+1] &\leq \binom{l}{k+1} \cdot \left(\frac{1}{n'}\right)^{k+1} \\ &\leq \left(\frac{e\cdot l}{k+1}\right)^{k+1} \cdot \left(\frac{c+2}{n}\right)^{k+1} < \left(\frac{1}{2}\right)^k \quad .\end{aligned}$$

A union bound yields

$$\begin{aligned}\mathbf{Pr}[\exists i \in [n] : L_i \geq k+1] &\leq n \cdot \left(\frac{1}{2}\right)^k \leq n \cdot \left(\frac{1}{2}\right)^{(5(c+1)+1)\log n} \\ &= \frac{1}{n^{5(c+1)}} \leq \frac{1}{(nk)^5} \quad .\end{aligned}$$

□

We prove Theorem 3.2.5 in two steps. First we prove the following lemma about how the profits of the virtual items in V are distributed, where the profit of a virtual item $x \in \{0,1\}^n$ is defined as $p \cdot x$. Observe that scaling all profits by the same factor does not affect the number of Pareto-optimal solutions. Hence, we can assume that the profits are chosen uniformly at random from the interval $[-a, a]$ for an arbitrary $a > 0$.

Lemma 3.2.7. *If the profits $p_1, \dots, p_n$ of the original items are chosen independently uniformly at random from the interval $[-n^{c+1}, n^{c+1}]$, then there exist constants $\rho > 0$ and $p > 0$ depending only on c such that with probability at least p, for each $j \in \{0, \dots, n^{c+1} - 1\}$, the set V contains at least n/ρ virtual items whose profits lie in the interval $(j, j+1)$.*

Furthermore, we adapt the lower bound of $\Omega(n^2)$ in [BV04] for the binary case from uniformly random profits to profits that are chosen only "nearly" uniformly at random. To make this more precise, consider a knapsack instance with n items in which the i-th item has weight 2^i and the profits of the items are chosen independently according to a probability distribution $F\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$. Assume that F consists of two components, that is, there exists a constant $\delta > 0$ such that $F = \delta \cdot U + (1 - \delta) \cdot G$ for two probability distributions U and G. Furthermore, assume that U has the property that for each $j \in \{0, \dots, T-1\}$ it holds $\mathbf{Pr}[X \in (j, j+1)] = 1/T$ for a random variable X distributed according to U and some $T \geq n$.

Lemma 3.2.8. *The expected number of Pareto-optimal solutions in the aforementioned scenario is lower bounded by $\delta^2 n^2/128$.*

Together Lemmas 3.2.6, 3.2.7, and 3.2.8 and the upper bound on the expected number of Pareto-optimal solutions presented in Theorem 3.2.1 imply Theorem 3.2.5.

Proof of Theorem 3.2.5. Assume that the ranking on the set of solutions is determined as described above, that is, the ranking is induced by l randomly chosen virtual items from V. Let $\mathcal{F}_1$ denote the event that there exists some $j \in \{0, \dots, n^{c+1} - 1\}$ for which less than n/ρ elements in V have a profit in $(j, j+1)$. Due to Lemma 3.2.7, the probability of the event $\mathcal{F}_1$ is bounded from above by $1 - p$. Intuitively, the failure event $\mathcal{F}_1$ occurs if the profits $p_1, \dots, p_n$ are chosen such that the profit distribution of the virtual items is not uniform enough. We first analyze the number q' of Pareto-optimal solutions in a different random experiment. In this random experiment, we do not care if the sequence of virtual items is valid, that is, we assume $\mathcal{D} = \{0, \dots, l\}$, for which the sequence is always valid.

Let X denote the random variable that describes the profit of a randomly chosen virtual item from V. Under the assumption that $\mathcal{F}_1$ does not occur, we can write the distribution of X in a form such that Lemma 3.2.8 is applicable. Since for every $j \in \{0, \dots, n^{c+1} - 1\}$ the interval $(j, j+1)$ contains at least n/ρ virtual items and since there are n'^{c+2} virtual items in V, the probability that one of the first n/ρ items in one of these intervals is chosen is $\delta = n^{c+2}/(\rho n'^{c+2}) = (c+2)^{c+2}/\rho$. Hence, with probability δ, X is chosen according to a distribution U that is uniform on the intervals $(j, j+1)$ with $j \in \{0, \dots, n^{c+1} - 1\}$. The number of virtual items in the sequence is $l = nk/(2e(c+2))$. These virtual items are chosen independently and hence we can apply Lemma 3.2.8, yielding that the expected number of Pareto-optimal solutions is lower bounded by $\delta^2 l^2/128 = \kappa' n^2 k^2$ for $\kappa' = \delta^2/(512e^2(c+2)^2)$. Altogether, we have shown $\mathbf{E}[q' \mid \neg\mathcal{F}_1] \geq \kappa' n^2 k^2$.

Now we take into account that the sequence of virtual items might not be a valid sequence for $\mathcal{D} = \{0, \dots, k\}$. Let $\mathcal{F}_2$ denote the event that the sequence

of virtual items is not allowed because it contains more than k instances of one item. Due to Lemma 3.2.6, we know that $\mathbf{Pr}[\mathcal{F}_2] \leq 1/(nk)^5$. Remember that if this failure event occurs, the ranking is set to an arbitrary ranking on $\mathcal{D}^n$. Let q denote the number of Pareto-optimal solutions. By definition of q' and the failure event $\mathcal{F}_2$, we know that $\mathbf{E}[q \mid \neg\mathcal{F}_2] = \mathbf{E}[q' \mid \neg\mathcal{F}_2]$. Furthermore, since $\mathcal{F}_2$ does not affect the choice of the profits, we can use Theorem 3.2.1 to bound $\mathbf{E}[q' \mid \mathcal{F}_2]$, but we have to take into account that in the modified random experiment for which q' is defined we have $\mathcal{D} = \{0, \ldots, l\}$. Hence, we obtain $\mathbf{E}[q' \mid \mathcal{F}_2] \leq \kappa'' n^4 k^2 \log(nk)$ for a sufficiently large constant κ''.

Putting these results together yields

$$\begin{aligned}
\mathbf{E}[q] &\geq \mathbf{Pr}[\neg\mathcal{F}_2] \cdot \mathbf{E}[q \mid \neg\mathcal{F}_2] \\
&= \mathbf{Pr}[\neg\mathcal{F}_2] \cdot \mathbf{E}\left[q' \mid \neg\mathcal{F}_2\right] \\
&= \mathbf{E}\left[q'\right] - \mathbf{Pr}[\mathcal{F}_2] \cdot \mathbf{E}\left[q' \mid \mathcal{F}_2\right] \\
&\geq \mathbf{Pr}[\neg\mathcal{F}_1] \cdot \mathbf{E}\left[q' \mid \neg\mathcal{F}_1\right] - \mathbf{Pr}[\mathcal{F}_2] \cdot \mathbf{E}\left[q' \mid \mathcal{F}_2\right] \\
&\geq p \cdot \kappa' n^2 k^2 - \frac{\kappa'' n^4 k^2 \log(nk)}{(nk)^5} \\
&\geq \kappa n^2 k^2
\end{aligned}$$

for a sufficiently large constant κ. □

Proof of Lemma 3.2.7

In order to prove Lemma 3.2.7, we analyze an auxiliary random experiment first. A well studied random process is the experiment of placing n balls uniformly and independently at random into m bins. In this random allocation process, the expected load of each bin is n/m and one can use Chernoff bounds to show that in the case $n \geq m$ it is unlikely that there exists a bin whose load deviates by more than a logarithmic factor from its expectation. In this section, we consider a random experiment in which the locations of the balls are chosen as linear combinations of independent random variables. Since the same random variables appear in linear combinations for different balls, the locations of the balls are dependent in a special way.

Let $c \in \mathbb{N}$ with $c \geq 2$ be an arbitrary constant and assume that we are given n independent random variables that are chosen uniformly at random from the interval $[-n^{c+1}, n^{c+1}]$. We assume that n is a multiple of $c+2$, and we partition the set of random variables into $c+2$ sets with $n' = n/(c+2)$ random variables each. For $i \in \{1, \ldots, c+2\}$ and $j \in \{1, \ldots, n'\}$, let p_j^i denote the j-th random variable in the i-th group.

For every $l \in [c+2]$, we consider a random experiment in which the set of balls is $[n']^l$ and the bins are the intervals $(-ln^{c+1}, -ln^{c+1}+1), \ldots, (ln^{c+1}-1, ln^{c+1})$. In the following, bin j denotes the interval $(j, j+1)$. Hence, the number of balls is $(n')^l$ and the number of bins is $2ln^{c+1}$. Instead of placing these balls independently in the bins, the location of a ball $a \in [n']^l$ is chosen to be $p_{a_1}^1 + \cdots + p_{a_l}^l$, that is, it is placed in bin $\lfloor p_{a_1}^1 + \cdots + p_{a_l}^l \rfloor$. We show that despite these dependencies, the allocation process generates a more or less balanced allocation with constant probability. We use the following weighted Chernoff bound whose proof can be found in Appendix B.2.

Lemma 3.2.9. *Let $X_1, \ldots, X_n$ be independent discrete random variables with values in $[0, z]$ for some $z > 0$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. Then for every $x > 0$,*

$$\mathbf{Pr}[X \geq x] < \left(\frac{e \cdot \mu}{x}\right)^{x/z} \quad \text{and} \quad \mathbf{Pr}[X \leq x] < \left(\frac{e^{1-\mu/x} \cdot \mu}{x}\right)^{x/z} .$$

Lemma 3.2.10. *For $l = c + 1$, the average number of balls per bin is a constant depending on c, and with probability $1 - o(1)$, the maximal number of balls in any bin is bounded from above by $\ln n$.*

Proof. Instead of assigning all balls to the bins at once, we consider $c + 1$ rounds separately. In round $i \in \{1, \ldots, c+1\}$, we consider the set of balls $[n']^i$. Each ball $a \in [n']^i$ is placed at position $p^1_{a_1} + \cdots + p^i_{a_i}$, that is, in the interval $(j, j+1)$ with $j = \lfloor p^1_{a_1} + \cdots + p^i_{a_i} \rfloor$. Intuitively, at the beginning of round i, we replace each ball from round $i-1$ by n' identical copies. Then these n' copies are moved, where the location of the j-th copy is obtained by adding p^i_j to the current location.

During the first round, n' balls are placed independently uniformly at random in $2n^{c+1}$ possible bins. We define $\mathcal{F}_1$ to be the event that there exists a bin that contains more than one ball after the first round. Since $c \geq 2$, the probability of $\mathcal{F}_1$ can be bounded by $(n')^2/(2n^{c+1}) = o(1)$.

Now we consider round i with $i \in \{2, \ldots, c\}$. Let $\mathcal{F}_i$ denote the event that after round i there exists a bin that contains more than $(2c+4)^{i-1}$ balls, and let X^i_j denote the number of balls in bin j after the i-th round. Assume that the random variables in the first $i-1$ groups are already fixed in such a way that the event $\mathcal{F}_{i-1}$ does not occur. Under this assumption, also the variables X^{i-1}_j are fixed and have values of at most $(2c+4)^{i-2}$. Consider a bin j after round $i-1$ and assume that to all elements in that bin the d-th element of the i-th group is added. The profits of the items obtained this way are in the interval $(j + \lfloor p^i_d \rfloor, j + \lfloor p^i_d \rfloor + 2)$, that is, they lie either in bin $j + \lfloor p^i_d \rfloor$ or $j + \lfloor p^i_d \rfloor + 1$. Hence, we can bound X^i_j by

$$X^i_j \leq \sum_{d=1}^{n'} Y^{i-1}_{j,p^i_d} \quad \text{with} \quad Y^{i-1}_{j,p^i_d} = X^{i-1}_{j-\lfloor p^i_d \rfloor} + X^{i-1}_{j-\lfloor p^i_d \rfloor - 1} .$$

Hence, when the random variables in the first $i-1$ groups are fixed such that $\mathcal{F}_{i-1}$ does not occur, then X^i_j is bounded by the sum of independent discrete random variables Y^{i-1}_{j,p^i_d} that take only values from the set $\{0, \ldots, 2(2c+4)^{i-2}\}$. The expected value of X^i_j is bounded from above by $(n')^i/(2n^{c+1}) < 1/n$. Altogether, this implies that we can use Lemma 3.2.9 to bound the probability that X^i_j exceeds its expectation. We obtain

$$\mathbf{Pr}\left[X^i_j \geq (2c+4)^{i-1} \,\middle|\, \neg\mathcal{F}_{i-1}\right] \leq \left(\frac{e}{(2c+4)^{i-1} n}\right)^{c+2} < n^{-(c+2)} .$$

Applying a union bound over all $2in^{c+1}$ bins j yields

$$\mathbf{Pr}[\mathcal{F}_i \mid \neg\mathcal{F}_{i-1}] \leq \mathbf{Pr}\left[\exists j : X^i_j \geq (2c+4)^{i-1} \,\middle|\, \neg\mathcal{F}_{i-1}\right] \leq (2in^{c+1}) \cdot n^{-(c+2)} = o(1) .$$

Now consider round $c+1$. The expected value of X_j^{c+1} is bounded from above by $(n')^{c+1}/(2n^{c+1}) < 1$ and similar arguments as for the previous rounds show that X_j^{c+1} can be bounded by the sum of independent random variables with values from the set $\{0, \ldots, 2(2c+4)^{c-1}\}$ when the random variables in the first c groups are fixed such that $\mathcal{F}_c$ does not occur. Hence, we can again apply Lemma 3.2.9 to obtain

$$\mathbf{Pr}\left[X_j^{c+1} \geq \ln n \,\middle|\, \neg\mathcal{F}_c \right] \leq \left(\frac{e}{\ln n}\right)^{\ln n/(2(2c+4)^{c-1})} \leq n^{-\ln\ln n/(2(2c+4)^{c-1})+1} \ .$$

Let $\mathcal{F}_{c+1}$ denote the event that after round $c+1$ there exists a bin that contains more than $\ln n$ balls. Applying a union bound over all $2(c+1)n^{c+1}$ bins yields

$$\mathbf{Pr}[\mathcal{F}_{c+1} \mid \neg\mathcal{F}_c] \leq (2(c+1)n^{c+1}) \cdot n^{-\ln\ln n/(2(2c+4)^{c-1})+1} = o(1) \ .$$

Now we can bound the probability that $\mathcal{F}_{c+1}$ occurs as

$$\begin{aligned}
\mathbf{Pr}[\mathcal{F}_{c+1}] &\leq \mathbf{Pr}[\mathcal{F}_c] + \mathbf{Pr}[\mathcal{F}_{c+1} \mid \neg\mathcal{F}_c] \\
&\leq \mathbf{Pr}[\mathcal{F}_{c-1}] + \mathbf{Pr}[\mathcal{F}_c \mid \neg\mathcal{F}_{c-1}] + o(1) \\
&\leq \mathbf{Pr}[\mathcal{F}_{c-2}] + \mathbf{Pr}[\mathcal{F}_{c-1} \mid \neg\mathcal{F}_{c-2}] + o(1) \\
&\leq \ldots = o(1) \ .
\end{aligned}$$

□

Based on Lemma 3.2.10, we prove the following lemma about the allocation after round $c+2$, which directly implies Lemma 3.2.7.

Lemma 3.2.11. *For every constant $c \geq 2$, there exist constants $\rho > 0$ and $p > 0$ such that with probability at least p the above described process yields an allocation of the $(n')^{c+2}$ balls to the $2(c+2)n^{c+1}$ bins in which every bin $j \in \{0, \ldots, n^{c+1}-1\}$ contains at least n/ρ balls.*

Proof. In order to analyze the last round, we need besides $\neg\mathcal{F}_{c+1}$ one additional property that has to be satisfied after round $c+1$. Let Y denote the number of balls after round $c+1$ that are assigned to bins j with $j \in \{0, \ldots, n^{c+1}-1\}$. The probability that a fixed ball $a \in [n']^{c+1}$ is placed in one of these bins is at least $1/(2(c+1))$. Hence, the expected value of Y is at least $(n')^{c+1}/(2(c+1))$. Let $\overline{Y}$ denote the number of balls after round $c+1$ that are not assigned to bins in $\{0, \ldots, n^{c+1}-1\}$. The expected value of $\overline{Y}$ is at most $(n')^{c+1}(2c+1)/(2c+2)$. Applying Markov's inequality yields

$$\mathbf{Pr}\left[Y \leq \frac{(n')^{c+1}}{4c+4} \right] \leq \mathbf{Pr}\left[\overline{Y} \geq \mathbf{E}\left[\overline{Y}\right] \frac{4c+3}{4c+2} \right] \leq \frac{4c+2}{4c+3} \ .$$

Let $\mathcal{G}$ denote the failure event that Y is less than $(n')^{c+1}/(4c+4)$. We have seen that $\neg\mathcal{G}$ occurs with constant probability.

Now we analyze round $c+2$ and assume that the random variables in the first $c+1$ groups are fixed in such a way that $\neg\mathcal{F}_{c+1} \cap \neg\mathcal{G}$ occurs. Consider a bin $j \in \{0, \ldots, n^{c+1}-1\}$. Since under the assumption $\neg\mathcal{G}$, there are together at least $(n')^{c+1}/(4c+4)$ balls in the bins in $\{0, \ldots, n^{c+1}-1\}$, the expected value of X_j^{c+2} is at least $(n')^{c+2}/((4c+4)2n^{c+1}) \geq n/(c+2)^{c+5}$. Furthermore, under the

assumption $\neg\mathcal{F}_{c+1}$, Y_{j,p_d}^{c+1} takes only values in the interval $\{0,\dots,2\ln n\}$. We apply Lemma 3.2.9 to bound the probability that X_j^{c+2} deviates from its mean:

$$\mathbf{Pr}\left[X_j^{c+2} \le \frac{n}{2(c+2)^{c+5}} \;\middle|\; \neg\mathcal{F}_{c+1} \cap \neg\mathcal{G}\right] \le \left(\frac{2}{e}\right)^{n/(4(c+2)^{c+5}\ln n)} .$$

Let $\mathcal{F}$ denote the event that there exists a bin $j \in \{0,\dots,n^{c+1}-1\}$ whose load is smaller than $n/(2(c+2)^{c+5})$. We can bound the probability of $\mathcal{F}$ by

$$\mathbf{Pr}[\mathcal{F} \mid \neg\mathcal{F}_{c+1} \cap \neg\mathcal{G}] \le n^{c+1} \cdot \left(\frac{2}{e}\right)^{n/(4(c+2)^{c+5}\ln n)} = o(1) .$$

Altogether this implies

$$\begin{aligned}\mathbf{Pr}[\mathcal{F}] &\le \mathbf{Pr}[\mathcal{F}_{c+1} \cup \mathcal{G}] + \mathbf{Pr}[\mathcal{F} \mid \neg\mathcal{F}_{c+1} \cap \neg\mathcal{G}] \\ &\le \mathbf{Pr}[\mathcal{F}_c] + \mathbf{Pr}[\mathcal{F}_{c+1} \mid \neg\mathcal{F}_c] + \frac{4c+2}{4c+3} + o(1) \\ &\le \mathbf{Pr}[\mathcal{F}_{c-1}] + \mathbf{Pr}[\mathcal{F}_c \mid \neg\mathcal{F}_{c-1}] + \frac{4c+2}{4c+3} + o(1) \\ &\le \dots \le \frac{4c+2}{4c+3} + o(1) ,\end{aligned}$$

which yields the lemma. □

Proof of Lemma 3.2.8

Beier and Vöcking [BV04] prove a lower bound of $\Omega(n^2)$ on the expected number of Pareto-optimal knapsack fillings for exponentially growing weights and profits that are chosen independently and uniformly at random from the interval $[0,1]$. In this section, we adapt their proof for a random experiment in which the profits are chosen only "nearly" uniformly at random. Assume that we are given n items and that the i-th item has weight $w_i = 2^i$. Furthermore, let $T \in \mathbb{N}$ be given and assume that $T \ge n$. In order to determine the profit p_i of the i-th item, first one of the intervals $(0,1),(1,2),\dots,(T-1,T)$ is chosen uniformly at random. Then an adversary is allowed to choose the exact profit within the randomly chosen interval. We call an item whose profit is chosen this way a *nearly uniform item*. We prove that also in this scenario the expected number of Pareto-optimal solutions is lower bounded by $\Omega(n^2)$.

Lemma 3.2.12. *For instances consisting of n nearly uniform items, the expected number of Pareto-optimal solutions is bounded from below by $n^2/16$.*

Proof. The proof follows the line of the proof of Theorem 3.2.4 for the binary case. Let $\mathcal{P}_j$ denote the set of Pareto-optimal solutions over the first j items, and let P_j denote the total profit of the first j items. For $j \in [n]$ and $\alpha \ge 0$, let X_α^j denote the number of Pareto-optimal solutions from $\mathcal{P}_j$ with profits in the interval $[P_j - \alpha, P_j)$. Observe that $p_j > \alpha$ implies $X_\alpha^j = X_\alpha^{j-1}$ and $p_j < \alpha$ implies $X_\alpha^j = X_{p_j}^{j-1} + X_{\alpha-p_j}^{j-1} + 1$. For integral $\alpha \in [T]$, the adversary cannot influence the event $p_j < \alpha$, as the interval from which he is allowed to pick a values for p_j lies

either completely left or completely right of α. Hence, for $\alpha \in [T]$ we can bound the expected value of X_α^j recursively as follows:

$$\begin{aligned}\mathbf{E}\left[X_\alpha^j\right] \geq\ & \mathbf{Pr}[p_j > \alpha] \cdot \mathbf{E}\left[X_\alpha^{j-1} \,\middle|\, p_j > \alpha\right] \\ & + \mathbf{Pr}[p_j < \alpha] \cdot \left(\mathbf{E}\left[X_{p_j}^{j-1} \,\middle|\, p_j < \alpha\right] + \mathbf{E}\left[X_{\alpha-p_j}^{j-1} \,\middle|\, p_j < \alpha\right] + 1\right) \ .\end{aligned}$$

As X_β^{j-1} is independent of p_j and X_β^{j-1} is monotone in β, we have

$$\begin{aligned}\mathbf{E}\left[X_\alpha^j\right] \geq\ & \mathbf{Pr}[p_j > \alpha] \cdot \mathbf{E}\left[X_\alpha^{j-1}\right] \\ & + \mathbf{Pr}[p_j < \alpha] \cdot \left(\mathbf{E}\left[X_{\lfloor p_j \rfloor}^{j-1} \,\middle|\, p_j < \alpha\right] + \mathbf{E}\left[X_{\lfloor \alpha-p_j \rfloor}^{j-1} \,\middle|\, p_j < \alpha\right] + 1\right) \ .\end{aligned} \tag{3.2.5}$$

In the following, we prove by induction on j that for every $\alpha \in [T]$,

$$\mathbf{E}\left[X_\alpha^j\right] \geq \frac{\alpha \cdot j}{2T} \ .$$

For $j = 1$ and $\alpha \in [T]$, we obtain

$$\mathbf{E}\left[X_\alpha^1\right] = \mathbf{Pr}[p_1 < \alpha] = \frac{\alpha}{T} \geq \frac{\alpha}{2T} \ .$$

Using the induction hypothesis and (3.2.5), we obtain, for $j \in [n] \setminus \{1\}$ and $\alpha \in [T]$, that $\mathbf{E}\left[X_\alpha^j\right]$ is lower bounded by

$$\begin{aligned}& \mathbf{Pr}[p_j > \alpha] \cdot \mathbf{E}\left[X_\alpha^{j-1}\right] \\ & + \mathbf{Pr}[p_j < \alpha] \left(\mathbf{E}\left[X_{\lfloor p_j \rfloor}^{j-1} \,\middle|\, p_j < \alpha\right] + \mathbf{E}\left[X_{\lfloor \alpha-p_j \rfloor}^{j-1} \,\middle|\, p_j < \alpha\right] + 1\right) \\ & \geq \frac{T-\alpha}{T} \cdot \frac{\alpha(j-1)}{2T} + \sum_{i=0}^{\alpha-1} \mathbf{Pr}[p_j \in (i, i+1)] \left(\mathbf{E}\left[X_i^{j-1}\right] + \mathbf{E}\left[X_{\alpha-i-1}^{j-1}\right] + 1\right) \\ & \geq \frac{T-\alpha}{T} \cdot \frac{\alpha(j-1)}{2T} + \sum_{i=1}^{\alpha} \frac{1}{T} \left(\frac{i(j-1)}{2T} + \frac{(\alpha-i-1)(j-1)}{2T} + 1\right) \\ & = \frac{T-\alpha}{T} \cdot \frac{\alpha(j-1)}{2T} + \frac{\alpha}{T} \left(\frac{(\alpha-1)(j-1)}{2T} + 1\right) \\ & = \frac{\alpha(j-1)}{2T} + \frac{\alpha}{T} \left(1 - \frac{j-1}{2T}\right) \\ & \geq \frac{\alpha(j-1)}{2T} + \frac{\alpha}{T} \cdot \frac{1}{2} \\ & = \frac{\alpha \cdot j}{2T} \ .\end{aligned}$$

This yields the following lower bound on the expected number of Pareto-optimal solutions:

$$\mathbf{E}[q] \geq \sum_{j=1}^{n} \mathbf{E}\left[X_{\lfloor p_j \rfloor}^{j-1}\right] \geq \sum_{j=1}^{n} \frac{\mathbf{E}[\lfloor p_j \rfloor] \cdot (j-1)}{2T} \geq \frac{T-1}{4T} \cdot \sum_{j=1}^{n} (j-1) \geq \frac{n^2}{16} \ . \qquad \square$$

We further generalize the scenario that we considered above and analyze the expected number of Pareto-optimal solutions for instances that do not only consist of nearly uniform items but also of some adversarial items. To be more precise, we assume that the profit of each item is chosen as follows: First a coin is tossed which comes up head with probability $\delta > 0$. If the coin comes up head, then the profit of the item is chosen as for nearly uniform items, that is, an interval is chosen uniformly at random and after that an adversary may choose an arbitrary profit in that interval. If the coin comes up tail, then an arbitrary non-integer profit can be chosen by an oblivious adversary who does not know the outcomes of the previous profits.

Proof of Lemma 3.2.8. First of all, we show that the presence of adversarial items does not affect the lower bound for the expected number of Pareto-optimal solutions. That is, we show that if there are $\hat{n}$ nearly uniform items and an arbitrary number of adversarial items, one can still apply Lemma 3.2.12 to obtain a lower bound of $\hat{n}^2/16$ on the expected number of Pareto optimal solutions. For this, consider the situation that the first j items are nearly uniform items and that item $j+1$ is an adversarial item. Due to Lemma 3.2.12, we obtain that the expected value of X_α^j is bounded from below by $j \cdot \alpha/(2T)$ for every $\alpha \in [T]$. We show that the expected value of X_α^{j+1} is lower bounded by the same value. For this, consider the two alternatives that the adversary has. He can either choose $p_{j+1} > \alpha$ or $p_{j+1} < \alpha$. In the former case, we have $X_\alpha^j = X_\alpha^{j+1}$. In the latter case, we have

$$\mathbf{E}\left[X_\alpha^{j+1}\right] \geq \mathbf{E}\left[X^j_{\lfloor p_{j+1} \rfloor}\right] + \mathbf{E}\left[X^j_{\lfloor \alpha - p_j \rfloor}\right] + 1 \geq \frac{(\alpha-1)j}{2T} + 1 \geq \frac{\alpha \cdot j}{2T} \ .$$

Hence, the adversarial profit of item $j+1$ does not affect the lower bound for the expected number of Pareto-optimal solutions. One can apply this argument inductively to show the desired lower bound of $\hat{n}^2/16$.

In expectation the number $\hat{n}$ of nearly uniform items is δn and applying a Chernoff bound yields that with high probability $\hat{n} \geq \delta n/2$. For sufficiently large n, we can bound the probability that $\hat{n} < \delta n/2$ from above by $1/2$. Hence, with probability $1/2$ the expected number of Pareto-optimal solutions is at least $(\delta n/2)^2/16$, and hence, the expected number of Pareto-optimal solutions is bounded from below by $(\delta n)^2/128$. □

3.3 Smoothed Complexity of Integer Programming

We start this section by summarizing some definitions and results about the smoothed complexity of binary and integer optimization problems due to Beier and Vöcking [BV06] and Röglin and Vöcking [RV07]. After that, we show that our upper bound on the expected number of Pareto-optimal solutions presented in Theorem 3.2.1 can be used to simplify and to improve the analysis of the smoothed complexity of integer optimization problems.

Beier and Vöcking [BV06] initiated the study of the smoothed complexity of binary optimization problems. They consider binary optimization problems in which an objective function $c\colon \mathcal{S} \to \mathbb{R}$ is to be minimized over a set of feasible solutions that is given as $\mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_m$, where $\mathcal{S} \subseteq \{0,1\}^n$ denotes a fixed ground set and $\mathcal{B}_i$ denotes a halfspace induced by a linear constraint of the form $w_{i,1}x_1 +$

$\cdots + w_{i,n}x_n \leq t_i$. Similar to the semi-random input model in which we analyzed the expected number of Pareto-optimal solutions, it is assumed that the coefficients $w_{i,j}$ are independent continuous random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. To avoid technical details that are not of interest for the results in this thesis, we make the same simplifying assumptions as in Section 1.2, that is, we assume that an adversary can specify the coefficients $w_{i,j}$ in the interval $[-1, 1]$ and that they are perturbed by adding Gaussian or uniform random variables with standard deviation σ and constant expected absolute value. Also the case that the objective function c is linear and its coefficients are random variables with the aforementioned properties is considered in [BV06]. Röglin and Vöcking [RV07] consider the same scenario for integer optimization problems, in which $\mathcal{S}$ is a subset of $\mathcal{D}^n$ for a finite set of integers $\mathcal{D}$.

In the following theorem, it is assumed that an integer optimization problem is given in which at least one component is linear and perturbed, that is, either the objective function is linear and its coefficients are perturbed or at least one constraint is linear with perturbed coefficients or both. The theorem is proven for the binary case in [BV06] and for the integer case in [RV07].

Theorem 3.3.1. *([BV06, RV07]) An integer optimization problem has polynomial smoothed complexity if and only if it can be solved in pseudopolynomial time with respect to the perturbed coefficients in the worst case.*

Before we define the term *polynomial smoothed complexity* formally, let us present two examples that illustrate Theorem 3.3.1. The constrained spanning tree problem, in which edges have lengths and weights and one has to find the shortest spanning tree whose weight lies below a given threshold, is NP-hard but it can be solved in pseudopolynomial time with respect to lengths and weights [RG96]. Given a graph $G = (V, E)$, we introduce a binary decision variable $x_e \in \{0, 1\}$ for each edge $e \in E$ and we let the ground set $\mathcal{S}$ be the set of all vectors from $\{0, 1\}^{|E|}$ that encode spanning trees. Furthermore, we have one linear constraint for the weights and $\mathcal{B}_1$ denotes all subsets of edges whose total weight is below the given threshold. Since there is an algorithm that is pseudopolynomial in the weights and lengths, Theorem 3.3.1 implies that the constrained spanning tree problem has polynomial smoothed complexity if both the weights and the lengths are perturbed. On the other hand, let us consider the traveling salesperson problem, in which we are interested in finding the cheapest Hamiltonian cycle in a given graph. Analogously, we can introduce a decision variable for each edge and let $\mathcal{S}$ denote the set of Hamiltonian cycles. It is well known that the TSP is strongly NP-hard and hence, there is no pseudopolynomial algorithm, unless P = NP. Hence, Theorem 3.3.1 implies that under the assumption P $\neq$ NP, the TSP does not have polynomial smoothed complexity if the costs of the edges are randomly perturbed, which coincides with the observation that the TSP is hard to solve even on instances that occur in practical applications.

An obvious way to define the term *polynomial smoothed complexity* would be to require that the expected running time of the considered algorithm must be polynomial in the input size and in the reciprocal of the standard deviation σ. Beier and Vöcking introduce a different definition because defining polynomial smoothed complexity based on the expected running time is not a sufficiently robust notion.

For example, an algorithm with expected polynomial running time on one machine model might have expected exponential running time on another machine model, even if in the latter model, the former can be simulated in polynomial time. In Beier and Vöcking's definition, a problem Π is said to have polynomial smoothed complexity if it admits a polynomial P and an algorithm $\mathcal{A}$ whose running time T satisfies for every instance I, for every $\varepsilon \in (0,1]$, and for every $\sigma \in (0,1]$,

$$\mathbf{Pr}\left[T(I_\sigma) \geq P\left(|I|, \sigma^{-1}, \varepsilon^{-1}\right)\right] \leq \varepsilon \ , \tag{3.3.1}$$

where $|I|$ denotes the length of instance I, and I_σ denotes the random instance obtained from I by adding to each coefficient in the linear components a Gaussian or uniform random variable with standard deviation σ and constant expected absolute value. That is, with probability at least $1-\varepsilon$ the running time of $\mathcal{A}$ is polynomially bounded in the input length $|I|$ and the reciprocals of the standard deviation σ and the failure probability ε. This definition of polynomial smoothed complexity follows more or less the way how polynomial complexity is defined in average-case complexity theory [Wan97], adding the requirement that the running time should be polynomially bounded not only in the input length but also in σ^{-1}. As described in Section 1.2, it is assumed that the algorithm can access the perturbed numbers by an oracle.

In the following, we summarize the main idea that is used in the proof of Theorem 3.3.1 to transform an algorithm with pseudopolynomial worst-case complexity into an algorithm with polynomial smoothed complexity.

3.3.1 Winner, Loser, and Feasibility Gap

A common algorithmic approach to find approximate solutions for NP-hard problems that admit a pseudopolynomial time algorithm is to round all numbers in the input and to apply the pseudopolynomial time algorithm to the rounded instance. If every rounded number can be represented by a logarithmic number of bits, then the pseudopolynomial time algorithm runs in polynomial time on the rounded instance. If all coefficients that need to be rounded occur in the objective function, then this approach often yields fully polynomial time approximation schemes, as it is the case for the knapsack problem and the constrained shortest path problem. The situation becomes more difficult if also coefficients in the constraints are rounded. Then the optimal solution with respect to the rounded coefficients might not even be feasible in the original unrounded instance. However, the crucial observation in the proof of Theorem 3.3.1 is that perturbed instances of integer optimization problems are typically very robust against rounding the coefficients. To be precise, the optimal solution of a perturbed instance stays optimal with high probability if each coefficient is rounded after a logarithmic number of bits. This is a remarkable observation because in the worst case even the slightest rounding can change the optimal solution. The proof of this observation is based on three structural properties of integer optimization problems, called *winner*, *loser*, and *feasibility gap*, which we introduce in the following.

Let us first consider integer optimization problems in which only the objective function is randomly perturbed. In this case, the set $\mathcal{S} \subseteq \mathcal{D}^n$ of feasible solutions is fixed and one is interested in minimizing a linear objective function $c\colon \mathcal{S} \to \mathbb{R}$

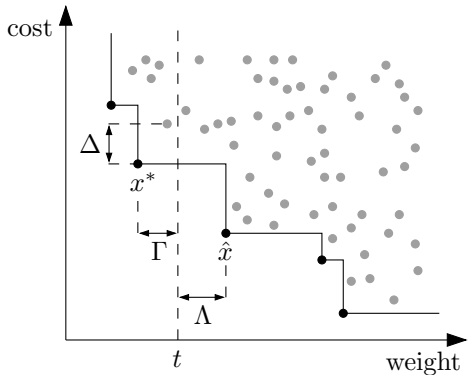


Figure 3.3.1: Illustration of the definitions of winner, loser, and feasibility gap.

of the form $c(x) = c_1x_1 + \cdots + c_nx_n$. The solution x^* that minimizes c over $\mathcal{S}$ is called the *winner*, and we denote by x^{**} the second best solution, that is,

$$x^{**} = \operatorname*{argmin}_{x \in \mathcal{S} \setminus \{x^*\}} c \cdot x \ .$$

The *winner gap* Δ is defined to be the difference between the best and the second best solution, that is,

$$\Delta = c \cdot x^{**} - c \cdot x^* \ .$$

Even though the winner gap can be arbitrarily small in the worst case, it is typically polynomially large if the coefficients in the objective function are perturbed. Following the notations in Section 3.1, we set $d = |\mathcal{D}|$.

Lemma 3.3.2. *([BV06, RV07]) Let the costs $c_1, \ldots, c_n$ be independent random variables whose densities are bounded by ϕ and let $\mathcal{S} \subseteq \mathcal{D}^n$ be chosen arbitrarily. For every $\varepsilon \geq 0$,*

$$\mathbf{Pr}[\Delta \leq \varepsilon] \leq \varepsilon \phi n d^2 \ .$$

Now let us consider the case that one linear constraint is perturbed. In this case, a function $c\colon \mathcal{S} \to \mathbb{R}$ is to be minimized over a set of feasible solutions that can be written as the intersection of a fixed ground set $\mathcal{S} \subseteq \mathcal{D}^n$ and a set $\mathcal{B}$ that contains all solutions that satisfy a constraint of the form $w_1x_1 + \cdots + w_nx_n \leq t$. Let x^* denote the winner, that is, the solution from $\mathcal{S} \cap \mathcal{B}$ that minimizes the function c. According to Beier and Vöcking [BV06], we define the *feasibility gap* Γ to be the slack of this solution from the threshold t, that is,

$$\Gamma = \begin{cases} t - w \cdot x^* & \text{if } \mathcal{S} \cap \mathcal{B} \neq \emptyset, \\ \bot & \text{otherwise.} \end{cases}$$

The set $\mathcal{L}$ of *losers* is defined to be the set of all solutions from $\mathcal{S}$ that have lower costs than x^* but are infeasible due to the linear constraint, that is,

$$\mathcal{L} = \{x \in \mathcal{S} \mid c(x) < c(x^*)\} \ .$$

The *minimal loser* $\hat{x}$ is defined to be the solution from $\mathcal{L}$ that has the smallest weight and the *loser gap* Λ denotes the distance of $\hat{x}$ to the threshold t, that is,

$$\Lambda = \begin{cases} w \cdot \hat{x} - t & \text{if } \mathcal{L} \neq \emptyset, \\ \bot & \text{otherwise.} \end{cases}$$

The definitions of winner, loser, and feasibility gap are illustrated in Figure 3.3.1. The following lemma shows that even though both the feasibility and the loser gap can be arbitrarily small in the worst case, they are typically polynomially large. We use the notations $d = |\mathcal{D}|$ and $d_{\max} = \max\{|a| \mid a \in \mathcal{D}\}$.

Lemma 3.3.3. *([BV06, RV07]) Let the weights $w_1, \ldots, w_n$ be independent random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. For arbitrary cost functions $c\colon \mathcal{S} \to \mathbb{R}$, arbitrary sets $\mathcal{S} \subseteq \mathcal{D}^n$ with $0^n \notin \mathcal{S}$, and every $\varepsilon \in [0, (32\mu n^5 d^7 d_{\max}\phi^2)^{-1}]$,*

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq 2(\varepsilon \cdot 32\mu n^5 d^7 d_{\max}\phi^2)^{1/3} \quad \textit{and} \quad \mathbf{Pr}[\Lambda \leq \varepsilon] \leq 2(\varepsilon \cdot 32\mu n^5 d^7 d_{\max}\phi^2)^{1/3}.$$

In the binary case (i.e., $\mathcal{D} = \{0,1\}$), it holds for every $\varepsilon \geq 0$,

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \varepsilon\phi n^2 \quad \textit{and} \quad \mathbf{Pr}[\Lambda \leq \varepsilon] \leq \varepsilon\phi n^2 \ .$$

How much Accuracy is Necessary?

The crucial observation is that whenever winner, loser, and feasibility gap are polynomially large, the optimal solution stays optimal even if all coefficients are rounded after a logarithmic number of bits. In order to see this, observe that rounding down a coefficient to a number that can be represented with b bits after the binary point, lowers its value by at most 2^{-b}. Hence, the total cost and the total weight of any solution $x \in \mathcal{S}$ is decreased due to the rounding by at most $nd_{\max}2^{-b}$. Let us first consider the case that the coefficients in the objective function, the costs, are rounded down. Let $x^* \in \mathcal{S}$ denote the optimal solution and let $y \in \mathcal{S}$ denote an arbitrary solution with $y \neq x^*$. If the winner gap is larger than $nd_{\max}2^{-b+1}$, then

$$c \cdot y - c \cdot x^* \geq \Delta > nd_{\max}2^{-b+1} \ .$$

If $\lfloor c \rfloor_b$ denotes the vector in which every cost is rounded down after the b-th bit after the binary point, then this implies

$$\begin{aligned} \lfloor c \rfloor_b \cdot y - \lfloor c \rfloor_b \cdot x^* &\geq \lfloor c \rfloor_b \cdot y - c \cdot x^* - nd_{\max}2^{-b} \\ &\geq c \cdot y - c \cdot x^* - 2nd_{\max}2^{-b} \geq \Delta - nd_{\max}2^{-b+1} > 0 \ . \end{aligned}$$

Hence, if the winner gap is larger than $nd_{\max}2^{-b+1}$, rounding every cost down after the b-th bit after the binary point does not change the optimal solution. Let us assume that, for example, we are given an integer optimization problem in which the costs are randomly perturbed and for which there exists an algorithm whose running time is pseudopolynomial with respect to the costs. For any desired failure probability $p \in (0,1]$, the above reasoning and Lemma 3.3.2 imply that, with probability at least $1 - p$, rounding down each cost after

$$b = \log\left(n^2 d^2 d_{\max}\phi p^{-1}\right) + 1$$

bits does not change the optimal solution. If we apply the pseudopolynomial time algorithm to this rounded instance, then its running time is polynomial in the input size, the maximal density ϕ, and the reciprocal of the failure probability p.

Now let us consider the case that the coefficients in the constraint, the weights, are rounded down. Let $x \in \mathcal{S}$ denote an arbitrary loser, that is, an arbitrary solution that has lower costs than the optimal solution x^*. If the loser gap is larger than $nd_{\max}2^{-b}$, then

$$w \cdot x - t \geq \Lambda > nd_{\max}2^{-b} \ .$$

If $\lfloor w \rfloor_b$ denotes the vector in which every weight is rounded down after the b-th bit after the binary point, then

$$\lfloor w \rfloor_b \cdot x - t \geq w \cdot x - t - nd_{\max}2^{-b} \geq \Lambda - nd_{\max}2^{-b} > 0 \ .$$

Hence, all solutions that have lower costs than the optimal solution x^* remain infeasible if all weights are rounded down after the b-th bit after the binary point. Similarly, one can argue that if the feasibility gap is larger than $nd_{\max}2^{-b}$, then the optimal solution x^* is still feasible after the rounding. Altogether, this implies that if both the loser and the feasibility gap exceed $nd_{\max}2^{-b}$, the optimal solution x^* is also optimal with respect to the rounded coefficients. Assume, for example, that we are given an integer optimization problem in which one linear constraint is perturbed and for which an algorithm exists that is pseudopolynomial with respect to the coefficients in this constraint. Then by similar reasoning as above and by Lemma 3.3.3, rounding down each coefficient after

$$b = \Theta(\log\left(ndd_{\max}\phi\mu p^{-1}\right))$$

bits does not change the optimal solution with probability at least $1 - p$.

Adaptive Rounding

The final ingredient for turning an algorithm with pseudopolynomial worst-case complexity into an algorithm with polynomial smoothed complexity is an *adaptive rounding procedure*. The main idea of this procedure is very simple: starting with $b = 1$, we round each coefficient after the b-th bit after the binary point, we apply the pseudopolynomial time algorithm to obtain an optimal solution x_b^* for the rounded instance, and we check whether this solution is also optimal for the original unrounded instance. If this is not the case, the precision b is increased by one and the aforementioned steps are repeated. In [BV06] and [RV07] it is described how it can be checked efficiently whether x_b^* is optimal for the original instance if the winner and the feasibility gap are large enough. Based on Lemmas 3.3.2 and 3.3.3 it is shown that this adaptive rounding procedure has polynomial smoothed complexity, that is, its running time satisfies (3.3.1).

3.3.2 Improved Analysis of Loser and Feasibility Gap

The main contribution in our article [RV07] is the analysis of the random variables loser and feasibility gap for integer optimization problems. In a rather technical and lengthy analysis, we generalize Lemma 3.3.3 from the binary to the integer case. In this section, we present a much simpler proof for the following improved version of the lemma. We use the notations $d = |\mathcal{D}|$ and $D = \max\{a - b \mid a, b \in \mathcal{D}\}$.

Lemma 3.3.4. *Let the weights $w_1, \dots, w_n$ be independent random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. There exists a constant κ such that for arbitrary cost functions $c\colon \mathcal{S} \to \mathbb{R}$, arbitrary sets $\mathcal{S} \subseteq \mathcal{D}^n$ with $0^n \notin \mathcal{S}$, and every $\varepsilon \geq 0$,*

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \varepsilon\kappa\phi^2\mu n^3 Dd \log^2 d \quad \textit{and} \quad \mathbf{Pr}[\Lambda \leq \varepsilon] \leq \varepsilon\kappa\phi^2\mu n^3 Dd \log^2 d \ .$$

The main improvement upon our previous analysis is that the bounds in Lemma 3.3.4 depend only linearly on ε instead of $\varepsilon^{1/3}$, which allows us to prove Theorem 1.2.8.

Proof of Lemma 3.3.4. Already in [BV06] it has been observed that the optimal solution x^* and the minimal loser $\hat{x}$ are always Pareto-optimal solutions. Based on this observation, we can rewrite the probabilities as

$$\mathbf{Pr}[\Gamma \leq \varepsilon] = \mathbf{Pr}[\exists x \in \mathcal{P} \mid w \cdot x \in [t - \varepsilon, t]]$$

and

$$\mathbf{Pr}[\Lambda \leq \varepsilon] = \mathbf{Pr}[\exists x \in \mathcal{P} \mid w \cdot x \in (t, t + \varepsilon]] \ ,$$

where $\mathcal{P}$ denotes the set of Pareto-optimal solutions. Hence, in both cases we have to estimate the probability that there exists a Pareto-optimal solution whose weight lies in a certain interval I of length ε, where the interval I is independent of the random variables w_i. In the following, let I denote an arbitrary interval of length ε.

Define $\mathcal{D}' = \mathcal{D} \setminus \{0\}$. For every $i \in [n]$ and $v \in \mathcal{D}'$, we define $\mathcal{S}^{x_i=v} = \{x \in \mathcal{S} \mid x_i = v\}$. Furthermore, we define $\mathcal{P}^{x_i=v} \subseteq \mathcal{S}^{x_i=v}$ to be the set of solutions from $\mathcal{S}^{x_i=v}$ that are not dominated by other solutions from $\mathcal{S}^{x_i=v}$. Since $0^n \notin \mathcal{S}$, every Pareto-optimal solution from $\mathcal{P}$ is contained in at least one set $\mathcal{P}^{x_i=v}$. Hence,

$$\mathbf{Pr}[\exists x \in \mathcal{P}\colon w \cdot x \in I] \leq \sum_{i \in [n]} \sum_{v \in \mathcal{D}'} \mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v}\colon w \cdot x \in I] \ .$$

It remains to analyze the probability that there exists a solution in $\mathcal{P}^{x_i=v}$ whose weight lies in the interval I. In order to bound this probability, we assume that all weights w_j with $j \neq i$ have already been chosen. Then the set $\mathcal{P}^{x_i=v}$ is fixed since w_i affects all solutions from $\mathcal{S}^{x_i=v}$ in the same fashion. In particular, a condition of the form $|\mathcal{P}^{x_i=v}| = a$ does not affect the random variable w_i and

$$\begin{aligned} &\mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v}\colon w \cdot x \in I] \\ &= \sum_{a=1}^{\infty} \mathbf{Pr}[|\mathcal{P}^{x_i=v}| = a] \cdot \mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v}\colon w \cdot x \in I \mid |\mathcal{P}^{x_i=v}| = a] \ . \end{aligned} \tag{3.3.2}$$

For every solution $x \in \mathcal{P}^{x_i=v}$, we can define an interval I_x of length $\varepsilon/|v|$ such that the weight of solution x lies in the interval I if and only if the weight w_i lies in the interval I_x. The union of all these intervals I_x has measure at most $\varepsilon|\mathcal{P}^{x_i=v}|/|v|$, and as the density of the random variable w_i is bounded by ϕ,

$$\mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v}\colon w \cdot x \in I \mid |\mathcal{P}^{x_i=v}| = a] \leq \frac{\varepsilon\phi a}{|v|} \ . \tag{3.3.3}$$

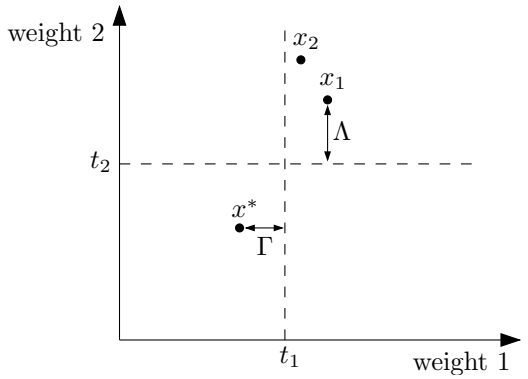


Figure 3.3.2: Let x^* be the winner and let x_1 and x_2 be the only losers, then x_1 is the minimal loser determining the loser gap Λ.

Together (3.3.2) and (3.3.3) yield

$$\mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v} \colon w \cdot x \in I] \leq \sum_{a=1}^{\infty} \mathbf{Pr}[|\mathcal{P}^{x_i=v}| = a] \cdot \frac{\varepsilon\phi a}{|v|} = \frac{\varepsilon\phi}{|v|} \cdot \mathbf{E}[|\mathcal{P}^{x_i=v}|] \quad .$$

From Theorem 3.2.1 we know that there exists a constant κ such that the expected value of $|\mathcal{P}^{x_i=v}|$ is bounded by $\kappa\phi\mu n^2 Dd \log d$ for every $i \in [n]$ and $v \in \mathcal{D}'$. Hence,

$$\begin{aligned}
\mathbf{Pr}[\exists x \in \mathcal{P} : w \cdot x \in I] &\leq \sum_{i\in[n]} \sum_{v\in\mathcal{D}'} \mathbf{Pr}[\exists x \in \mathcal{P}^{x_i=v} : w \cdot x \in I] \\
&\leq \sum_{i\in[n]} \sum_{v\in\mathcal{D}'} \frac{\varepsilon\phi}{|v|} \kappa\phi\mu n^2 Dd \log d \\
&\leq 2\varepsilon\kappa\phi^2\mu n^3 Dd \log^2 d \ ,
\end{aligned}$$

which implies the lemma. □

Loser and Feasibility Gap for Multiple Constraints

Now we consider briefly the case that the set of feasible solutions is determined by multiple linear constraints. That is, we assume that a function $c\colon \mathcal{S} \to \mathbb{R}$ is to be minimized over a set $\mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_m$, where $\mathcal{S} \subseteq \mathcal{D}^n$ denotes a fixed ground set and $\mathcal{B}_i$ denotes a halfspace induced by a linear constraint of the form $w_{i,1}x_1 + \cdots + w_{i,n}x_n \leq t_i$. Again, we denote by x^* the optimal solution and we define the feasibility gap as the minimal slack of the winner x^* to one of the thresholds, that is,

$$\Gamma = \begin{cases} \min_{i\in[m]}(t_i - (w_{i,1}x_1^* + \cdots + w_{i,n}x_n^*)) & \text{if } \mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_m \neq \emptyset, \\ \bot & \text{otherwise.} \end{cases}$$

A solution in $\mathcal{S}$ is called a loser if it has a lower cost than x^*, and we denote by $\mathcal{L}$ the set of all losers. Observe that a loser only needs to be infeasible with respect to one of the m constraints. In particular, it is not true that the weight values of each loser are likely to be far away from the corresponding thresholds

t_i; not even if we consider only those constraints for which the respective loser is infeasible. Fortunately, however, we do not need such a property in the application of the loser gap. For every loser, one only needs a single constraint that renders it infeasible. Therefore, we define the loser gap Λ for multiple constraints as

$$\Lambda = \begin{cases} \min_{x \in \mathcal{L}} \max_{i \in [m]} (w_{i,1}x_1 + \cdots + w_{i,n}x_n - t_i) & \text{if } \mathcal{L} \neq \emptyset, \\ \perp & \text{otherwise.} \end{cases}$$

These definitions are illustrated in Figure 3.3.2.

In [BV06] it is shown how Lemma 3.3.3 gives rise to bounds on the sizes of loser and feasibility gap for multiple constraints. By applying the same arguments, our bounds in Lemma 3.3.4 yield the following lemma.

Lemma 3.3.5. *Let the weights $w_{i,j}$ be independent random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. There exists a constant κ such that for arbitrary cost functions $c\colon \mathcal{S} \to \mathbb{R}$, arbitrary sets $\mathcal{S} \subseteq \mathcal{D}^n$ with $0^n \notin \mathcal{S}$, and every $\varepsilon \geq 0$,*

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \varepsilon\kappa\phi^2\mu mn^3 Dd \log^2 d \quad \text{and} \quad \mathbf{Pr}[\Lambda \leq \varepsilon] \leq \varepsilon\kappa\phi^2\mu mn^3 Dd \log^2 d \ .$$

Proof. First we show the bound for the feasibility gap. Let x^* denote the winner and suppose $\Gamma \leq \varepsilon$. Then there exists a constraint $i \in [m]$ such that x^* has distance at most ε from the threshold t_i, and hence,

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \sum_{i \in [m]} \mathbf{Pr}[t_i - (w_{i,1}x_1^* + \cdots + w_{i,n}x_n^*) \leq \varepsilon] \ .$$

For each individual $i \in [m]$, we can apply Lemma 3.3.4, assuming that the set of feasible solutions with respect to all other constraints is fixed as the coefficients in constraint i are stochastically independent from the other constraints. This way, we obtain

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \varepsilon\kappa\phi^2\mu mn^3 Dd \log^2 d \ .$$

Next, we turn our attention to the loser gap. Unfortunately, we cannot generalize the bound on the loser gap from one to multiple constraints in the same way as we generalized the feasibility gap since the loser gap for multiple constraints does not correspond to the minimal loser gap over the individual constraints. Instead we make use of the result for the feasibility gap established above. Assume $\Lambda \leq \varepsilon$. Then there exists a loser x satisfying

$$\forall i \in [m]\colon w_{i,1}x_1 + \cdots + w_{i,n}x_n - t_i \leq \varepsilon \ .$$

Let x denote a loser that has the smallest cost among all losers with this property. We consider a relaxed instance in which the thresholds of all constraints are increased by ε. Observe that x is feasible in the relaxed instance, and by the definition of x, no solution with smaller cost is feasible. Thus, x is the winner in the relaxed instance. Since for some $i \in [m]$

$$t_i < w_{i,1}x_1 + \cdots + w_{i,n}x_n \leq t_i + \varepsilon \ ,$$

the feasibility gap Γ' of the relaxed problem is smaller than ε. Hence, $\Lambda \leq \varepsilon$ implies $\Gamma' \leq \varepsilon$. Finally, applying the bound derived in the first part of the proof yields

$$\mathbf{Pr}[\Lambda \leq \varepsilon] \leq \mathbf{Pr}\left[\Gamma' \leq \varepsilon\right] \leq \varepsilon\kappa\phi^2\mu mn^3 Dd \log^2 d \ . \qquad \square$$

Expected Polynomial Running Time

The aforementioned definition of polynomial smoothed complexity yields a notion that does not vary among classes of machines admitting polynomial time simulations among each other. The drawback of this definition is, however, that polynomial smoothed complexity does not imply *expected polynomial running time*. For the binary case, it is shown in [BV06] that problems that admit a pseudolinear algorithm, i.e., an algorithm whose running time is bounded by $O(\text{poly}(N) \cdot W)$, where N denotes the input size and W the largest absolute value of one of the coefficients in the input, can be solved in expected polynomial time on semi-random instances. For this, one has to assume that the tail functions of the random coefficients exhibit an exponential decay. More precisely, we assume that there exists a constant $E \in \mathbb{R}_{\geq 0}$ such that $\mathbf{Pr}[|X| > \alpha E] \leq 2^{-\alpha}$ for every $\alpha \geq 2$ and for every coefficient X. For example, the Gaussian and the exponential distribution have this property as well as all distributions with finite domain. Based on Lemma 3.3.5, we prove the following theorem, which is a generalization of Theorem 1.2.8.

Theorem 3.3.6. *Assume that there exists a constant $E > 0$ such that for every random coefficient X it holds $\mathbf{Pr}[|X| > \alpha E] \leq 2^{-\alpha}$ for every $\alpha \geq 2$. If an integer optimization problem can be solved in the worst case in time $O(\text{poly}(N) \cdot W)$, then it allows an algorithm with expected polynomial running time on perturbed instances.*

Proof. We analyze the expected running time of the adaptive rounding procedure described in Section 3.3.1. Remember that in this procedure, we start by revealing the first bit of each random coefficient after the binary point. Then we compute the solution that is optimal with respect to the rounded coefficients and check whether it is also optimal with respect to the original non-rounded coefficients. If this is not the case, we increase the number b of revealed bits after the binary point by one and repeat the aforementioned steps. Let us consider one iteration of this algorithm in which b bits after the binary point of each number have already been revealed. The running time of this iteration is made up of the running time for finding the optimal solution of the rounded instance and the running time for checking whether it is also optimal for the non-rounded instance. In [RV07], we show that this check can be performed by solving $O(n)$ rounded instances. In these rounded instances, each number is rounded after b bits after the binary point, but some numbers are rounded up while others are rounded down. The largest number in the rounded instances in the considered iteration is $W = W_1 W_2$, where $W_1 = 2^b$ and W_2 denotes the largest absolute integer part of any of the coefficients. The factor W_1 is due to the fact that we have to scale the rounded coefficients by 2^b in order to make them integers. Hence, if the running time of the pseudopolynomial time algorithm is $O(N^{l-1}W)$ for some constant l, the running time of the considered iteration is $O(nN^{l-1}W_1W_2) = O(N^l 2^b W_2)$.

By similar arguments as in Section 3.3.1, it is shown in [RV07] that more than b bits after the binary point need only be revealed if, in the case that the objective function is perturbed, the winner gap is smaller than $nd_{\max}2^{b+1}$ or if, in the case that constraints are perturbed, the loser or the feasibility gap is smaller than $(nd_{\max}+1)2^{b+1}$. We denote by k the number of perturbed linear expressions, that is, $k = m$ if only m constraints are perturbed, and $k = m+1$ if m constraints

and the objective function are perturbed. Due to Lemmas 3.3.2 and 3.3.5, we can upper bound the probability of this event by

$$\text{poly}(\phi, k, n, d_{\max}) \cdot 2^{-b} \ , \tag{3.3.4}$$

where we used the fact that the expected absolute values of the coefficients are bounded by some constant depending on E.

We define $\mathcal{F}$ to be the failure event that either at least $n \log d$ bits after the binary point of each random coefficient need to be revealed or that $W_2 > (n \log d)E$. Due to the assumption about the probability distributions of the coefficients, the probability for the latter event can be bounded by a union bound from above by

$$nk \cdot 2^{-n \log d} = (nk)d^{-n} \ .$$

Together with (3.3.4) this implies that the probability of the failure event $\mathcal{F}$ is bounded from above by

$$\text{poly}(\phi, k, n, d_{\max}) \cdot 2^{-n \log d} + (nk)d^{-n} = \text{poly}(\phi, k, n, d_{\max}) \cdot d^{-n} \ .$$

Before the adaptive rounding procedure is started, it is checked whether the absolute value of one of the coefficients is larger than $(n \log d)E$. If this is the case, the problem is solved by a brute force enumeration over all d^n possible solutions. If this is not the case, we start the adaptive rounding procedure. If this procedure reveals $n \log d$ bits after the binary point of each coefficient without finding the optimal solution, it is stopped and again replaced by a brute force enumeration. In this case, the running time of the unsuccessful adaptive rounding procedure is

$$\sum_{b=1}^{n \log d} O\left(N^l 2^b (n \log d)E\right) = O\left(N^l d^n (n \log d)E\right) \ .$$

Hence, in case of the failure event $\mathcal{F}$, the running time is bounded from above by $\text{poly}(\phi, k, N, d_{\max}) \cdot d^n$. Since the failure event occurs only with probability at most $\text{poly}(\phi, k, n, d_{\max}) \cdot d^{-n}$, this implies that the contribution of the failure event to the expected running time is polynomial.

It remains to consider the case that the failure event $\mathcal{F}$ does not occur. Let b denote the number of bits that have been revealed of each number in the last iteration of the adaptive rounding procedure and let $W_1 = 2^b$. We define the random variable W_1' as

$$W_1' = \begin{cases} W_1 & \text{in case of } \neg\mathcal{F}, \\ 0 & \text{otherwise.} \end{cases}$$

By (3.3.4), it follows that

$$\begin{aligned} \mathbf{E}\left[W_1'\right] &\leq \sum_{i=0}^{n \log d} \mathbf{Pr}\left[W_1' = 2^i\right] 2^i \leq \sum_{i=0}^{n \log d} \mathbf{Pr}\left[W_1 = 2^i\right] 2^i \\ &\leq \sum_{i=0}^{n \log d} \text{poly}(\phi, k, n, d_{\max}) \cdot 2^{-i} \cdot 2^i = \text{poly}(\phi, k, n, d_{\max}) \ . \end{aligned} \tag{3.3.5}$$

Let T denote the running time of this algorithm. Since

$$\mathbf{E}\left[W_1'\right] = \mathbf{Pr}[\neg\mathcal{F}]\,\mathbf{E}\left[W_1' \,\middle|\, \neg\mathcal{F}\right] = \mathbf{Pr}[\neg\mathcal{F}]\,\mathbf{E}[W_1 \mid \neg\mathcal{F}]$$

and, in case of $\neg\mathcal{F}$, the running time of the adaptive rounding procedure is always dominated by the running time of the last iteration, we obtain

$$\begin{aligned}\mathbf{E}[T] &\leq \mathbf{Pr}[\neg\mathcal{F}] \cdot \mathbf{E}[T \mid \neg\mathcal{F}] + \mathbf{Pr}[\mathcal{F}] \cdot \mathbf{E}[T \mid \mathcal{F}] \\ &\leq N^l((n \log d)E) \cdot \mathbf{Pr}[\neg\mathcal{F}] \cdot \mathbf{E}[W_1 \mid \neg\mathcal{F}] + \mathrm{poly}(\phi, k, N, d_{\max}) \\ &\leq \mathrm{poly}(\phi, k, N, d_{\max}) \ ,\end{aligned}$$

which concludes the proof. □

Observe that the previous proof does not yield expected polynomial running time, when the old bounds derived in [RV07] instead of our new bounds in Lemma 3.3.5 are used. When the old bounds are used, the running time in (3.3.5) is not polynomial anymore.

3.4 Enumeration of the Pareto Set

In this section, we consider bicriteria binary optimization problems in which an adversary can choose an arbitrary set $\mathcal{S} \subseteq \{0,1\}^n$ of feasible solutions and in which the costs $c\colon \mathcal{S} \to \mathbb{R}$ and the weights $w\colon \mathcal{S} \to \mathbb{R}$ are to be minimized. We consider semi-random instances in which both the cost and the weight function are linear with perturbed coefficients. That is, we assume that $c(x) = c_1x_1 + \cdots + c_nx_n$ and $w(x) = w_1x_1 + \cdots + w_nx_n$ and that the costs c_i and the weights w_i are independent random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. We make the additional assumption that all weights w_i are non-negative.

For a few problems like, for example, the shortest path problem, algorithms are known that compute the Pareto set in polynomial time in the input size and the size of the Pareto set. For many other well-studied problems like the spanning tree and the perfect matching problem, it is still unknown whether such algorithms exist. In this section, we consider problems for which the Pareto set can be computed in pseudopolynomial time in the worst case. For these problems, we present an algorithmic approach that allows use to compute with probability at least $1-p$, for every desired failure probability $p > 0$, the Pareto set of semi-random instances in polynomial time. The assumption that the Pareto set can be computed in pseudopolynomial time is not very restrictive. This assumption is, for instance, satisfied for every problem whose exact single-criterion version (i.e., the question, "Is there a solution with cost exactly x?") can be solved in pseudopolynomial time, including the spanning tree and the perfect matching problem.

In [ANRV07], we show that f-Π problems are NP-hard even for simple polynomial objective functions and we show that it is hard to approximate them for rapidly increasing objective functions if Π is the bicriteria spanning tree, shortest path, or perfect matching problem. The optimal solution of an f-Π problem is a Pareto-optimal solution if f is non-decreasing. Hence, our aforementioned result

implies that we can, for any problem Π whose Pareto set can be computed in pseudopolynomial time and for every non-decreasing function f, devise an algorithm for the f-Π problem that is efficient on semi-random inputs.

In the previous section, we have shown that integer optimization problems are robust against rounding of the coefficients. In this section, we show that the Pareto set of bicriteria binary optimization problems is also robust against rounding. To be precise, we show that if one rounds each coefficient in the linear objective functions, that is, each weight and each cost, after a logarithmic number of bits, then with high probability every solution that is Pareto-optimal in the original instance is also Pareto-optimal in the rounded instance. Hence, with high probability we can generate the complete Pareto set by taking into account only a logarithmic number of bits of each input number. This way, an algorithm with pseudopolynomial worst-case complexity for generating the Pareto set can be turned into a polynomial time algorithm that succeeds with high probability on semi-random instances.

Outline of the Approach

We show how a pseudopolynomial time algorithm $\mathcal{A}$ for generating the Pareto set can be turned into a polynomial time algorithm that succeeds with probability at least $1-p$, for arbitrary $p > 0$, on semi-random inputs. In order to apply $\mathcal{A}$ efficiently it is necessary to round the costs and weights such that they are only polynomially large after the rounding, i.e., such that the lengths of their representations are only logarithmic in the input size. Let $\lfloor c \rfloor_b$ and $\lfloor w \rfloor_b$ denote the costs and weights rounded down to b bits after the binary point. We denote by $\mathcal{P}$ the Pareto set with respect to the objective functions c and w and by $\mathcal{P}_b$ the Pareto set with respect to the rounded objective functions $\lfloor c \rfloor_b$ and $\lfloor w \rfloor_b$. The following lemma is the main result in this section.

Lemma 3.4.1. *Let the costs $c_1, \ldots, c_n$ and the non-negative weights $w_1, \ldots, w_n$ be independent random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. There exists a constant κ such that for every $p > 0$ and for $b = \kappa \log\left(n\phi\mu p^{-1}\right)$, the Pareto set $\mathcal{P}$ is a subset of $\mathcal{P}_b$ with probability at least $1-p$.*

This means, we can round the coefficients after only a logarithmic number of bits and use the pseudopolynomial time algorithm, which runs on the rounded input in polynomial time, to obtain $\mathcal{P}_b$. With probability at least $1-p$, the set $\mathcal{P}_b$ contains all Pareto-optimal solutions from $\mathcal{P}$. It can contain polynomially many additional solutions that are not Pareto-optimal w.r.t. c and w. By removing these superfluous solutions, we obtain the set $\mathcal{P}$ with probability at least $1-p$. This implies Theorem 1.2.7 stated in the introduction.

From the definition of a Pareto-optimal solution, it follows that the optimal solution x^* of a constrained problem, i.e., the solution with the smallest cost among all solutions fulfilling a weight constraint $w \cdot x \leq t$, is always Pareto-optimal. Otherwise, if there were a solution x that dominates x^*, then x would also be a better solution to the constrained problem. We show that with sufficiently large probability, for every $x \in \mathcal{P}$, we can find a threshold t such that x is the optimal solution to the constrained problem in which the function $\lfloor c \rfloor_b \cdot x$ is to be minimized

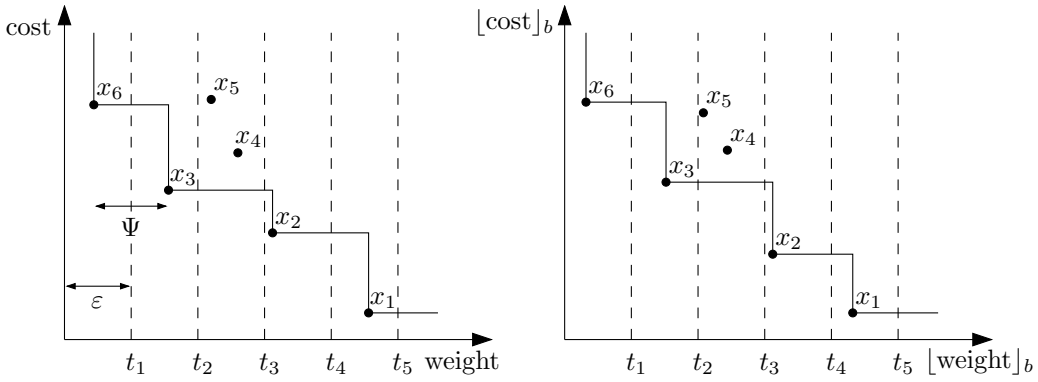


Figure 3.4.1: In this illustration, none of the failure events occurs. We have $\varepsilon < \Psi$ and the weight of every solution lies in the interval $[0, z\varepsilon]$. Additionally, $x^{(1)} = x_b^{(1)} = x_6$, $x^{(2)} = x_b^{(2)} = x^{(3)} = x_b^{(3)} = x_3$, $x^{(4)} = x_b^{(4)} = x_2$, $x^{(5)} = x_b^{(5)} = x_1$.

over $\mathcal{S}$ with the additional constraint that $\lfloor w \rfloor_b \cdot x \leq t$. This implies that with sufficiently large probability, every $x \in \mathcal{P}$ is Pareto-optimal w.r.t. the rounded coefficients.

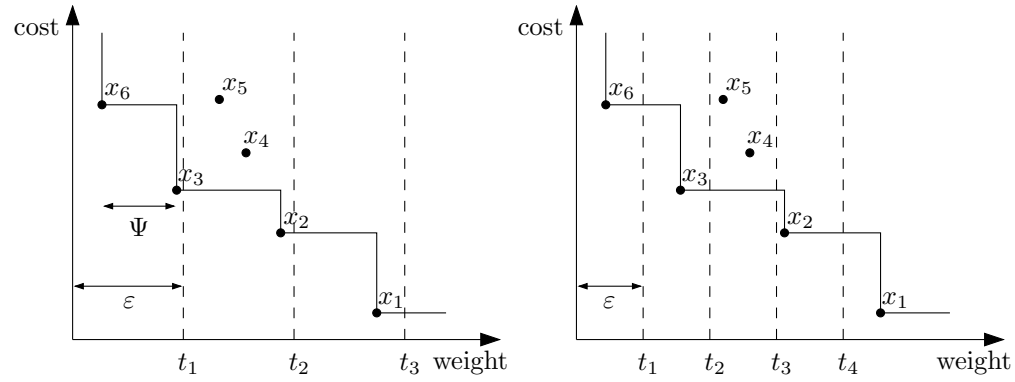
To be more precise, we consider, for appropriately chosen $z \in \mathbb{N}$ and $\varepsilon > 0$, z many constrained problems with costs $\lfloor c \rfloor_b$, weights $\lfloor w \rfloor_b$, and thresholds $t_i = i\varepsilon$, for $i \in [z]$. We denote the minimal weight difference between two different Pareto-optimal solutions by Ψ, i.e.,

$$\Psi = \min_{\substack{x,y \in \mathcal{P} \\ x \neq y}} |w \cdot x - w \cdot y| \ .$$

If Ψ is larger than ε and the weight of all solutions lies in the interval $[0, z\varepsilon]$, then $\mathcal{P}$ consists only of solutions to constrained problems of the form $\min(c \cdot x)$, w.r.t. $x \in \mathcal{S}$ and $w \cdot x \leq t_i$ because in that case we do not miss a Pareto-optimal solution by our choice of thresholds. We prove that, for each $i \in [z]$, the solution $x^{(i)}$ to the constrained problem $\min(c \cdot x)$ w.r.t. $x \in \mathcal{S}$ and $w \cdot x \leq t_i$ is the same as the solution $x_b^{(i)}$ to the constrained problem $\min(\lfloor c \rfloor_b \cdot x)$ w.r.t. $x \in \mathcal{S}$ and $\lfloor w \rfloor_b \cdot x \leq t_i$ with sufficiently large probability. Thus, if $\varepsilon < \Psi$, the weights of all solutions lie in the interval $[0, z\varepsilon]$, and $x^{(i)} = x_b^{(i)}$ for all $i \in [z]$, then $\mathcal{P} \subseteq \mathcal{P}_b$. See Figure 3.4.1 for an illustration of this approach.

We do not know how to determine Ψ in polynomial time, but we can show a lower bound for Ψ that holds with a certain probability. Based on this lower bound, we can appropriately choose ε. Then we choose z sufficiently large, so that $w \cdot x \in [0, z\varepsilon]$ holds with sufficiently large probability for every solution x. Thus, our analysis fails only if one of the following three failure events occurs (cf. Figures 3.4.2 and 3.4.3):

- $\mathcal{F}_1$: Ψ is smaller than the chosen ε.
- $\mathcal{F}_2$: For one $i \in [z]$, the solution $x^{(i)}$ to $\min(c \cdot x)$ w.r.t. $x \in \mathcal{S}$ and $w \cdot x \leq t_i$ does not equal the solution $x_b^{(i)}$ to $\min(\lfloor c \rfloor_b \cdot x)$ w.r.t. $x \in \mathcal{S}$ and $\lfloor w \rfloor_b \cdot x \leq t_i$.
- $\mathcal{F}_3$: There exists a solution x with $w \cdot x > z\varepsilon$.



(a) Failure event $\mathcal{F}_1$: $\varepsilon > \Psi$. x_6 is not a solution to any of the constrained problems.

(b) Failure event $\mathcal{F}_3$: For $z = 4$, $w \cdot x_1 > z \cdot \varepsilon$, and hence, x_1 is not a solution to any of the constrained problems.

Figure 3.4.2: Failure events $\mathcal{F}_1$ and $\mathcal{F}_3$.

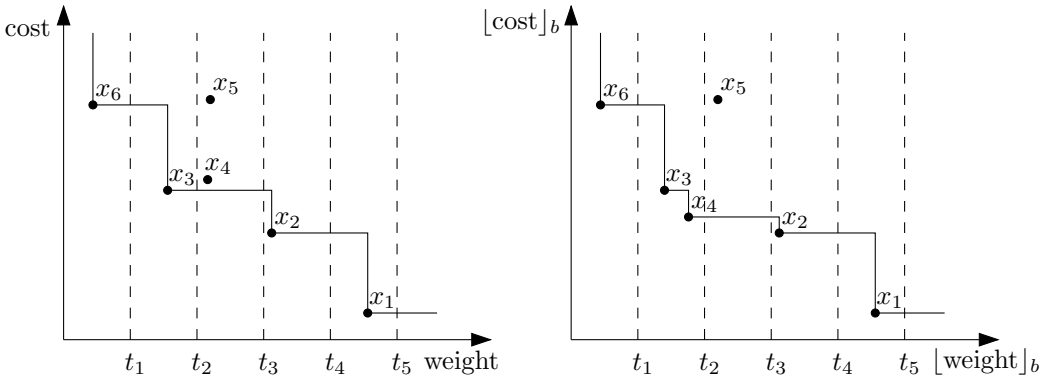


Figure 3.4.3: Failure event $\mathcal{F}_2$: $x^{(2)} = x_3 \neq x_4 = x_b^{(2)}$.

In order to prove Lemma 3.4.1, we first have to estimate the probability of the failure events. Depending on these failure probabilities, we can choose appropriate values for z, ε, and b yielding the theorem. We start by estimating the probability of the first failure event, which is the most involved part of the proof.

Estimating the Size of the Smallest Gap

For bounding Ψ, it is not necessary that the costs are chosen at random because the bound that we prove holds for every deterministic choice of the costs. Thus, we assume the costs to be fixed arbitrarily.

Let $y^{(1)}, \ldots, y^{(l)}$ denote a sequence containing all elements from $\mathcal{S}$ ordered in increasing order of costs, that is, $c \cdot y^{(1)} \leq \cdots \leq c \cdot y^{(l)}$. For $j \in \{2, \ldots, l\}$, we define

$$\Psi_j = \min_{i \in [j-1]} (w \cdot y^{(i)}) - \min_{i \in [j]} (w \cdot y^{(i)}) \ .$$

Observe that a solution $y^{(j)}$, for $j \in \{2, \ldots, l\}$, is Pareto-optimal if and only if $\Psi_j > 0$ and that Ψ_j describes how much less $y^{(j)}$ weighs compared to the solutions

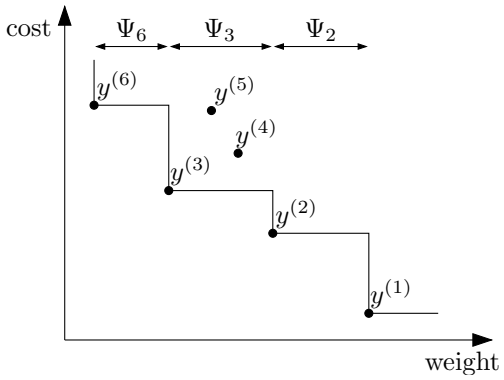


Figure 3.4.4: In this example, $\Psi_4 = \Psi_5 = 0$ and $\Psi = \Psi_6$.

$y^{(i)}$ with $i < j$ (see Figure 3.4.4). Thus, we can write Ψ as

$$\Psi = \min_{j \in [l] \setminus \{1\} : \Psi_j > 0} \Psi_j \ .$$

Our goal is to bound the probability that Ψ lies below a given value ε. Therefore, we rewrite the probability of the event $\Psi < \varepsilon$ as

$$\begin{aligned} \mathbf{Pr}[\Psi < \varepsilon] &= \mathbf{Pr}[\exists j \in [l] \setminus \{1\} \colon 0 < \Psi_j < \varepsilon] \\ &\leq \sum_{j \in [l] \setminus \{1\}} \mathbf{Pr}[\Psi_j > 0] \cdot \mathbf{Pr}[\Psi_j < \varepsilon \mid \Psi_j > 0] \ . \end{aligned} \tag{3.4.1}$$

Let us assume that we can bound the conditional probability $\mathbf{Pr}[\Psi_j < \varepsilon \mid \Psi_j > 0]$ from above for every j by some term a. Then we have

$$\begin{aligned} \mathbf{Pr}[\Psi < \varepsilon] &\leq a \cdot \sum_{j \in [l] \setminus \{1\}} \mathbf{Pr}[\Psi_j > 0] \\ &= a \cdot (\mathbf{E}[q] - 1) \leq a \cdot \mathbf{E}[q] \ , \end{aligned}$$

where q denotes the number of Pareto-optimal solutions.

We can apply Theorem 3.2.1 to obtain a polynomial upper bound on the expected number of Pareto-optimal solutions. The crucial point in Beier and Vöcking's analysis [BV04] of the expected number of Pareto-optimal solutions is a lower bound on $\mathbf{E}[\Psi_j \mid \Psi_j > 0]$ for every $j \in [l] \setminus \{1\}$. Unfortunately, we cannot apply their results directly to bound the conditional probability $\mathbf{Pr}[\Psi_j < \varepsilon \mid \Psi_j > 0]$ since in general, a bound on the conditional expectation does not imply a bound on the conditional probability. Nonetheless, we prove the following result.

Lemma 3.4.2. *Let the weights be independent positive random variables whose densities are bounded by ϕ and whose expected absolute values are bounded by μ. Then, for every $\varepsilon \leq (6n^5\phi^2\mu)^{-1}$,*

$$\mathbf{Pr}[\Psi < \varepsilon] \leq 2(6\varepsilon n^5 \phi^2 \mu)^{1/3} \ .$$

Analogously to the analysis in [BV04], we also look at long-tailed distributions first, and after that we use the results for long-tailed distributions to analyze the general case.

Long-Tailed Distributions. One can classify continuous probability distributions by comparing their tails with the tail of the exponential distribution. In principle, if the tail function of a distribution can be lower bounded by the tail function of an exponential function, then we say the distribution has a "long tail".

Of special interest to us is the behavior of the tail function under a logarithmic scale. Given any continuous probability distribution with density $f\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, the *tail function* $T\colon \mathbb{R}_{\geq 0} \to [0,1]$ is defined by $T(t) = \int_t^\infty f(x)\,dx$. We define the *slope* of T at $x \in \mathbb{R}_{\geq 0}$ to be the first derivative of the function $-\ln(T(\cdot))$ at x, i.e., $\operatorname{slope}_T(x) = -\frac{d}{dx}\ln(T(x))$. For example, the tail function of the exponential distribution with parameter λ is $T(x) = \exp(-\lambda x)$ so that the slope of this function is $\operatorname{slope}_T(x) = \lambda$ for every $x \geq 0$. The tail of a continuous probability distribution is defined to be *long* if there exists a constant $\alpha > 0$ such that $\operatorname{slope}_T(x) \leq \alpha$ for every $x \geq 0$.

For $i \in [n]$, we denote by T_i the tail function of the weight w_i and by f_i the corresponding density function. Beier and Vöcking prove the following theorem on the expected number of Pareto-optimal solutions.

Theorem 3.4.3 ([BV04])**.** *Let the weights $w_1, \ldots, w_n$ be positive long-tailed random variables with expected values at most μ, and let α be a positive real number such that $\operatorname{slope}_{T_i}(x) \leq \alpha$ for every $x \geq 0$ and every $i \in [n]$. For the number q of Pareto-optimal solutions, it holds*

$$\mathbf{E}[q] \leq \alpha\mu n^2 + 1 \ .$$

In order to bound the conditional probability $\mathbf{Pr}[\,\Psi_j < \varepsilon \mid \Psi_j > 0]$, we have to take a closer look into the proof of Theorem 3.4.3. The following lemma is implicitly contained in this proof.

Lemma 3.4.4 ([BV04])**.** *Let α and μ be defined as in Theorem 3.4.3. Then for every $j \in [l] \setminus \{1\}$, it holds*

$$\mathbf{Pr}[\,\Psi_j < \varepsilon \mid \Psi_j > 0] \leq 1 - \exp(-n\alpha\varepsilon) \ .$$

Let $\varepsilon \geq 0$ be fixed arbitrarily. Combining Theorem 3.4.3 and Lemma 3.4.4 with equation (3.4.1) yields

$$\begin{aligned}
\mathbf{Pr}[\Psi < \varepsilon] &\leq \sum_{j \in [l]\setminus\{1\}} \mathbf{Pr}[\Psi_j > 0] \cdot \mathbf{Pr}[\,\Psi_j < \varepsilon \mid \Psi_j > 0] \\
&\leq (1 - \exp(-n\alpha\varepsilon)) \cdot \mathbf{E}[q] \\
&\leq \varepsilon \cdot n\alpha \cdot \mathbf{E}[q] \\
&\leq \varepsilon \cdot 2n^3\alpha^2\mu \ .
\end{aligned}$$

Thus, we obtain the following lemma.

Lemma 3.4.5. *Let α and μ be defined as in Theorem 3.4.3. Then for every $\varepsilon \geq 0$,*

$$\mathbf{Pr}[\Psi < \varepsilon] \leq \varepsilon \cdot 2n^3\alpha^2\mu \ .$$

General Distributions with Bounded Mean and Bounded Density. For general distributions, a statement like Lemma 3.4.4 is not true anymore. Nonetheless, Lemma 3.4.2 is also true for short-tailed distributions.

Proof of Lemma 3.4.2. For every weight $i \in [n]$, we define a random variable $W_i = T_i(w_i)$. For any $a > 0$, let $\mathcal{F}_a$ denote the event that, for at least one weight $i \in [n]$, it holds $W_i \leq a$. We show that we can apply the analysis for long-tailed distributions if $\mathcal{F}_a$ does not occur. We estimate the probability of the event that Ψ is smaller than ε as

$$\mathbf{Pr}[\Psi < \varepsilon] \leq \mathbf{Pr}[\mathcal{F}_a] + \mathbf{Pr}[\Psi < \varepsilon \wedge \neg\mathcal{F}_a] \ . \tag{3.4.2}$$

The random variables W_i are uniformly distributed over $[0, 1]$ because for every $z \in [0, 1]$,

$$\begin{aligned}\mathbf{Pr}[W_i \leq z] &= \mathbf{Pr}\left[w_i \geq T_i^{-1}(z)\right] \\ &= \int_{T_i^{-1}(z)}^{\infty} f_i(x)\,dx \\ &= T_i(T_i^{-1}(z)) = z \ .\end{aligned}$$

Thus, we obtain

$$\mathbf{Pr}[\mathcal{F}_a] = \mathbf{Pr}[\exists i \in [n] \colon x_i \leq a] \leq na \ . \tag{3.4.3}$$

We would like to estimate $\mathbf{Pr}[\Psi < \varepsilon \wedge \neg\mathcal{F}_a]$ in such a way that we get rid of the event $\neg\mathcal{F}_a$ since, under the condition $\neg\mathcal{F}_a$, the random variables w_i are short-tailed instead of long-tailed. If the event $\mathcal{F}_a$ does not occur, the distribution of a weight w_i for values larger than $T_i^{-1}(a)$ is not important. Thus, we can replace the tail function T_i by a tail function T_i^* with

$$T_i^*(x) = \begin{cases} T_i(x) & \text{if } x \leq T_i^{-1}(a), \\ a \cdot \exp\left(-\phi n \left(x - T_i^{-1}(a)\right)\right) & \text{otherwise.} \end{cases}$$

We denote by w_i^* a random variable drawn according to the tail function T_i^*. Furthermore, we denote by Ψ^* the random variable equivalent to Ψ but w.r.t. the weights w_i^* drawn according to the tail functions T_i^* instead of T_i. Then

$$\mathbf{Pr}[\Psi < \varepsilon \wedge \neg\mathcal{F}_a] = \mathbf{Pr}[\Psi^* < \varepsilon \wedge \neg\mathcal{F}_a] \leq \mathbf{Pr}[\Psi^* < \varepsilon] \ . \tag{3.4.4}$$

Let f_i^* denote a density corresponding to the tail function T_i^*. The random variable w_i^* is long-tailed since

$$\operatorname{slope}_{T_i^*}(x) = -\frac{d}{dx}\ln(T_i^*(x)) = \frac{f_i^*(x)}{T_i^*(x)} \leq \begin{cases} \phi/a & \text{if } x \leq T_i^{-1}(a), \\ \phi n & \text{otherwise.} \end{cases}$$

For $a \leq 1/n$, this implies

$$\operatorname{slope}_{T_i^*}(x) \leq \phi/a \ .$$

Before we can apply Lemma 3.4.5, we have to calculate the expectations of the random variables w_i^* drawn according to the tail functions T_i^*. It holds

$$\begin{aligned}\int_0^\infty x f_i^*(x)\,dx &= \int_0^{T_i^{-1}(a)} x f_i(x)\,dx + \int_{T_i^{-1}(a)}^\infty x f_i^*(x)\,dx \\ &\leq \mu + a\phi n \int_0^\infty (x + T_i^{-1}(a)) e^{-\phi n x}\,dx \\ &\leq \mu + a\phi n \left(\int_0^\infty x e^{-\phi n x}\,dx + \int_0^\infty T_i^{-1}(a) e^{-\phi n x}\,dx \right) \\ &\leq \mu + \frac{a}{\phi n} + a \cdot T_i^{-1}(a)\ .\end{aligned}$$

An application of Markov's inequality yields $T_i(\mu/a) = \mathbf{Pr}[w_i \geq \mu/a] \leq a$, and hence, $T_i^{-1}(a) \leq \mu/a$. Therefore, we have

$$\int_0^\infty x f_i^*(x)\,dx \leq \mu + \frac{a}{\phi n} + \mu \leq 2\mu + 1 \leq 3\mu\ .$$

Applying Lemma 3.4.5 with $\alpha' = \phi/a$ and $\mu' = 3\mu$ yields, for every $\varepsilon \geq 0$,

$$\mathbf{Pr}[\Psi^* < \varepsilon] \leq \frac{6\varepsilon n^3 \phi^2 \mu}{a^2}\ . \tag{3.4.5}$$

Equations (3.4.2) to (3.4.5) result in the following bound:

$$\mathbf{Pr}[\Psi < \varepsilon] \leq na + \frac{6\varepsilon n^3 \phi^2 \mu}{a^2}\ .$$

We choose $a = (6\varepsilon n^2 \phi^2 \mu)^{1/3}$ and obtain

$$\mathbf{Pr}[\Psi < \varepsilon] \leq 2(6\varepsilon n^5 \phi^2 \mu)^{1/3}\ .$$

We assumed a to be less or equal to $1/n$. Thus, we have to choose ε such that $(6\varepsilon n^2 \phi^2 \mu)^{1/3} \leq 1/n$. This is equivalent to $\varepsilon \leq (6n^5 \phi^2 \mu)^{-1}$. □

Proof of Lemma 3.4.1

In the following, fix some $i \in [z]$ and let $\mathcal{F}_2^{(i)}$ denote the event that the solution $x^{(i)}$ does not equal the solution $x_b^{(i)}$. The situation is very similar to the situation considered in Section 3.3. We have a binary optimization problem and we need to bound the probability that rounding the coefficients in the objective function and the constraint changes the optimal solution. The crucial observation in Section 3.3 is that whenever the winner, loser, and feasibility gap are large enough, the optimal solution of the constrained problem stays optimal even w.r.t. the rounded coefficients. To be precise, we have seen that for the binary case, rounding down each coefficient after the b-th bit after the binary point cannot change the optimal solution if the winner gap is larger than $n2^{-b+1}$ and if both loser and feasibility gap are larger than $n2^{-b}$. In fact, for the binary case, the optimal solution $x^{(i)}$ cannot become infeasible due to the rounding because all coefficients are rounded down and hence, the weight of $x^{(i)}$ can only decrease by the rounding. This means, for the binary case, the feasibility gap is not important. By this reasoning, using Lemmas 3.3.2 and 3.3.3 yields the following lemma.

Lemma 3.4.6. *The probability of the failure event $\mathcal{F}_2$ is bounded from above by*

$$z \cdot 2^{-b+2} n^3 \phi \ .$$

Proof. We show that for every $i \in [z]$, the probability of the event $\mathcal{F}_2^{(i)}$ is bounded from above by $2^{-b+2} n^3 \phi$. Then a union bound over all $i \in [z]$ yields the lemma.

Let $i \in [z]$ be fixed. Let Λ denote the loser gap of the constrained problem $\min(c \cdot x)$, w.r.t. $x \in \mathcal{S}$ and $w \cdot x \leq t_i$. And let Δ denote the winner gap of the constrained problem $\min(c \cdot x)$, w.r.t. $x \in \mathcal{S}$ and $\lfloor w \rfloor_b \cdot x \leq t_i$, in which the weights are already rounded. From the argumentation in Section 3.3 it follows that the winners $x^{(i)}$ and $x_b^{(i)}$ can only differ if either $\Delta \leq n 2^{-b+1}$ or $\Lambda \leq n 2^{-b}$. Combining this observation with Lemmas 3.3.2 and 3.3.3 yields

$$\begin{aligned} \mathbf{Pr}\left[\mathcal{F}_2^{(i)}\right] &\leq \mathbf{Pr}\left[\Delta \leq n 2^{-b+1}\right] + \mathbf{Pr}\left[\Lambda \leq n 2^{-b}\right] \\ &\leq n^2 \cdot 2^{-b+1} \cdot \phi + n^3 \cdot 2^{-b} \cdot \phi \\ &\leq 2^{-b+2} n^3 \phi \ , \end{aligned}$$

which implies the lemma. □

Now we use the Lemmas 3.4.2 and 3.4.6 to prove Lemma 3.4.1.

Proof of Lemma 3.4.1. We would like to choose ε, z and b in such a way that each failure probability $\mathbf{Pr}[\mathcal{F}_i]$ is bounded by $p/3$. By Lemma 3.4.2, choosing $\varepsilon = p^3 (1296 n^5 \phi^2 \mu)^{-1}$ yields $\mathbf{Pr}[\mathcal{F}_1] \leq p/3$. By a simple application of Markov's inequality, we obtain that choosing

$$z = \frac{3888 n^6 \phi^2 \mu^2}{p^4}$$

implies $\mathbf{Pr}[\mathcal{F}_3] \leq p/3$. Due to Lemma 3.4.6, setting $b = \log(\alpha n^9 \phi^3 \mu^2 p^{-5})$, for an appropriate constant α, yields $\mathbf{Pr}[\mathcal{F}_2] \leq p/3$.

This proves the lemma since for $b = \log(\alpha n^9 \phi^3 \mu^2 p^{-5}) = \Theta(\log(n \phi \mu p^{-1}))$, the failure probability is at most p. □

Chapter 4

Local Optima

In this chapter, we prove the results about the 2-Opt heuristic for the traveling salesperson problem that we described in Section 1.3. After stating some preliminaries and notations, we start by presenting the lower bounds, showing that the 2-Opt heuristic can take an exponential number of steps on two-dimensional instances for each L_p metric. Then we present the upper bounds on the expected running time and the expected approximation ratio.

4.1 Preliminaries

An *instance of the TSP* consists of a set $V = \{v_1, \ldots, v_n\}$ of *vertices* (depending on the context, synonymously referred to as *points*) and a symmetric *distance function* $\mathsf{d}\colon V \times V \to \mathbb{R}_{\geq 0}$ that associates with each pair $\{v_i, v_j\}$ of distinct vertices a distance $\mathsf{d}(v_i, v_j) = \mathsf{d}(v_j, v_i)$. The goal is to find a Hamiltonian cycle of minimum length. We also use the term *tour* to denote a Hamiltonian cycle. For a natural number $n \in \mathbb{N}$, we denote the set $\{1, \ldots, n\}$ by $[n]$.

A pair (V, d) of a nonempty set V and a function $\mathsf{d}\colon V \times V \to \mathbb{R}_{\geq 0}$ is called a *metric space* if for all $x, y, z \in V$ the following properties are satisfied:

(a) $\mathsf{d}(x, y) = 0$ if and only if $x = y$ *(reflexivity)*,

(b) $\mathsf{d}(x, y) = \mathsf{d}(y, x)$ *(symmetry)*,

(c) $\mathsf{d}(x, z) \leq \mathsf{d}(x, y) + \mathsf{d}(y, z)$ *(triangle inequality)*.

If (V, d) is a metric space, then d is called a *metric on* V. A TSP instance with vertices V and distance function d is called *metric TSP instance* if (V, d) is a metric space.

A well-known class of metrics on $\mathbb{R}^d$ is the class of L_p *metrics*. For $p \in \mathbb{N}$, the distance $\mathsf{d}_p(x, y)$ of two points $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^d$ with respect to the L_p metric is given by $\mathsf{d}_p(x, y) = \sqrt[p]{|x_1 - y_1|^p + \cdots + |x_d - y_d|^p}$. The L_1 metric is often called *Manhattan metric*, and the L_2 metric is well-known as *Euclidean metric*. For $p \to \infty$, the L_p metric converges to the L_∞ metric defined by the distance function $\mathsf{d}_\infty(x, y) = \max\{|x_1 - y_1|, \ldots, |x_d - y_d|\}$. A TSP instance (V, d) with $V \subseteq \mathbb{R}^d$ in which d equals d_p restricted to V is called an L_p *instance*. We also use the terms *Manhattan instance* and *Euclidean instance* to denote L_1 and L_2 instances, respectively. Furthermore, if p is clear from context, we write d instead of d_p.

A *tour construction heuristic* for the TSP incrementally constructs a tour and stops as soon as a valid tour is created. Usually, a tour constructed by such a

heuristic is used as the initial solution 2-Opt starts with. A well-known class of tour construction heuristics for metric TSP instances are so-called *insertion heuristics*. These heuristics insert the vertices into the tour one after another, and every vertex is inserted between two consecutive vertices in the current tour where it fits best. To make this more precise, let T_i denote a subtour on a subset S_i of i vertices, and suppose $v \notin S_i$ is the next vertex to be inserted. If (x, y) denotes an edge in T_i that minimizes $\mathsf{d}(x, v) + \mathsf{d}(v, y) - \mathsf{d}(x, y)$, then the new tour T_{i+1} is obtained from T_i by deleting the edge (x, y) and adding the edges (x, v) and (v, y). Depending on the order in which the vertices are inserted into the tour, one distinguishes between several different insertion heuristics. Rosenkrantz et al. [RSI77] show an upper bound of $\lceil \log n \rceil + 1$ on the approximation factor of any insertion heuristic on metric TSP instances. Furthermore, they show that two variants which they call *nearest insertion* and *cheapest insertion* achieve an approximation ratio of 2 for metric TSP instances. The nearest insertion heuristic always inserts the vertex with the smallest distance to the current tour, and the cheapest insertion heuristic always inserts the vertex whose insertion leads to the cheapest tour T_{i+1}.

4.2 Exponential Lower Bounds

In this section, we answer Chandra, Karloff, and Tovey's question [CKT99] whether it is possible to construct TSP instances in the Euclidean plane on which 2-Opt can take an exponential number of steps. We present, for every $p \in \mathbb{N} \cup \{\infty\}$, a family of two-dimensional L_p instances with exponentially long sequences of improving 2-changes. In Section 4.2.1, we present our construction for the Euclidean plane, and in Section 4.2.2 we extend this construction to general L_p metrics.

4.2.1 Exponential Lower Bound for the Euclidean Plane

In Lueker's construction [Lue75] many of the 2-changes remove two edges that are far apart in the current tour in the sense that many vertices are visited between them, no matter of how the direction of the tour is chosen. Our construction differs significantly from the previous one as the 2-changes in our construction affect the tour only locally. The instances we construct are composed of gadgets of constant size. Each of these gadgets has a *zero state* and a *one state*, and there exists a sequence of improving 2-changes starting in the zero state and eventually leading to the one state. Let $G_0, \ldots, G_{n-1}$ denote these gadgets. If gadget G_i with $i > 0$ has reached state one, then it can be reset to its zero state by gadget G_{i-1}. The crucial property of our construction is that whenever a gadget G_{i-1} changes its state from zero to one, it resets gadget G_i twice. Hence, if in the initial tour, gadget G_0 is in its zero state and every other gadget is in state one, then for every i with $0 \leq i \leq n-1$, gadget G_i performs 2^i state changes from zero to one as, for $i > 0$, gadget G_i is reset 2^i times.

Every gadget is composed of 2 subgadgets, which we refer to as *blocks*. Each of these blocks consists of 4 vertices that are consecutively visited in the tour. For $i \in \{0, \ldots, n-1\}$ and $j \in [2]$, let $\mathcal{B}_1^i$ and $\mathcal{B}_2^i$ denote the blocks of gadget G_i and let A_j^i, B_j^i, C_j^i, and D_j^i denote the four points $\mathcal{B}_j^i$ consist of. If one ignores certain intermediate configurations that arise when one gadget resets another one,

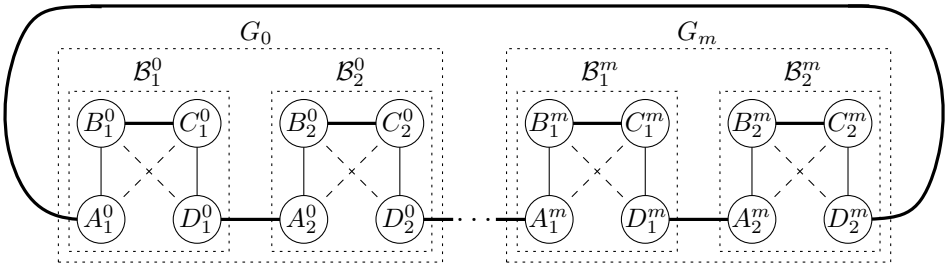


Figure 4.2.1: In the illustration, we use m to denote $n-1$. Every tour that occurs in the sequence of 2-changes contains the thick edges. For each block, either both solid or both dashed edges are contained. In the former case the block is in its short state; in the latter case the block is in its long state.

our construction ensures the following properties: The points A_j^i, B_j^i, C_j^i, and D_j^i are always consecutive in the tour, the edge between B_j^i and C_j^i is contained in every tour, and B_j^i and C_j^i are always the inner points of the block. That is, if one excludes the intermediate configurations, only the configurations $A_j^i B_j^i C_j^i D_j^i$ and $A_j^i C_j^i B_j^i D_j^i$ occur during the sequence of 2-changes. Observe that the change from one of these configurations to the other corresponds to a single 2-change in which the edges $A_j^i B_j^i$ and $C_j^i D_j^i$ are replaced by the edges $A_j^i C_j^i$ and $B_j^i D_j^i$, or vice versa. In the following, we assume that the sum $\mathsf{d}(A_j^i, B_j^i) + \mathsf{d}(C_j^i, D_j^i)$ is strictly smaller than the sum $\mathsf{d}(A_j^i, C_j^i) + \mathsf{d}(B_j^i, D_j^i)$, and we refer to the configuration $A_j^i B_j^i C_j^i D_j^i$ as the *short state* of the block and to the configuration $A_j^i C_j^i B_j^i D_j^i$ as the *long state*. Another property of our construction is that neither the order in which the blocks are visited nor the order of the gadgets is changed during the sequence of 2-changes. Again with the exception of the intermediate configurations, the order in which the blocks are visited is $\mathcal{B}_1^0 \mathcal{B}_2^0 \mathcal{B}_1^1 \mathcal{B}_2^1 \dots \mathcal{B}_1^{n-1} \mathcal{B}_2^{n-1}$ (see Figure 4.2.1).

Due to the aforementioned properties, we can describe every non-intermediate tour that occurs during the sequence of 2-changes completely by specifying for every block if it is in its short state or in its long state. In the following, we denote the state of a gadget G_i by a pair (x_1, x_2) with $x_j \in \{S, L\}$, meaning that block $\mathcal{B}_j^i$ is in its short state if and only if $x_j = S$. Since every gadget consists of two blocks, there are four possible states for each gadget. However, only three of them appear in the sequence of 2-changes, namely (L, L), (S, L), and (S, S). We call state (L, L) the *zero state* and state (S, S) the *one state*. In order to guarantee the existence of an exponentially long sequence of 2-changes, the gadgets we construct possess the following property.

Property 4.2.1. *If, for $i \in \{0, \dots, n-2\}$, gadget G_i is in state (L, L) or (S, L) and gadget G_{i+1} is in state (S, S), then there exists a sequence of seven consecutive 2-changes terminating with gadget G_i being in state (S, L) or (S, S), respectively, and gadget G_{i+1} in state (L, L). In this sequence only edges of and between the gadgets G_i and G_{i+1} are involved.*

If this property is satisfied and if in the initial tour gadget G_0 is in its zero state (L, L) and every other gadget is in its one state (S, S), then there exists an exponentially long sequence of 2-changes in which gadget G_i changes 2^i times from state zero to state one, as the following lemma shows.

Lemma 4.2.2. *If, for $i \in \{0, \dots, n-2\}$, gadget G_i is in the zero state (L, L) and all gadgets G_j with $j > i$ are in the one state (S, S), then there exists a sequence of $2^{n+3-i} - 14$ consecutive 2-changes in which only edges of and between the gadgets G_j with $j \geq i$ are involved and that terminates in a state in which all gadgets G_j with $j \geq i$ are in the one state.*

Proof. We prove the lemma by induction on i. If gadget G_{n-1} is in state (L, L), then it can change its state with two 2-changes to (S, S) without affecting the other gadgets. Hence, the lemma is true for $i = n - 1$. Now assume that the lemma is true for $i + 1$ and consider a state in which gadget G_i is in state (L, L) and all gadgets G_j with $j > i$ are in state (S, S). Due to Property 4.2.1, there exists a sequence of seven consecutive 2-changes in which only edges of and between G_i and G_{i+1} are involved terminating with G_i being in state (S, L) and G_{i+1} being in state (L, L). By the induction hypothesis there exists a sequence of $2^{n+2-i} - 14$ 2-changes after which all gadgets G_j with $j > i$ are in state (S, S). Then, due to Property 4.2.1, there exists a sequence of seven consecutive 2-changes in which only G_i changes its state from (S, L) to (S, S) while resetting gadget G_{i+1} again from (S, S) to (L, L). Hence, we can apply the induction hypothesis again, yielding that after another $2^{n+2-i} - 14$ 2-changes all gadgets G_j with $j \geq i$ are in state (S, S). This concludes the proof as the number of 2-changes performed is $14 + 2(2^{n+2-i} - 14) = 2^{n+3-i} - 14$. □

In particular, this implies that, given Property 4.2.1, one can construct instances consisting of $2n$ gadgets, i.e., $16n$ points, whose transition graphs contain paths of length $2^{2n+3} - 14 > 2^{n+4} - 22$, as desired in Theorem 1.3.1.

Detailed Description of the Sequence of Steps

Now we describe in detail how a sequence of 2-changes satisfying Property 4.2.1 can be constructed. First, we assume that gadget G_i is in state (S, L) and that gadget G_{i+1} is in state (S, S). Under this assumption, there are three consecutive blocks, namely $\mathcal{B}_2^i$, $\mathcal{B}_1^{i+1}$, and $\mathcal{B}_2^{i+1}$, such that the leftmost one is in its long state, and the other blocks are in their short states. We need to find a sequence of 2-changes in which only edges of and between these three blocks are involved and after which the first block is in its short state and the other blocks are in their long states. Remember that when the edges $\{u_1, u_2\}$ and $\{v_1, v_2\}$ are removed from the tour and the vertices appear in the order u_1, u_2, v_1, v_2 in the current tour, then the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ are added to the tour and the subtour between u_1 and v_2 is visited in reverse order. If, e.g., the current tour corresponds to the permutation $(1, 2, 3, 4, 5, 6, 7)$ and the edges $\{1, 2\}$ and $\{5, 6\}$ are removed, then the new tour is $(1, 5, 4, 3, 2, 6, 7)$. The following sequence of 2-changes has the desired properties. Brackets indicate the edges that are removed from the tour.

1)	$[A_2^i$	$C_2^i]$	B_2^i	D_2^i	A_1^{i+1}	B_1^{i+1}	C_1^{i+1}	D_1^{i+1}	A_2^{i+1}	B_2^{i+1}	$[C_2^{i+1}$	$D_2^{i+1}]$
2)	A_2^i	C_2^{i+1}	$[B_2^{i+1}$	$A_2^{i+1}]$	D_1^{i+1}	C_1^{i+1}	B_1^{i+1}	A_1^{i+1}	$[D_2^i$	$B_2^i]$	C_2^i	D_2^{i+1}
3)	A_2^i	C_2^{i+1}	$[B_2^{i+1}$	$D_2^i]$	A_1^{i+1}	B_1^{i+1}	$[C_1^{i+1}$	$D_1^{i+1}]$	A_2^{i+1}	B_2^i	C_2^i	D_2^{i+1}
4)	A_2^i	C_2^{i+1}	B_2^{i+1}	C_1^{i+1}	$[B_1^{i+1}$	$A_1^{i+1}]$	D_2^i	D_1^{i+1}	A_2^{i+1}	B_2^i	$[C_2^i$	$D_2^{i+1}]$
5)	$[A_2^i$	$C_2^{i+1}]$	B_2^{i+1}	C_1^{i+1}	B_1^{i+1}	C_2^i	$[B_2^i$	$A_2^{i+1}]$	D_1^{i+1}	D_2^i	A_1^{i+1}	D_2^{i+1}
6)	A_2^i	B_2^i	C_2^i	B_1^{i+1}	$[C_1^{i+1}$	$B_2^{i+1}]$	C_2^{i+1}	A_2^{i+1}	D_1^{i+1}	D_2^i	$[A_1^{i+1}$	$D_2^{i+1}]$
7)	A_2^i	B_2^i	$[C_2^i$	$B_1^{i+1}]$	C_1^{i+1}	A_1^{i+1}	$[D_2^i$	$D_1^{i+1}]$	A_2^{i+1}	C_2^{i+1}	B_2^{i+1}	D_2^{i+1}
	A_2^i	B_2^i	C_2^i	D_2^i	A_1^{i+1}	C_1^{i+1}	B_1^{i+1}	D_1^{i+1}	A_2^{i+1}	C_2^{i+1}	B_2^{i+1}	D_2^{i+1}

If gadget G_i is in state (L, L) instead of state (S, L), a sequence of steps that satisfies Property 4.2.1 can be constructed analogously. Additionally, one has to take into account that the three involved blocks $\mathcal{B}_1^i$, $\mathcal{B}_1^{i+1}$, and $\mathcal{B}_2^{i+1}$ are not consecutive in the tour but that block $\mathcal{B}_2^i$ lies between them. However, one can easily verify that this block is not affected by the sequence of 2-changes, as after the seven 2-changes have been performed, the block is in the same state and at the same position as before.

Embedding the Construction into the Euclidean Plane

The only missing step in the proof of Theorem 1.3.1 for the Euclidean plane is to find points such that all of the 2-changes that we described in the previous section are improving. We specify the positions of the points of gadget G_{n-1} and give a rule how the points of gadget G_i can be derived when all points of gadget G_{i+1} have already been placed. In our construction it happens that different points have exactly the same coordinates. This is only for ease of notation; if one wants to obtain a TSP instance in which distinct points have distinct coordinates, one can slightly move these points without affecting the property that all 2-changes are improving. For $j \in [2]$, we choose $A_j^{n-1} = (0, 0)$, $B_j^{n-1} = (1, 0)$, $C_j^{n-1} = (-0.1, 1.4)$, and $D_j^{n-1} = (-1.1, 4.8)$. Then $A_j^{n-1}B_j^{n-1}C_j^{n-1}D_j^{n-1}$ is the short state because

$$\mathsf{d}(A_j^{n-1}, C_j^{n-1}) + \mathsf{d}(B_j^{n-1}, D_j^{n-1}) - (\mathsf{d}(A_j^{n-1}, B_j^{n-1}) + \mathsf{d}(C_j^{n-1}, D_j^{n-1})) > 2.09 \ .$$

We place the points of gadget G_i as follows (see Figure 4.2.2):

1. Start with the coordinates of the points of gadget G_{i+1}.
2. Rotate these points around the origin by $3\pi/2$.
3. Scale each coordinate with a factor of 3.
4. Translate the points by the vector $(-1.2, 0.1)$.

For $j \in [2]$, this yields $A_j^{n-2} = (-1.2, 0.1)$, $B_j^{n-2} = (-1.2, -2.9)$, $C_j^{n-2} = (3, 0.4)$, and $D_j^{n-2} = (13.2, 3.4)$.

From this construction it follows that each gadget is a scaled, rotated, and translated copy of gadget G_{n-1}. If one has a set of points in the Euclidean plane that admit certain improving 2-changes, then these 2-changes are still improving if one scales, rotates, and translates all points in the same manner. Hence, it suffices to show that the sequences in which gadget G_{n-2} resets gadget G_{n-1} from

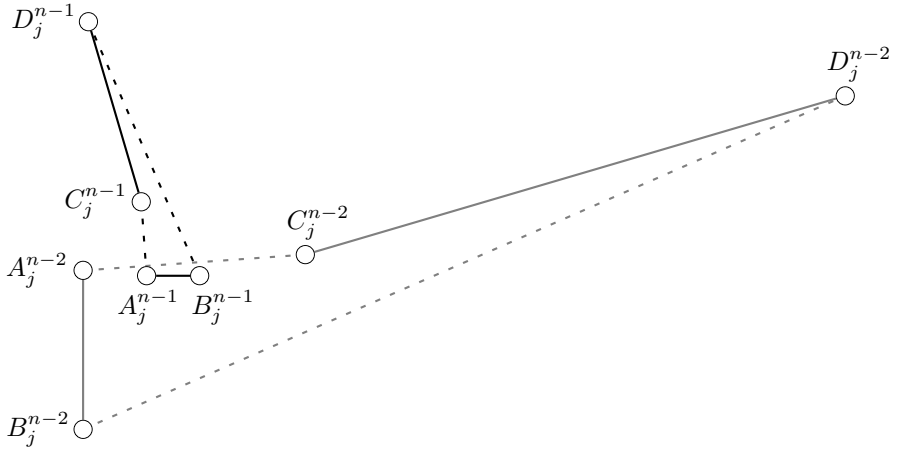


Figure 4.2.2: This illustration shows the points of the gadgets G_{n-1} and G_{n-2}. One can see that G_{n-2} is a scaled, rotated, and translated copy of G_{n-1}.

(S, S) to (L, L) are improving. There are two such sequences; in the first one, gadget G_{n-2} changes its state from (L, L) to (S, L), in the second one, gadget G_{n-2} changes its state from (S, L) to (S, S). Since the coordinates of the points in both blocks of gadget G_{n-2} are the same, the inequalities for both sequences are also identical. The improvements made by the steps in these sequences are bounded from below by 0.03, 0.91, 0.06, 0.05, 0.43, 0.06, and 0.53. This concludes the proof of Theorem 1.3.1 for the Euclidean plane as it shows that all 2-changes in Lemma 4.2.2 are improving.

4.2.2 Exponential Lower Bound for L_p Metrics

We were not able to find a set of points in the plane such that all 2-changes in Lemma 4.2.2 are improving with respect to the Manhattan metric. Therefore, we modify the construction of the gadgets and the sequence of 2-changes. Our construction for the Manhattan metric is based on the construction for the Euclidean plane, but it does not possess the property that every gadget resets its neighboring gadget twice. This property is only true for half of the gadgets. To be more precise, we construct two different types of gadgets which we call *reset gadgets* and *propagation gadgets*. Reset gadgets perform the same sequence of 2-changes as the gadgets that we constructed for the Euclidean plane. Propagation gadgets also have the same structure as the gadgets for the Euclidean plane, but when such a gadget changes its state from (L, L) to (S, S), it resets its neighboring gadget only once. Due to this relaxed requirement it is possible to find points in the Manhattan plane whose distances satisfy all necessary inequalities. Instead of n gadgets, our constructions consists of $2n$ gadgets, namely n propagation gadgets $G_0^P, \ldots, G_{n-1}^P$ and n reset gadgets $G_0^R, \ldots, G_{n-1}^R$. The order in which these gadgets appear in the tour is $G_0^P G_0^R G_1^P G_1^R \ldots G_{n-1}^P G_{n-1}^R$.

As before, every gadget consists of two blocks and the order in which the blocks and the gadgets are visited does not change during the sequence of 2-changes. Consider a reset gadget G_i^R and its neighboring propagation gadget G_{i+1}^P. Then Property 4.2.1 is still satisfied. That is, if G_i^R is in state (L, L) or (S, L) and G_{i+1}^P is in state (S, S), then there exists a sequence of seven consecutive 2-changes

resetting gadget G^P_{i+1} to state (L, L) and leaving gadget G^R_i in state (S, L) or (S, S), respectively. The situation is different for a propagation gadget G^P_i and its neighboring reset gadget G^R_i. In this case, if G^P_i is in state (L, L), it first changes its state with a single 2-change to (S, L). After that, gadget G^P_i changes its state to (S, S) while resetting gadget G^R_i from state (S, S) to state (L, L) by a sequence of seven consecutive 2-changes. In both cases, the sequences of 2-changes in which one block changes from its long to its short state while resetting two blocks of the neighboring gadget from their short to their long states are chosen analogously to the ones for the Euclidean plane described in Section 4.2.1.

In the initial tour, only gadget G^P_0 is in state (L, L) and every other gadget is in state (S, S). With similar arguments as for the Euclidean plane, we can show that gadget G^R_i is reset from its one state (S, S) to its zero state (L, L) 2^i times and that the total number of steps is $2^{n+4} - 22$.

Embedding the Construction into the Manhattan Plane

Similar to the construction in the Euclidean plane, the points in both blocks of a reset gadget G^R_i have the same coordinates. Also in this case one can slightly move all the points without affecting the inequalities if one wants distinct coordinates for distinct points. Again, we choose points for the gadgets G^P_{n-1} and G^R_{n-1} and describe how the points of the gadgets G^P_i and G^R_i can be chosen when the points of the gadgets G^P_{i+1} and G^R_{i+1} are already chosen. For $j \in [2]$, we choose $A^{n-1}_{R,j} = (0, 1)$, $B^{n-1}_{R,j} = (0, 0)$, $C^{n-1}_{R,j} = (-0.7, 0.1)$, and $D^{n-1}_{R,j} = (-1.2, 0.08)$. Furthermore, we choose $A^{n-1}_{P,1} = (-2, 1.8)$, $B^{n-1}_{P,1} = (-3.3, 2.8)$, $C^{n-1}_{P,1} = (-1.3, 1.4)$, $D^{n-1}_{P,1} = (1.5, 0.9)$, $A^{n-1}_{P,2} = (-0.7, 1.6)$, $B^{n-1}_{P,2} = (-1.5, 1.2)$, $C^{n-1}_{P,2} = (1.9, -1.5)$, and $D^{n-1}_{P,2} = (-0.8, -1.1)$.

Before we describe how the points of the other gadgets are chosen, we first show that the 2-changes within and between the gadgets G^P_{n-1} and G^R_{n-1} are improving. For $j \in [2]$, $A^{n-1}_{R,j} B^{n-1}_{R,j} C^{n-1}_{R,j} D^{n-1}_{R,j}$ is the short state because

$$\mathsf{d}(A^{n-1}_{R,j}, C^{n-1}_{R,j}) + \mathsf{d}(B^{n-1}_{R,j}, D^{n-1}_{R,j}) - (\mathsf{d}(A^{n-1}_{R,j}, B^{n-1}_{R,j}) + \mathsf{d}(C^{n-1}_{R,j}, D^{n-1}_{R,j})) = 1.36 \ .$$

Also the 2-change in which G^P_{n-1} changes its state from (L, L) to (S, L) is improving because

$$\mathsf{d}(A^{n-1}_{P,1}, C^{n-1}_{P,1}) + \mathsf{d}(B^{n-1}_{P,1}, D^{n-1}_{P,1}) - (\mathsf{d}(A^{n-1}_{P,1}, B^{n-1}_{P,1}) + \mathsf{d}(C^{n-1}_{P,1}, D^{n-1}_{P,1})) = 2.2 \ .$$

The improvements made by the 2-changes in the sequence in which G^P_{n-1} changes its state from (S, L) to (S, S) while resetting G^R_{n-1} are 0.04, 0.4, 0.04, 0.16, 0.4, 0.04, and 0.6.

Again, our construction possesses the property that each pair of gadgets G^P_i and G^R_i is a scaled and translated version of the pair G^P_{n-1} and G^R_{n-1}. Since we have relaxed the requirements for the gadgets, we do not even need rotations here. We place the points of G^P_i and G^R_i as follows:

1. Start with the coordinates specified for the points of gadgets G^P_{i+1} and G^R_{i+1}.
2. Scale each coordinate with a factor of 7.7.
3. Translate the points by the vector $(1.93, 0.3)$.

For $j \in [2]$, this yields $A_{R,j}^{n-2} = (1.93, 8)$, $B_{R,j}^{n-2} = (1.93, 0.3)$, $C_{R,j}^{n-2} = (-3.46, 1.07)$, and $D_{R,j}^{n-2} = (-7.31, 0.916)$. Similar to our construction for the Euclidean plane, it suffices to show that the sequences in which gadget G_{n-2}^R resets gadget G_{n-1}^P from (S, S) to (L, L) are improving. As the coordinates of the points in the two blocks of gadget G_{n-2}^R are the same, the inequalities for both sequences are also identical. The improvements made by the steps in these sequences are 1.06, 1.032, 0.168, 1.14, 0.06, 0.4, and 0.012. This concludes the proof of Theorem 1.3.1 for the Manhattan metric as it shows that all 2-changes are improving.

Let us remark that this also implies Theorem 1.3.1 for the L_∞ metric because distances w.r.t. the L_∞ metric coincide with distances w.r.t. the Manhattan metric if one rotates all points by $\pi/4$ around the origin and scales every coordinate with $1/\sqrt{2}$.

Embedding the Construction into General L_p Metrics

It is also possible to embed our construction into the L_p metric for $p \geq 3$. For $j \in [2]$, we choose $A_{R,j}^{n-1} = (0, 1)$, $B_{R,j}^{n-1} = (0, 0)$, $C_{R,j}^{n-1} = (3.5, 3.7)$, and $D_{R,j}^{n-1} = (7.8, -3.2)$. Moreover, we choose $A_{P,1}^{n-1} = (-2.5, -2.4)$, $B_{P,1}^{n-1} = (-4.7, -7.3)$, $C_{P,1}^{n-1} = (-8.6, -4.6)$, $D_{P,1}^{n-1} = (3.7, 9.8)$, $A_{P,2}^{n-1} = (3.2, 2)$, $B_{P,2}^{n-1} = (7.2, 7.2)$, $C_{P,2}^{n-1} = (-6.5, -1.6)$, and $D_{P,2}^{n-1} = (-1.5, -7.1)$. We place the points of G_i^P and G_i^R as follows:

1. Start with the coordinates specified for the points of gadgets G_{i+1}^P and G_{i+1}^R.
2. Rotate these points around the origin by π.
3. Scale each coordinate with a factor of 7.8.
4. Translate the points by the vector $(7.2, 5.3)$.

It can be calculated that the distances of these points when measured according to the L_p metric for any $p \geq 3$ satisfy all necessary inequalities.

4.3 Expected Number of 2-Changes

We analyze the expected number of 2-changes on random d-dimensional Manhattan and Euclidean instances, for an arbitrary constant dimension $d \geq 2$, and on general TSP instances. The previous results on the expected number of 2-changes due to Kern [Ker89] and Chandra, Karloff, and Tovey [CKT99] are based on the analysis of the improvement made by the smallest improving 2-change. If the smallest improvement is not too small, then the number of improvements cannot be large. In our analyses for the Manhattan and the Euclidean metric, we consider not only a single step but certain pairs of steps. We show that the smallest improvement made by any such pair is typically much larger than the improvement made by a single step, which yields our improved bounds. Our approach is not restricted to pairs of steps. One could also consider sequences of steps of length k for any small enough k. In fact, for general ϕ-perturbed graphs with m edges, we consider sequences of length $\sqrt{\log m}$. The reason why we can analyze longer sequences for general graphs is that these inputs possess more randomness than ϕ-perturbed Manhattan and Euclidean instances because every edge length is a

random variable that is independent of the other edge lengths. Hence, the analysis for general ϕ-perturbed graphs demonstrates the limits of our approach under optimal conditions. For Manhattan and Euclidean instances, the gain of considering longer sequences is small due to the dependencies between the edge lengths.

4.3.1 Manhattan Instances

In this section, we analyze the expected number of 2-changes on ϕ-perturbed Manhattan instances. First we prove a weaker bound than the one in Theorem 1.3.2. The proof of this weaker bound illustrates our approach and reveals the problems one has to tackle in order to improve the upper bounds. It is solely based on an analysis of the smallest improvement made by any of the possible 2-Opt steps. If with high probability every 2-Opt step decreases the tour length by a polynomially large amount, then with high probability only polynomially many 2-Opt steps are possible before a local optimum is reached.

Theorem 4.3.1. *Starting with an arbitrary tour, the expected number of steps performed by 2-Opt on ϕ-perturbed Manhattan instances with n vertices is $O(n^6 \cdot \log n \cdot \phi)$.*

Proof. In order to prove the desired bound on the expected convergence time, we only need two simple observations. First, the initial tour can have length at most dn as the number of edges is n and every edge has length at most d. And second, every 2-Opt step decreases the length of the tour by a polynomially large amount with high probability. The latter can be shown by a union bound over all possible 2-Opt steps. Consider a fixed 2-Opt step S, let e_1 and e_2 denote the edges removed from the tour in step S, and let e_3 and e_4 denote the edges added to the tour. Then the improvement $\Delta(S)$ of step S can be written as

$$\Delta(S) = \mathsf{d}(e_1) + \mathsf{d}(e_2) - \mathsf{d}(e_3) - \mathsf{d}(e_4) \ . \tag{4.3.1}$$

Without loss of generality let $e_1 = (v_1, v_2)$ be the edge between the vertices v_1 and v_2, and let $e_2 = (v_3, v_4)$, $e_3 = (v_1, v_3)$, and $e_4 = (v_2, v_4)$. Furthermore, for $i \in \{1, \dots 4\}$, let $x^i \in \mathbb{R}^d$ denote the coordinates of vertex v_i. Then the improvement $\Delta(S)$ of step S can be written as

$$\Delta(S) = \sum_{i=1}^{d} \left(|x_i^1 - x_i^2| + |x_i^3 - x_i^4| - |x_i^1 - x_i^3| - |x_i^2 - x_i^4| \right) \ .$$

Depending on the order of the coordinates, $\Delta(S)$ can be written as linear combination of the coordinates. If, e.g., for all $i \in [d]$, $x_i^1 \geq x_i^2 \geq x_i^3 \geq x_i^4$, then the improvement $\Delta(S)$ can be written as $\sum_{i=1}^{d}(-2x_i^2 + 2x_i^3)$. There are $(4!)^d$ such orders and each one gives rise to a linear combination of the x_i^j's with integer coefficients. For each of these linear combinations, the probability that it takes a value in the interval $(0, \varepsilon]$ is bounded from above by $\varepsilon\phi$, following, e.g., from Lemma B.3.1. Since $\Delta(S)$ can only take a value in the interval $(0, \varepsilon]$ if one of the linear combinations takes a value in this interval, the probability of the event $\Delta(S) \in (0, \varepsilon]$ can be upper bounded by $(4!)^d \varepsilon\phi$.

Let $\Delta_{\min}$ denote the improvement of the smallest improving 2-Opt step S, i.e., $\Delta_{\min} = \min\{\Delta(S) \mid \Delta(S) > 0\}$. We can estimate $\Delta_{\min}$ by a union bound, yielding

$$\mathbf{Pr}[\Delta_{\min} \leq \varepsilon] \leq (4!)^d \varepsilon n^4 \phi$$

as there are at most n^4 different 2-Opt steps. Let T denote the random variable describing the number of 2-Opt steps before a local optimum is reached. Observe that T can only exceed a given number t if the smallest improvement $\Delta_{\min}$ is less than dn/t, and hence

$$\mathbf{Pr}[T \geq t] \leq \mathbf{Pr}\left[\Delta_{\min} \leq \frac{dn}{t}\right] \leq \frac{d(4!)^d n^5 \phi}{t} \ .$$

Since there are at most $(n!)$ different TSP tours and none of these tours can appear twice during the local search, T is always bounded by $(n!)$. Altogether, we can bound the expected value of T by

$$\mathbf{E}[T] = \sum_{t=1}^{n!} \mathbf{Pr}[T \geq t] \leq \sum_{t=1}^{n!} \frac{d(4!)^d n^5 \phi}{t} \ .$$

Since we assumed the dimension d to be a constant, bounding the n-th harmonic number by $\ln(n) + 1$ and using $\ln(n!) = O(n \log n)$ yields

$$\mathbf{E}[T] = O(n^6 \cdot \log n \cdot \phi) \ . \qquad \square$$

The bound in Theorem 4.3.1 is only based on the smallest improvement $\Delta_{\min}$ made by any of the 2-Opt steps. Intuitively, this is too pessimistic since most of the steps performed by 2-Opt yield a larger improvement than $\Delta_{\min}$. In particular, two consecutive steps yield an improvement of at least $\Delta_{\min}$ plus the improvement $\Delta'_{\min}$ of the second smallest step. This observation alone, however, does not suffice to improve the bound substantially. Instead, we regroup the 2-changes to pairs such that each pair of 2-changes is *linked* by an edge, i.e., one edge added to the tour in the first 2-change is removed from the tour in the second 2-change, and we analyze the smallest improvement made by any pair of linked 2-Opt steps. Obviously, this improvement is at least $\Delta_{\min} + \Delta'_{\min}$ but one can hope that it is much larger because it is unlikely that the 2-change that yields the smallest improvement and the 2-change that yields the second smallest improvement form a pair of linked steps. We show that this is indeed the case and use this result to prove the bound on the expected length of the longest path in the transition graph of 2-Opt on ϕ-perturbed Manhattan instances claimed in Theorem 1.3.2.

Construction of Pairs of Linked 2-Changes

Consider an arbitrary sequence of consecutive 2-changes of length t. The following lemma guarantees that the number of disjoint linked pairs of 2-changes in every such sequence increases linearly with the length t.

Lemma 4.3.2. *In every sequence of t consecutive 2-changes, the number of disjoint pairs of 2-changes that are linked by an edge, i.e., pairs such that there exists an edge added to the tour in the first 2-change of the pair and removed from the tour in the second 2-change of the pair, is at least $t/3 - n(n-1)/4$.*

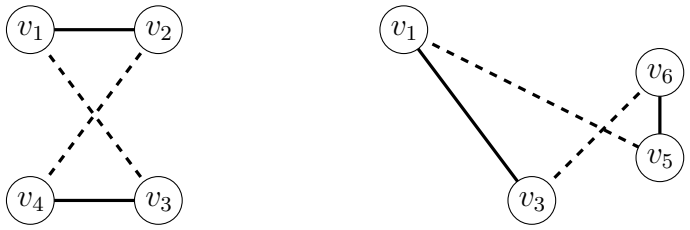


Figure 4.3.1: A pair of type 1.

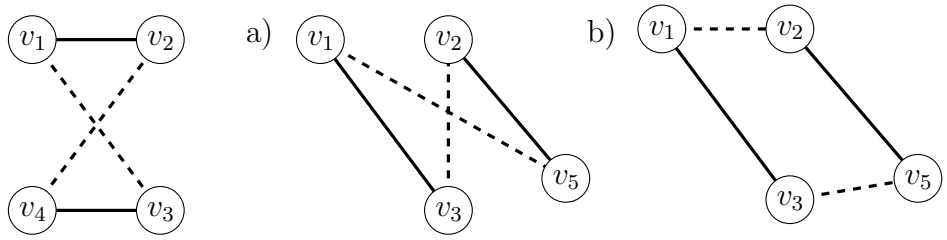


Figure 4.3.2: Pairs of type 2.

Proof. Let $S_1, \ldots, S_t$ denote an arbitrary sequence of consecutive 2-changes. The sequence is processed step by step and a list $\mathcal{L}$ of disjoint linked pairs of 2-changes is created. Assume that the 2-changes $S_1, \ldots, S_{i-1}$ have already been processed and that now 2-change S_i has to be processed. Assume further that in step S_i the edges e_1 and e_2 are exchanged with the edges e_3 and e_4. Let j denote the smallest index with $j > i$ such that edge e_3 is removed from the tour in step S_j if such a step exists, and let j' denote the smallest index with $j' > i$ such that edge e_4 is removed from the tour in step $S_{j'}$ if such a step exists. If the index j is defined, the pair (S_i, S_j) is added to the constructed list $\mathcal{L}$. If the index j is not defined but the index j' is defined, the pair $(S_i, S_{j'})$ is added to the constructed list $\mathcal{L}$. After that, both steps S_j and $S_{j'}$ (if defined) are removed from the sequence of 2-changes, that is, they are not processed in the following in order to guarantee disjointness of the pairs in $\mathcal{L}$.

If one 2-change is processed, it excludes at most two other 2-changes from being processed. Hence, the number of pairs added to $\mathcal{L}$ is at least $t/3 - n(n-1)/4$ because there can be at most $\lfloor n(n-1)/4 \rfloor$ steps S_i for which neither j nor j' is defined. □

Consider a fixed pair of 2-changes linked by an edge. Without loss of generality assume that in the first step the edges $\{v_1, v_2\}$ and $\{v_3, v_4\}$ are exchanged with the edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$, for distinct vertices $v_1, \ldots, v_4$. Also without loss of generality assume that in the second step the edges $\{v_1, v_3\}$ and $\{v_5, v_6\}$ are exchanged with the edges $\{v_1, v_5\}$ and $\{v_3, v_6\}$. However, note that the vertices v_5 and v_6 are not necessarily distinct from the vertices v_2 and v_4. We distinguish between three different types of pairs.

1. $|\{v_2, v_4\} \cap \{v_5, v_6\}| = 0$. This case is illustrated in Figure 4.3.1.
2. $|\{v_2, v_4\} \cap \{v_5, v_6\}| = 1$. We can assume w.l.o.g. that $v_2 \in \{v_5, v_6\}$. We have to distinguish between two subcases: a) The edges $\{v_1, v_5\}$ and $\{v_2, v_3\}$

are added to the tour in the second step. b) The edges $\{v_1, v_2\}$ and $\{v_3, v_5\}$ are added to the tour in the second step. These cases are illustrated in Figure 4.3.2.

3. $|\{v_2, v_4\} \cap \{v_5, v_6\}| = 2$. The case $v_2 = v_5$ and $v_4 = v_6$ cannot appear as it would imply that the tour is not changed by performing the considered pair of steps. Hence, for pairs of this type, we must have $v_2 = v_6$ and $v_4 = v_5$.

When distances are measured according to the Euclidean metric, pairs of type 3 result in vast dependencies and hence the probability that there exists a pair of this type in which both steps are improvements by at most ε w.r.t. the Euclidean metric cannot be bounded appropriately. In order to reduce the number of cases we have to consider and in order to prepare the analysis of ϕ-perturbed Euclidean instances, we exclude pairs of type 3 from our probabilistic analysis by leaving out all pairs of type 3 when constructing the list $\mathcal{L}$ in the proof of Lemma 4.3.2.

We only need to show that there are always enough pairs of type 1 or 2. Consider two steps S_i and S_j with $i < j$ that form a pair of type 3. Assume that in step S_i the edges $\{v_1, v_2\}$ and $\{v_3, v_4\}$ are replaced by the edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$, and that in step S_j these edges are replaced by the edges $\{v_1, v_4\}$ and $\{v_2, v_3\}$. Now consider the next step S_l with $l > j$ in which the edge $\{v_1, v_4\}$ is removed from the tour if such a step exists and the next step $S_{l'}$ with $l' > j$ in which the edge $\{v_2, v_3\}$ is removed from the tour if such a step exists. Observe that neither (S_j, S_l) nor $(S_j, S_{l'})$ can be a pair of type 3 because otherwise the improvement of one of the steps S_i, S_j, and S_l, or $S_{l'}$, respectively, must be negative. In particular, we must have $l \neq l'$.

If we encounter a pair (S_i, S_j) of type 3 in the construction of the list $\mathcal{L}$, we mark step S_i as being processed without adding a pair of 2-changes to $\mathcal{L}$ and without removing S_j from the sequence of steps to be processed. Let x denote the number of pairs of type 3 that we encounter during the construction of the list $\mathcal{L}$. Our argument above shows that the number of pairs of type 1 or 2 that are added to $\mathcal{L}$ is at least $x - n(n-1)/4$. This implies $t \geq 2x - n(n-1)/4$ and $x \leq t/2 + n(n-1)/8$. Hence, the number of relevant steps reduces from t to $t' = t - x \geq t/2 - n(n-1)/8$. Using this estimate in Lemma 4.3.2 yields the following lemma.

Lemma 4.3.3. *In every sequence of t consecutive 2-changes the number of disjoint pairs of 2-changes of type 1 or 2 is at least $t/6 - 7n(n-1)/24$.* □

Analysis of Pairs of Linked 2-Changes

The following lemma gives a bound on the probability that there exists a pair of type 1 or 2 in which both steps are small improvements.

Lemma 4.3.4. *In a ϕ-perturbed Manhattan instance with n vertices, the probability that there exists a pair of type 1 or type 2 in which both 2-changes are improvements by at most ε is bounded by $O(n^6 \cdot \varepsilon^2 \cdot \phi^2)$.*

Proof. First, we consider pairs of type 1. We assume that in the first step the edges $\{v_1, v_2\}$ and $\{v_3, v_4\}$ are replaced by the edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$ and that in the second step the edges $\{v_1, v_3\}$ and $\{v_5, v_6\}$ are replaced by the edges

$\{v_1, v_5\}$ and $\{v_3, v_6\}$. For $i \in [6]$, let $x^i \in \mathbb{R}^d$ denote the coordinates of vertex v_i. Furthermore, let Δ_1 denote the (possibly negative) improvement of the first step and let Δ_2 denote the (possibly negative) improvement of the second step. The random variables Δ_1 and Δ_2 can be written as

$$\Delta_1 = \sum_{i=1}^{d} |x_i^1 - x_i^2| + |x_i^3 - x_i^4| - |x_i^1 - x_i^3| - |x_i^2 - x_i^4|$$

and

$$\Delta_2 = \sum_{i=1}^{d} |x_i^1 - x_i^3| + |x_i^5 - x_i^6| - |x_i^1 - x_i^5| - |x_i^3 - x_i^6| \ .$$

For any fixed order of the coordinates, Δ_1 and Δ_2 can be expressed as linear combinations of the coordinates with integer coefficients. For $i \in [d]$, let σ_i denote an order of the coordinates $x_i^1, \ldots, x_i^6$, let $\sigma = (\sigma_1, \ldots, \sigma_d)$, and let Δ_1^σ and Δ_2^σ denote the corresponding linear combinations. We denote by $\mathcal{A}$ the event that both Δ_1 and Δ_2 take values in the interval $(0, \varepsilon]$, and we denote by $\mathcal{A}^\sigma$ the event that both linear combinations Δ_1^σ and Δ_2^σ take values in the interval $(0, \varepsilon]$. Obviously $\mathcal{A}$ can only occur if for at least one σ, the event $\mathcal{A}^\sigma$ occurs. Hence, we obtain

$$\mathbf{Pr}[\mathcal{A}] \leq \sum_{\sigma} \mathbf{Pr}[\mathcal{A}^\sigma] \ .$$

Since there are $(6!)^d$ different orders σ, which is constant for constant dimension d, it suffices to show that for every tuple of orders σ, the probability of the event $\mathcal{A}^\sigma$ is bounded from above by $O(\varepsilon^2\phi^2)$. Then a union bound over all possible pairs of linked 2-changes of type 1 yields the lemma for pairs of type 1.

We divide the set of possible pairs of linear combinations $(\Delta_1^\sigma, \Delta_2^\sigma)$ into three classes. We say that a pair of linear combinations belongs to class A if at least one of the linear combinations equals 0, we say that it belongs to class B if $\Delta_1^\sigma = -\Delta_2^\sigma$, and we say that it belongs to class C if Δ_1^σ and Δ_2^σ are linearly independent. For tuple of orders σ that yield pairs from class A or B, the event $\mathcal{A}^\sigma$ can never occur because in both cases the value of at least one linear combination is at most 0. For tuples σ that yield pairs from class C, we can apply Lemma B.3.1 from Appendix B.3, which shows that the probability of the event $\mathcal{A}^\sigma$ is bounded from above by $(\varepsilon\phi)^2$. Hence, we only need to show that every pair $(\Delta_1^\sigma, \Delta_2^\sigma)$ of linear combinations belongs either to class A, B, or C.

Consider a fixed tuple of orders $\sigma = (\sigma_1, \ldots, \sigma_d)$. We split Δ_1^σ and Δ_2^σ into d parts that correspond to the d dimensions. To be precise, for $j \in [2]$, we write $\Delta_j^\sigma = \sum_{i \in [d]} X_i^{\sigma_i, j}$, where $X_i^{\sigma_i, j}$ is a linear combination of the variables $x_i^1, \ldots, x_i^6$. For $i \in [d]$, we show that the pair of linear combinations $(X_i^{\sigma_i,1}, X_i^{\sigma_i,2})$ belongs either to class A, B, or C. This directly implies that also $(\Delta_1^\sigma, \Delta_2^\sigma)$ must belong to one of these classes.

Assume that the pair of linear combinations $(X_i^{\sigma_i,1}, X_i^{\sigma_i,2})$ is linearly dependent for the fixed order σ_i. Observe that this can only happen if $X_i^{\sigma_i,1}$ does not contain x_i^2 and x_i^4 and if $X_i^{\sigma_i,2}$ does not contain x_i^5 and x_i^6. The former can only happen if either $x_i^3 \geq x_i^4$, $x_i^2 \geq x_i^4$, and $x_i^2 \geq x_i^1$ or if $x_i^3 \leq x_i^4$, $x_i^2 \leq x_i^4$, and $x_i^2 \leq x_i^1$.

The latter can only happen if either $x_i^5 \geq x_i^6$, $x_i^3 \geq x_i^6$, and $x_i^5 \geq x_i^1$ or if $x_i^5 \leq x_i^6$, $x_i^3 \leq x_i^6$, and $x_i^5 \leq x_i^1$.

If one chooses the order such that x_i^2, x_i^4, x_i^5, and x_i^6 cancel out and such that $x_i^1 \geq x_i^3$, one can verify by a case distinction that $X_i^{\sigma_i,1} \in \{0, -2x_i^1 + 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, 2x_i^1 - 2x_i^3\}$. Hence, in this case the resulting pair of linear combinations belongs either to class A or B. Analogously, if one chooses the order such that x_i^2, x_i^4, x_i^5, and x_i^6 cancel out and such that $x_i^3 \geq x_i^1$, we have $X_i^{\sigma_i,1} \in \{0, 2x_i^1 - 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, -2x_i^1 + 2x_i^3\}$. Hence, also in this case, the pair of resulting linear combinations belongs either to class A or B.

With similar arguments we prove the lemma for pairs of type 2. We first prove the lemma for pairs of type 2 a). Using the same notations as for pairs of type 1, we can write the improvement Δ_2 as

$$\Delta_2 = \sum_{i \in [d]} |x_i^1 - x_i^3| + |x_i^2 - x_i^5| - |x_i^1 - x_i^5| - |x_i^2 - x_i^3| \ .$$

Again we show that, for every $i \in [d]$ and every order σ_i, the pair of linear combinations $(X_i^{\sigma_i,1}, X_i^{\sigma_i,2})$ belongs either to class A, B, or C. Assume that the pair is linearly dependent for the fixed order σ_i. Observe that this can only happen if $X_i^{\sigma_i,1}$ does not contain x_i^4 and if $X_i^{\sigma_i,2}$ does not contain x_i^5. The former can only happen if either $x_i^3 \geq x_i^4$ and $x_i^2 \geq x_i^4$ or if $x_i^3 \leq x_i^4$ and $x_i^2 \leq x_i^4$. The latter can only happen if either $x_i^2 \geq x_i^5$ and $x_i^1 \geq x_i^5$ or if $x_i^2 \leq x_i^5$ and $x_i^1 \leq x_i^5$.

If one chooses the order such that x_i^4 and x_i^5 cancel out and such that $x_i^1 \geq x_i^3$, one can verify by a case distinction that $X_i^{\sigma_i,1} \in \{0, -2x_i^1 + 2x_i^2, -2x_i^1 + 2x_i^3, -2x_i^2 + 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, 2x_i^1 - 2x_i^2, 2x_i^1 - 2x_i^3, 2x_i^2 - 2x_i^3\}$. Hence, under the assumption that the linear combinations $X_i^{\sigma_i,1}$ and $X_i^{\sigma_i,2}$ are linearly dependent, the pair of resulting linear combinations in the case $x_i^1 \geq x_i^3$ belongs either to class A or B. Analogously, if one chooses the order such that x_i^4 and x_i^5 cancel out and such that $x_i^3 \geq x_i^1$, we have $X_i^{\sigma_i,1} \in \{0, 2x_i^1 - 2x_i^2, 2x_i^1 - 2x_i^3, 2x_i^2 - 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, -2x_i^1 + 2x_i^2, -2x_i^1 + 2x_i^3, -2x_i^2 + 2x_i^3\}$. Hence, also in this case, the pair of resulting linear combinations belongs either to class A or B.

It remains to consider pairs of type 2 b). For these pairs, we can write Δ_2 as

$$\Delta_2 = \sum_{i=1}^{d} |x_i^1 - x_i^3| + |x_i^2 - x_i^5| - |x_i^1 - x_i^2| - |x_i^3 - x_i^5| \ .$$

Assume that the pair of linear combinations $(X_i^{\sigma_i,1}, X_i^{\sigma_i,2})$ is linearly dependent for the fixed order σ_i. Observe that this can only happen if $X_i^{\sigma_i,1}$ does not contain x_i^4 and if $X_i^{\sigma_i,2}$ does not contain x_i^5. As we have already seen for pairs of type 2 a), the former can only happen if either $x_i^3 \geq x_i^4$ and $x_i^2 \geq x_i^4$ or if $x_i^3 \leq x_i^4$ and $x_i^2 \leq x_i^4$. The latter can only happen if either $x_i^2 \geq x_i^5$ and $x_i^3 \geq x_i^5$ or if $x_i^2 \leq x_i^5$ and $x_i^3 \leq x_i^5$.

If one chooses the order such that x_i^4 and x_i^5 cancel out and such that $x_i^1 \geq x_i^3$, one can verify by a case distinction that $X_i^{\sigma_i,1} \in \{0, -2x_i^1 + 2x_i^2, -2x_i^1 + 2x_i^3, -2x_i^2 + 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, 2x_i^1 - 2x_i^2, 2x_i^1 - 2x_i^3, 2x_i^2 - 2x_i^3\}$. Hence, under the assumption that the linear combinations $X_i^{\sigma_i,1}$ and $X_i^{\sigma_i,2}$ are linearly dependent, the pair of resulting linear combinations belongs either to class A or B. If one chooses the order such that x_i^4 and x_i^5 cancel out and such that $x_i^3 \geq x_i^1$, one can verify by

a case distinction that $X_i^{\sigma_i,1} \in \{0, 2x_i^1 - 2x_i^2, 2x_i^1 - 2x_i^3, 2x_i^2 - 2x_i^3\}$ and $X_i^{\sigma_i,2} \in \{0, -2x_i^1 + 2x_i^2, -2x_i^1 + 2x_i^3, -2x_i^2 + 2x_i^3\}$. Hence, also in this case, the pair of resulting linear combinations belongs either to class A or B. □

Expected Number of 2-Changes

Based on Lemmas 4.3.3 and 4.3.4, we are now able to prove part a) of Theorem 1.3.2.

Proof of Theorem 1.3.2 a). Let T denote the random variable that describes the length of the longest path in the transition graph. If $T \geq t$, then there must exist a sequence $S_1, \ldots, S_t$ of t consecutive 2-changes in the transition graph. We start by identifying a set of linked pairs of type 1 and 2 in this sequence. Due to Lemma 4.3.3, we know that we can find at least $t/6 - 7n(n-1)/24$ such pairs. Let $\Delta^*_{\min}$ denote the smallest improvement made by any pair of improving 2-Opt steps of type 1 or 2. For $t > 2n^2$, we have $t/6 - 7n(n-1)/24 > t/48$ and hence due to Lemma 4.3.4,

$$\mathbf{Pr}[T \geq t] \leq \mathbf{Pr}\left[\Delta^*_{\min} \leq \frac{48dn}{t}\right] = O\left(\min\left\{\frac{n^8\phi^2}{t^2}, 1\right\}\right) \ .$$

Since T cannot exceed $(n!)$, this implies the following bound on the expected number of 2-changes:

$$\mathbf{E}[T] \leq 2n^2 + \sum_{t=2n^2+1}^{n!} O\left(\min\left\{\frac{n^8\phi^2}{t^2}, 1\right\}\right) = O(n^4 \cdot \phi) \ .$$

This concludes the proof of part a) of the theorem. □

Chandra, Karloff, and Tovey [CKT99] show that for every metric that is induced by a norm on $\mathbb{R}^d$, and for any set of n points in the unit hypercube $[0,1]^d$, the optimal tour visiting all n points has length $O(n^{(d-1)/d})$. Furthermore, every insertion heuristic finds an $O(\log n)$-approximation [RSI77]. Hence, if one starts with a solution computed by an insertion heuristic, the initial tour has length $O(n^{(d-1)/d} \cdot \log n)$. Using this observation yields part a) of Theorem 1.3.3.

Proof of Theorem 1.3.3 b). Since the initial tour has length $O(n^{(d-1)/d} \cdot \log n)$, we obtain for an appropriate constant c and $t > 2n^2$,

$$\begin{aligned}\mathbf{Pr}[T \geq t] &\leq \mathbf{Pr}\left[\Delta^*_{\min} \leq \frac{c \cdot n^{(d-1)/d} \cdot \log n}{t}\right] \\ &= O\left(\min\left\{\frac{n^{8-2/d} \cdot \log^2 n \cdot \phi^2}{t^2}, 1\right\}\right) \ .\end{aligned}$$

This yields

$$\mathbf{E}[T] \leq 2n^2 + \sum_{t=2n^2+1}^{n!} O\left(\min\left\{\frac{n^{8-2/d} \cdot \log^2 n \cdot \phi^2}{t^2}, 1\right\}\right) = O(n^{4-1/d} \cdot \log n \cdot \phi) \ .$$

□

4.3.2 Euclidean Instances

In this section, we analyze the expected number of 2-changes on ϕ-perturbed Euclidean instances. The analysis is similar to the analysis of Manhattan instances in the previous section, only Lemma 4.3.4 needs to be replaced by its equivalent version for the L_2 metric.

Lemma 4.3.5. *For ϕ-perturbed L_2 instances, the probability that there exists a pair of type 1 or type 2 in which both 2-changes are improvements by at most $\varepsilon \leq 1/2$ is bounded by $O(n^6 \cdot \phi^5 \cdot \varepsilon^2 \cdot \log^2(1/\varepsilon)) + O(n^5 \cdot \phi^4 \cdot \varepsilon^{3/2} \cdot \log(1/\varepsilon))$.*

The bound that this lemma provides is slightly weaker than its L_1 counterpart, and hence also the bound on the expected running time is slightly worse for L_2 instances. The crucial step to prove Lemma 4.3.5 is to gain a better understanding of the random variable that describes the improvement of a single fixed 2-change. In the next section, we analyze this random variable under several conditions, e.g., under the condition that the length of one of the involved edges is fixed. With the help of these results, pairs of linked 2-changes can easily be analyzed. Let us mention that our analysis of a single 2-change yields a bound of $O(n^7 \cdot \log^2(n) \cdot \phi^3)$ for the expected number of 2-changes. For Euclidean instances in which all points are distributed uniformly at random over the unit square, this bound already improves the best previously known bound of $O(n^{10} \cdot \log n)$.

Analysis of a Single 2-Change

We analyze a 2-change in which the edges $\{O, Q_1\}$ and $\{P, Q_2\}$ are exchanged with the edges $\{O, Q_2\}$ and $\{P, Q_1\}$ for some vertices O, P, Q_1, and Q_2. In the input model we consider, each of these points has a probability distribution over the unit hypercube according to which it is chosen. In this section, we consider a simplified random experiment in which O is chosen to be the origin and P, Q_1, and Q_2 are chosen independently and uniformly at random from a d-dimensional hyperball with radius $\sqrt{d}$ centered at the origin. In the next section, we argue that the analysis of this simplified random experiment helps to analyze the actual random experiment that occurs in the probabilistic input model.

Due to the rotational symmetry of the simplified model, we assume without loss of generality that P lies at position $(0, T)$ for some $T \geq 0$. For $i \in [2]$, Let Z_i denote the difference $\mathsf{d}(O, Q_i) - \mathsf{d}(P, Q_i)$. Then the improvement Δ of the 2-change can be expressed as $Z_1 - Z_2$. The random variables Z_1 and Z_2 are identically distributed, and they are independent if T is fixed. We denote by $f_{Z|T=\tau,R=r}$ the density of Z_1 and Z_2 under the conditions that $\mathsf{d}(O, Q_1) = r$ and $\mathsf{d}(O, Q_2) = r$, respectively, and $T = \tau$.

Lemma 4.3.6. *For $\tau, r \in (0, \sqrt{d}]$, and $z \in (-\tau, \min\{\tau, 2r - \tau\})$,*

$$f_{Z|T=\tau,R=r}(z) \leq \begin{cases} \sqrt{\frac{2}{\tau^2 - z^2}} & \text{if } r \geq \tau, \\ \sqrt{\frac{2}{(\tau+z)(2r-\tau-z)}} & \text{if } r \leq \tau. \end{cases}$$

For $z \notin [-\tau, \min\{\tau, 2r - \tau\}]$, the density $f_{Z|T=\tau,R=r}(z)$ is 0.

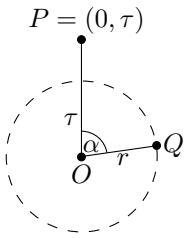


Figure 4.3.3: The random variable Z is defined as $r - \mathsf{d}(P, Q)$.

Proof. We denote by Z the random variable $\mathsf{d}(O, Q) - \mathsf{d}(P, Q)$, where Q is a point chosen uniformly at random from a d-dimensional hyperball with radius $\sqrt{d}$ centered at the origin. In the following, we assume that the plane spanned by the points O, P, and Q is fixed arbitrarily, and we consider the random experiment conditioned on the event that Q lies in this plane. To make the calculations simpler, we use polar coordinates to describe the location of Q. Since the radius $\mathsf{d}(O, Q) = r$ is given, the point Q is completely determined by the angle α between the y-axis and the line between O and Q (see Figure 4.3.3). Hence, the random variable Z can be written as

$$Z = r - \sqrt{r^2 + \tau^2 - 2r\tau \cdot \cos\alpha} \ .$$

It is easy to see that Z can only take values in the interval $[-\tau, \min\{\tau, 2r - \tau\}]$, and hence the density $f_{Z|T=\tau,R=r}(z)$ is 0 outside this interval.

Since Q is chosen uniformly at random, the angle α is chosen uniformly at random from the interval $[0, 2\pi)$. For symmetry reasons, we can assume that α is chosen uniformly from the interval $[0, \pi)$. When α is restricted to the interval $[0, \pi)$, then there exists a unique inverse function mapping Z to α, namely

$$\alpha(z) = \arccos\left(\frac{\tau^2 + 2zr - z^2}{2r\tau}\right) \ .$$

The density $f_{Z|T=\tau,R=r}$ can be expressed as

$$f_{Z|T=\tau,R=r}(z) = f_\alpha(\alpha(z)) \cdot \left|\frac{d}{dz}\alpha(z)\right| = -\frac{1}{\pi} \cdot \frac{d}{dz}\alpha(z) \ ,$$

where f_α denotes the density of α, i.e., the uniform density over $[0, \pi)$. For $|x| < 1$, the derivative of the arc cosine is

$$(\arccos(x))' = -\frac{1}{\sqrt{1 - x^2}} \ .$$

Hence, the derivative of $\alpha(z)$ equals

$$\frac{r - z}{r\tau} \cdot \frac{-1}{\sqrt{1 - \frac{(\tau^2 + 2zr - z^2)^2}{4r^2\tau^2}}} = \frac{2(z - r)}{\sqrt{4r^2\tau^2 - 4r^2z^2 - 4r\tau^2 z + 4rz^3 - \tau^4 + 2\tau^2 z^2 - z^4}} \ .$$

In order to prove the lemma, we distinguish between the cases $r \geq \tau$ and $r \leq \tau$.

First case: $r \geq \tau$.

In this case, it suffices to show

$$4r^2\tau^2 - 4r^2z^2 - 4r\tau^2 z + 4rz^3 - \tau^4 + 2\tau^2 z^2 - z^4 \geq 2(z - r)^2(\tau^2 - z^2) \ ,$$

which is implied by

$$\begin{aligned}&4r^2\tau^2 - 4r^2z^2 - 4r\tau^2 z + 4rz^3 - \tau^4 + 2\tau^2z^2 - z^4 - 2(z-r)^2(\tau^2 - z^2)\\ &= 2r^2(\tau^2 - z^2) - \tau^4 + z^4 \geq 2\tau^2(\tau^2 - z^2) - \tau^4 + z^4 = (\tau^2 - z^2)^2 \geq 0 \ .\end{aligned}$$

Second case: $r \leq \tau$**.**
In this case, it suffices to show

$$4r^2\tau^2 - 4r^2z^2 - 4r\tau^2 z + 4rz^3 - \tau^4 + 2\tau^2z^2 - z^4 \geq 2(z-r)^2(\tau + z)(2r - \tau - z) \ ,$$

which is equivalent to

$$\begin{aligned}&(-2r + z + \tau)(\tau + z)(z^2 + 2\tau z - 2rz + 2r^2 - \tau^2 - 2\tau r) \geq 0\\ \iff\ &z^2 + 2\tau z - 2rz + 2r^2 - \tau^2 - 2\tau r \leq 0 \ ,\end{aligned}$$

which follows from

$$\begin{aligned}&z^2 + 2\tau z - 2rz + 2r^2 - \tau^2 - 2\tau r\\ =&z^2 + 2z(\tau - r) + 2r^2 - \tau^2 - 2\tau r\\ \leq&(2r - \tau)^2 + 2(2r - \tau)(\tau - r) + 2r^2 - \tau^2 - 2\tau r\\ =&2(r^2 - \tau^2) \leq 0 \ .\end{aligned}$$

□

Based on Lemma 4.3.6, the density of the random variable $\Delta = Z_1 - Z_2$ under the conditions $R_1 := \mathsf{d}(O, Q_1) = r_1$, $R_2 := \mathsf{d}(O, Q_2) = r_2$, and $T := \mathsf{d}(O, P) = \tau$ can be computed as the convolution (cf. Appendix B.1) of the density $f_{Z|T=\tau,R=r}$ with itself.

Lemma 4.3.7. *Let $\tau, r_1, r_2 \in (0, \sqrt{d}]$, and let Z_1 and Z_2 be independent random variables drawn according to the densities $f_{Z|T=\tau,R=r_1}$ and $f_{Z|T=\tau,R=r_2}$, respectively. For every $\delta \in (0, 1/2]$ and a sufficiently large constant κ, the density $f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)$ of the random variable $\Delta = Z_1 - Z_2$ is bounded from above by*

$$\begin{cases} \frac{\kappa}{\tau} \cdot \ln \delta^{-1} & \text{if } \tau \leq r_1, \tau \leq r_2,\\ \frac{\kappa}{\sqrt{r_1 r_2}} \cdot \left(\ln \delta^{-1} + \ln |2(r_1 - r_2) - \delta|^{-1} + \kappa\right) & \text{if } r_1 \leq \tau, r_2 \leq \tau,\\ \frac{\kappa}{\sqrt{\tau r_1}} \cdot \ln \delta^{-1} & \text{if } r_1 \leq \tau \leq r_2,\\ \frac{\kappa}{\sqrt{\tau r_2}} \cdot \left(\ln \delta^{-1} + \ln |2(\tau - r_2) - \delta|^{-1} + \kappa\right) & \text{if } r_2 \leq \tau \leq r_1. \end{cases}$$

The simple but somewhat tedious calculation that yields Lemma 4.3.7 is deferred to Appendix B.4.1. In order to prove Lemma 4.3.5, we need bounds on the densities of the random variables Δ, Z_1, and Z_2 under certain conditions. We summarize these bounds in the following lemma.

Lemma 4.3.8. *Let $\tau, r \in (0, \sqrt{d}]$, $\delta \in (0, 1/2]$, and let κ denote a sufficiently large constant.*

a) For $i \in [2]$, the density of Δ under the condition $R_i = r$ is bounded by

$$f_{\Delta|R_i=r}(\delta) \leq \frac{\kappa}{\sqrt{r}} \cdot \ln \delta^{-1} \ .$$

b) The density of Δ under the condition $T = \tau$ is bounded by

$$f_{\Delta|T=\tau}(\delta) \leq \frac{\kappa}{\tau} \cdot \ln \delta^{-1} \ .$$

c) The density of Δ is bounded by

$$f_{\Delta}(\delta) \leq \kappa \cdot \ln \delta^{-1} \ .$$

d) For $i \in [2]$, the density of Z_i under the condition $T = \tau$ is bounded by

$$f_{Z_i|T=\tau}(z) \leq \frac{\kappa}{\sqrt{\tau^2 - z^2}}$$

if $|z| < \tau$. Since Z_i takes only values in the interval $[-\tau, \tau]$, the conditional density $f_{Z_i|T=\tau}(z)$ is 0 for $z \notin [-\tau, \tau]$.

Lemma 4.3.8 follows from Lemmas 4.3.6 and 4.3.7 by integrating over all values of the unconditioned distances. The proof can be found in Appendix B.4.2.

Simplified Random Experiments

In the previous section we did not analyze the random experiment that really takes place. Instead of choosing the points according to the given density functions, we simplified their distributions by placing point O in the origin and by giving the other points P, Q_1, and Q_2 uniform distributions centered around the origin. In our input model, however, each of these points is described by a density function over the unit hypercube. We consider the probability of the event $\Delta \in [0, \varepsilon]$ in both the original input model as well as in the simplified random experiment. In the following, we denote this event by $\mathcal{E}$. We claim that the simplified random experiment that we analyze is only slightly dominated by the original random experiment, in the sense that the probability of the event $\mathcal{E}$ in the simplified random experiment is smaller by at most some factor depending on ϕ.

In order to compare the probabilities in the original and in the simplified random experiment, consider the original experiment and assume that the point O lies at position $x \in [0,1]^d$. Then one can identify a region $\mathcal{R}_x \subseteq \mathbb{R}^{3d}$ with the property that the event $\mathcal{E}$ occurs if and only if the random vector (P, Q_1, Q_2) lies in $\mathcal{R}_x$. No matter of how the position x of O is chosen, this region always has the same shape, only its position is shifted. Let $\mathcal{V} = \sup_{x \in [0,1]^d} \mathrm{Vol}(\mathcal{R}_x \cap [0,1]^{3d})$. Then the probability of $\mathcal{E}$ can be bounded from above by $\phi^3 \cdot \mathcal{V}$ in the original random experiment. Since Δ is invariant under translating O, P, Q_1, and Q_2 by the same vector, we obtain

$$\begin{aligned} \mathrm{Vol}\left(\mathcal{R}_x \cap [0,1]^{3d}\right) &= \mathrm{Vol}\left(\mathcal{R}_{0^d} \cap ([-x_1, 1-x_1] \times \cdots \times [-x_d, 1-x_d])^3\right) \\ &\leq \mathrm{Vol}\left(\mathcal{R}_{0^d} \cap [-1,1]^{3d}\right) \ . \end{aligned}$$

Hence, $\mathcal{V} \leq \mathcal{V}' := \mathrm{Vol}(\mathcal{R}_{0^d} \cap [-1,1]^{3d})$. Since the hyperball centered around the origin with radius $\sqrt{d}$ contains the hypercube $[-1,1]^d$ completely, the probability of $\mathcal{E}$ in the simplified random experiment can be bounded from below by $\mathcal{V}'/V_d(\sqrt{d})^3$, where $V_d(\sqrt{d})$ denotes the volume of a d-dimensional hyperball with radius $\sqrt{d}$.

Since this hyperball is contained in a hypercube with side length $2\sqrt{d}$, its volume can be bounded from above by $(4d)^{d/2}$. Hence, the probability of $\mathcal{E}$ in the simplified random experiment is smaller by at most a factor of $\phi^3(4d)^{3d/2}$ compared to the original random experiment.

Taking into account this factor and using Lemma 4.3.8 c) and a union bound over all possible 2-changes yields the following lemma about the improvement of a single 2-change.

Lemma 4.3.9. *The probability that there exists an improving 2-change whose improvement is at most $\varepsilon \leq 1/2$ is bounded from above by $O(n^4 \cdot \phi^3 \cdot \varepsilon \cdot \log(1/\varepsilon))$.*

Using similar arguments as in the proof of Theorem 4.3.1 yields the following upper bound on the expected number of 2-changes.

Theorem 4.3.10. *Starting with an arbitrary tour, the expected number of steps performed by 2-Opt on ϕ-perturbed Euclidean instances is $O(n^7 \cdot \log^2(n) \cdot \phi^3)$.*

Pairs of Type 1. In order to improve upon Theorem 4.3.10, we consider pairs of linked 2-changes as in the analysis of ϕ-perturbed Manhattan instances. Since our analysis of pairs of linked 2-changes is based on the analysis of a single 2-change that we presented in the previous section, we also have to consider simplified random experiments when analyzing pairs of 2-changes. For a fixed pair of type 1, we assume that point v_3 is chosen to be the origin and the other points v_1, v_2, v_4, v_5, and v_6 are chosen uniformly at random from a hyperball with radius $\sqrt{d}$ centered at v_3. Let $\mathcal{E}$ denote the event that both Δ_1 and Δ_2 lie in the interval $[0, \varepsilon]$, for some given ε. With the same arguments as above, one can see that the probability of $\mathcal{E}$ in the simplified random experiment is smaller compared to the original experiment by at most a factor of $((4d)^{d/2}\phi)^5$.

Pairs of Type 2. For a fixed pair of type 2, we consider the simplified random experiment in which v_2 is placed in the origin and the other points v_1, v_3, v_4, and v_5 are chosen uniformly at random from a hyperball with radius $\sqrt{d}$ centered at v_2. In this case, the probability in the simplified random experiment is smaller by at most a factor of $((4d)^{d/2}\phi)^4$.

Analysis of Pairs of Linked 2-Changes

Finally, we can prove Lemma 4.3.5.

Proof of Lemma 4.3.5. We start by considering pairs of type 1. We consider the simplified random experiment in which v_3 is chosen to be the origin and the other points are drawn uniformly at random from a hyperball with radius $\sqrt{d}$ centered at v_3. If the position of the point v_1 is fixed, then the events $\Delta_1 \in [0, \varepsilon]$ and $\Delta_2 \in [0, \varepsilon]$ are independent as only the vertices v_1 and v_3 appear in both the first and the second step. In fact, because the densities of the points v_2, v_4, v_5, and v_6 are rotationally symmetric, the concrete position of v_1 is not important in our simplified random experiment anymore, but only the distance R between v_1 and v_3 is of interest.

For $i \in [2]$, we determine the conditional probability of the event $\Delta_i \in [0, \varepsilon]$ under the condition that the distance $\mathsf{d}(v_1, v_3)$ is fixed with the help of Lemma 4.3.8 a) and obtain

$$\begin{aligned}\mathbf{Pr}[\Delta_i \in [0,\varepsilon] \mid \mathsf{d}(v_1,v_3) = r] &= \int_0^{\varepsilon} f_{\Delta \mid R_i = r}(\delta)\, d\delta \\ &\leq \int_0^{\varepsilon} \frac{\kappa}{\sqrt{r}} \ln \delta^{-1}\, d\delta \\ &\leq \frac{3\kappa}{\sqrt{r}} \cdot \varepsilon \cdot \ln(1/\varepsilon)\ . \end{aligned} \tag{4.3.2}$$

Since for fixed distance $\mathsf{d}(v_1, v_3)$ the random variables Δ_1 and Δ_2 are independent, we obtain

$$\mathbf{Pr}[\Delta_1, \Delta_2 \in [0,\varepsilon] \mid \mathsf{d}(v_1,v_3) = r] \leq \frac{9\kappa^2}{r} \cdot \varepsilon^2 \cdot \ln^2(1/\varepsilon)\ . \tag{4.3.3}$$

For $r \in [0, \sqrt{d}]$, the density $f_{\mathsf{d}(v_1,v_3)}$ of the random variable $\mathsf{d}(v_1, v_3)$ in the simplified random experiment is $r^{d-1}/d^{d/2-1}$. Combining this observation with the bound given in (4.3.3) yields

$$\mathbf{Pr}[\Delta_1, \Delta_2 \in [0,\varepsilon]] \leq \int_0^{\sqrt{d}} \frac{9\kappa^2 \cdot r^{d-2}}{d^{d/2-1}} \cdot \varepsilon^2 \cdot \ln^2(1/\varepsilon)\, dr = O\left(\varepsilon^2 \cdot \ln^2(1/\varepsilon)\right)\ .$$

There are $O(n^6)$ different pairs of type 1, hence a union bound over all of them concludes the first part of the proof when taking into account the factor $((4d)^{d/2}\phi)^5$ that results from considering the simplified random experiment.

It remains to consider pairs of type 2. We consider the simplified random experiment in which v_2 is chosen to be the origin and the other points are drawn uniformly at random from a hyperball with radius $\sqrt{d}$ centered at v_2. In contrast to pairs of type 1, pairs of type 2 exhibit larger dependencies as only 5 different vertices are involved in these pairs. Fix one pair of type 2. The two 2-changes share the whole triangle consisting of v_1, v_2, and v_3. In the second step, there is only one new vertex, namely v_5. Hence, there is not enough randomness contained in a pair of type 2 such that Δ_1 and Δ_2 are nearly independent as for pairs of type 1.

We start by considering pairs of type 2 a). First, we analyze the probability that Δ_1 lies in the interval $[0, \varepsilon]$. After that, we analyze the probability that Δ_2 lies in the interval $[0, \varepsilon]$ under the condition that the points v_1, v_2, v_3, and v_4 have already been chosen. In the analysis of the second step we cannot make use of the fact that the distances $\mathsf{d}(v_1, v_3)$ and $\mathsf{d}(v_2, v_3)$ are random variables anymore since we exploited their randomness already in the analysis of the first step. The only distances whose randomness we can exploit are the distances $\mathsf{d}(v_1, v_5)$ and $\mathsf{d}(v_2, v_5)$. We pessimistically assume that the distances $\mathsf{d}(v_1, v_3)$ and $\mathsf{d}(v_2, v_3)$ have been chosen by an adversary. This means the adversary can determine an interval of length ε in which the random variable $\mathsf{d}(v_2, v_5) - \mathsf{d}(v_1, v_5)$ must lie in order for Δ_2 to lie in $[0, \varepsilon]$.

Analogously to (4.3.2), the probability of the event $\Delta_1 \in [0, \varepsilon]$ under the condition $\mathsf{d}(v_1, v_2) = r$ can be bounded by

$$\mathbf{Pr}[\Delta_1 \in [0,\varepsilon] \mid \mathsf{d}(v_1,v_2) = r] \leq \frac{3\kappa}{\sqrt{r}} \cdot \varepsilon \cdot \ln(1/\varepsilon)\ . \tag{4.3.4}$$

Due to Lemma 4.3.8 d), the conditional density of the random variable $Z = \mathsf{d}(v_2, v_5) - \mathsf{d}(v_1, v_5)$ under the condition $\mathsf{d}(v_1, v_2) = r$ can be bounded by

$$f_{Z|\,\mathsf{d}(v_1,v_2)=r}(z) \leq \frac{\kappa}{\sqrt{r^2 - z^2}}$$

for $|z| \leq r$.

The intervals the adversary can specify which have the highest probability of Z falling into them are $[-r, -r+\varepsilon]$ and $[r-\varepsilon, r]$. Hence, the conditional probability of the event $\Delta_2 \in [0, \varepsilon]$ under the condition $\mathsf{d}(v_1, v_2) = r$ and for fixed points v_3 and v_4 is bounded from above by

$$\int_{\max\{r-\varepsilon,-r\}}^{r} \frac{\kappa}{\sqrt{r^2 - z^2}}\, dz \leq \frac{\kappa}{\sqrt{r}} \cdot \int_{\max\{r-\varepsilon,-r\}}^{r} \frac{1}{\sqrt{r - |z|}}\, dz \leq \frac{\kappa'\sqrt{\varepsilon}}{\sqrt{r}}$$

for a sufficiently large constant κ'. Combining this inequality with (4.3.4) yields

$$\mathbf{Pr}[\Delta_1, \Delta_2 \in [0, \varepsilon] \mid \mathsf{d}(v_1, v_2) = r] \leq \frac{3\kappa\kappa'}{r}\varepsilon^{3/2} \cdot \ln(1/\varepsilon)\ .$$

In order to get rid of the condition $\mathsf{d}(v_1, v_2) = r$, we integrate over all possible values the random variable $\mathsf{d}(v_1, v_2)$ can take, yielding

$$\mathbf{Pr}[\Delta_1, \Delta_2 \in [0, \varepsilon]] \leq \int_0^{\sqrt{d}} \frac{3\kappa\kappa' \cdot r^{d-2}}{d^{d/2} - 1} \cdot \varepsilon^{3/2} \cdot \ln(1/\varepsilon)\, dr = O\left(\varepsilon^{3/2} \cdot \ln(1/\varepsilon)\right)\ .$$

Applying a union bound over all $O(n^5)$ possible pairs of type 2 a) concludes the proof when one takes into account the factor $((4d)^{d/2}\phi)^4$ due to considering the simplified random experiment.

For pairs of type 2 b), the situation looks somewhat similar. We analyze the first step and in the second step, we can only exploit the randomness of the distances $\mathsf{d}(v_2, v_5)$ and $\mathsf{d}(v_3, v_5)$. Due to Lemma 4.3.8 b) and similar to (4.3.2), the probability of the event $\Delta_1 \in [0, \varepsilon]$ under the condition $\mathsf{d}(v_2, v_3) = \tau$ can be bounded by

$$\mathbf{Pr}[\Delta_1 \in [0, \varepsilon] \mid \mathsf{d}(v_2, v_3) = \tau] \leq \frac{3\kappa}{\tau} \cdot \varepsilon \cdot \ln(1/\varepsilon)\ . \tag{4.3.5}$$

The remaining analysis of pairs of type 2 b) can be carried out completely analogously to the analysis of pairs of type 2 a). □

The Expected Number of 2-Changes

Based on Lemmas 4.3.3 and 4.3.5, we are now able to prove part b) of Theorem 1.3.2.

Proof of Theorem 1.3.2 b). We use the same notations as in the proof of part a) of the theorem. For $t > 2n^2$, we have $t/6 - 7n(n-1)/24 > t/48$ and hence due to Lemma 4.3.5,

$$\begin{aligned}\mathbf{Pr}[T \geq t] &\leq \mathbf{Pr}\left[\Delta^*_{\min} \leq \frac{48\sqrt{d}n}{t}\right] \\ &= O\left(\min\left\{\frac{n^8 \cdot \log^2(t) \cdot \phi^5}{t^2}, 1\right\}\right) + O\left(\min\left\{\frac{n^{13/2} \cdot \log(t) \cdot \phi^4}{t^{3/2}}, 1\right\}\right)\ .\end{aligned}$$

This implies that the expected length $\mathbf{E}[T]$ of the longest path in the transition graph is bounded from above by

$$2n^2 + \sum_{t=2n^2+1}^{n!} O\left(\min\left\{\frac{n^8 \cdot \log^2(t) \cdot \phi^5}{t^2}, 1\right\}\right) + \sum_{t=2n^2+1}^{n!} O\left(\min\left\{\frac{n^{13/2} \cdot \log(t) \cdot \phi^4}{t^{3/2}}, 1\right\}\right) .$$

Splitting the sums at $t = n^4 \cdot \log(n\phi) \cdot \phi^{5/2}$ and $t = n^{13/3} \cdot \log^{2/3}(n\phi) \cdot \phi^{8/3}$, respectively, yields

$$\mathbf{E}[T] = O\left(n^4 \cdot \log(n\phi) \cdot \phi^{5/2}\right) + O\left(n^{13/3} \cdot \log^{2/3}(n\phi) \cdot \phi^{8/3}\right) .$$

This concludes the proof of part b) of the theorem. □

Using the same observations as in the proof of Theorem 1.3.3 a) also yields part b).

Proof of Theorem 1.3.3 b). Estimating the length of the initial tour by $O(n^{(d-1)/d} \cdot \log n)$ instead of $O(n)$ improves the upper bound on the expected number of 2-changes by a factor of $\Theta(n^{1/d}/\log n)$ compared to Theorem 1.3.2 b). This observation yields the bound claimed in Theorem 1.3.3 b). □

4.3.3 General Graphs

In this section, we analyze the expected number of 2-changes on ϕ-perturbed graphs. Observe that such graphs contain more randomness than ϕ-perturbed Manhattan or Euclidean instances because each edge length is a random variable that is independent of the other edge lengths. It is easy to obtain a polynomial bound on the expected number of local improvements by just estimating the smallest improvement made by any of the 2-changes. For Manhattan and Euclidean instances we improved this simple bound by considering pairs of linked 2-changes. For ϕ-perturbed graphs we pursue the same approach but due to the larger amount of randomness, we are now able to consider not only pairs of linked steps but longer sequences of linked steps.

We know that every sequence of steps that contains k distinct 2-changes shortens the tour by at least $\Delta^{(k)} = \Delta^{(1)}_{\min} + \cdots + \Delta^{(k)}_{\min}$, where $\Delta^{(i)}_{\min}$ denotes the i-th smallest improvement. This observation alone, however, does not suffice to improve the simple bound substantially. Instead we show that one can identify in every long enough sequence of consecutive 2-changes, subsequences that are *linked*, where a sequence $S_1, \ldots, S_k$ of 2-changes is called *linked* if for every $i \in [k-1]$, there exists an edge that is added to the tour in step S_i and removed from the tour in step S_{i+1}. We analyze the smallest improvement of a linked sequence that consists of k distinct 2-Opt steps. Obviously, this improvement must be at least $\Delta^{(k)}$ as in the worst-case, the linked sequence consists of the k smallest improvements. Intuitively, one can hope that it is much larger than $\Delta^{(k)}$ because it is unlikely that the k smallest improvements form a sequence of linked steps. We show that

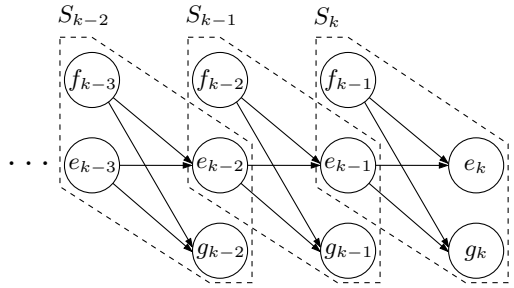


Figure 4.3.4: Illustration of the notations used in Definitions 4.3.11, 4.3.12, and 4.3.13. Every node in the DAG corresponds to an edge involved in one of the 2-changes. An arc from a node u to a node v indicates that in one of the 2-changes, the edge corresponding to node u is removed from the tour and the edge corresponding to node v is added to the tour. Hence, every arc is associated with one 2-change.

this is indeed the case and use this result to prove the desired upper bound on the expected number of 2-changes.

We introduce the notion of *witness sequences*, i.e., linked sequences of 2-changes that satisfy some additional technical properties. We show that the smallest total improvement made by a witness sequence yields an upper bound on the running time. That is, whenever the 2-Opt heuristic needs many local improvement steps to find a locally optimal solution, there must be a witness sequence whose total improvement is small. Furthermore, our probabilistic analysis reveals that it is unlikely that there exists a witness sequence whose total improvement is small. Together, these results yield the desired bound on the expected number of 2-changes.

Definition of Witness Sequences

In this section, we give a formal definition of the notion of a *k-witness sequence*. As mentioned above, a witness sequence $S_1, \dots, S_k$ has to be linked, i.e., for $i \in [k-1]$, there must exist an edge that is added to the tour in step S_i and removed from the tour in step S_{i+1}. Let m denote the number of edges in the graph. Then there are at most $4^{k-1} \cdot m^{k+1}$ such linked sequences as there are at most m^2 different choices for S_1, and once S_i is fixed, there are at most $4m$ different choices for S_{i+1}. For a fixed 2-change, the probability that it is an improvement by at most ε is bounded by $\varepsilon\phi$. We would like to show an upper bound of $(\varepsilon\phi)^k$ on the probability that each step in the witness sequence $S_1, \dots, S_k$ is an improvement by at most ε. For general linked sequences, this is not true as the steps can be dependent in various ways. Hence, we need to introduce further restrictions on witness sequences.

In the following definitions, we assume that a linked sequence $S_1, \dots, S_k$ of 2-changes is given. For $i \in [k]$, in step S_i the edges e_{i-1} and f_{i-1} are removed from the tour and the edges e_i and g_i are added to the tour, i.e., for $i \in [k-1]$, e_i denotes an edge added to the tour in step S_i and removed from the tour in step S_{i+1}. These definitions are illustrated in Figure 4.3.4.

Definition 4.3.11 (witness sequences of type 1)**.** *If for every $i \in [k]$, the edge e_i does not occur in any step S_j with $j < i$, then $S_1, \dots, S_k$ is called a* k-witness sequence of type 1.

Intuitively, witness sequences of type 1 possess enough randomness as every step introduces an edge that has not been seen before. Based on this observation, we prove in Lemma 4.3.14 the desired bound of $(\varepsilon\phi)^k$ on the probability that every step is an improvement by at most ε for these sequences.

Definition 4.3.12 (witness sequences of type 2)**.** *Assume that for every $i \in [k]$, the edge e_i does not occur in any step S_j with $j < i$. If both endpoints of f_{k-1} occur in steps S_j with $j < k$, then $S_1, \ldots, S_k$ is called a* k-witness sequence of type 2*.*

Also for witness sequences of type 2, we obtain the desired bound of $(\varepsilon\phi)^k$ on the probability that every step is an improvement by at most ε. Due to the additional restriction on f_{k-1}, there are less than $4^{k-1}m^{k+1}$ witness sequences of type 2. As the two endpoints of f_{k-1} must be chosen among those vertices that occur in steps S_j with $j < k$, there are only $O(k^2)$ choices for the last step S_k. This implies that the number of k-witness sequences of type 2 can be upper bounded by $O(4^k k^2 m^k)$.

Definition 4.3.13 (witness sequences of type 3)**.** *Assume that for every $i \in [k-1]$, the edge e_i does not occur in any step S_j with $j < i$. If the edges e_k and g_k occur in steps S_j with $j < k$ and if f_{k-1} does not occur in any step S_j with $j < k$, then $S_1, \ldots, S_k$ is called a* k-witness sequence of type 3*.*

Also witness sequences of type 3 possess enough randomness to bound the probability that every step is an improvement by at most ε by $(\varepsilon\phi)^k$ as also the last step introduces a new edge, namely f_{k-1}.

Improvement made by Witness Sequences

In this section, we analyze the probability that there exists a k-witness sequence in which every step is an improvement by at most ε.

Lemma 4.3.14. *The probability that there exists a k-witness sequence in which every step is an improvement by at most ε*

a) is bounded from above by $4^{k-1}m^{k+1}(\varepsilon\phi)^k$ for k-witness sequences of type 1.

b) is bounded from above by $k^2 4^{k-1}m^k(\varepsilon\phi)^k$ for k-witness sequences of type 2.

c) is bounded from above by $k^2 4^k m^k(\varepsilon\phi)^k$ for k-witness sequences of type 3.

Proof. We use a union bound to estimate the probability that there exists a witness sequence in which every step is a small improvement.

a) We consider k-witness sequences of type 1 first. As already mentioned in the previous section, the number of such sequences is bounded by $4^{k-1}m^{k+1}$ as there are at most m^2 choices for the first step S_1, and once S_i is fixed, there are at most $4m$ choices for step S_{i+1}. The number $4m$ follows since if S_i is fixed, there are two choices for the edge added to the tour in step S_i and removed from the tour in step S_{i+1}, there are at most m choices for the other edge removed in step S_{i+1}, and once these edges are determined, there are two possible 2-Opt steps in which these edges are removed from the tour.

Now fix an arbitrary k-witness sequence $S_1, \ldots, S_k$ of type 1. We use the same notations as in Figure 4.3.4 to denote the edges involved in this sequence. In the

first step, the edges e_0 and f_0 are replaced by the edges e_1 and g_1. We assume that the lengths of the edges e_0, f_0, and g_1 are determined by an adversary. The improvement of step S_1 can be expressed as a simple linear combination of the lengths of the involved edges. Hence, for fixed lengths of e_0, f_0, and g_1, the event that S_1 is an improvement by at most ε corresponds to the event that the length $\mathsf{d}(e_1)$ of e_1 lies in some fixed interval of length ε. Since the density of $\mathsf{d}(e_1)$ is bounded by ϕ, the probability that $\mathsf{d}(e_1)$ takes a value in the given interval is bounded by $\varepsilon\phi$. Now consider a step S_i and assume that arbitrary lengths for the edges e_j and f_j with $j < i$ and for g_j with $j \leq i$ are chosen. Since the edge e_i is not involved in any step S_j with $j < i$, its length is not determined. Hence, analogously to the first step, the probability that step S_i is an improvement by at most ε is bounded by $\varepsilon\phi$ independent of the improvements of the steps S_j with $j < i$. Applying this argument to every step S_i yields the desired bound of $(\varepsilon\phi)^k$.

b) In witness sequences of type 2, there are at most m^2 choices for step S_1. Analogously to witness sequences of type 1, the number of possible choices for S_i with $1 < i < k-1$ is at most $4m$. The number of different vertices involved in steps S_j with $j < k$ is at most $4 + 2(k-2) = 2k$ as the first step introduces four new vertices and every other step at most 2. Since the endpoints of the edge f_{k-1} must be chosen among those vertices that have been involved in the steps S_j with $j < k$, there are at most $4k^2$ possible choices for step S_{k-1}. This implies that the number of different k-witness sequences of type 2 is bounded by $4 \cdot k^2 4^{k-2} m^k = k^2 4^{k-1} m^k$.

For a fixed witness sequence of type 2, applying the same arguments as for witness sequences of type 1, yields a probability of at most $(\varepsilon\phi)^k$ that every step is an improvement by at most ε.

c) The number of different edges involved in the steps S_i with $i < k$ is at most $4 + 3(k-2) < 3k$. Hence, the number of k-witness sequences of type 3 is bounded by $9k^2 4^{k-2} m^k < k^2 4^k m^k$. Furthermore, similar to witness sequences of type 1, we can bound the probability that a fixed k-witness sequence of type 3 consists only of improvements by at most ε by $(\varepsilon\phi)^k$ since the last step introduces an edge which does not occur in the steps S_i with $i < k$, namely f_{k-1}. □

Definition 4.3.15. *In the following, we use the term* k-witness sequence *to denote a k-witness sequence of type 1 or an i-witness sequence of type 2 or 3 with $i \leq k$. We call a k-witness sequence* improving *if every 2-change in the sequence is an improvement. Moreover, by $\Delta_{\mathrm{ws}}^{(k)}$ we denote the smallest total improvement made by any improving k-witness sequence.*

Due to Lemma 4.3.14, it is unlikely that there exists an improving witness sequence whose total improvement is small.

Corollary 4.3.16. *For $k \in \mathbb{N}$ and $0 < \varepsilon \leq (4m^{(k-1)/(k-2)}\phi)^{-1}$,*

$$\mathbf{Pr}\left[\Delta_{\mathrm{ws}}^{(k)} \leq \varepsilon\right] \leq 64k^3(m\varepsilon\phi)^2 \ .$$

Proof. Due to Lemma 4.3.14 and the fact that witness sequences of type 2 or 3 must consist of at least two steps, applying a union bound yields the following bound on the probability that there exists an improving k-witness sequence whose

total improvement is at most ε:

$$\begin{aligned}\mathbf{Pr}\left[\Delta_{\mathrm{ws}}^{(k)} \leq \varepsilon\right] \leq & 4^{k-1}m^{k+1}(\varepsilon\phi)^k + \sum_{i=2}^{k} i^2 4^{i-1} m^i (\varepsilon\phi)^i + \sum_{i=2}^{k} i^2 4^i m^i (\varepsilon\phi)^i \\ \leq & 4^{k-1}m^{k+1}(\varepsilon\phi)^k + 2k^2 \sum_{i=2}^{k} (4m\varepsilon\phi)^i \ .\end{aligned}$$

Since $4m\varepsilon\phi < 1$, we can bound the sum by

$$\mathbf{Pr}\left[\Delta_{\mathrm{ws}}^{(k)} \leq \varepsilon\right] \leq 4^{k-1}m^{k+1}(\varepsilon\phi)^k + 2k^3(4m\varepsilon\phi)^2 \ ,$$

which implies the corollary because for $\varepsilon \leq (4m^{(k-1)/(k-2)}\phi)^{-1}$, the second term in the sum is at least as large as first one. □

Finding Witness Sequences

In the previous section, we have shown an upper bound on the probability that there exists an improving k-witness sequence whose total improvement is small. In this section, we show that in every long enough sequence of consecutive 2-changes, one can identify a certain number of disjoint k-witness sequences. This way, we obtain a lower bound on the improvement made by any long enough sequence of consecutive 2-changes in terms of $\Delta_{\mathrm{ws}}^{(k)}$.

Lemma 4.3.17. *Let $k \in \mathbb{N}$, and let $S_1, \ldots, S_t$ denote a sequence of consecutive 2-changes performed by the 2-Opt heuristic with $t > n4^{k+1}$. The sequence $S_1, \ldots, S_t$ shortens the tour by at least $t/4^{k+3} \cdot \Delta_{\mathrm{ws}}^{(k)}$.*

Basically, we have to show that one can find $t/4^{k+3}$ disjoint k-witness sequences in the given sequence of consecutive 2-changes. Therefore, we first introduce a so-called *witness DAG* (directed acyclic graph) which represents the sequence $S_1, \ldots, S_t$ of 2-changes. In order to not confuse the constructed witness DAG W with the input graph G, we use the terms *nodes* and *arcs* when referring to the DAG W and the terms *vertices* and *edges* when referring to G. Nodes of W correspond to edges of G combined with a time stamp. The construction is started by adding the edges of the initial tour as nodes into W. These nodes get the time stamps $1, \ldots, n$ in an arbitrary order. Then the sequence $S_1, \ldots, S_t$ is processed step by step. Assume that the steps $S_1, \ldots S_{i-1}$ have already been processed and that step S_i is to be processed next. Furthermore, assume that in step S_i the edges e_{i-1} and f_{i-1} are exchanged with the edges e_i and g_i. Since the edges e_{i-1} and f_{i-1} are contained in the tour after the steps $S_1, \ldots, S_{i-1}$, there are nodes in W corresponding to these edges. Let u_1 and u_2 denote the nodes with the most recent time stamps corresponding to e_{i-1} and f_{i-1}, respectively. We create two new nodes u_3 and u_4 corresponding to the edges e_i and g_i, each with time stamp $n + i$. Finally, four new arcs are added to W, namely the arcs (u_1, u_3), (u_1, u_4), (u_2, u_3), and (u_2, u_4). We refer to these four arcs as *twin arcs*. Observe that each node in W has indegree and outdegree at most 2. We call the resulting DAG W a *t-witness DAG*.

By the *height* of a node u, we denote the length of a shortest path from u to a leaf of W. After the witness DAG has been completely constructed, we associate

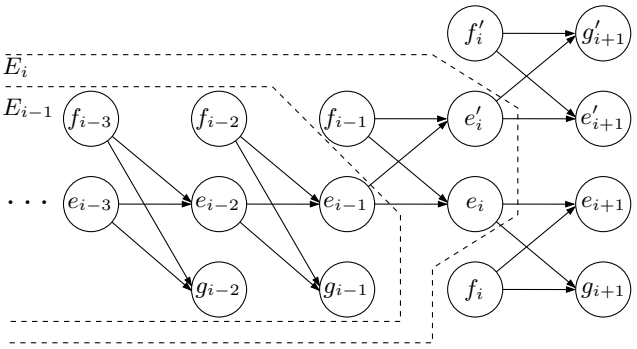


Figure 4.3.5: Construction of a path in the witness DAG: The path has been constructed up to step S_i and now it has to be decided whether to continue it along e_i or e'_i.

with each node u with height at least k a sub-DAG of W. The sub-DAG W_u associated with such a node u is the induced sub-DAG of those nodes of W that can be reached from u by traversing at most k arcs. The following two lemmas imply Lemma 4.3.17.

Lemma 4.3.18. *For every sub-DAG W_u, the 2-changes represented by the arcs in W_u and their twin arcs yield a total improvement of at least $\Delta_{\mathrm{ws}}^{(k)}$.*

Lemma 4.3.19. *For $t > n4^{k+1}$, every t-witness DAG contains at least $t/4^{k+2}$ nodes u whose corresponding sub-DAGs W_u are pairwise disjoint.*

Proof of Lemma 4.3.18. Assume that a sub-DAG W_u with root u is given. Since node u has height k, one can identify 2^{k-1} distinct sequences of linked 2-changes of length k in the sub-DAG W_u. In the following, we show that at least one of these sequences is a k-witness sequence or a sequence whose total improvement is as large as the total improvement of one of the k-witness sequences. We give a recursive algorithm Sequ that constructs such a sequence step by step. It is initialized with the sequence which consists only of the first step S_1 that is represented by the two outgoing arcs of the root u and their twin arcs.

Assume that Sequ is called with a sequence of steps $S_1, \ldots, S_i$ that has been constructed so far. Given this sequence, it has to decide if the sequence is continued with a step S_{i+1} such that S_i and S_{i+1} are linked or if the construction is stopped since a k-witness sequence is found. In Figure 4.3.5, we summarize the notations that we use in the following. In step S_j for $j \leq i-1$ and $j = i+1$, the edges e_{j-1} and f_{j-1} are exchanged with the edges e_j and g_j. In step S_i, the edges e_{i-1} and f_{i-1} are exchanged with the edges e_i and e'_i, and in step S'_{i+1}, the edges e'_i and f'_i are exchanged with the edges e'_{i+1} and g'_{i+1}. We denote by E_i all edges that are involved in steps S_j with $j \leq i$. Similarly, by E_{i-1} we denote all edges that are involved in steps S_j with $j \leq i-1$.

Our construction ensures that whenever algorithm Sequ is called with a sequence $S_1, \ldots, S_i$ as input, then at least one of the edges that is added to the tour in step S_i is not contained in E_{i-1}. In the following, assume without loss of generality that $e_i \notin E_{i-1}$. When we call the algorithm recursively with the sequence $S_1, \ldots, S_{i+1}$ or with the sequence $S_1, \ldots, S_i, S'_{i+1}$, then either the recursive call never gives back a return value since a witness sequence is found in the recursive

call, which immediately stops the construction, or a 2-change S is returned. Whenever a 2-change S is returned, the meaning is as follows: There exists a sequence of linked 2-changes in the sub-DAG W_u starting with S_{i+1} or S'_{i+1}, respectively, whose net effect equals the 2-change S. That is, after all steps in the sequence have been performed, the same two edges as in S are removed from the tour, the same two edges are added to the tour, and all other edges either stay in or out of the tour. In this case, we can virtually replace step S_{i+1} or S'_{i+1}, respectively, by the new step S.

When Sequ is called with the sequence $S_1, \ldots, S_i$, then it first identifies the steps S_{i+1} and S'_{i+1} based on the last step S_i. If $i = k$, then $S_1, \ldots, S_i$ is a k-witness sequence of type 1, and Sequ stops. Otherwise, the following steps are performed, where we assume that whenever Sequ has identified a witness sequence, it immediately stops the construction.

1. Type 2 Sequence: If $f_{i-1} \in E_{i-1}$, then $S_1, \ldots, S_i$ is a witness sequence of type 2 because we assumed $e_i \notin E_{i-1}$.
2. Continuation: If $e_{i+1} \notin E_i$ or $g_{i+1} \notin E_i$, then call Sequ recursively with $\mathsf{Sequ}(S_1, \ldots, S_{i+1})$. If $e'_i \notin E_{i-1}$ and ($e'_{i+1} \notin E_i$ or $g'_{i+1} \notin E_i$), then call Sequ recursively with $\mathsf{Sequ}(S_1, \ldots, S_i, S'_{i+1})$.

   If in one of the recursive calls a step S is returned, which happens only in case 3 (c), then replace the corresponding step S_{i+1} or S'_{i+1} virtually by the returned step. That is, in the following steps of the algorithm, assume that S_{i+1} or S'_{i+1} equals step S. The algorithm ensures that the edges that are added to the tour in the new step S are always chosen from the set E_i.
3. No Continuation I: $e'_i \in E_{i-1}$ and $e_{i+1}, g_{i+1} \in E_i$ and $f_{i-1} \notin E_{i-1}$
   (a) If $f_i \notin E_i$, then $S_1, \ldots, S_{i+1}$ is a witness sequence of type 3.
   (b) If $e_{i+1}, g_{i+1} \in E_{i-1}$, then $S_1, \ldots, S_i$ is a witness sequence of type 2 since one endpoint of f_{i-1} equals one endpoint of e'_i and the other one equals one endpoint of either e_{i+1} or g_{i+1}.
   (c) If $f_i \in E_i$ and ($e_{i+1} \in E_i \setminus E_{i-1}$ or $g_{i+1} \in E_i \setminus E_{i-1}$), then one can assume w.l.o.g. that $g_{i+1} = f_{i-1}$ and $e_{i+1} \in E_{i-1}$ since $e_{i+1} \neq e'_i$ and $g_{i+1} \neq e'_i$ (e_{i+1} and g_{i+1} share one endpoint with e_i; e'_i does not share an endpoint with e_i.) In this case, return the step $S = (e_{i-1}, f_i) \to (e_{i+1}, e'_i)$. Observe that $e_{i+1}, e'_i \in E_{i-1}$, as desired.
4. No Continuation II: $e'_i \notin E_{i-1}$ and $e_{i+1}, g_{i+1}, e'_{i+1}, g'_{i+1} \in E_i$ and $f_{i-1} \notin E_{i-1}$
   (a) If $e_{i+1}, g_{i+1}, e'_{i+1}, g'_{i+1} \in E_{i-1}$, then $S_1, \ldots, S_i$ is a witness sequence of type 2.
   (b) If $f_i \notin E_i$, then $S_1, \ldots, S_{i+1}$ is a witness sequence of type 3.
   (c) If $f'_i \notin E_i$, then $S_1, \ldots, S_i, S'_{i+1}$ is a witness sequence of type 3.
   (d) If $f_i, f'_i \in E_i$ and ($e_{i+1} \in E_i \setminus E_{i-1}$ or $g_{i+1} \in E_i \setminus E_{i-1}$) and ($e'_{i+1} \in E_i \setminus E_{i-1}$ or $g'_{i+1} \in E_i \setminus E_{i-1}$), then as in case 3 (c), assume w.l.o.g. $g_{i+1} = g'_{i+1} = f_{i-1}$ and $e_{i+1}, e'_{i+1} \in E_{i-1}$. In this case, it must be $f_i \neq e'_i$ and $f'_i \neq e_i$ as otherwise step S_i would be reversed in step S_{i+1} or S'_{i+1}.

Hence, $f_i, f_i' \in E_{i-1}$, and $S_1, \ldots, S_i$ is a witness sequence of type 2 since one endpoint of f_{i-1} equals one endpoint of f_i and the other endpoint equals one endpoint of f_i'.

(e) If $|\{e_{i+1}, e_{i+1}', g_{i+1}, g_{i+1}'\} \cap (E_i \setminus E_{i-1})| = 1$, we can assume w.l.o.g. that $e_{i+1}, g_{i+1}, e_{i+1}' \in E_{i-1}$ and $g_{i+1}' = f_{i-1}$. As in the previous case, it must $f_i' \in E_{i-1}$. For the step $S = (e_{i-1}, f_i') \to (e_i, e_{i+1}')$, the sequence $S_1, \ldots, S_{i-1}, S$ is a witness sequence of type 2 as $f_i' \in E_{i-1}$. Observe that the original sequence $S_1, \ldots, S_i$ together with the step S_{i+1}' yields the same net effect and hence the same improvement as the sequence $S_1, \ldots, S_{i-1}, S$.

Observe that basically Sequ just constructs a path through the DAG starting at node u. When a path corresponding to the sequence $S_1, \ldots, S_i$ of 2-changes has been constructed, Sequ decides to either stop the construction since a witness sequence has been found, or, if possible, to continue the path with an arc corresponding to a step S_{i+1} or S_{i+1}'. In some situations, it can happen that Sequ has not found a witness sequence yet but cannot continue the construction. In such cases, step S_i is pruned and Sequ reconsiders the path $S_1, \ldots, S_{i-1}$. Based on the pruned step S_i it can then either decide that a witness sequence has been found, that also S_{i-1} has to be pruned, or it can decide to continue the path with S_i' instead of S_i.

This concludes the proof as the presented algorithm always identifies a k-witness sequence whose total improvement is at most as large as the improvement made by the steps in the sub-DAG W_u. □

Proof of Lemma 4.3.19. A t-witness DAG W consists of $n+2t$ nodes and n of these nodes are leaves. Since the indegree and the outdegree of every node is bounded by 2, there are at most $n2^k$ nodes in W whose height is less than k. Hence, there are at least $n + 2t - n2^k \geq t$ nodes in W with an associated sub-DAG. We construct a set of disjoint sub-DAGs in a greedy fashion: We take an arbitrary sub-DAG W_u and add it to the set of disjoint sub-DAGs that we construct. After that, we remove all nodes, arcs, and twin arcs of W_u from the DAG W. We repeat these steps until no sub-DAG W_u is left in W.

In order to see that the constructed set consists of at least $t/4^{k+2}$ disjoint sub-DAGs, observe that each sub-DAG consists of at most $2^{k+1} - 1$ nodes as its height is k. Hence, it contains at most $2^k - 1$ pairs of twin arcs, and there are at most $2^{k+2} - 4$ arcs that belong to the sub-DAG or that have a twin arc belonging to the sub-DAG. Furthermore, observe that each of these arcs can be contained in at most $2^k - 1$ sub-DAGs. Hence, every sub-DAG W_u can only be non-disjoint from at most $2^{2k+2} = 4^{k+1}$ other sub-DAGs. Thus, the number of disjoint sub-DAGs must be at least $\lfloor t/4^{k+1} \rfloor > t/4^{k+2}$, where the last inequality follows because we assumed $t > n4^{k+1}$. □

The Expected Number of 2-Changes

Now we can prove Theorem 1.3.2.

Proof of Theorem 1.3.2 c). We combine Corollary 4.3.16 and Lemma 4.3.17 to obtain an upper bound on the probability that the length T of the longest path

in the transition graph exceeds t. For $t > n4^{k+1}$, the tour is shortened by the sequence of 2-changes by at least $t/4^{k+2} \cdot \Delta_{\mathrm{ws}}^{(k)}$. Hence, for $t > n4^{k+1}$,

$$\mathbf{Pr}[T \geq t] \leq \mathbf{Pr}\left[\frac{t}{4^{k+2}} \cdot \Delta_{\mathrm{ws}}^{(k)} \leq n\right] = \mathbf{Pr}\left[\Delta_{\mathrm{ws}}^{(k)} \leq \frac{n \cdot 4^{k+2}}{t}\right] .$$

For $t \geq 4^{k+3} \cdot n \cdot \phi \cdot m^{(k-1)/(k-2)}$, combining the previous inequality with Corollary 4.3.16 yields

$$\mathbf{Pr}[T \geq t] \leq 64k^3 \left(\frac{4^{k+2} \cdot n \cdot m \cdot \phi}{t}\right)^2 .$$

Hence, we can bound the expected number of 2-changes by

$$\mathbf{E}[T] \leq 4^{k+3} \cdot n \cdot \phi \cdot m^{(k-1)/(k-2)} + \sum_{t=1}^{n!} \min\left\{64k^3 \left(\frac{4^{k+2} \cdot n \cdot m \cdot \phi}{t}\right)^2, 1\right\} .$$

Splitting the sum at $t = k^{3/2} \cdot 4^{k+2} \cdot n \cdot m \cdot \phi$ yields

$$\mathbf{E}[T] = O\left(k^{3/2} \cdot 4^k \cdot n \cdot m^{(k-1)/(k-2)} \cdot \phi\right) .$$

Setting $k = \sqrt{\log m}$ yields the theorem. □

4.4 Expected Approximation Ratio

In this section, we consider the expected approximation ratio of the solution found by 2-Opt on ϕ-perturbed L_p instances. Chandra, Karloff, and Tovey [CKT99] show that if one has a set of n points in the unit hypercube $[0,1]^d$ and the distances are measured according to a metric that is induced by a norm, then every locally optimal solution has length at most $c \cdot n^{(d-1)/d}$ for an appropriate constant c depending on the dimension d and the metric. Hence, it follows for every L_p metric that 2-Opt yields a tour of length at most $O(n^{(d-1)/d})$ on ϕ-perturbed L_p instances. This implies that, in order to bound the expected approximation ratio of 2-Opt on these instances, we just need to upper bound the expected value of $1/\mathrm{Opt}$, where Opt denotes the length of the shortest tour.

Lemma 4.4.1. *Let $p \in \mathbb{N} \cup \{\infty\}$. For ϕ-perturbed L_p instances with n points,*

$$\mathbf{E}\left[\frac{1}{\mathrm{Opt}}\right] = O\left(\frac{\sqrt[d]{\phi}}{n^{(d-1)/d}}\right) .$$

Proof. Let $v_1, \ldots, v_n \in \mathbb{R}^d$ denote the points of the ϕ-perturbed instance. We partition the unit hypercube into $k = \lceil n\phi \rceil$ smaller hypercubes with volume $1/k$ each and analyze how many of these smaller hypercubes contain at least one of the points. Assume that X of these hypercubes contain a point, then the optimal tour must have length at least $X/(3^d \sqrt[d]{k})$. In order to see this, we construct a set $P \subseteq \{v_1, \ldots, v_n\}$ of points as follows: Consider the points $v_1, \ldots, v_n$ one after another, and insert a point v_i into P if P does not contain a point in the same hypercube as v_i or in one of its $3^d - 1$ neighboring hypercubes yet. Due to the

triangle inequality, the optimal tour on P is at most as long as the optimal tour on $v_1, \ldots, v_n$. Furthermore, P contains at least $X/3^d$ points and every edge between two points from P has length at least $1/\sqrt[d]{k}$ since P does not contain two points in the same or in two neighboring hypercubes.

Hence, it remains to analyze the random variable X. For each hypercube i with $1 \leq i \leq k$, we define a random variables X_i which takes value 0 if hypercube i is empty and value 1 if hypercube i contains at least one point. The density functions that specify the locations of the points induce for each pair of hypercube i and point j a probability p_i^j such that point j falls into hypercube i with probability p_i^j. Hence, one can think of throwing n balls into k bins in a setting where each ball has its own probability distribution over the bins. Due to the bounded density, we have $p_i^j \leq \phi/k$. For each hypercube i, let M_i denote the probability mass associated with the hypercube, that is

$$M_i = \sum_{j=1}^{n} p_i^j \leq \frac{n\phi}{k} \ .$$

We can write the expected value of the random variable X_i as

$$\mathbf{E}[X_i] = \mathbf{Pr}[X_i = 1] = 1 - \prod_{j=1}^{n}(1 - p_i^j) \geq 1 - \left(1 - \frac{M_i}{n}\right)^n$$

as, under the constraint $\sum_j (1 - p_i^j) = n - M_i$, the term $\prod_j (1 - p_i^j)$ is maximized if all p_i^j are equal. Due to linearity of expectation, the expected value of X is

$$\mathbf{E}[X] \geq \sum_{i=1}^{k} 1 - \left(1 - \frac{M_i}{n}\right)^n = k - \sum_{i=1}^{k} \left(1 - \frac{M_i}{n}\right)^n \ .$$

Observe that $\sum_i M_i = n$. Thus, the sum $\sum_i (1 - M_i/n)$ becomes maximal if the M_i's are chosen as unbalanced as possible. Hence, we assume that $\lceil k/\phi \rceil$ of the M_i's take their maximal value of $n\phi/k$ and the other M_i's are zero. This yields, for sufficiently large n,

$$\begin{aligned}
\mathbf{E}[X] \geq & k - \left(\left\lceil \frac{k}{\phi} \right\rceil \left(1 - \frac{\phi}{k}\right)^n + \left(k - \left\lceil \frac{k}{\phi} \right\rceil\right)\right) \\
\geq & \frac{k}{\phi} - \frac{2k}{\phi}\left(1 - \frac{\phi}{k}\right)^n \\
\geq & \frac{k}{\phi}\left(1 - 2\left(1 - \frac{1}{n+1}\right)^n\right) \geq \frac{n}{5} \ .
\end{aligned}$$

Hence, we obtain the following bound on the expected length of the optimal tour

$$\mathbf{E}[\mathrm{Opt}] \geq \frac{\mathbf{E}[X]}{3^d \sqrt[d]{k}} \geq \frac{n}{5 \cdot 3^d \sqrt[d]{k}} \geq \frac{n^{(d-1)/d}}{5 \cdot 3^d \sqrt[d]{\phi + 1}} \ .$$

We still need to determine the expected value of the random variable $1/\mathrm{Opt}$. Therefore, we first show that X is sharply concentrated around its mean value. The random variable X is the sum of n 0-1-random variables. If these random

variables were independent, we could simply use a Chernoff bound to bound the probability that X takes a value that is smaller than its mean value. The X_i's are negatively associated, in the sense that whenever we already know that some of the X_i's are zero, then the probability of the event that another X_i also takes the value zero becomes smaller. Hence, intuitively, the dependencies can only help to bound the probability that X takes a value smaller than its mean value. Dubhashi and Ranjan [DR98] formalize this intuition by introduction the notion of negative dependence and by showing that in the case of negative dependent random variables, one can still apply a Chernoff bound. This yields

$$\mathbf{Pr}\left[X \leq \frac{n}{10}\right] \leq e^{-n/40} \ .$$

Thus, as $1/X \leq 1$ with certainty, for sufficiently large n,

$$\mathbf{E}\left[\frac{1}{X}\right] \leq (1 - e^{-n/40}) \cdot \frac{10}{n} + e^{-n/40} \leq \frac{11}{n}$$

Altogether, this implies

$$\mathbf{E}\left[\frac{1}{\mathrm{Opt}}\right] \leq \mathbf{E}\left[\frac{3^d \cdot \sqrt[d]{\lceil n\phi \rceil}}{X}\right] = O\left(\frac{\sqrt[d]{\phi}}{n^{(d-1)/d}}\right) \ . \qquad \square$$

If one combines Lemma 4.4.1 with the result of Chandra, Karloff, and Tovey that every locally optimal solution has length $O(n^{(d-1)/d})$, one obtains Theorem 1.3.4.

4.5 Smoothed Analysis

In Spielman and Teng's smoothed analysis of the Simplex algorithm, the adversary specifies an arbitrary linear program which is perturbed by adding an independent Gaussian random variable to each number in the constraints. Our probabilistic analysis of 2-Opt can also be seen as a smoothed analysis in which an adversary can choose the distributions for the points over the unit hypercube. The adversary is restricted to distributions that can be represented by densities that are bounded by ϕ. Our model cannot handle Gaussian perturbations directly because the support of Gaussian random variables is not bounded.

Assume that every point $v_1, \ldots, v_n$ is described by a density whose support is restricted to the hypercube $[-\alpha, \alpha]^d$ for some $\alpha \geq 1$. Then after appropriate scaling and translating, we can assume that all supports are restricted to the unit hypercube $[0, 1]^d$. Thereby, the maximal density ϕ increases by at most a factor of $(2\alpha)^d$. Hence, after appropriate scaling and translating, Theorems 1.3.2, 1.3.3, and 1.3.4 can still be applied if one takes into account the increased densities.

One possibility to cope with Gaussian perturbations is to consider *truncated Gaussian perturbations*. In such a perturbation model, the coordinates of each point are initially chosen from $[0, 1]^d$ and then perturbed by adding Gaussian random variables with some standard deviation σ to them that are conditioned to lie in $[-\alpha, \alpha]$ for some $\alpha \geq 1$. The maximal density of such truncated Gaussian random variables for $\sigma \leq 1$ is bounded from above by

$$\frac{1/(\sigma\sqrt{2\pi})}{1 - \sigma \cdot \exp(-\alpha^2/(2\sigma^2))} \ .$$

After such a truncated perturbation, all points lie in the hypercube $[-\alpha, 1+\alpha]^d$. Hence, one can apply Theorems 1.3.2, 1.3.3, and 1.3.4 with

$$\phi = \frac{(2\alpha+1)^d}{(\sigma\sqrt{2\pi} - \sigma^2\sqrt{2\pi}\exp(-\alpha^2/(2\sigma^2)))^d} = O\left(\frac{\alpha^d}{\sigma^d}\right) \quad .$$

It is not necessary to truncate the Gaussian random variables if the standard deviation is small enough. For $\sigma \leq \min\{\alpha/\sqrt{2(n+1)\ln n + 2\ln d}, 1\}$, the probability that one of the Gaussian random variables has an absolute value larger than $\alpha \geq 1$ is bounded from above by n^{-n}. In this case, even if one does not truncate the random variables, Theorems 1.3.2, 1.3.3, and 1.3.4 can be applied with $\phi = O(\alpha^d/\sigma^d)$. To see this, it suffices to observe that the worst-case bound for the number of 2-changes is $(n!)$ and the worst-case approximation ratio is $O(\log n)$ [CKT99]. Multiplying these values with the failure probability of n^{-n} constitutes less than 1 to the expected values. In particular, this implies that the expected length of the longest path in the transition graph is bounded by $O(\text{poly}(n, 1/\sigma))$.

Chapter 5

Conclusions and Open Problems

In a standard combinatorial optimization problem, a decision-maker has to optimize an objective function over a set of feasible solutions. Even though this model is very appealing from a theoretical point of view, decision-making in practice is often more complex for various reasons such as the presence of selfish agents and multiple objectives. Additional difficulties arise because many combinatorial optimization problems are intractable if $\mathsf{P} \neq \mathsf{NP}$. In order to cope with these difficulties, several solution concepts have been introduced, including Nash equilibria, Pareto-optimal solutions, and local optima, which we have considered in this thesis. In particular for the last two mentioned solution concepts, previous analyses left a wide gap between theoretical and practical results, the reason being that the theoretical results are overly pessimistic as they are based on the typical worst-case perspective of theoretical computer science. We have narrowed the gap between theory and practice by considering Pareto-optimal solutions and local optima in the probabilistic framework of smoothed analysis, which combines features of typical worst-case and average-case analyses. Now that we can partly explain the good behavior in practice, an interesting direction for future research is the question whether the insights gained from the theoretical analyses can be used to further improve the behavior of the considered heuristics in practice.

We believe that another very interesting and challenging direction for future research is the combination of smoothed analysis and algorithmic game theory. We have shown that computing a pure Nash equilibrium is PLS-complete for various classes of congestion games and that coordination in two-sided markets is necessary since the random better and best response dynamics can take an exponential number of steps to reach an equilibrium. These are, however, worst-case results based on carefully designed instances. Due to preliminary experimental results (see also [PS06]), we believe that in typical instances, the best response dynamics reaches a pure Nash equilibrium quickly in congestion games as well as in two-sided markets. There has been a significant amount of research on properties of Nash equilibria such as the price of anarchy (see, e.g., [CK05]) and the price of stability (see, e.g., [ADK+04]). These properties have also been studied almost exclusively from a worst-case perspective. Studying them in an appropriate probabilistic model might yield new insights.

In the following, we summarize some concrete open questions concerning the considered solutions concepts.

5.1 Nash Equilibria

We have shown that computing a pure Nash equilibrium in threshold congestion games is PLS-complete. The strategy spaces in such congestion games are extremely simple as each player has just two options: she can either allocate her private resource or a fixed subset of resources. Let k denote the cardinality of the latter subset (without loss of generality, we can assume that this cardinality is the same for every player). In our reductions, k has to grow linearly with the number of players. It is an open question how large k has to be chosen in order to prove PLS-completeness. Since we have shown that finding a Nash equilibrium in threshold congestion games is PLS-complete by a reduction from Max-Cut, this question is closely related to the minimal node degree for which Max-Cut is PLS-complete. To the best of our knowledge, there is still a considerable gap between the known results: On the one hand, the degree of the vertices in the Max-Cut instances constructed in the PLS-completeness proof in [JPY88] grows linearly with the number of vertices. On the other hand, Poljak [Pol95] gives a polynomial time algorithm to find a locally optimal partition for cubic graphs.

In our model of congestion games with priorities, players displace other players with lower priorities. As we have already mentioned in the introduction, this is only reasonable if players control streams of jobs rather than single ones. Hence, we believe that finding and analyzing different models in which jobs are only slowed down by jobs with higher priorities, that is, models in which they incur a large but finite delay, might yield new insights into scheduling scenarios in which each player has only one job that is to be processed.

It might also yield new insights to study other solution concepts than Nash equilibria. For example, in many situations that can be modeled as congestion games or two-sided matching markets, it is unreasonable to assume that every player has complete knowledge about the current state and the other players' behavior and preferences. In such situations, one might, for instance, study Bayesian Nash equilibria and consider questions similar to the ones that we have considered for Nash equilibria.

5.2 Pareto-Optimal Solutions

We have proven the first bounds on the expected number of Pareto-optimal solutions in bicriteria integer optimization problems. These results can be seen as an explanation why, in practice, enumerating the Pareto set is often feasible even for large-scale instances of many optimization problems. In particular, our results imply that if an algorithm for a bicriteria problem has a worst-case running time of $O(\text{poly}(n) \cdot q)$, where n denotes the input size and q the number of Pareto-optimal solutions, then its expected running time on semi-random instances is polynomial. However, if the worst-case running time grows superlinearly with the number of Pareto-optimal solutions (e.g., $\Theta(\text{poly}(n) \cdot q^2)$), then our bound on the expected number of Pareto-optimal solutions does not imply expected polynomial running time. An interesting question is whether also the variance is polynomially bounded. In the affirmative case, this would imply that also algorithms whose running times depend quadratically on the number of Pareto-optimal solutions have an expected polynomial running time.

In practice, often problems with more than two criteria occur. We conjecture that also for these problems, the expected number of Pareto-optimal solutions is polynomially bounded, where the degree of the polynomial grows with the number of criteria. We believe that bounding the expected number of Pareto-optimal solutions for multi-criteria optimization problems is a challenging open problem.

5.3 Local Optima

We have shown several new results on the running time and the approximation ratio of the 2-Opt heuristic. However, there is still a variety of open problems regarding this algorithm. Our lower bounds only show that there exist families of instances on which 2-Opt takes an exponential number of steps if it uses a particular pivot rule. It would be interesting to analyze the diameter of the transition graph and to either present instances on which every pivot rule needs an exponential number of steps or to prove that there is always an improvement sequence of polynomial length to a locally optimal solution. Also the worst number of local improvements for some natural pivot rules like, e.g., the one that always makes the largest possible improvement or the one that always chooses a random improving 2-change, is not known yet. Furthermore, the complexity of computing locally optimal solutions is open. The only result in this regard is due to Krentel [Kre89] who shows that it is PLS-complete to compute a local optimum for the metric TSP for k-Opt for some constant k. It is not known whether his construction can be embedded into the Euclidean metric and whether it is PLS-complete to compute locally optimal solutions for 2-Opt. Fischer and Torenvliet [FT95] show, however, that for the general TSP, it is PSPACE-hard to compute a local optimum for 2-Opt that is reachable from a given initial tour.

The obvious open question concerning the probabilistic analysis is how the gap between experiments and theory can be narrowed further. In order to tackle this question, new methods seem to be necessary. Our approach, which is solely based on analyzing the smallest improvement made by a sequence of linked 2-changes, seems to yield too pessimistic bounds. Another interesting area to explore is the expected approximation ratio of 2-Opt. In experiments, approximation ratios close to 1 are observed. For instances that are chosen uniformly at random, the bound on the expected approximation ratio is a constant but unfortunately a large one. It seems to be a very challenging problem to improve this constant to a value that matches the experimental results.

Besides 2-Opt, there are also other local search algorithms that are successful for the traveling salesperson problem. In particular, the Lin-Kernighan heuristic [LK73] is one of the most successful local search algorithm for the symmetric TSP. It is a variant of k-Opt in which k is not fixed and it can roughly be described as follows: Each local modification starts by removing one edge $\{a, b\}$ from the current tour, which results in a Hamiltonian path with the two endpoints a and b. Then an edge $\{b, c\}$ is added, which forms a cycle; there is a unique edge $\{c, d\}$ incident to c whose removal breaks the cycle, producing a new Hamiltonian path with endpoints a and d. This operation is called a rotation. Now either a new Hamiltonian cycle can be obtained by adding the edge $\{a, d\}$ to the tour or another rotation can be performed. There are a lot of different variants and heuristic

improvements of this basic scheme, but little is known theoretically. Papadimitriou [Pap92] shows for a variant of the Lin-Kernighan heuristic that computing a local optimum is PLS-complete, which is a sharp contrast to the experimental results. Since the Lin-Kernighan heuristic is widely used in practice, a theoretical explanation for its good behavior in practice is of great interest. Our analysis of 2-Opt relies crucially on the fact that there is only a polynomial number of different 2-changes. For the Lin-Kernighan heuristic, however, the number of different local improvements is exponential. Hence, it is an interesting question whether nonetheless the smallest possible improvement is polynomially large or whether different methods yield a polynomial upper bound on the expected running time of the Lin-Kernighan heuristic.

Another interesting question concerning the analysis of local search algorithms is raised by Arthur and Vassilvitskii [AV06b]. They consider the k-means algorithm, a well-known clustering algorithm, and they show that its expected running time is polynomially bounded in n^k and σ^{-1} for semi-random inputs with n points that are perturbed by adding Gaussian random vectors with standard deviation σ. Experiments suggest, however, that the degree of the polynomial does not depend on the number k of clusters. A theoretical explanation of this behavior is of great interest because the k-means algorithm is widely used in practice.

Appendix A

Some Facts about Matroids

In this chapter, we briefly introduce *matroids* and we summarize the properties that we used in Sections 2.2.3 and 2.3.3. For a detailed introduction to matroids, we refer the reader to Schrijver's book [Sch03].

A pair $(\mathcal{R}, \mathcal{I})$ is called a *matroid* if $\mathcal{R}$ is a finite set and $\mathcal{I} \subseteq 2^{\mathcal{R}}$ is a nonempty collection of subsets of $\mathcal{R}$ satisfying

(i) if $X \in \mathcal{I}$ and $Y \subseteq X$, then $Y \in \mathcal{I}$,

(ii) if $X, Y \in \mathcal{I}$ and $|Y| < |X|$, then there exists an $x \in X$ with $Y \cup \{x\} \in \mathcal{I}$.

Given a matroid $(\mathcal{R}, \mathcal{I})$, we call a set $X \subseteq \mathcal{R}$ *independent* if $X \in \mathcal{I}$, and *dependent* otherwise. A *basis* of a matroid $(\mathcal{R}, \mathcal{I})$ is an inclusion-maximal set $X \in \mathcal{I}$. It is a well-known property of matroids that every basis has the same cardinality, which is called the *rank* of the matroid. A matroid is called *weighted* if a weight function $w\colon \mathcal{R} \to \mathbb{N}$ is given. Usually one is interested in finding a basis of maximal or minimal weight, where the weight of a subset is defined as the sum of the weights of its elements.

In the following, we list some examples of matroids. Some of these examples are rather uninteresting from an optimization point of view but they lead to rich combinatorial structures when various players with possibly different strategy spaces are involved in a congestion game or two-sided market.

- Let $\mathcal{R}$ be a finite set and let $k \in \mathbb{N}$. If the set $\mathcal{I}$ contains all subsets of $\mathcal{R}$ with cardinality at most k, then $(\mathcal{R}, \mathcal{I})$ is a *uniform matroid*.
- Let $\mathcal{R}_1, \dots, \mathcal{R}_l$ be disjoint finite sets, let $\mathcal{R} = \mathcal{R}_1 \cup \dots \cup \mathcal{R}_l$, and let $k_1, \dots, k_l \in \mathbb{N}$. We define that a set $X \subseteq \mathcal{R}$ belongs to $\mathcal{I}$ if $X \cap \mathcal{R}_i \leq k_i$ for every $i \in \{1, \dots, l\}$. The resulting set system $(\mathcal{R}, \mathcal{I})$ is a *partition matroid*.
- Assume that E is the set of edges of some graph $G = (V, E)$ and let $\mathcal{I}$ denote the set of all subsets of edges that do not contain a cycle. The set system $(E, \mathcal{I})$ is a *graphical matroid*.
- Let A be an $m \times n$ matrix. Let $\mathcal{R} = \{1, \dots, n\}$ and let $\mathcal{I}$ be the collection of all those subsets I of $\mathcal{R}$ such that the columns of A with index in I are linearly independent. Then the set system $(\mathcal{R}, \mathcal{I})$ is a *linear matroid*.

In Section 2.2.3, we claim that in any state of a matroid two-sided market, there exists a lazy best response. This follows from the following lemma.

Lemma A.1. *Let a matroid $(\mathcal{R},\mathcal{I})$ with weights $w\colon \mathcal{R} \to \mathbb{N}$ be given and let $B \in \mathcal{I}$. There exists a basis B^* of maximum weight and a sequence $B_1,\dots,B_l$ of independent sets with $B_1 = B$, $B_l = B^*$, and such that $|B_{i+1} \setminus B_i| = 1$ and $w(B_{i+1}) > w(B_i)$ for all $i \in \{1,\dots,l-1\}$.*

Proof. In order to prove the lemma, set $B_1 = B$. If B is a basis of maximum weight, then we are done. Otherwise, the following lemma implies the existence of an independent set B_2 with $|B_2 \setminus B_1| = 1$ and $w(B_2) > w(B_1)$.

Lemma A.2. *(Corollary 39.12b in [Sch03]) Let a matroid $(\mathcal{R},\mathcal{I})$ with weights $w\colon \mathcal{R} \to \mathbb{N}$ be given. An independent set $B \in \mathcal{I}$ is of maximum weight if and only if there exists no independent set $B^* \in \mathcal{I}$ with $|B^* \setminus B| = 1$ and $w(B^*) > w(B)$.*

We can iterate this argument until a basis of maximum weight is reached, yielding a sequence $B_1,\dots,B_l$ of independent sets with the desired properties. □

In the proof of Theorem 2.3.7, we made use of the following lemma.

Lemma A.3. *Let $(\mathcal{R},\mathcal{I})$ be a matroid with weights $w\colon \mathcal{R} \to \mathbb{N}$ and let B be a basis of minimum weight. If the weight of a single resource $r \in B$ is increased such that B is no longer of minimum weight, then, in order to obtain a basis of minimum weight, it suffices to exchange r with a resource $r' \in \mathcal{R}$ of minimum weight such that $B \cup \{r'\} \setminus \{r\}$ is a basis.*

Proof. In order to prove the lemma, we use the following matroid property.

Lemma A.4. *(Corollary 39.12a in [Sch03]). Let $(\mathcal{R},\mathcal{I})$ be a matroid, and let $I,J \in \mathcal{I}$ with $|I| = |J|$ be independent sets. The bipartite graph $G(I\Delta J) = (V,E)$ with $V = (I \setminus J) \cup (J \setminus I)$ and $E = \{\{i,j\} \mid i \in I \setminus J, j \in J \setminus I, I \cup \{j\} \setminus \{i\} \in \mathcal{I}\}$ contains a perfect matching.*

Let B' be a basis of minimum weight w.r.t. the increased weight of r. Let P be a perfect matching of the graph $G(B\Delta B')$ and denote by e the edge from P that contains r. For every edge $\{i,j\} \in P \setminus \{e\}$, it holds $w(i) \leq w(j)$ as, otherwise, if $w(i) > w(j)$, the basis $B \cup \{j\} \setminus \{i\}$ would have smaller weight than B.

Now denote by r' the resource that is matched with r, i.e., the resource such that $e = \{r,r'\} \in P$. As $w(i) \leq w(j)$ for every $\{i,j\} \in P \setminus \{e\}$, the weight of $B \setminus \{r\}$ is bounded from above by the weight of $B' \setminus \{r'\}$. By the definition of the matching P, $B \cup \{r'\} \setminus \{r\}$ is a basis. By our arguments above, the weight of this basis is bounded from above by the weight of B'. Hence, this basis is optimal w.r.t. the increased weight of r. □

In the proofs of Theorem 2.2.11 and 2.3.7, we made use of two standard matroid operations: *deletion* and *contraction*. We describe these operation briefly and refer the reader for more details to [Sch03]. Let $\mathcal{M} = (\mathcal{R},\mathcal{I})$ denote a matroid and let $T \subseteq \mathcal{R}$ be an arbitrary set. The set system $\mathcal{M}' = (\mathcal{R} \setminus T, \mathcal{I}')$ with $\mathcal{I}' = \{X \in \mathcal{I} \mid X \cap T = \emptyset\}$ is a matroid, the so-called *restriction of $\mathcal{M}$ to $\mathcal{R} \setminus T$*. We say that $\mathcal{M}'$ is obtained from $\mathcal{M}$ by *deleting* the elements in T.

Let $\mathcal{M} = (\mathcal{R},\mathcal{I})$ denote a matroid, let $T \subseteq \mathcal{R}$ be arbitrary, and let $B \subseteq T$ denote an inclusion-maximal independent subset of T, that is, $B \in \mathcal{I}$ and $B \cup \{x\} \notin \mathcal{I}$ for all $x \in T \setminus B$. The set system $(\mathcal{R} \setminus T, \mathcal{I}')$ with $\mathcal{I}' = \{X \subseteq \mathcal{R} \setminus T \mid X \cup B \in \mathcal{I}\}$ is a matroid, the so-called *contraction* M/T of M by T.

Chapter B

Some Probability Theory

B.1 Continuous Random Variables

In this section, we give a brief introduction to continuous random variables. We do not give a formal treatment, which would include analytical difficulties such as measurability questions, but our aim is to present an intuitive description of the notions that are used in this thesis. For a formal treatment, we refer the reader to the book by Feller [Fel91].

A random variable X that takes only values from a countable set, say $\mathbb{N}$, can be described by its *probability mass function* $f\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$, which satisfies $\sum_{i\in\mathbb{N}} f(i) = 1$. Given a natural number $i \in \mathbb{N}$, the probability that X takes the value i equals the value $f(i)$, i.e., $\mathbf{Pr}[X = i] = f(i)$. Random variables whose possible outcomes form an uncountable set, say $\mathbb{R}$, cannot be described by a probability mass function because it might be the case that the probability of the event $X = a$ is 0 for every $a \in \mathbb{R}$. Instead such random variables are described by a *distribution function* $F\colon \mathbb{R} \to [0,1]$, which has to be non-decreasing, right-continuous, and which has to satisfy $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = 1$. Given such a distribution function, the probability of the event $X \in [a,b]$ is $F(b) - F(a)$ for every interval $[a,b]$. If the distribution F is continuous, we say that X is a *continuous random variable*. Given a random variable X with distribution F, we say that a function $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a *density* of X if for all $a \in \mathbb{R}$,

$$F(a) = \int_{-\infty}^{a} f(x)\,dx \ . \tag{B.1.1}$$

Similarly, one can also define distributions and densities for *random vectors* that take values in $\mathbb{R}^d$. Let $F\colon \mathbb{R}^d \to [0,1]$ be a distribution and let $f\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be a density of a random vector X. Then for all $a \in \mathbb{R}^d$,

$$\mathbf{Pr}[X \leq a] = F(a) = \int_{-\infty}^{a_1} \cdots \int_{-\infty}^{a_d} f(x_1,\ldots,x_d)\,dx_d \ldots dx_1 \ .$$

The continuous random variables occurring in this thesis are usually described by density functions. An important consequence of (B.1.1) that we use several times implicitly is that if the density function of a random variable X is bounded from above by some value $\phi > 0$, then the probability that X takes a value in a fixed interval of length ε is bounded from above by $\varepsilon\phi$. In the following, we list a few important examples of continuous random variables.

- A random variable is said to be *uniformly distributed* over some interval $[a, b]$ if it can be described by the density function $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ with

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } x \in [a, b], \\ 0 & \text{otherwise.} \end{cases}$$

- A *Gaussian* or *normal distribution* with mean μ and standard deviation σ can be described by the density function $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ with

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \ .$$

  At some places, we use the notion of a *d-dimensional Gaussian random vector* with mean $\mu \in \mathbb{R}^d$ and standard deviation σ. In that cases, we mean a d-dimensional random vector whose i-th entry is an independent Gaussian random variable with mean μ_i and standard deviation σ

- The *exponential distribution* with rate parameter λ can be described by the density function $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ with

$$f(x) = \begin{cases} \lambda \cdot \exp(-\lambda x) & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$$

Given two random variables X and Y with densities f_X and f_Y, respectively, it is sometimes important to compute the density f_{X+Y} of the random variable $X + Y$. In this thesis, this problem occurs in Section 4.3.2. This density can be obtained as *convolution* of f_X and f_Y, that is, for every $a \in \mathbb{R}$,

$$f_{X+Y}(a) = \int_{-\infty}^{\infty} f_X(x) \cdot f_Y(a-x)\, dx \ .$$

An important technique used in the proof of Lemma B.3.1 is a *change of variables*. Let $X = (X_1, \ldots, X_d)$ denote a random vector whose entries have a joint density $f\colon \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, let $\Phi\colon \mathbb{R}^d \to \mathbb{R}^d$ denote a bijection, and let Y denote the random vector $\Phi(X)$. The following lemma states how the density of Y can be computed from the density f of X and the bijection Φ.

Lemma B.1.1. *Let g denote the density of the random vector Y. For all $y \in \mathbb{R}^d$,*

$$g(y) = \left|\det\left(\frac{\partial \Phi^{-1}(y)}{\partial y}\right)\right| \cdot f\left(\Phi^{-1}(y)\right) \ ,$$

where $\left(\frac{\partial \Phi^{-1}(y)}{\partial y}\right)$ denotes the Jacobian matrix *of Φ^{-1}, i.e.,*

$$\left(\frac{\partial \Phi^{-1}(y)}{\partial y}\right) = \begin{pmatrix} \frac{\partial \Phi_1^{-1}(y)}{\partial y_1} & \cdots & \frac{\partial \Phi_1^{-1}(y)}{\partial y_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_d^{-1}(y)}{\partial y_1} & \cdots & \frac{\partial \Phi_d^{-1}(y)}{\partial y_d} \end{pmatrix} \ .$$

B.2 Weighted Chernoff Bound

Typically, Chernoff bounds are formulated for sums of independent Bernoulli trials. In this section, we derive a Chernoff bound for general discrete random variables. The proof is based very closely on the one for sums of Bernoulli trials in [MU05]. In fact, the only part which needs to be exchanged is an upper bound on the moment generating function.

For a random variable X, let $M_X(t) = \mathbf{E}\left[e^{tX}\right]$ denote its *moment generating function*. Assume that X is the sum of independent random variables $X_1, \ldots, X_n$, where each X_i is a discrete random variable taking only values in $[0, 1]$. Fix an index i and consider the random variable X_i, and let $p\colon W \to \mathbb{R}_{\geq 0}$ be its distribution, where $W \subseteq [0, 1]$ is a countable set. We can write the moment generating function of X_i as follows:

$$\begin{aligned} M_{X_i}(t) &= \mathbf{E}\left[e^{tX_i}\right] \\ &= \sum_{w \in W} p(w) \cdot e^{tw} \\ &\leq \sum_{w \in W} p(w) \cdot (we^t + 1 - w) \ , \end{aligned} \tag{B.2.1}$$

where the last inequality follows from the convexity of the function $f(x) = e^{tx}$ because

$$e^{tw} = f(w) = f(w \cdot 1 + (1 - w) \cdot 0) \leq w \cdot f(1) + (1 - w) \cdot f(0) = w \cdot e^t + 1 - w \ .$$

Inequality (B.2.1) yields

$$\begin{aligned} M_{X_i}(t) &\leq 1 + \sum_{w \in W} p(w) \cdot w \cdot (e^t - 1) \\ &= 1 + \mathbf{E}[X_i] \cdot (e^t - 1) \\ &\leq \exp\left(\mathbf{E}[X_i] \cdot (e^t - 1)\right) \ , \end{aligned}$$

where the last inequality follows from the fact that for any $y \in \mathbb{R}$, $1 + y \leq e^y$.

Since the random variables $X_1, \ldots, X_n$ are assumed to be independent, the moment generating function of X is simply the product of the moment generating functions of the X_i's. Hence, we obtain

$$\begin{aligned} M_X(t) &= \prod_{i=1}^{n} M_{X_i}(t) \\ &\leq \prod_{i=1}^{n} \exp\left(\mathbf{E}[X_i] \cdot (e^t - 1)\right) \\ &\leq \exp\left(\sum_{i=1}^{n} \mathbf{E}[X_i] \cdot (e^t - 1)\right) \\ &= \exp\left(\mathbf{E}[X] \cdot (e^t - 1)\right) \ . \end{aligned}$$

Now we are ready to prove the following Chernoff bound.

Theorem B.2.1. *Let $X_1, \ldots, X_n$ be independent discrete random variables with values in $[0,1]$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. Then for every $x > 0$,*

$$\mathbf{Pr}[X \geq x] < \left(\frac{e \cdot \mu}{x}\right)^x \quad .$$

Proof. Applying Markov's inequality yields for any $\delta > 0$ and $t > 0$,

$$\begin{aligned}\mathbf{Pr}[X \geq (1+\delta)\mu] &= \mathbf{Pr}\left[e^{tX} \geq e^{t(1+\delta)\mu}\right] \\ &\leq \frac{\mathbf{E}\left[e^{tX}\right]}{e^{t(1+\delta)\mu}} \\ &\leq \frac{e^{(e^t-1)\mu}}{e^{t(1+\delta)\mu}} \quad .\end{aligned}$$

For any $\delta > 0$, we can set $t = \ln(1+\delta) > 0$ to get

$$\mathbf{Pr}[X \geq (1+\delta) \cdot \mu] \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu \quad .$$

This yields the theorem since

$$\begin{aligned}\mathbf{Pr}[X \geq t] &= \mathbf{Pr}\left[X \geq \left(1 + \left(\frac{x}{\mu} - 1\right)\right) \cdot \mu\right] \\ &\leq \left(\frac{e^{x/\mu - 1}}{(x/\mu)^{x/\mu}}\right)^\mu \\ &\leq \left(\frac{e \cdot \mu}{x}\right)^x \quad .\end{aligned}$$

□

By appropriate scaling, we obtain the following variant of Theorem B.2.1.

Corollary B.2.2. *Let $X_1, \ldots, X_n$ be independent discrete random variables with values in $[0,z]$ for some $z > 0$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. Then for every $x > 0$,*

$$\mathbf{Pr}[X \geq x] < \left(\frac{e \cdot \mu}{x}\right)^{x/z} \quad .$$

By similar calculations, one obtains the following corollary.

Corollary B.2.3. *Let $X_1, \ldots, X_n$ be independent discrete random variables with values in $[0,z]$ for some $z > 0$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbf{E}[X]$. Then for every $x > 0$,*

$$\mathbf{Pr}[X \leq x] < \left(\frac{e^{1-\mu/x} \cdot \mu}{x}\right)^{x/z} \quad .$$

B.3 Linear Combinations of Random Variables

Lemma B.3.1. *Let $X^1, \dots, X^n$ be independent d-dimensional random row vectors, and, for $i \in [n]$ and some $\phi \geq 1$, let $f_i \colon [0,1]^d \to [0, \phi]$ denote the joint density of the entries of X^i. Furthermore, let $\lambda^1, \dots, \lambda^k \in \mathbb{Z}^{dn}$ be linearly independent row vectors. For $i \in [n]$ and a fixed $\varepsilon \geq 0$, we denote by $\mathcal{A}_i$ the event that $\lambda^i \cdot X$ takes a value in the interval $[0, \varepsilon]$, where X denotes the vector $X = (X^1, \dots, X^n)^{\mathrm{T}}$. Under these assumptions,*

$$\mathbf{Pr}\left[\bigcap_{i=1}^{k} \mathcal{A}_i\right] \leq (\varepsilon\phi)^k \ .$$

Proof. The main tool for proving the lemma is a change of variables. Instead of using the canonical basis of the dn-dimensional vector space $\mathbb{R}^{dn}$, we use the given linear combinations as basis vectors. To be more precise, the basis $\mathcal{B}$ that we use consists of two parts: it contains the vectors $\lambda^1, \dots, \lambda^k$ and it is completed by some vectors from the canonical basis $\{e^1, \dots, e^{dn}\}$, where e^i denotes the i-th canonical row vector, i.e., $e^i_i = 1$ and $e^i_j = 0$ for $j \neq i$. That is, the basis $\mathcal{B}$ can be written as $\{\lambda^1, \dots, \lambda^k, e^{\pi(1)}, \dots, e^{\pi(dn-k)}\}$, for some injective function $\pi \colon [dn-k] \to [dn]$.

Let $\Phi \colon \mathbb{R}^{dn} \to \mathbb{R}^{dn}$ be defined by $\Phi(x) = Ax$, where A denotes the $(dn) \times (dn)$-matrix

$$\begin{pmatrix} \lambda^1 \\ \vdots \\ \lambda^k \\ e^{\pi(1)} \\ \vdots \\ e^{\pi(dn-k)} \end{pmatrix} .$$

Since $\mathcal{B}$ is a basis of $\mathbb{R}^{dn}$, the function Φ is a bijection. We define the vector $Y = (Y_1, \dots, Y_{dn})^{\mathrm{T}}$ as $Y = \Phi(X)$, and for $i \in [n]$, we denote by Y^i the vector $(Y_{d(i-1)+1}, \dots, Y_{di})$. Let $f \colon \mathbb{R}^{dn} \to \mathbb{R}$ denote the joint density of the entries of the random vectors $X^1, \dots, X^n$, and let $g \colon \mathbb{R}^{dn} \to \mathbb{R}$ denote the joint density of the entries of the random vectors $Y^1, \dots, Y^n$. Due to the independence of the random vectors $X^1, \dots, X^n$, we have $f(x_1, \dots, x_{dn}) = f_1(x_1, \dots, x_d) \cdot \dots \cdot f_n(x_{d(n-1)+1}, \dots, x_{dn})$. We can express the joint density g as

$$g(y_1, \dots, y_{dn}) = |\det_{\partial} \Phi^{-1}(y_1, \dots, y_{dn})| \cdot f(\Phi^{-1}(y_1, \dots, y_{dn})) \ ,$$

where $\det_{\partial}$ denotes the determinant of the Jacobian matrix of Φ^{-1}.

The matrix A is invertible as $\mathcal{B}$ is a basis of $\mathbb{R}^{dn}$. Hence, for $y \in \mathbb{R}^{dn}$, $\Phi^{-1}(y) = A^{-1}y$ and the Jacobian matrix of Φ^{-1} equals A^{-1}. Thus, $\det_{\partial} \Phi^{-1} = \det A^{-1} = (\det A)^{-1}$. Since all entries of A are integers, also its determinant must be an integer, and since it has rank dn, we know that $\det A \neq 0$. Hence, $|\det A| \geq 1$ and $|\det A^{-1}| \leq 1$. For $y \in \mathbb{R}^{dn}$, we decompose $\Phi^{-1}(y) \in \mathbb{R}^{dn}$ into n subvectors with d entries each, i.e., $\Phi^{-1}(y) = (\Phi_1^{-1}(y), \dots, \Phi_n^{-1}(y))$ with $\Phi_i^{-1}(y) \in \mathbb{R}^d$ for $i \in [n]$. This yields

$$g(y) = |\det A^{-1}| \cdot f(\Phi^{-1}(y)) \leq f_1(\Phi_1^{-1}(y)) \cdot \dots \cdot f_n(\Phi_n^{-1}(y)) \ .$$

The probability we want to estimate can be written as

$$\mathbf{Pr}\left[\bigcap_{i=1}^{k}\mathcal{A}_i\right] = \int_{y_1=0}^{\varepsilon}\cdots\int_{y_k=0}^{\varepsilon}\int_{y_{k+1}=-\infty}^{\infty}\cdots\int_{y_{dn}=-\infty}^{\infty} g(y_1,\ldots,y_{dn})\,dy_{dn}\ldots dy_1 \ . \tag{B.3.1}$$

Since all entries of the vectors $X^1,\ldots,X^n$ take only values in the interval $[0,1]$ and since for $i \in \{k+1,\ldots,dn\}$, the random variable Y_i coincides with one of these entries, (B.3.1) simplifies to

$$\mathbf{Pr}\left[\bigcap_{i=1}^{k}\mathcal{A}_i\right] = \int_{y_1=0}^{\varepsilon}\cdots\int_{y_k=0}^{\varepsilon}\int_{y_{k+1}=0}^{1}\cdots\int_{y_{dn}=0}^{1} g(y_1,\ldots,y_{dn})\,dy_{dn}\ldots dy_1 \ . \tag{B.3.2}$$

Without loss of generality, we assume that $\{i \mid \nexists j \in [dn-k]\colon \pi(j)=i\} \subseteq [dk]$, i.e., only vectors e^i from the canonical basis with $i \le dk$ are replaced by the vectors $\lambda^1,\ldots,\lambda^k$ in the basis $\mathcal{B}$. Furthermore, we can assume w.l.o.g. $\pi(i)=i$, for $i > dk$. Under these assumptions, the density g can be upper bounded as follows:

$$g(y_1,\ldots,y_{dn}) \le \phi^k \cdot f_{k+1}(y_{dk+1},\ldots,y_{d(k+1)})\cdot\cdots\cdot f_n(y_{d(n-1)+1},\ldots,y_{dn}) \ . \tag{B.3.3}$$

Putting together (B.3.2) and (B.3.3) yields

$$\begin{aligned}\mathbf{Pr}\left[\bigcap_{i=1}^{k}\mathcal{A}_i\right] &\le (\varepsilon\phi)^k \cdot \int_{y_{dk+1}=0}^{1}\cdots\int_{y_{d(k+1)}=0}^{1} f_{k+1}(y_{dk+1},\ldots,y_{d(k+1)}) \\ &\qquad \cdots\int_{y_{d(n-1)+1}=0}^{1}\int_{y_{dn}=0}^{1} f_n(y_{d(n-1)+1},\ldots,y_{dn})\,dy_{dn}\ldots dy_{dk+1} \\ &= (\varepsilon\phi)^k \ ,\end{aligned}$$

where the last equation follows because $f_{k+1},\ldots,f_n$ are density functions. □

B.4 Proofs of some Lemmas from Section 4.3.2

B.4.1 Proof of Lemma 4.3.7

Let $a,c \in (0,C]$ for some $C>0$. In the following proof, we use the following two identities:

$$\int_0^c \frac{1}{\sqrt{z(c-z)}}\,dz = \pi$$

and

$$\begin{aligned}\int_0^a \frac{1}{\sqrt{z(z+c)}}\,dz &= \ln\left(\frac{c}{2}+a+\sqrt{a(a+c)}\right)+\ln\left(\frac{2}{c}\right) \\ &\le \ln\left(\frac{3}{2}c+2a\right)+\ln\left(\frac{2}{c}\right) \le \ln(4C)+\ln\left(\frac{2}{c}\right) \ .\end{aligned}$$

Proof of Lemma 4.3.7. The conditional density of Δ can be calculated as convolution of the conditional densities of Z_1 and Z_2 as follows:

$$f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta) = \int_{-\infty}^{\infty} f_{Z|T=\tau,R=r_1}(z)\cdot f_{Z|T=\tau,R=r_2}(z-\delta)\,dz \ .$$

In order to estimate this integral, we distinguish between several cases. In the following, let κ denote a sufficiently large constant.

First case: $\tau \le r_1$ and $\tau \le r_2$.
As Z_i takes only values in the interval $[-\tau, \tau]$, we can assume $0 < \delta \le \min\{1/2, 2\tau\}$ and

$$f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta) = \int_{-\tau+\delta}^{\tau} f_{Z|T=\tau,R=r_1}(z) \cdot f_{Z|T=\tau,R=r_2}(z-\delta)\, dz \ .$$

Due to Lemma 4.3.6, we can estimate the densities of Z_1 and Z_2 by

$$f_{Z|T=\tau,R=r_i}(z) \le \sqrt{\frac{2}{\tau^2 - z^2}} \le \sqrt{\frac{2}{\tau(\tau - |z|)}} \le \sqrt{\frac{2}{\tau}} \left(\frac{1}{\sqrt{\tau - z}} + \frac{1}{\sqrt{\tau + z}} \right) \ .$$

For $\delta \in (0, \min\{1/2, 2\tau\}]$, we obtain the following upper bound on the density of Δ:

$$\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta) \\
\le &\frac{2}{\tau} \int_{-\tau+\delta}^{\tau} \left(\frac{1}{\sqrt{\tau - z}} + \frac{1}{\sqrt{\tau + z}} \right) \left(\frac{1}{\sqrt{\tau - z + \delta}} + \frac{1}{\sqrt{\tau + z - \delta}} \right) dz \\
= &\frac{2}{\tau} \left(\int_0^{2\tau-\delta} \frac{1}{\sqrt{z(z+\delta)}}\, dz + \int_{\delta}^{2\tau} \frac{1}{\sqrt{z(2\tau+\delta-z)}}\, dz \right. \\
&\left. + \int_0^{2\tau-\delta} \frac{1}{\sqrt{z(2\tau-\delta-z)}}\, dz + \int_0^{2\tau-\delta} \frac{1}{\sqrt{z(z+\delta)}}\, dz \right) \\
\le &\frac{2}{\tau} \left(2\pi + 2\ln(8\sqrt{d}) + 2\ln\left(2\delta^{-1}\right) \right) \le \frac{\kappa}{\tau} \cdot \ln \delta^{-1} \ .
\end{aligned}$$

Second case: $r_1 \le \tau$ and $r_2 \le \tau$.
Since Z_i takes only values in the interval $[-\tau, 2r_i - \tau]$, we can assume $0 < \delta \le \min\{1/2, 2r_1\}$ and

$$f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta) = \int_{-\tau+\delta}^{\min\{2r_1-\tau, 2r_2-\tau+\delta\}} f_{Z|T=\tau,R=r_1}(z) \cdot f_{Z|T=\tau,R=r_2}(z-\delta)\, dz \ .$$

Due to Lemma 4.3.6, we can estimate the densities of Z_1 and Z_2 by

$$\begin{aligned}
f_{Z|T=\tau,R_i=r_i}(z) \le &\sqrt{\frac{2}{(\tau+z)(2r_i-\tau-z)}} \le \begin{cases} \sqrt{\frac{2}{r_i(\tau+z)}} & \text{if } z \le r_i - \tau \\ \sqrt{\frac{2}{r_i(2r_i-\tau-z)}} & \text{if } z \ge r_i - \tau \end{cases} \\
\le &\sqrt{\frac{2}{r_i}} \left(\frac{1}{\sqrt{\tau+z}} + \frac{1}{\sqrt{2r_i-\tau-z}} \right) \ .
\end{aligned}$$

Case 2.1: $\delta \in (\max\{0, 2(r_1 - r_2)\}, 2r_1]$.

We obtain the following upper bound on the density of Δ:

$$\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\\
\leq&\tfrac{2}{\sqrt{r_1 r_2}}\int_{-\tau+\delta}^{2r_1-\tau}\left(\tfrac{1}{\sqrt{\tau+z}}+\tfrac{1}{\sqrt{2r_1-\tau-z}}\right)\left(\tfrac{1}{\sqrt{\tau+z-\delta}}+\tfrac{1}{\sqrt{2r_2-\tau-z+\delta}}\right)dz\\
=&\frac{2}{\sqrt{r_1 r_2}}\left(\int_0^{2r_1-\delta}\frac{1}{\sqrt{z(z+\delta)}}\,dz+\int_0^{2r_1-\delta}\frac{1}{\sqrt{z(2r_1-\delta-z)}}\,dz\right.\\
&\left.+\int_\delta^{2r_1}\frac{1}{\sqrt{z(2r_2+\delta-z)}}\,dz+\int_0^{2r_1-\delta}\frac{1}{\sqrt{z(2(r_2-r_1)+\delta+z)}}\,dz\right)\\
\leq&\frac{2}{\sqrt{r_1 r_2}}\left(2\pi+2\ln(8\sqrt{d})+\ln\left(2\delta^{-1}\right)+\ln\left(2(2(r_2-r_1)+\delta)^{-1}\right)\right)\\
\leq&\frac{\kappa}{\sqrt{r_1 r_2}}\left(\ln\delta^{-1}+\ln\left((2(r_2-r_1)+\delta)^{-1}\right)+\kappa\right)\quad.
\end{aligned}$$

Case 2.2: $\delta\in(0,\max\{0,2(r_1-r_2)\}]$**.**
We obtain the following upper bound on the density of Δ:

$$\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\\
\leq&\tfrac{2}{\sqrt{r_1 r_2}}\int_{-\tau+\delta}^{2r_2-\tau+\delta}\left(\tfrac{1}{\sqrt{\tau+z}}+\tfrac{1}{\sqrt{2r_1-\tau-z}}\right)\left(\tfrac{1}{\sqrt{\tau+z-\delta}}+\tfrac{1}{\sqrt{2r_2-\tau-z+\delta}}\right)dz\\
=&\frac{2}{\sqrt{r_1 r_2}}\left(\int_0^{2r_2}\frac{1}{\sqrt{z(z+\delta)}}\,dz+\int_0^{2r_2}\frac{1}{\sqrt{z(2r_1-\delta-z)}}\,dz\right.\\
&\left.+\int_0^{2r_2}\frac{1}{\sqrt{z(2r_2+\delta-z)}}\,dz+\int_0^{2r_2}\frac{1}{\sqrt{z(2(r_1-r_2)-\delta+z)}}\,dz\right)\\
\leq&\frac{2}{\sqrt{r_1 r_2}}\left(2\pi+2\ln(8\sqrt{d})+\ln\left(2\delta^{-1}\right)+\ln\left(2(2(r_1-r_2)-\delta)^{-1}\right)\right)\\
\leq&\frac{\kappa}{\sqrt{r_1 r_2}}\left(\ln\delta^{-1}+\ln\left((2(r_1-r_2)-\delta)^{-1}\right)+\kappa\right)\quad.
\end{aligned}$$

Third case: $r_1\leq\tau\leq r_2$**.**
Since Z_1 takes only values in the interval $[-\tau,2r_1-\tau]$ and Z_2 takes only values in the interval $[-\tau,\tau]$, we can assume $0<\delta\leq\min\{1/2,2r_1\}$ and

$$f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)=\int_{-\tau+\delta}^{2r_1-\tau}f_{Z|T=\tau,R=r_1}(z)\cdot f_{Z|T=\tau,R=r_2}(z-\delta)\,dz\quad.$$

For $\delta\in(0,\min\{1/2,2r_1\}]$, we obtain the following upper bound on the density of

Δ:

$$
\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\\
\leq&\frac{2}{\sqrt{\tau r_1}}\int_{-\tau+\delta}^{2r_1-\tau}\left(\frac{1}{\sqrt{\tau+z}}+\frac{1}{\sqrt{2r_1-\tau-z}}\right)\left(\frac{1}{\sqrt{\tau-z+\delta}}+\frac{1}{\sqrt{\tau+z-\delta}}\right)dz\\
=&\frac{2}{\sqrt{\tau r_1}}\left(\int_{\delta}^{2r_1}\frac{1}{\sqrt{z(2\tau+\delta-z)}}\,dz+\int_{0}^{2r_1-\delta}\frac{1}{\sqrt{z(2(\tau-r_1)+\delta+z)}}\,dz\right.\\
&\left.+\int_{0}^{2r_1-\delta}\frac{1}{\sqrt{z(z+\delta)}}\,dz+\int_{0}^{2r_1-\delta}\frac{1}{\sqrt{z(2r_1-\delta-z)}}\,dz\right)\\
\leq&\frac{2}{\sqrt{\tau r_1}}\left(2\pi+2\ln(8\sqrt{d})+\ln\left(2\delta^{-1}\right)+\ln\left(2(2(\tau-r_1)+\delta)^{-1}\right)\right)\\
\leq&\frac{\kappa}{\sqrt{\tau r_1}}\cdot\ln\delta^{-1}\ .
\end{aligned}
$$

Fourth case: $r_2\leq\tau\leq r_1$**.**
Since Z_1 takes only values in the interval $[-\tau,\tau]$ and Z_2 takes only values in the interval $[-\tau,2r_2-\tau]$, we can assume $0<\delta\leq\min\{1/2,2\tau\}$ and

$$
f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)=\int_{-\tau+\delta}^{\min\{2r_2-\tau+\delta,\tau\}}f_{Z|T=\tau,R=r_1}(z)\cdot f_{Z|T=\tau,R=r_2}(z-\delta)\,dz\ .
$$

Case 4.1: $\delta\in(0,2(\tau-r_2)]$**.**
We obtain the following upper bound on the density of Δ:

$$
\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\\
\leq&\frac{2}{\sqrt{\tau r_2}}\int_{-\tau+\delta}^{2r_2-\tau+\delta}\left(\frac{1}{\sqrt{\tau-z}}+\frac{1}{\sqrt{\tau+z}}\right)\left(\frac{1}{\sqrt{\tau+z-\delta}}+\frac{1}{\sqrt{2r_2-\tau-z+\delta}}\right)dz\\
=&\frac{2}{\sqrt{\tau r_2}}\left(\int_{0}^{2r_2}\frac{1}{\sqrt{z(z+\delta)}}\,dz+\int_{0}^{2r_2}\frac{1}{\sqrt{z(2\tau-\delta-z)}}\,dz\right.\\
&\left.+\int_{0}^{2r_2}\frac{1}{\sqrt{z(2r_2+\delta-z)}}\,dz+\int_{0}^{2r_2}\frac{1}{\sqrt{z(2(\tau-r_2)-\delta+z)}}\,dz\right)\\
\leq&\frac{2}{\sqrt{\tau r_2}}\left(2\pi+2\ln(8\sqrt{d})+\ln\left(2\delta^{-1}\right)+\ln\left(2(2(\tau-r_2)-\delta)^{-1}\right)\right)\\
\leq&\frac{\kappa}{\sqrt{\tau r_2}}\left(\ln\delta^{-1}+\ln\left((2(\tau-r_2)-\delta)^{-1}\right)+\kappa\right)\ .
\end{aligned}
$$

Case 4.2: $\delta\in(2(\tau-r_2),2\tau]$**.**

We obtain the following upper bound on the density of Δ:

$$\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\\
\leq&\frac{2}{\sqrt{\tau r_2}}\int_{-\tau+\delta}^{\tau}\left(\frac{1}{\sqrt{\tau-z}}+\frac{1}{\sqrt{\tau+z}}\right)\left(\frac{1}{\sqrt{\tau+z-\delta}}+\frac{1}{\sqrt{2r_2-\tau-z+\delta}}\right)dz\\
=&\frac{2}{\sqrt{\tau r_2}}\left(\int_0^{2\tau-\delta}\frac{1}{\sqrt{z(2\tau-\delta-z)}}\,dz+\int_0^{2\tau-\delta}\frac{1}{\sqrt{z(z+\delta)}}\,dz\right.\\
&\left.+\int_0^{2\tau-\delta}\frac{1}{\sqrt{z(2(r_2-\tau)+\delta+z)}}\,dz+\int_\delta^{2\tau}\frac{1}{\sqrt{z(2r_2+\delta-z)}}\,dz\right)\\
\leq&\frac{2}{\sqrt{\tau r_2}}\left(2\pi+2\ln(8\sqrt{d})+\ln\left(2\delta^{-1}\right)+\ln\left(2(2(r_2-\tau)+\delta)^{-1}\right)\right)\\
\leq&\frac{\kappa}{\sqrt{\tau r_2}}\left(\ln\delta^{-1}+\ln\left((2(r_2-\tau)+\delta)^{-1}\right)+\kappa\right)\quad.
\end{aligned}$$

Altogether, this yields the lemma. □

B.4.2 Proof of Lemma 4.3.8

First, we derive the following lemma, which gives bounds on the conditional density of the random variable Δ when only one of the radii R_1 and R_2 is given.

Lemma B.4.1. *Let $r_1, r_2, \tau \in (0, \sqrt{d})$ and $\delta \in (0, 1/2]$. In the following, let κ denote a sufficiently large constant.*

a) The density of Δ under the conditions $T = \tau$ and $R_1 = r_1$ is bounded by

$$f_{\Delta|T=\tau,R_1=r_1}(\delta) \leq \begin{cases} \frac{\kappa}{\sqrt{\tau r_1}} \cdot \ln\delta^{-1} & \text{if } r_1 \leq \tau,\\ \frac{\kappa}{\tau} \cdot \ln\delta^{-1} & \text{if } r_1 \geq \tau. \end{cases}$$

b) The density of Δ under the conditions $T = \tau$ and $R_2 = r_2$ is bounded by

$$f_{\Delta|T=\tau,R_2=r_2}(\delta) \leq \begin{cases} \frac{\kappa}{\sqrt{\tau r_2}} \cdot (\ln\delta^{-1} + \ln|2(\tau-r_2)-\delta|^{-1} + \kappa) & \text{if } r_2 \leq \tau,\\ \frac{\kappa}{\tau} \cdot \ln\delta^{-1} & \text{if } r_2 \geq \tau. \end{cases}$$

Proof. a) We can write the density of Δ under the conditions $T = \tau$ and $R_1 = r_1$ as

$$\begin{aligned}
f_{\Delta|T=\tau,R_1=r_1}(\delta) &= \int_0^{\sqrt{d}} f_{R_2}(r_2) \cdot f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\,dr_2\\
&= \int_0^{\sqrt{d}} \frac{r_2^{d-1}}{d^{d/2-1}} \cdot f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\,dr_2\quad,
\end{aligned}$$

where f_{R_2} denotes the density of the length $R_2 = \mathsf{d}(O, Q_2)$. We use Lemma 4.3.7 to bound this integral. For $r_1 \leq \tau$ and sufficiently large constants κ' and κ'', we

obtain

$$\begin{aligned}
&f_{\Delta|T=\tau,R_1=r_1}(\delta)\\
&\leq \int_0^{\tau} \frac{r_2^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{r_1 r_2}} \left(\ln \delta^{-1} + \ln |2(r_1 - r_2) - \delta|^{-1} + \kappa\right) dr_2\\
&+ \int_{\tau}^{\sqrt{d}} \frac{r_2^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r_1}} \cdot \ln \delta^{-1}\, dr_2\\
&\leq \frac{\kappa'}{\sqrt{r_1}} \cdot \ln \delta^{-1} + \frac{\kappa'}{\sqrt{r_1}} \int_0^{\sqrt{d}} \ln |2(r_1 - r_2) - \delta|^{-1}\, dr_2 + \frac{\kappa}{\sqrt{\tau r_1}} \cdot \ln \delta^{-1}\\
&\leq \frac{\kappa''}{\sqrt{\tau r_1}} \cdot \ln \delta^{-1} \ .
\end{aligned}$$

For $\tau \leq r_1$ and a sufficiently large constant κ', we obtain analogously

$$\begin{aligned}
f_{\Delta|T=\tau,R_1=r_1}(\delta) &\leq \int_0^{\tau} \frac{r_2^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r_2}} \left(\ln \delta^{-1} + \ln |2(\tau - r_2) - \delta|^{-1} + \kappa\right) dr_2\\
&+ \int_{\tau}^{\sqrt{d}} \frac{r_2^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, dr_2\\
&\leq \frac{\kappa'}{\tau} \cdot \ln \delta^{-1} \ .
\end{aligned}$$

b) We can write the density of Δ under the conditions $T = \tau$ and $R_2 = r_2$ as

$$f_{\Delta|T=\tau,R_2=r_2}(\delta) = \int_0^{\sqrt{d}} \frac{r_1^{d-1}}{d^{d/2-1}} \cdot f_{\Delta|T=\tau,R_1=r_1,R_2=r_2}(\delta)\, dr_1 \ .$$

For $r_2 \leq \tau$ and sufficiently large constants κ' and κ'', we obtain

$$\begin{aligned}
f_{\Delta|T=\tau,R_2=r_2}(\delta) &\leq \int_0^{\tau} \frac{r_1^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{r_1 r_2}} \left(\ln \delta^{-1} + \ln |2(r_1 - r_2) - \delta|^{-1} + \kappa\right) dr_1\\
&+ \int_{\tau}^{\sqrt{d}} \frac{r_1^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r_2}} \left(\ln \delta^{-1} + \ln |2(\tau - r_2) - \delta|^{-1} + \kappa\right) dr_1\\
&\leq \frac{\kappa'}{\sqrt{r_2}} (\ln \delta^{-1} + \kappa) + \frac{\kappa'}{\sqrt{r_2}} \int_0^{\sqrt{d}} \ln |2(r_1 - r_2) - \delta|^{-1}\, dr_1\\
&+ \frac{\kappa}{\sqrt{\tau r_2}} \left(\ln \delta^{-1} + \ln |2(\tau - r_2) - \delta|^{-1} + \kappa\right)\\
&\leq \frac{\kappa''}{\sqrt{\tau r_2}} (\ln \delta^{-1} + \ln |2(\tau - r_2) - \delta|^{-1} + \kappa) \ .
\end{aligned}$$

For $\tau \leq r_2$ and a sufficiently large constant κ', we obtain

$$\begin{aligned}
f_{\Delta|T=\tau,R_2=r_2}(\delta) &\leq \int_0^{\tau} \frac{r_1^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r_1}} \cdot \ln \delta^{-1}\, dr_1 + \int_{\tau}^{\sqrt{d}} \frac{r_1^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, dr_1\\
&\leq \frac{\kappa'}{\tau} \cdot \ln \delta^{-1} \ .
\end{aligned}$$

□

Now we are ready to prove Lemma 4.3.8.

Proof of Lemma 4.3.8. a) In order to prove part a), we integrate $f_{\Delta|T=\tau,R_1=r}(\delta)$ over all values τ that T can take:

$$\begin{aligned} f_{\Delta|R_1=r}(\delta) &= \int_0^{\sqrt{d}} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot f_{\Delta|T=\tau,R_1=r}(\delta)\, d\tau \\ &\leq \int_0^{r} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, d\tau + \int_r^{\sqrt{d}} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r}} \cdot \ln \delta^{-1}\, d\tau \\ &\leq \frac{\kappa'}{\sqrt{r}} \cdot \ln \delta^{-1} \ . \end{aligned}$$

Furthermore, we integrate $f_{\Delta|T=\tau,R_2=r}(\delta)$ over all values τ that T can take:

$$\begin{aligned} f_{\Delta|R_2=r}(\delta) &= \int_0^{\sqrt{d}} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot f_{\Delta|T=\tau,R_2=r}(\delta)\, d\tau \\ &\leq \int_0^{r} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, d\tau \\ &+ \int_r^{\sqrt{d}} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r}} (\ln \delta^{-1} + \ln |2(\tau - r) - \delta|^{-1} + \kappa)\, d\tau \\ &\leq \frac{\kappa'}{\sqrt{r}} \cdot \ln \delta^{-1} \ . \end{aligned}$$

b) For a sufficiently large constant κ',

$$\begin{aligned} f_{\Delta|T=\tau}(\delta) &= \int_0^{\sqrt{d}} \frac{r^{d-1}}{d^{d/2-1}} \cdot f_{\Delta|T=\tau,R_1=r}(\delta)\, dr \\ &\leq \int_0^{\tau} \frac{r^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\sqrt{\tau r}} \cdot \ln \delta^{-1}\, dr + \int_\tau^{\sqrt{d}} \frac{r^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, dr \\ &\leq \frac{\kappa'}{\tau} \cdot \ln \delta^{-1} \ . \end{aligned}$$

c) For a sufficiently large constant κ',

$$f_\Delta(\delta) = \int_0^{\sqrt{d}} \frac{\tau^{d-1}}{d^{d/2-1}} \cdot \frac{\kappa}{\tau} \cdot \ln \delta^{-1}\, d\tau \leq \kappa' \cdot \ln \delta^{-1} \ .$$

d) For sufficiently large constants κ' and κ'',

$$\begin{aligned} f_{Z|T=\tau}(z) &= \int_{r=0}^{\tau} \frac{r^{d-1}}{d^{d/2-1}} \sqrt{\frac{2}{(\tau + z)(2r - \tau - z)}}\, dr + \int_{r=\tau}^{\sqrt{d}} \frac{r^{d-1}}{d^{d/2-1}} \sqrt{\frac{2}{\tau^2 - z^2}}\, dr \\ &\leq \sqrt{\frac{2}{\tau + z}} d^{1/2} \int_{r=0}^{\tau} \sqrt{\frac{1}{2r - \tau - r}}\, dr + \sqrt{\frac{2}{\tau^2 - z^2}} \\ &\leq \frac{\kappa'}{\sqrt{\tau + z}} + \sqrt{\frac{2}{\tau^2 - z^2}} \leq \frac{\kappa''}{\sqrt{\tau^2 - z^2}} \ . \end{aligned}$$

□

Bibliography

[ADK+04] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 295–304, 2004.

[AEGH98] Pankaj K. Agarwal, David Eppstein, Leonidas J. Guibas, and Monika Rauch Henzinger. Parametric and kinetic minimum spanning trees. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 596–605, 1998.

[AGM+07a] Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. Uncoordinated two-sided markets. Technical Report AIB-2007-22, RWTH Aachen, 2007.

[AGM+07b] Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. A unified approach to congestion games and two-sided markets. In *Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE)*, pages 30–41, 2007.

[AGP97] Edward J. Anderson, Celia A. Glass, and Chris N. Potts. Machine scheduling. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*. John Wiley and Sons, 1997.

[ALMO07] David Abraham, Ariel Levavi, David Manlove, and Gregg O'Malley. The stable roommates problem with globally-ranked pairs. In *Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE)*, pages 431–444, 2007.

[ANRV07] Heiner Ackermann, Alantha Newman, Heiko Röglin, and Berthold Vöcking. Decision-making based on approximate and smoothed Pareto curves. *Theoretical Computer Science*, 378(3):253–270, 2007.

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[ARV06a] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 613–622, 2006.

[ARV06b] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. Pure Nash equilibria in player-specific and weighted congestion games. In *Proceedings of the 2nd International Workshop on Internet and Network Economics (WINE)*, pages 50–61, 2006.

[AV06a] David Arthur and Sergei Vassilvitskii. How slow is the k-means method? In *Proceedings of the 22nd ACM Symposium on Computational Geometry (SoCG)*, pages 144–153, 2006.

[AV06b] David Arthur and Sergei Vassilvitskii. Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 153–164, 2006.

[Bei04] René Beier. *Probabilistic Analysis of Discrete Optimization Problems*. PhD thesis, Universität des Saarlandes, 2004.

[Boc58] F. Bock. An algorithm for solving traveling-salesman and related network optimization problems. Unpublished manuscript associated with talk presented at the 14th ORSA National Meeting, 1958.

[BP87] Francisco Barahona and William R. Pulleyblank. Exact arborescences, matchings and cycles. *Discrete Applied Mathematics*, 16:91–99, 1987.

[BRV07] René Beier, Heiko Röglin, and Berthold Vöcking. The smoothed number of Pareto optimal solutions in bicriteria integer optimization. In *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 53–67, 2007.

[BV04] René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences*, 69(3):306–329, 2004.

[BV06] René Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM Journal on Computing*, 35(4):855–881, 2006.

[CK05] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–73, 2005.

[CKT99] Barun Chandra, Howard J. Karloff, and Craig A. Tovey. New results on the old k-Opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999.

[CM85] H. William Corley and I. Douglas Moon. Shortest paths in networks with vector weights. *Journal of Optimization Theory and Application*, 46(1):79–86, 1985.

[Cou38] Antoine Augustin Cournot. *Recherches sur les Principes Mathématiques de la Théorie des Richesses*. Hachette, 1838. English Translation by Nathaniel T. Bacon: *Researches into the Mathematical Principles of the Theory of Wealth*, 1897.

[Cro58] G. A. Croes. A method for solving traveling salesman problems. *Operations Research*, 6(6):792–812, 1958.

[Dey97] Tamal K. Dey. Improved bounds on planar k-sets and k-levels. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 156–161, 1997.

[DR98] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *Random Structures and Algorithms*, 13(2):99–124, 1998.

[DY07] Ilias Diakonikolas and Mihalis Yannakakis. Small approximate pareto sets for bi-objective shortest paths and other problems. In *Proceedings of the 10th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 74–88, 2007.

[EDKM03] Eyal Even-Dar, Alexander Kesselman, and Yishay Mansour. Convergence time to Nash equilibria. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.

[EG02] Matthias Ehrgott and Xavier Gandibleux, editors. *Multiple Criteria Optimization – State of the Art Annotated Bibliographic Surveys*. Kluwer Academic Publishers, 2002.

[Ehr99] Matthias Ehrgott. Integer solutions of multicriteria network flow problems. *Investigacao Operacional*, 19:229–243, 1999.

[Ehr00] Matthias Ehrgott. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems. Springer, 2000.

[EO06] Federico Echenique and Jorge Oviedo. A theory of stability in many-to-many matching markets. *Theoretical Economics*, 1(2):233–273, 2006.

[ERV07] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1295–1304, 2007.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley and Sons, 1968.

[Fel91] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. John Wiley and Sons, 1991.

[Fle03] Tamás Fleiner. A fixed-point approach to stable matchings and some applications. *Mathematics of Operations Research*, 28(1):103–126, 2003.

[FPT04] Alex Fabrikant, Christos H. Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[FT91] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.

[FT95] Sophie Fischer and Leen Torenvliet. The malleability of TSP_{2opt}. In *Proceedings of the 21st International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 152–166, 1995.

[GI89] Dan Gusfield and Robert W. Irving. *The stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[GLMT04] Michel X. Goemans, Erran L. Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 55–66, 2004.

[GS62] David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–15, 1962.

[Han80] Pierre Hansen. Bicriterion path problems. In *Multiple Criteria Decision Making: Theory and Applications*, volume 177 of *Lecture Notes in Economics and Mathematical Systems*, pages 109–127, 1980.

[Hen86] Mordechai I. Henig. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291, 1986.

[HT90] Reiner Horst and Hoang Tuy. *Global Optimization*. Springer, 1990.

[IMMM99] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 443–452, 1999.

[IMN+05] Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, pages 489–494, 2005.

[JM97] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study in local optimization. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*. John Wiley and Sons, 1997.

[JPY88] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

[Kat92] Naoki Katoh. Bicriteria network optimization problems. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E75-A(3):321–329, 1992.

[KC82] Alexander S. Kelso and Vincent P. Crawford. Job matchings, coalition formation, and gross substitute. *Econometrica*, 50:1483–1504, 1982.

[Ker89] Walter Kern. A probabilistic analysis of the switching algorithm for the Euclidean TSP. *Mathematical Programming*, 44(2):213–219, 1989.

[KL70] Brian W. Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

[Knu76] Donald E. Knuth. *Marriage Stables et leurs relations avec d'autres problèmes Combinatories*. Les Presses de l'Université de Montréal, 1976.

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.

[Kre89] Mark W. Krentel. Structure in locally optimal solutions. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 216–221, 1989.

[KW00] Kathrin Klamroth and Margaret M. Wiecek. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics*, 47(1):57–76, 2000.

[LK73] Shen Lin and Brian W. Kernighan. An effective heuristic for the traveling salesman problem. *Operations Research*, 21:489–516, 1973.

[Lue75] George S. Lueker. Unpublished manuscript, 1975. Princeton University.

[MG98] Adli Mustafa and Mark Goh. Finding integer efficient solutions for bicriteria and tricriteria network flow problems using dinas. *Computers & Operations Research*, 25(2):139–157, 1998.

[MHW01] Matthias Müller-Hannemann and Karsten Weihe. Pareto shortest paths is often feasible in practice. In *Proceedings of the 5th International Workshop on Algorithm Engineering (WAE)*, pages 185–198, 2001.

[Mil96] Igal Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111–124, 1996.

[Mit99] Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–114, 1987.

[Nas50] John F. Nash. *Non-Cooperative Games*. PhD thesis, Priceton University, 1950.

[NU69] George L. Nemhauser and Zev Ullmann. Discrete dynamic programming and capital allocation. *Management Science*, 15(9):494–505, 1969.

[Owe95] Guillermo Owen. *Game Theory*. Academic Press, 1995.

[Pap77] Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

[Pap92] Christos H. Papadimitriou. The complexity of the lin-kernighan heuristic for the traveling salesman problem. *SIAM Journal on Computing*, 21(3):450–465, 1992.

[Pol95] Svatopluk Poljak. Integer linear programs and local search for max-cut. *SIAM Journal on Computing*, 24(4):822–839, 1995.

[PS06] Panagiota N. Panagopoulou and Paul G. Spirakis. Algorithms for pure nash equilibria in weighted congestion games. *ACM Journal of Experimental Algorithmics*, 11, 2006.

[PY82] Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.

[PY00] Christos H. Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 86–92, 2000.

[Rei91] Gerhard Reinelt. TSPLIB – A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[RG96] Ram Ravi and Michel X. Goemans. The constrained minimum spanning tree problem (extended abstract). In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 66–75, 1996.

[RL01] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling (3DIM)*, pages 145–152, 2001.

[Ros73] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[Rot84] Alvin E. Roth. The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, 92:991–1016, 1984.

[Rot96] Alvin E. Roth. The national residency matching program as a labor market. *Journal of American Medical Association*, 275(13):1054–1056, 1996.

[RS90] Alvin E. Roth and Marilda A. Oliveira Sotomayor. *Two-sided Matching: A study in game-theoretic modeling and analysis*. Cambridge University Press, 1990.

[RSI77] Daniel J. Rosenkrantz, Richard Edwin Stearns, and Philip M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

[RV90] Alvin E. Roth and John H. Vande Vate. Random paths to stability in two-sided matching. *Econometrica*, 58(6):1475–1480, 1990.

[RV07] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007.

[SA00] Anders J. V. Skriver and Kim Allan Andersen. A label correcting approach for solving bicriterion shortest-path problems. *Computers & Operations Research*, 27(6):507–524, 2000.

[SAZ+04] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet indirection infrastructure. *IEEE/ACM Transactions on Networking*, 12(2):205–218, 2004.

[Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer, 2003.

[Spi] Daniel A. Spielman. Smoothed Analysis Homepage. `http://www.cs.yale.edu/homes/spielman/SmoothedAnalysis/index.html`.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[SY91] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

[TZ04] George Tsaggouris and Christos Zaroliagis. Non-additive shortest paths. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA)*, pages 822–834, 2004.

[Vaz04] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2004.

[vLS81] Jan van Leeuwen and Anneke A. Schoon. Untangling a traveling salesman tour in the plane. In *Proceedings of the 7th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 87–98, 1981.

[vNM47] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[VY05] Sergei Vassilvitskii and Mihalis Yannakakis. Efficiently computing succinct trade-off curves. *Theoretical Computer Science*, 348(2–3):334–356, 2005.

[Wan97] Jie Wang. Average-case computational complexity theory. In *Complexity Theory Retrospective II*, pages 295–328. Springer, 1997.

Index

Curriculum Vitae

Name:	Heiko Röglin
Geburtsdatum:	10. April 1981
Geburtsort:	Dortmund

Ausbildung

1987 – 1991	Besuch der Graf-Konrad-Grundschule in Dortmund
1991 – 2000	Besuch des Helmholtz-Gymnasiums in Dortmund
06/2000	Abitur (Durchschnittsnote: 1,2)
10/2000 – 08/2004	Informatikstudium an der Universität Dortmund
08/2004	Diplom in Informatik mit Nebenfach Mathematik (Gesamtnote: mit Auszeichnung)

Berufliche Tätigkeiten

10/2001 – 08/2002	Studentische Hilfskraft an der Universität Dortmund
10/2003 – 08/2004	Studentische Hilfskraft an der Universität Dortmund
seit 10/2004	Wissenschaftlicher Mitarbeiter am Lehrstuhl für Algorithmen und Komplexität der RWTH Aachen

Auszeichnungen

11/2003	Förderpreis des Vereins Deutscher Ingenieure (VDI) für besondere Leistungen im Grundstudium
01/2005	Jahrgangsbestenpreis 2004 der Gesellschaft der Freunde der Universität Dortmund e. V.