On approximation algorithms and polyhedral relaxations for knapsack problems, and clustered planarity testing

Thèse N° 9277

Présentée le 15 mars 2019

à la Faculté des sciences de base
Chaire d'optimisation discrète
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Igor MALINOVIC

Acceptée sur proposition du jury

Prof. K. Hess Bellwald, présidente du jury
Prof. F. Eisenbrand, Prof. Y. Faenza, directeurs de thèse
Prof. M. Mastrolilli, rapporteur
Prof. S. Fiorini, rapporteur
Prof. O. Svensson, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Врећа! Врећа се ово зове, а ово у врећи,
ово се јазавац зове.
— Петар Кочић, "Јазавац пред судом"

Мојој породици и пријатељима…

Acknowledgements

I would like to say many thanks to my supervisor Fritz Eisenbrand, for the opportunity to be here and his generous support from the beginning of my master studies until today, for his research advices, involving me into teaching, student projects, interesting political discussions, and guiding me to become independent.

I am very grateful to my co-supervisor Yuri Faenza, for a lot of patience, time and effort that he invested in working with me and my development towards becoming a researcher. Thanks to him for being as well a life coach, great support, football teammate, and a kind friend. Thanks to Yuri and Fanny for hosting me in NY, and the great time there.

Thanks to many researchers for transferring their knowledge and experience, in particular to Monaldo Mastrolilli, Ola Svensson, Samuel Fiorini, Daniel Bienstock, Andrey Kupavskii, János Pach, Radoslav Fulek, István Tomon, Jan Kynčl, and Dömötör Pálvölgyi. Also, to Kathryn Hess Bellwald for being the jury president for both, my candidacy exam and the thesis defense.

Thanks for love and most devoted support to my family, and friends: "Sina i mi ostali" – Boka, Laza, Semba, Sina, Oza; "Momci mladi u širem sastavu" – Anči, Bura, DeDe, Đole, Gadža, Lemi, Miloš, Peđa, Staniša; "Lausannois++": Andreicheck, Andres, Aria, Alfonso, Christoph, Gonzalo, Jovanče, Lilli, Manuel, Mauro, Mića, Nicolas, Renaud, Slobo, Vido. Thanks to all DISOPT and DCG members, especially to Jocelyne (for keeping me out of prison and more), and to all football teammates.

Many wonderful people were around for the time of my studies, and although not mentioned by name, I would like to thank them.

Lausanne, December 12, 2018 I. M.

Abstract

Knapsack problems give a simple framework for decision making. A classical example is the min-knapsack problem (MINKNAP): choose a subset of items with minimum total cost, whose total profit is above a given threshold. While this model successfully generalizes to problems in scheduling, network design and capacited location, its dynamic programming approaches do not. One often relies on strong polyhedral relaxations for corresponding integer programs instead. Among other results, we construct such a relaxation for the time-invariant incremental knapsack problem (IIK), and study classes of valid inequalities for MINKNAP.

IIK is covered in the first part of this thesis. It is a generalization of the max-knapsack problem to a discrete multi-period setting. At each time, capacity increases and items can be added, but not removed from the knapsack. The goal is to maximize the sum of profits over all times. IIK models scenarios in specific financial markets and governmental decision processes. It is known to be strongly NP-hard and there has been work on approximation algorithms for special cases. We settle the complexity of IIK by designing a PTAS, and provide several extensions of the technique.

The second part is on MINKNAP and divided into two chapters. One is motivated by a recent work on disjunctive relaxations for MINKNAP with fixed objective function, where we reduce the size of the construction. The other focuses on a class of bounded pitch inequalities, that generalize the unweighted cover inequalities for MINKNAP. While separating over pitch-1 inequalities is NP-hard, we show that approximate separation over the set of pitch-1 and pitch-2 inequalities can be done in polynomial time. We also investigate integrality gaps of linear relaxations for MINKNAP when these inequalities are added. Consequently we show that, for any fixed t, the t-th CG closure of the natural relaxation has unbounded gap.

The last chapter deals with questions in clustered planarity testing. The Hanani–Tutte theorem is a classical result that characterizes planar graphs as graphs that admit a drawing in the plane in which every pair of edges not sharing a vertex cross an even number of times. We generalize this result to clustered graphs with two disjoint clusters, and show that a straightforward extension to flat clustered graphs with three or more disjoint clusters is not possible. For general clustered graphs we show a variant of the Hanani–Tutte theorem in the case when each cluster induces a connected subgraph. We conclude by a short proof, using matroid intersection, for a result by Di Battista and Frati on embedded clustered graphs.

Keywords: approximation algorithms, polyhedral relaxations, time-invariant incremental knapsack, bounded pitch, min-knapsack, clustered planarity, Hanani–Tutte theorem

Résumé

Les problèmes de sac à dos, ou knapsack, offrent un cadre simple pour la prise de décision. Un exemple classique est le problème min-knapsack (MinKnap) : choisir un sous-ensemble d'articles avec un coût total minimum, dont le bénéfice total est supérieur à un seuil donné. Alors que ce modèle est généralisable aux problèmes d'ordonnancement, de conception de réseau et de localisation, sa solution par programmation dynamique ne l'est pas. On se base souvent sur des relaxations polyhédrales pour le problème d'optimisation en nombre entier correspondant. Parmi d'autres résultats, nous construisons une telle relaxation pour le problème de sac à dos incrémental et invariant dans le temps (IIK) et étudions des classes d'inégalités valables pour MinKnap.

IIK est l'objet de la première partie de la thèse. C'est une généralisation du problème max-knapsack à un cas discret et à périodes multiples. À chaque temps, la capacité augmente et des articles peuvent être ajoutés mais pas retirés du knapsack. L'objectif est de maximiser la somme des bénéfices sur tous les temps. IIK modélise des scénarios spécifiques sur des marchés financiers et dans des processus de décision gouvernementaux. C'est un problème NP-difficile et des algorithmes d'approximation ont été développés pour des cas particuliers. Nous déterminons la complexité de IIK en concevant un PTAS et fournissons plusieurs extensions de la technique.

La deuxième partie traite de MinKnap et est divisée en deux chapitres. L'un est motivé par un travail récent sur les relaxations disjonctives pour le problème MinKnap à fonction objectif fixe et nous y réduisons la taille de la construction. L'autre se concentre sur une classe d'inégalités à pitch borné, qui généralise les inégalités de couverture non pondérées pour MinKnap. Alors que la séparation sur les inégalités de pitch-1 est NP-difficile, nous montrons qu'une séparation approximative sur l'ensemble des inégalités de pitch-1 et de pitch-2 peut être effectuée en temps polynomial. Nous étudions également les écarts d'intégralité des relaxations linéaires pour MinKnap lorsque ces inégalités sont ajoutées. Finalement nous montrons que, pour tout t fixe, la t-ème fermeture CG de la relaxation naturelle a un saut non-borné.

Le dernier chapitre examine une classe de problèmes différente. Le théorème Hanani-Tutte est un résultat classique qui caractérise les graphes planaires en tant que graphes admettant un dessin dans le plan dans lequel chaque paire d'arêtes ne partageant pas un sommet se croisent un nombre pair de fois. Nous généralisons ce résultat aux graphes partitionnés constitués de deux partitions disjointes et montrons qu'une extension directe à des graphes

partitionnés plats comportant au moins trois partitions disjointes est impossible. Pour les graphes partitionnés généraux, nous montrons une variante du théorème de Hanani-Tutte dans le cas où chaque partition induit un sous-graphe connecté. Nous concluons par une courte démonstration, utilisant l'intersection de matroïdes, d'un résultat de Di Battista et Frati sur les tests de planarité par partition.

Mots-clés : algorithmes d'approximation, relaxations polyhédrales, sac à dos incrémental invariant dans le temps, pitch borné, min-knapsack, graphes partitionnés, théorème Hanani–Tutte

Zusammenfassung

Knapsack-Probleme geben Entscheidungsverfahren eine einfache Struktur. Ein klassisches Beispiel ist das Min-Knapsack-Problem (MINKNAP): wählen Sie eine Teilmenge von Elementen mit minimalen Gesamtkosten aus, deren Gesamtgewinn über einem bestimmten Schwellenwert liegt. Dieses Modell wurde erfolgreich auf Planungsprobleme, Netzwerkdesign und kapazitierte Standortprobleme verallgemeinert. Hingegen können dynamische Programmieransätze nicht für diese Probleme benutzt werden. Stattdessen werden häufig starke polyedrische Relaxationen für entsprechende ganzzahlige lineare Optimierungen verwendet. Unter anderem stellen wir eine solche Relaxion für das zeitinvariante inkrementelle Knapsack-Problem (IIK) her und studieren gültige Ungleichungen für MINKNAP.

IIK behandeln wir im ersten Teil der Arbeit. Es verallgemeinert das Max-Knapsack-Problem auf mehrere Perioden: Nach jedem Zeitintervall erhöht sich die Kapazität und es können Elemente hinzugefügt, jedoch nicht entfernt werden. Ziel ist es, die Summe der Gewinne über alle Zeitintervalle zu maximieren. Mit IIK kann man gewisse Szenarien in Finanzmärkten und staatlichen Entscheidungsprozessen modellieren. Es ist bekannt, dass IIK stark NP-schwer ist, in speziellen Fällen wurde aber an Näherungsalgorithmen gearbeitet. Wir legen die Komplexität von IIK durch den Entwurf eines PTAS fest und geben mehrere Verallgemeinerungen dieser Technik an.

Die nächsten zwei Kapitel befassen sich mit MINKNAP und bilden den zweiten Teil der Arbeit. Das erste Kapitel ist motiviert durch eine kürzlich durchgeführte Arbeit über disjunktive Relaxationen für MINKNAP mit fester Zielfunktion. Bei dieser Relaxion reduzieren wir die Größe der Konstruktion. Das zweite Kapitel konzentriert sich auf eine Klasse begrenzter pitch-Ungleichungen, welche die ungewichteten cover-Ungleichheiten für MINKNAP verallgemeinern. Pitch-1-Ungleichungen separieren ist NP-schwer, wir zeigen aber, dass eine approximierte Separierung der Menge von pitch-1 und pitch-2-Ungleichungen in Polynomialzeit erfolgen kann. Wir untersuchen auch die Integritätslücken von linearen Relaxationen für MINKNAP, wenn diese Ungleichungen hinzugefügt werden. Folglich zeigen wir, dass der t-te CG-Verschluss der natürlichen Relaxation für jedes feste t eine unbeschränkte Lücke aufweist.

Das letzte Kapitel untersucht eine andere Klasse von Problemen. Das Hanani–Tutte Theorem ist ein klassisches Resultat, welches planare Graphen durch eine mögliche Zeichnung in der Ebene charakterisiert, in der jedes Kantenpaar, das keinen Scheitelpunkt hat, eine gerade Anzahl von Kreuzungen hat. Wir verallgemeinern dieses Ergebnis auf Cluster-Diagramme mit zwei disjunkten Clustern und zeigen, dass eine einfache Verallgemeinerung auf flache

Cluster-Diagramme mit drei oder mehr disjunkten Clustern nicht möglich ist. Für allgemeine Cluster-Diagramme zeigen wir eine Variante des Satzes von Hanani - Tutte in dem Fall, in dem jedes Cluster einen zusammenhängenden Untergraph induziert. Mit Matroid-Schnittpunkten geben wir einen kurzen Beweis für ein Ergebnis von Di Battista und Frati über gruppierte Planaritätstests.

Schlüsselwörter: Approximationsalgorithmen, polyedrische Relaxationen, zeitinvarianter inkrementeller Knapsack, beschränkte pitch, min-Knapsack, gruppierte Planarität, Satz von Hanani – Tutte

Contents

Abstract (English/Français/Deutsch) **vii**

1 Introduction **1**

1.1 Knapsack problems 2

1.2 Clustered planarity 4

1.3 Our results and organization of the chapters 4

2 A PTAS for the time-invariant incremental knapsack **7**

2.1 Overview of basic techniques 9

2.2 A PTAS for IIK 10

2.2.1 Reducing IIK to special instances and solutions 10

2.2.2 A disjunctive relaxation 13

2.2.3 Rounding 16

2.3 Generalizations 21

3 An improved disjunctive relaxation for the min-knapsack **25**

3.1 Overview of the technique 25

3.2 The disjunctive relaxation 27

4 On bounded pitch inequalities for the min-knapsack **31**

4.1 Basics 31

4.2 A weak separation oracle for pitch-1 and pitch-2 inequalities 33

4.2.1 Restricting the set of valid pitch-2 inequalities 33

4.2.2 An oracle 35

4.2.3 Separating inequalities of pitch-3 and larger, with fixed support 37

4.3 Integrality gap for MINKNAP with bounded pitch inequalities 38

4.3.1 When p=c 38

4.3.2 CG closures of bounded rank of the natural MINKNAP relaxation 39

4.3.3 When all knapsack cover inequalities are added 41

4.3.4 When all bounded pitch and knapsack cover inequalities are added . . . 42

5 Clustered planarity testing **45**

5.1 Basic definitions and an overview of results 45

5.2 Algorithm 48

5.3 Weak Hanani–Tutte for two-clustered graphs 51
5.3.1 Proof of Theorem 26 52
5.4 Strong Hanani–Tutte for two-clustered graphs 53
5.4.1 Proof of Theorem 27 54
5.5 Strong Hanani–Tutte for c-connected clustered graphs 58
5.6 Counterexample on three clusters 59
5.6.1 Proof of Theorem 29 64
5.7 Small faces 65
5.7.1 Proof of Theorem 30 67
5.8 Concluding remarks 68

6 Future directions **71**

A **73**
A.1 Background on disjunctive programming 73
A.2 IIK, MKP, and UFP 73

Bibliography **80**

Curriculum Vitae **81**

1 Introduction

When making decisions, we are regularly deciding among a set of possibilities, according to our experience and emotional preferences. One can think of planing daily activities subject to the time limit. There are many factors, and it is not easy to quantify the utility of each activity for us. We are subjective and sometimes even irrational. However, many decisions in industrial applications are made in a rigorously controlled environment. Each possible choice has an assigned value and the goal is to maximize the total utility while respecting well-defined constraints. A major problem is that the number of choices can be very large.

Knapsack problems give a straightforward way to model decision making. A classical example is the maximum knapsack problem (MAXKNAP): given a threshold and a set of alternatives, where each alternative has its weight and profit, provide a subset maximizing the cumulative profit whose weight does not exceed the threshold. Verifying that a chosen set satisfies the latter condition, and brings profit above a certain value, requires only a small number of arithmetic operations. However, it is believed that finding a maximum set satisfying the condition, and thus solving MAXKNAP, cannot be done efficiently in general [1].

Approximation algorithms are a class of efficient methods to obtain problem solutions which are not necessarily optimal but have a close-to-optimal guarantee. In the 50s, George Dantzig provided an algorithm returning a solution to MAXKNAP with profit of at least half the optimal value, using slightly more operations then reading the problem data. Dantzig was a pioneer in the field of linear programming, which aims at maximizing a linear function subject to linear constraints. Geometrically, the corresponding set of feasible solutions is defined with a convex polyhedron which can be seen as an intersection of half-spaces. While the set of solutions for MAXKNAP is discrete, polyhedral relaxations form a convex envelope around this set. It is desired that an extreme point of this envelope can be rounded to a MAXKNAP solution with value sufficiently close to the optimum. Approximation algorithms designed in such a way are benefiting from the easiness of adding extra constraints and adaptation to similar problems, as well as maturity of linear programming solvers in practice.

[1] It boils down to the "P=NP?" question and the exponential time hypothesis.

Clustered planrity relates to another aspect important for decision making, namely data representation. A sequence of tasks that are to be performed in a large information system can be visualized by a diagram. Each task is shown as a node and there is an arc between each pair of tasks sharing data. In addition, tasks are grouped into clusters drawn as circular discs, e.g., indicating tasks that are to be executed in parallel. Clustered planarity works with questions whether certain diagrams can be drawn on a sheet of paper without line intersections. Such a drawing gives a cleaner overview of the system and makes managing decisions easier.

1.1 Knapsack problems

Knapsack problems are among the most fundamental and well-studied in discrete optimization. Some variants forego the development of modern optimization theory, dating back to 1896 [39]. MAXKNAP was classified as an NP-complete problem already in the initial Karp's list in the 70's [31]. This was closely followed by a *fully polynomial-time approximation scheme* (FPTAS) for the same problem which was derived by Ibarra and Kim [28]. Many classical algorithmic techniques including greedy, dynamic programming, backtracking/branch-and-bound have been studied by means of solving knapsack problems, see e.g. [32]. The algorithm of Martello and Toth [37] has been known to be the fastest in practice for exactly solving MAXKNAP instances [2].

However, pure knapsack problems rarely appear in applications. One aims at developing techniques that remain valid when less structured constraints are added on top of the original knapsack one. This can be achieved by providing *strong* linear relaxations for the problem: then, any additional linear constraint can be added to the formulation, providing a good starting point for any branch-and-bound procedure. The most common way to measure the strength of a linear relaxation is by measuring its *integrality gap*, i.e. the maximum ratio between the optimal solutions of the linear and the integer programs (inverse for minimization problems) over all the objective functions. The standard linear relaxation for MAXKNAP has integrality gap 2, and further reducing the gap requires an extended space [21]. For $\epsilon > 0$, an *extended formulation* with gap bounded by $(1+\epsilon)$ and size $n^{\tilde{O}(1/\epsilon)}$ is either obtained by *disjunctive programming* [9] or with $1/\epsilon$ rounds of the *Lassere hierarchy* [30]. Disjunctive programming (See Appendix A.1) provides a very general approach to find strong relaxations for integral sets, and it has been exploited in practice to produce *disjunctive cuts* for MILP.

The time-invariant incremental knapsack problem (IIK) is a natural extension of MAXKNAP to the scenario where the knapsack capacity increases in a predictable manner over a finite set of times. An item can be added to knapsack at an arbitrary time, and once placed it cannot be removed. The profit of an item is multiplied by the number of times it appears in the knapsack. The goal is to maximize profit while respecting the capacity at each time. The natural relaxation of IIK has unbounded integrality gap [11], and the disjunctive programming technique [10] has been applied to obtain strong relaxations and approximation algorithms for its variants [11, 44].

The min-knapsack problem (MINKNAP) is a close relative of MAXKNAP where one aims at finding a minimum cost subset of items with profit above a given threshold. NP-completeness of MINKNAP immediately follows from the NP-completeness of MAXKNAP. Moreover, it is not hard to show that the classical FPTAS for MAXKNAP [28, 36] can be adapted to work for MINKNAP, thus completely settling its complexity. MINKNAP is an important problem appearing as a substructure in many IPs. Valid inequalities for MINKNAP – like *the knapsack cover inequalities* [15] – have been generalized to problems in scheduling, network design and facility location. In contrast to MAXKNAP, the standard linear relaxation for MINKNAP has unbounded integrality gap, and this remains true even after $\Theta(n)$ rounds of the Lasserre hierarchy [35]. It is an open question whether MINKNAP admits a poly-size extended formulation with constant integrality gap. Adding all (exponentially many) knapsack cover inequalities reduces the gap to 2, and those can be approximately separated [15]. This bound on the gap is tight, even when the profit vector is equal to the cost vector. Recent results showed the existence [6] and gave an explicit construction [22] of an extended formulation for MINKNAP of quasi-polynomial size with integrality gap $2+\epsilon$, for $\epsilon > 0$.

Besides the knapsack cover inequalities, one can look for other classes of well-behaved inequalities for MINKNAP with the goal to reduce the gap below 2. This would further improve the approximation ratio of algorithms for problems containing MINKNAP as a subproblem. A prominent family of relaxations for covering problems is given by the so called *bounded pitch inequalities* [13], see Section 4.1 for the definition. Intuitively, the *pitch* is a parameter measuring the complexity of an inequality. The associated separation problem is NP-Hard already for pitch-1. The pitch-1 inequalities are often known in the literature as *unweighted cover inequalities* (see e.g. [6]). Bienstock & Zuckerberg showed that the t-th Chvátal-Gomory (CG) closure of any linear relaxation can be $(1+\epsilon)$-approximated by strengthening the relaxation with all valid inequalities of pitch at most $\theta(t/\epsilon)$. They developed a strong hierarchy [12] for 0/1-covering problems, which was simplified by Mastrolilli [38] with an augmented version of SOS hierarchy.

The work by Fiorini et al. [22] further extends this line of research. Given $k \in \mathbb{N}$, they present a procedure for obtaining linear relaxations satisfying all pitch-k inequalities valid for a given 0/1 set. The construction is based on a boolean formula defining the set. Take MINKNAP with n items as an example, there is a corresponding boolean threshold function which maps each feasible 0/1 vector to 1 and each infeasible to 0. Every function of this kind admits a monotone boolean formula using $n^{O(\log n)}$ operators [7]. The construction of Fiorini et al. [22] results with a MINKNAP relaxation of size $n^{O(k \log n)}$ satisfying all pitch-k. Applying it for $k=1$, to a series of residual MINKNAP problems, gives the above mentioned MINKNAP relaxation with quasi-polynomial size and gap arbitrary close to 2. Such a result is in a sense counterintuitive to the fact that the knapsack cover inequalities can be of pitch as high as n, and that the natural relaxation with all pitch-1 has unbounded gap. One immediate question can be whether applying pitch-$\frac{1}{\epsilon}$ to adequately chosen residual MINKNAP problems can reduce the gap to $1+\epsilon$, for $\epsilon > 0$.

1.2 Clustered planarity

Investigation of graph planarity can be traced back to the 1930s and developments accomplished at that time by Hanani [66], Kuratowski [71], Whitney [83] and others. The Hanani–Tutte theorem [66, 82] is a classical result that provides an algebraic characterization of planarity with interesting theoretical and algorithmic consequences; see Section 5.2. The (strong) Hanani–Tutte theorem says that a graph is planar if it can be drawn in the plane so that no pair of independent edges crosses an odd number of times. Moreover, its variant known as the weak Hanani–Tutte theorem [48, 75, 78] states that if G has a drawing $\mathcal{D}$ where every pair of edges cross an even number of times, then G has an embedding that preserves the cyclic order of edges at vertices in $\mathcal{D}$. Note that the weak variant does not directly follow from the strong Hanani–Tutte theorem. For sub-cubic graphs, the weak variant implies the strong variant. Other variants of the Hanani–Tutte theorem were proved for surfaces of higher genus [77, 79], x-monotone drawings [62, 76], partially embedded planar graphs, and several special cases of simultaneously embedded planar graphs [81]. See [80] for a recent survey on applications of the Hanani–Tutte theorem and related results.

With the advent of computing, a linear-time algorithm for graph planarity was discovered [68]. Nowadays, a polynomial-time algorithm for testing whether a graph admits a crossing-free drawing in the plane could almost be considered a folklore result. Nevertheless, many variants of planarity are still only poorly understood. As a consequence of this state of affairs, the corresponding decision problems for these variants had neither been shown to be polynomial nor NP-hard. *Clustered planarity* is one of the most prominent [50] of such planarity notions. Roughly speaking, an instance of this problem is a graph whose vertices are partitioned into clusters. The question is whether the graph can be drawn in the plane so that the vertices in the same cluster belong to the same simple closed region and no edge crosses the boundary of a particular region more than once.

1.3 Our results and organization of the chapters

In Chapter 2, we focus on the time-invariant incremental knapsack problem (IIK) and use disjunctive programming techniques. IIK has been addressed by various people [42, 27, 11, 44] and partial solutions were provided. We improve upon all the previous results giving a PTAS for the problem. This is essentially the best possible (unless P=NP), since the problem is shown to be strongly NP-hard [11]. In order to obtain a PTAS for IIK with arbitrary number of descrete times, we tailor the disjunctive approach to incremental problems. We show that, with a negligable profit loss, one can assume that item insertions happen only at a logaritmic number of times. Still, directly applaying the construction of Bienstock et al. [11] gives a formulation which is super-polynomial in size. Thus, we develop a set of patterns describing how the maximum profit of the inserted items evolves over time. Each pattern induces a union of polytopes, and each of those polytopes has integrality gap arbitrary close to 1. We perform rounding by extending the classical greedy algorithm for MAXKNAP by Dantzig [17].

In Chapter 3, we study polyhedral relaxations for MINKNAP with a disjunctive programming approach for the case when the objective function is fixed. We improve on the result of Bienstock and McClosky [10], providing a smaller relaxation while preserving most of the structural properties. Both results are based on the technique for grouping items into cost classes (this is why it is important that the objective function is fixed). The main difference is that our classes are non-uniform, and we exploit that there is an optimal point in the relaxation with at most two fractional components. This property holds for any item partitioning in such disjunctive formulations, and could potentially be used for more general MINKNAP settings.

In Chapter 4, we investigate structural properties of bounded pitch inequalities for MINKNAP, and their strength in reducing the integrality gap. In particular, we give a simple characterization of pitch-2 inequalities and an algorithm for their approximate separation. However, there is an example showing that inequalities of pitch-3 and higher have more complex structure. We have shown that pitch-1 and pitch-2 inequalities reduce the gap of MINKNAP to 3/2 in the case when the cost vector is equal to the profit vector. However, this is false in general considering the natural relaxation of MINKNAP. We prove that adding all the knapsack cover and inequalities of pitch at most k, for constant k, still gives the gap of 2. Moreover, bounded pitch alone can be much weaker than KC: we show that, for each fixed k, the integrality gap may be unbounded even if all pitch-k inequalities are added. Using the relation between bounded pitch and Chvátal-Gomory (CG) closures established in [13], this implies that, for each fixed q, the integrality gap of the q-th CG closure can be unbounded. A recent work by Bienstock and Zuckerberg [14] generalizes our result on approximate pitch-2 separation to inequalities with bounded coefficients.

Chapter 5 offers an alternative perspective to clustered planarity. We prove a variant of the strong Hanani–Tutte theorem for flat clustered graphs with two clusters and c-connected clustered graphs. As a byproduct, we immediately obtain an algorithm for testing c-planarity in those cases. It essentially consists of solving a linear system of equations over $\mathbb{Z}_2$. We remark that there exist more efficient algorithms for planarity testing using the Hanani–Tutte theorem such as those in [58, 59], which run in linear time. Moreover, in the case of x-monotone drawings a computational study [49] showed that the Hanani–Tutte approach [62] performs well in practice. This should come as no surprise, since Hanani–Tutte theory seems to provide solid theoretical foundations for graph planarity that bring together its combinatorial, algebraic, and computational aspects [81]. As a negative result, we construct a family of examples which shows that a straightforward extension of strong Hanani–Tutte theorem to flat clustered graphs with more than two clusters is not possible.

Chapter 2 contains a joint work with Yuri Faenza published in [19], while Chapter 4 is joint with Yuri Faenza, Monaldo Mastrolilli, and Ola Svensson [20]. Chapter 5 is a collaboration with Radoslav Fulek, Jan Kynčl, and Dömötör Pálvölgyi [24].

2 A PTAS for the time-invariant incremental knapsack

In order to model scenarios arising in real-world applications, a variety of knapsack problems have been introduced (see [32] for a survey) and recent works studied extensions of classical combinatorial optimization problems to multi-period settings, see e.g. [27, 42, 43]. At the intersection of those two streams of research, Bienstock et al. [11] proposed a generalization of MAXKNAP to a multi-period setting that they dubbed *Time-Invariant Incremental Knapsack* (IIK). In IIK, we are given a set of items $[n]$ with profits $p : [n] \to \mathbb{R}_{>0}$ and weights $w : [n] \to \mathbb{R}_{>0}$ and a knapsack with non decreasing capacity b_t over time $t \in [T]$. We can add items at each time as long as the capacity constraint is not violated, and once inserted, an item cannot be removed from the knapsack. The goal is to maximize the total profit, which is defined to be the sum, over $t \in [T]$, of profits of items in the knapsack at time t.

IIK models a scenario where available resources (e.g. money, labour force) augment over time in a predictable way, allowing to grow our portfolio. Take e.g. a bond market with an extremely low level of volatility, where all coupons render profit only at their common maturity time T (*zero-coupon* bonds) and an increasing budget over time that allows buying more and more (differently sized and priced) packages of those bonds. For variations of MAXKNAP that have been used to model financial problems, see [32]. A different application arises in government-type decision processes, where items are assets of public utility (schools, parks, etc.) that can be built at a given cost and give a yearly benefit (both constant over the years), and the community will profit each year those assets are available.

Previous work on IIK. Although the first publication on IIK appeared just very recently [18], it was previously studied in [11] and several PhD theses [27, 42, 44]. Here we summarize all those results. In [11], IIK is shown to be strongly NP-hard and an instance showing that the natural LP relaxation has unbounded integrality gap is provided. In the same paper, a PTAS is designed for $T = O(\log n)$. This improves over [42], where a PTAS for the special case $p = w$ is given when T is a constant. Again when $p = w$, a $1/2$-approximation algorithm for generic T is provided in [27]. Results from [44] can be adapted to give an algorithm that solves IIK in time polynomial in n and of order $(\log T)^{O(\log T)}$ for a fixed approximation guarantee ε [41]. The authors in [18] provide an alternative PTAS for IIK with constant T, and a 1/2-approximation for arbitrary T with under the assumption that every item alone fits into the knapsack at $t = 1$.

Our contributions. In this chapter, we give an algorithm for computing a $(1-\varepsilon)$-approximated solution for IIK that depends polynomially on the number n of items and, for any fixed ε, also polynomially on the number of times T. In particular, our algorithm provides a PTAS for IIK, regardless of T.

Theorem 1. *Given $\varepsilon \in \mathbb{R}_{>0}$ and an instance $\mathscr{I}$ of* IIK *with n items and $T \geq 2$ times, there exists an algorithm that produces a $(1-\varepsilon)$-approximation to the optimal solution of $\mathscr{I}$ in time required to solve $O(nT^{h(\varepsilon)})$ LP problems with $\Theta(nT)$ variables and constraints. Here, $h : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 1}$ is a function depending on ε only. In particular, there exists a PTAS for* IIK.

Theorem 1 dominates all previous results on IIK [11, 18, 27, 42, 44] and, due to the hardness results in [11], settles the complexity of the problem. Interestingly, it is based on designing a disjunctive formulation – a tool mostly common among integer programmers and practitioners[1] – and then rounding the solution to its linear relaxation with a greedy-like algorithm. We see Theorem 1 as an important step towards the understanding of the complexity landscape of knapsack problems over time. Theorem 1 is proved in Section 2.2: see the end of the current section for a sketch of the techniques we use and a detailed summary of Section 2.2. In Section 2.3, we show some extensions of Theorem 1 to more general problems.

Related work on other knapsack problems. Bienstock et al. [11] discuss the relation between IIK and the generalized assignment problem (GAP), highlighting the differences between those problems. In particular, there does not seem to be a direct way to apply to IIK the $(1-1/e-\varepsilon)$ approximation algorithm [23] for GAP. Other generalizations of MAXKNAP related to IIK, but whose current solving approaches do not seem to extend, are the multiple knapsack (MKP) and unsplittable flow on a path (UFP) problems. In Appendix A.2 we discuss those problems in order to highlight the new ingredients introduced by our approach.

Notation and basic assumptions. We refer to [40] for basic definitions and facts on approximation algorithms and polytopes. Given an integer k, we write $[k] := \{1, \ldots, k\}$ and $[k]_0 := [k] \cup \{0\}$. Given a polyhedron $Q \subseteq \mathbb{R}^n$, a *relaxation* $P \subseteq \mathbb{R}^n$ is a polyhedron such that $Q \subseteq P$ and the integer points in P and Q coincide. The *size* of a polyhedron is the minimum number of facets in an extended formulation for it, which is well-known to coincide with the minimum number of inequalities in any linear description of the extended formulation.

We assume that expressions $\frac{1}{\varepsilon}$, $(1+\varepsilon)^j$, $\log_{1+\varepsilon} \frac{T}{\varepsilon}$ and similar are to be rounded up to the closest integer. This is just done for simplicity of notation and can be achieved by replacing ε with an appropriate constant fraction of it, which will not affect the asymptotic running time.

[1] See Appendix A.1 for a discussion on disjunctive programming.

2.1 Overview of basic techniques

In order to illustrate the ideas behind the proof of Theorem 1, let us first recall one of the PTAS for the classical MAXKNAP with capacity β, n items, profit and weight vector p and w respectively. Recall the greedy algorithm for knapsack:

1. Sort items so that $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$.

2. Set $\bar{x}_i = 1$ for $i = 1, \ldots, \bar{\imath}$, where $\bar{\imath}$ is the maximum integer s.t. $\sum_{1 \leq i \leq \bar{\imath}} w_i \leq \beta$.

It is well-known that $p^T \bar{x} \geq p^T x^* - \max_{i \geq \bar{\imath}+1} p_i$, where x^* is the optimal solution to the linear relaxation. A PTAS for MAXKNAP can then be obtained as follows: "guess" a set S_0 of $\frac{1}{\varepsilon}$ items with $w(S_0) \leq \beta$ and consider the "residual" knapsack instance $\mathscr{I}$ obtained removing items in S_0 and items ℓ with $p_\ell > \min_{i \in S_0} p_i$, and setting the capacity to $\beta - w(S_0)$. Apply the greedy algorithm to $\mathscr{I}$ as to obtain solution S. Clearly $S_0 \cup S$ is a feasible solution to the original knapsack problem. The best solutions generated by all those guesses can be easily shown to be a $(1-\varepsilon)$-approximation to the original problem.

Recall that IIK can be defined as follows.

$$
\begin{array}{lll}
\max & \sum_{t \in [T]} p^T x_t & \\
\text{s.t.} & w^T x_t \leq b_t \quad \forall t \in [T] & \\
& x_t \leq x_{t+1} \quad \forall t \in [T-1] & \\
& x_t \in \{0,1\}^n \quad \forall t \in [T]. &
\end{array} \tag{2.1}
$$

By definition, $0 < b_t \leq b_{t+1}$ for $t \in [T-1]$. We also assume wlog that $1 = p_1 \geq p_2 \geq \ldots \geq p_n$.

When trying to extend the PTAS above for MAXKNAP to IIK, we face two problems. First, we have multiple times, and a standard guessing over all times will clearly be exponential in T. Second, when inserting an item into the knapsack at a specific time, we are clearly imposing this decision on all times that succeed it, and it is not clear a priori how to take this into account.

We solve these issues by proposing an algorithm that, in a sense, still follows the general scheme of the greedy algorithm sketched above: after some preprocessing, guess items (and insertion times) that give high profit, and then fill the remaining capacity with an LP-driven integral solution. However, the way of achieving this is different from the PTAS above. In particular, some of the techniques we introduced are specific for IIK and not to be found in methods for solving non-incremental knapsack problems.

An overview of the algorithm:

(i) *Sparsification and other simplifying assumptions.* We first show that by losing at most a 2ε fraction of the profit, we can assume the following (see Section 2.2.1): item 1, which has the maximum profit, is inserted into the knapsack at some time; the capacity of the knapsack only increases and hence the insertion of items can only happen at $J = O(\frac{1}{\varepsilon}\log T)$ times (we call them *significant*); and the profit of each item is either much smaller than $p_1 = 1$ or it takes one of $K = O(\frac{1}{\varepsilon}\log\frac{T}{\varepsilon})$ possible values (we call them *profit classes*).

(ii) *Guessing of a stairway.* The operations in the previous step give a $J \times K$ grid of "significant times" vs "profit classes" with $O(\frac{1}{\varepsilon^2}\log^2\frac{T}{\varepsilon})$ entries in total. One could think of the following strategy: for each entry (j, k) of the grid, guess how many items of profit class k are inserted in the knapsack at time t_j. However, those entries are still too many to perform guessing over all of them. Instead, we proceed as follows: we guess, for each significant time t_j, *which is the class* k of maximum profit that has an element in the knapsack at time t_j. Then, for profit class k and carefully selected profit classes "close" to k, we either *guess exactly how many items* are in the knapsack at time t_j or if these are *at least* $\frac{1}{\varepsilon}$. Each of the guesses leads to a natural IP. The optimal solution to one of the IPs is an optimal solution to our original problem. Clearly, the number of possible guesses affects the number of the IPs, hence the overall complexity. We introduce the concept of "stairway" to show that these guesses are polynomially many for fixed ϵ. See Section 2.2.2 for details. We remark that, from this step on, we substantially differ from the approach of [11], which is also based on a disjunctive formulation.

(iii) *Solving the linear relaxations and rounding.* Fix an IP generated at the previous step, and let x^* be the optimal solution of its linear relaxation. A classical rounding argument relies on LP solutions having a small number of fractional components. Unfortunately, x^* is not as simple as that. However, we show that, after some massaging, we can control the entries of x^* where "most" fractional components appear, and conclude that the profit of $\lfloor x^* \rfloor$ is close to that of x^*. See Section 2.2.3 for details. Hence, looping over all guessed IPs and outputting vector $\lfloor x^* \rfloor$ of maximum profit concludes the algorithm.

2.2 A PTAS for IIK

2.2.1 Reducing IIK to special instances and solutions

Our first step will be to show that we can reduce IIK, without loss of generality, to solutions and instances with a special structure. The first reduction is immediate: we restrict to solutions where the highest profit item is inserted in the knapsack at some time. We call these 1-*in* *solutions.* This can be assumed by guessing which is the highest profit item that is inserted in the knapsack, and reducing to the instance where all higher profit items have been excluded. Since we have n possible guesses, the running time is scaled by a factor $O(n)$.

Observation 2.2.1. *Suppose there exists a function* $f : \mathbb{N} \times \mathbb{N} \times \mathbb{R}_{>0}$ *such that, for each* $n, T \in \mathbb{N}$, $\varepsilon > 0$, *and any instance of* IIK *with* n *items and* T *times, we can find a* $(1-\varepsilon)$*-approximation to a* 1*-in solution of highest profit in time* $f(n, T, \varepsilon)$. *Then we can find a* $(1-\varepsilon)$*-approximation to any instance of* IIK *with* n *items and* T *times in time* $O(n) \cdot f(n, T, \varepsilon)$.

Now, let $\mathscr{I}$ be an instance of IIK with n items, let $\varepsilon > 0$. We say that $\mathscr{I}$ is ε*-well-behaved* if it satisfies the following properties.

(ε1) For all $i \in [n]$, one has $p_i = (1+\varepsilon)^{-j}$ for some $j \in \{0, 1, \ldots, \log_{1+\varepsilon} \frac{T}{\varepsilon}\}$, or $p_i \leq \frac{\varepsilon}{T}$.

(ε2) $b_t = b_{t-1}$ for all $t \in [T]$ such that $(1+\varepsilon)^{j-1} < T - t + 1 < (1+\varepsilon)^j$ for some $j \in \{0, 1, \ldots, \log_{1+\varepsilon} T\}$, where we set $b_0 = 0$.

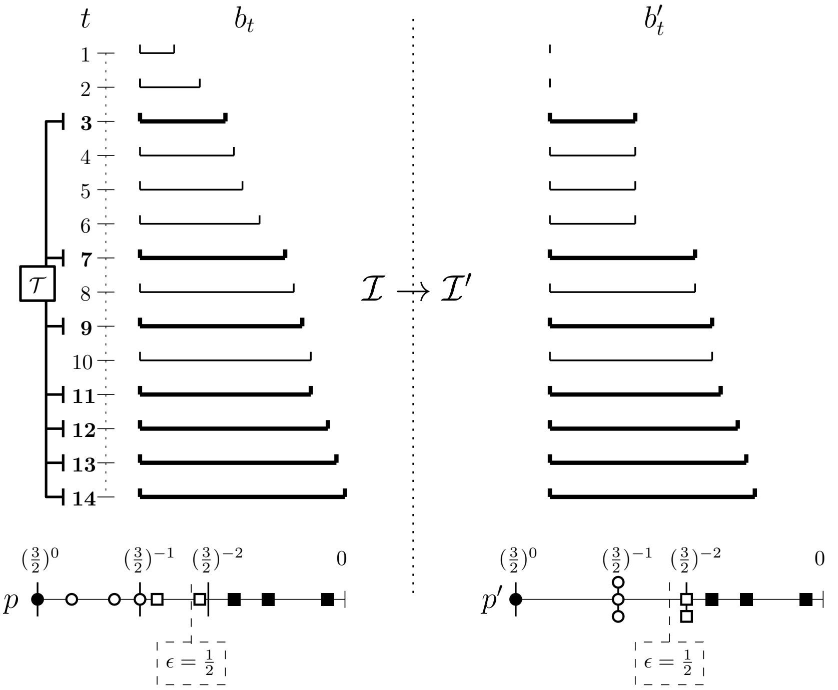

Figure 2.1 – An example of obtaining an ε-well-behaved instance for $\varepsilon = \frac{1}{2}$ and $T = 14$.

See Figure 2.1 for an example. Note that condition (ε2) implies that the capacity can change only during the set of times $\mathscr{T} := \{t \in [T] : t = T + 1 - (1+\varepsilon)^j \text{ for some } j \in \mathbb{N}\}$, with $|\mathscr{T}| = O(\log_{1+\varepsilon} T)$. $\mathscr{T}$ clearly gets sparser as t becames smaller. Note that for T not being a degree of $(1+\varepsilon)$ there will be a small fraction of times t at the beginning with capacity 0; see Figure 2.1.

Next theorem implies that we can, wlog, assume that our instances are ε-well-behaved (and our solutions are 1-in).

Theorem 2. *Suppose there exists a function $g : \mathbb{N} \times \mathbb{N} \times \mathbb{R}_{>0}$ such that, for each $n, T \in \mathbb{N}$, $\varepsilon > 0$, and any ε-well-behaved instance of* IIK *with n items and T times, we can find a $(1-2\varepsilon)$-approximation to a* 1*-in solution of highest profit in time $g(n, T, \varepsilon)$. Then we can find a $(1-4\varepsilon)$-approximation to any instance of* IIK *with n items and T times in time $O(T + n(n + g(n, T, \varepsilon))$.*

Fix an IIK instance $\mathscr{I}$. The reason why we can restrict ourselves to finding a 1-in solution is Observation 2.2.1. Denote with $\mathscr{I}'$ the instance with n items having the same weights as in $\mathscr{I}$, T times, and the other parameters defined as follows:

- For $i \in [n]$, if $(1+\varepsilon)^{-j} \le p_i < (1+\varepsilon)^{-j+1}$ for some $j \in \{0, 1, \ldots, \log_{1+\varepsilon} \frac{T}{\varepsilon}\}$, set $p'_i := (1+\varepsilon)^{-j}$; otherwise, set $p'_i := p_i$. Note that we have $1 = p'_1 \ge p'_2 \ge \ldots \ge p'_n$.
- For $t \in [T]$ and $(1+\varepsilon)^{j-1} < T - t + 1 \le (1+\varepsilon)^j$ for some $j \in \{0, 1, \ldots, \log_{1+\varepsilon} T\}$, set $b'_t := b_{T-(1+\varepsilon)^j+1}$, with $b'_0 := 0$.

One easily verifies that $\mathscr{I}'$ is ε-well-behaved. Moreover, $b'_t \le b_t$ for all $t \in [T]$ and $\frac{p_i}{1+\varepsilon} \le p'_i \le p_i$ for $i \in [n]$, so we deduce:

Claim 1. *Any solution $\bar{x}$ feasible for $\mathscr{I}'$ is also feasible for $\mathscr{I}$, and $p(\bar{x}) \ge p'(\bar{x})$.*

We also prove the following.

Claim 2. *Let x^* be a* 1*-in feasible solution of highest profit for $\mathscr{I}$. There exists a* 1*-in feasible solution x' for $\mathscr{I}'$ such that $p'(x') \ge (1-\varepsilon)^2 p(x^*)$.*

Proof. Define $x' \in \{0,1\}^{Tn}$ as follows:

$$x'_t := x^*_{T-(1+\varepsilon)^j+1} \quad \text{if } (1+\varepsilon)^{j-1} < T - t + 1 \le (1+\varepsilon)^j$$
$$\text{for } j \in \{0, 1, \ldots, \log_{1+\varepsilon} T\}, \text{with } x^*_0 = 0.$$

In order to prove the claim we first show that x' is a feasible 1-in solution for $\mathscr{I}'$. Indeed, it is 1-in, since by construction $x'_{T,1} = x^*_{T,1} = 1$. It is feasible, since for t such that $(1+\varepsilon)^{j-1} < T - t + 1 \le (1+\varepsilon)^j$, $j \in \{0, 1, \ldots, \log_{1+\varepsilon} T\}$ we have

$$w^T x'_t = w^T x^*_{T-(1+\varepsilon)^j+1} \le b_{T-(1+\varepsilon)^j+1} = b'_t.$$

Comparing $p'(x')$ and $p(x^*)$ gives

$$\begin{array}{rclcl} p'(x') & \ge & \sum_{t\in[T]} \sum_{i\in[n]} p'_i x'_{t,i} & = & \sum_{i\in[n]} (T - t_{i,\min}(x') + 1) p'_i \\ & \ge & \sum_{i\in[n]} \frac{1}{1+\varepsilon} (T - t_{i,\min}(x^*) + 1) p'_i & \ge & \sum_{i\in[n]} \frac{1}{(1+\varepsilon)^2} (T - t_{i,\min}(x^*) + 1) p_i \\ & = & (\frac{1}{1+\varepsilon})^2 p(x^*) & \ge & (1-\varepsilon)^2 p(x^*), \end{array}$$

where $t_{i,\min}(v) := \min\{t \in [T] : v_{t,i} = 1\}$ for $v \in \{0,1\}^{Tn}$. □

Proof of Theorem 2. Let $\hat{x}$ be a 1-in solution of highest profit for $\mathscr{I}'$ and $\bar{x}$ is a solution to $\mathscr{I}'$ that is a $(1-\varepsilon)$-approximation to $\hat{x}$. Claim 1 and Claim 2 imply that $\bar{x}$ is feasible for $\mathscr{I}$ and we deduce:

$$p(\bar{x}) \geq p'(\bar{x}) \geq (1-2\varepsilon)p'(\hat{x}) \geq (1-2\varepsilon)p'(x') \geq (1-2\varepsilon)(1-\varepsilon)^2 p(x^*) \geq (1-4\varepsilon)p(x^*).$$

In order to compute the running time, it is enough to bound the time required to produce $\mathscr{I}'$. Vector p' can be produced in time $O(n)$, while vector b' in time T. Moreover, the construction of the latter can be performed before fixing the highest profit object that belongs to the knapsack (see Observation 2.2.1). The thesis follows.

■

2.2.2 A disjunctive relaxation

Fix $\varepsilon > 0$. Because of Theorem 2, we can assume that the input instance $\mathscr{I}$ is ε-well-behaved. We call all times from $\mathscr{T}$ *significant*. Note that a solution over the latter times can be naturally extended to a global solution by setting $x_t = x_{t-1}$ for all non-significant times t. We denote significant times by $t(1) < t(2) < \cdots < t(|\mathscr{T}|)$. In this section, we describe an IP over feasible 1-in solutions of an ε-well-behaved instance of IIK. The feasible region of this IP is the union of different regions, each corresponding to a partial assignment of items to significant times. In Section 2.2.3 we give a strategy to round an optimal solution of the LP relaxation of the IP to a feasible integral solution with a $(1-2\varepsilon)$-approximation guarantee. Together with Theorem 2 (taking $\varepsilon' = \frac{\varepsilon}{4}$), this implies Theorem 1.

In order to describe those partial assignments, we introduce some additional notation. We say that items having profit $(1+\varepsilon)^{-k}$ for $k \in [\log_{1+\varepsilon} \frac{T}{\varepsilon}]$, belong to *profit class k*. Hence bigger profit classes correspond to items with smaller profit. All other items are said to belong to the *small* profit class. Note that there are $O(\frac{1}{\varepsilon}\log\frac{T}{\varepsilon})$ profit classes (some of which could be empty). Our partial assignments will be induced by special sets of vertices of a related graph called *grid*.

Definition 3. *Let $J \in \mathbb{Z}_{>0}, K \in \mathbb{Z}_{\geq 0}$, a grid of dimension $J \times (K+1)$ is the graph $G_{J,K} = ([J] \times [K]_0, E)$, where*

$$\begin{aligned} E := \{\{u,v\} :\ & u,v \in [J] \times [K]_0,\ u = (j,k) \\ & \textit{and either } v = (j+1,k) \textit{ or } v = (j,k+1)\ \}. \end{aligned}$$

Definition 4. *Given a grid $G_{J,K}$, we say that*

$$S := \{(j_1,k_1),(j_2,k_2),\ldots,(j_{|S|},k_{|S|})\} \subseteq V(G_{J,K})$$

is a stairway *if $j_h > j_{h+1}$ and $k_h < k_{h+1}$ for all $h \in [|S|-1]$.*

Lemma 5. *There are at most 2^{K+J+1} distinct stairways in $G_{J,K}$.*

Proof. The first coordinate of any entry of a stairway can be chosen among J values, the second coordinate from $K+1$ values. By Definiton 4, each stairway correspond to exactly one choice of sets $J_1 \subseteq [J]$ for the first coordinates and $K_1 \subseteq [K]_0$ for the second, with $|K_1| = |J_1|$.

□

Now consider the grid graph with $J := |\mathcal{T}| = \theta(\frac{1}{\varepsilon}\log T)$, $K = \log_{1+\varepsilon}\frac{T}{\varepsilon}$, and a stairway S with $k_1 = 0$. See Figure 2.2 for an example. This corresponds to a partial assignment that can be informally described as follows. Let $(j_h, k_h) \in S$ and $t_h := t(j_h)$. In the corresponding partial assignment no item belonging to profit classes $k_h \le k < k_{h+1}$ is inside the knapsack at any time $t < t_h$, while the first time an item from profit class k_h is inserted into the knapsack is at time t_h (if $j_{|S|} > 1$ then the only items that the knapsack can contain at times $1,\dots,t_{|S|}-1$ are the items from the small profit class). Moreover, for each $h \in [|S|]$, we focus on the family of profit classes $\mathcal{K}_h := \{k \in [K] : \ k_h \le k \le k_h + C_\varepsilon\}$ with $C_\varepsilon = \log_{1+\varepsilon}\frac{1}{\varepsilon}$. For each $k \in \mathcal{K}_h$ and every (significant) time t in the set $\mathcal{T}_h := \{t \in \mathcal{T} : \ t_{h-1} < t \le t_h\}$, we will either specify exactly the number of items taken from profit class k at time t, or impose that there are at least $\frac{1}{\varepsilon}+1$ of those items (this is established by map ρ_h below). Note that we can assume that the items taken within a profit class are those with minimum weight: this may exclude some feasible 1-in solutions, but it will always keep at least a feasible 1-in solution of maximum profit. No other constraint is imposed.

More formally, set $k_{|S|+1} = K+1$ and for each $h = 1,\dots,|S|$:

i) Set $x_{t,i} = 0$ for all $t \in [t_h - 1]$ and each item i in a profit class $k \in [k_{h+1}-1]$.

ii) Fix a map $\rho_h : \mathcal{T}_h \times \mathcal{K}_h \to \{0,1,\dots,\frac{1}{\varepsilon}+1\}$ such that for all $t \in \mathcal{T}_h$ one has $\rho_h(t,k_h) \ge 1$ and $\rho_h(\bar{t},k) \ge \rho_h(t,k), \ \forall (\bar{t},k) \in \mathcal{T}_h \times \mathcal{K}_h, \ \bar{t} \ge t$.

Additionally, we require $\rho_h(\bar{t},k) \ge \rho_{h+1}(t,k)$ for all $h \in [|S|-1]$, $k \in \mathcal{K}_h \cap \mathcal{K}_{h+1}$, $\bar{t} \in \mathcal{T}_h$, $t \in \mathcal{T}_{h+1}$. Thus, we can merge all ρ_h into a function $\rho : \cup_{h\in[|S|]}(\mathcal{T}_h \times \mathcal{K}_h) \to \{0,1,\dots,\frac{1}{\varepsilon}+1\}$. For each profit class $k \in [K]$ we assume that items from this class are $I_k = \{1(k),\dots,|I_k|(k)\}$, so that $w_{1(k)} \le w_{2(k)} \le \dots \le w_{|I_k|(k)}$. Based on our choice (S,ρ) we define the polytope:

$$
\begin{array}{llll}
P(S,\rho) = \{x \in \mathbb{R}^{Tn} : & & w^T x_t \le b_t & \forall t \in [T] \\
& & x_t \le x_{t+1} & \forall t \in [T-1] \\
& & 0 \le x_t \le 1 & \forall t \in [T] \\
& \forall h \in [|S|] : & & \\
& & x_{t,i(k)} = 0, & \forall t < t_h, \ \forall k < k_{h+1}, \ \forall i(k) \in I_k \\
& & x_{t,i(k)} = 1, & \forall t \in \mathcal{T}_h, \ \forall k \in \mathcal{K}_h, \ \forall i(k) : i \le \rho(t,k) \\
& & x_{t,i(k)} = 0, & \forall t \in \mathcal{T}_h, \ \forall k \in \mathcal{K}_h : \rho(t,k) \le \frac{1}{\varepsilon}, \\
& & & \forall i(k) : i > \rho(t,k)\}.
\end{array}
$$

The linear inequalities are those from the IIK formulation. The first set of equations impose

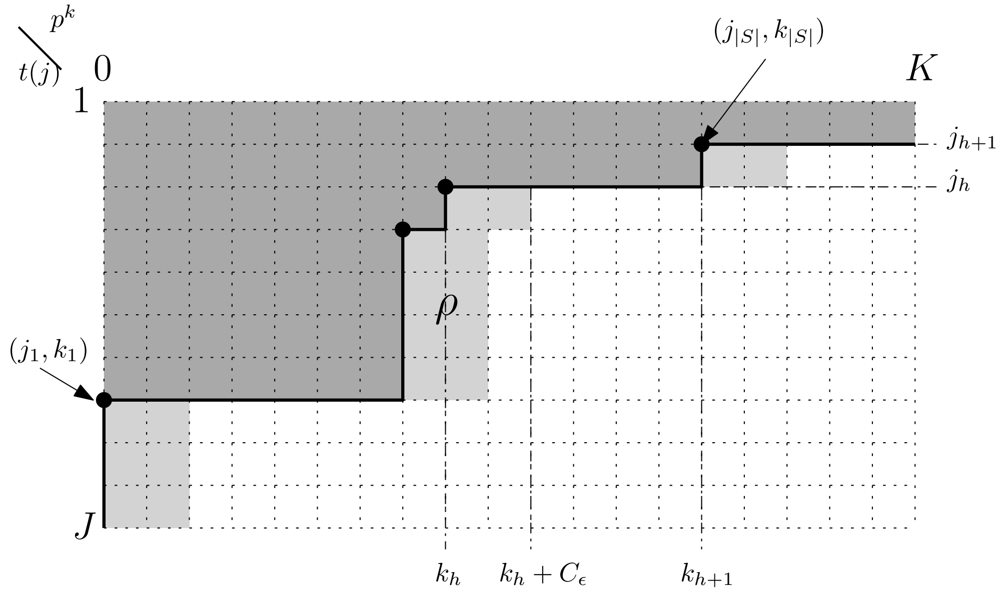

Figure 2.2 – An example of a stairway S, given by thick black dots. Entries (j, k) lying in the light grey area are those for which a value ρ is specified. No item corresponding to the entries in the dark grey area is taken, except on the boundary in bold.

that, at each time t, we do not take any object from a profit class k, if we guessed that the highest profit object in the solution at time t belongs to a profit class $k' > k$ (those are entries corresponding to the dark grey area in Figure 2.2). The second set of equations impose that for each time t and class k for which a guess $\rho(t, k)$ was made (light grey area in Figure 2.2), we take the $\rho(t, k)$ items of smallest weight. As mentioned above, this is done without loss of generality: since profits of objects from a given profit class are the same, we can assume that the optimal solution insert first those of smallest weight. The last set of equations imply that no other object of class k is inserted in time t if $\rho(t, k) \le \frac{1}{\varepsilon}$.

Note that some choices of S, ρ may lead to empty polytopes. Fix S, ρ, an item i and some time t. If, for some $t' \le t$, $x_{t',i} = 1$ explicitly appears in the definition of $P(S, \rho)$ above, then we say that i is *t-included*. Conversely, if $x_{\bar{t},i} = 0$ explicitly appears for some $\bar{t} \ge t$, then we say that i is *t-excluded*.

Theorem 6. *Any optimal solution of*

$$\max \sum_{t \in [T]} p_t^T x_t \quad s.t. \quad x \in \left(\cup_{S,\rho} P(S, \rho) \right) \cap \{0,1\}^{Tn}$$

is a 1*-in solution of maximum profit for* $\mathscr{I}$*. Moreover, the the number of constraints of the associated LP relaxation is at most* $nT^{f(\varepsilon)}$ *for some function* $f : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ *depending on* ε *only.*

Proof. Note that one of the choices of (S, ρ) will be the correct one, i.e. it will predict the stairway S associated to an optimal 1-in solution, as well as the number of items that this solution takes for each entry of the grid it guessed. Then there exists an optimal solution that takes, for each time t and class k for which a guess $\rho(t,k)$ was made, the $\rho(t,k)$ items of smallest weight from this class, and no other object if $\rho(t,k) \leq \frac{1}{\varepsilon}$. These are exactly the constraints imposed in $P(S,\rho)$. The second part of the statement follows from the fact that the possible choices of (S,ρ) are

$$\begin{aligned} &(\#\text{ stairways}) \cdot (\#\text{ possible values in each entry of } \rho)^{(\#\text{ entries of a vector } \rho)} \\ &= 2^{O(\frac{1}{\varepsilon}\log\frac{T}{\varepsilon})} \cdot O(\tfrac{1}{\varepsilon})^{O(\frac{1}{\varepsilon}\log\frac{T}{\varepsilon})C_\varepsilon} \\ &= (\tfrac{T}{\varepsilon})^{O(\frac{1}{\varepsilon})} \cdot (\tfrac{T}{\varepsilon})^{O((\frac{1}{\varepsilon})^3)}, \end{aligned}$$

and each (S,ρ) has $g(\varepsilon)O(Tn)$ constraints, where g depends on ε only. □

2.2.3 Rounding

By convexity, there is a choice of S and ρ given as in the previous section such that any optimal solution of

$$\max \sum_{t\in[T]} p^T x_t \quad \text{s.t.} \quad x \in P(S,\rho) \tag{2.2}$$

is also an optimal solution to

$$\max \sum_{t\in[T]} p^T x_t \quad \text{s.t.} \quad x \in \mathrm{conv}(\cup_{S,\rho} P(S,\rho)).$$

Hence, we can focus on rounding an optimal solution x^* of (2.2). We assume that the items are ordered so that $\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$. Moreover, let $\mathscr{I}^t$ (resp. $\mathscr{E}^t$) be the set of items from $[n]$ that are t-included (resp. t-excluded) for $t \in [T]$, and let $W_t := w^T x_t^*$.

Algorithm 1

1: Set $\bar{x}_0 = \mathbf{0}$.

2: For $t = 1, \ldots, T$:

   (a) Set $\bar{x}_t = \bar{x}_{t-1}$.

   (b) Set $\bar{x}_{t,i} = 1$ for all $i \in \mathscr{I}^t$.

   (c) While $W_t - w^T \bar{x}_t > 0$:

      (i) Select the smallest $i \in [n]$ such that $i \notin \mathscr{E}^t$ and $\bar{x}_{t,i} < 1$.

      (ii) Set $\bar{x}_{t,i} = \bar{x}_{t,i} + \min\{1 - \bar{x}_{t,i}, \frac{W_t - w^T \bar{x}_t}{w_i}\}$.

Respecting the choices of S and ρ, i.e. included/excluded items at each time t, Algorithm

1 greedly adds objects into the knapsack, until the total weight is equal to W_t. Recall that in MAXKNAP one obtains a rounded solution which differs from the fractional optimum by the profit of at most one item. Here the fractionality pattern is more complex, but still under control. In fact, as we show below, $\bar{x}$ is such that $\sum_{t\in[T]} p^T \bar{x}_t = \sum_{t\in[T]} p^T x_t^*$ and, for each $h \in [|S|]$ and $t \in [T]$ such that $t_h \leq t < t_{h-1}$, vector $\bar{x}_t$ has at most $|S| - h + 1$ fractional components that do not correspond to items in profit classes $k \in K$ with at least $\frac{1}{\epsilon}+1$ t-included items. We use this fact to show that $\lfloor \bar{x} \rfloor$ is an integral solution that is $(1-2\epsilon)$-optimal.

Theorem 7. *Let x^* be an optimal solution to* (2.2). *Algorithm 1 produces, in time $O(T+n)$, a vector $\bar{x} \in P(S,\rho)$ such that $\sum_{t\in[T]} p^T \lfloor \bar{x}_t \rfloor \geq (1-2\epsilon) \sum_{t\in[T]} p^T x_t^*$.*

Theorem 7 will be proved in a series of intermediate steps.

Claim 3. *Let $t \in [T-1]$. Then:*

(i) $\mathscr{I}^t \subseteq \mathscr{I}^{t+1}$ *and* $\mathscr{E}^t \supseteq \mathscr{E}^{t+1}$.

(ii) $\mathscr{I}^{t+1} \setminus \mathscr{I}^t \subseteq \mathscr{E}^t$.

Proof. (i) Immediately from the definition.

(ii) If $\mathscr{I}^{t+1} \setminus \mathscr{I}^t \neq \emptyset$, we deduce $t+1 \in \mathscr{T}$. Let $h \in [|S|]$ be such that $t_h \leq t < t_{h-1}$, where for completeness $t_0 = T+1$. By construction, the items $\mathscr{I}^{t+1} \setminus \mathscr{I}^t$ can only be in buckets $k : k_h \leq k < k_{h+1} + C_\varepsilon$ where either $k < k_{h+1}$ or $k \in \mathscr{K}_{h+1}$ and $\rho(t,k) \leq \frac{1}{\varepsilon}$. Hence, all items from $\mathscr{I}^{t+1} \setminus \mathscr{I}^t$ are t-excluded.

□

Recall that, for $t \in [T]$, $W_t := w^T x_t^*$. The proof of the following claim easily follows by construction.

Claim 4. *(i) For any $h \in [|S|]$, $t \in [t_h - 1]$, $k < k_{h+1}$ and $i \in I_k$, one has $x_{t,i}^* = \bar{x}_{t,i} = 0$.*

(ii) For $t \in [T-1]$ and $i \in [n]$, one has $\bar{x}_{t+1,i} \geq \bar{x}_{t,i} \geq 0$.

(iii) For $t \in [T]$, one has: $x_{t,i}^* = \bar{x}_{t,i} = 1$ for $i \in \mathscr{I}^t$ and $x_{t,i}^* = \bar{x}_{t,i} = 0$ for $i \in \mathscr{E}^t$.

Define $\mathscr{F}_t := \{i \in [n] : \ 0 < \bar{x}_{t,i} < 1\}$ to be the set of fractional components of $\bar{x}_t$ for $t \in [T]$. Recall that Algorithm 1 sorts items by monotonically decreasing profit/weight ratio. For items from a given profit class $k \in [K]$, this induces the order $i(1) < i(2) < \ldots$ – i.e. by monotonically increasing weight – since all $i(k) \in I_k$ have the same profit.

The following claim shows that $\bar{x}$ is in fact an optimal solution to $\max\{x : x \in P(S,\rho)\}$.

Claim 5. *For each $t \in [T]$, one has $w^T \bar{x}_t = w^T x_t^*$ and $p^T \bar{x}_t = p^T x_t^*$.*

Proof. We first prove the statement on the weights by induction on t, the basic step being trivial. Suppose it is true up to time $t-1$. The total weight of solution $\bar{x}_t$ after step (b) is

$$\begin{aligned} w^T\bar{x}_{t-1} + \textstyle\sum_{i\in\mathscr{I}^t\setminus\mathscr{I}^{t-1}} w_i(1-\bar{x}_{t-1,i}) &= W_{t-1} + \textstyle\sum_{i\in\mathscr{I}^t\setminus\mathscr{I}^{t-1}} w_i(1-x^*_{t-1,i}) \\ &= W_{t-1} + \textstyle\sum_{i\in\mathscr{I}^t\setminus\mathscr{I}^{t-1}} w_i \overset{(*)}{\leq} W_t, \end{aligned}$$

where the equations follow by induction, Claim 4.(iii), and Claim 3.(ii), and $(*)$ follows by observing $w^Tx^*_t - w^Tx^*_{t-1} \geq \sum_{i\in\mathscr{I}^t\setminus\mathscr{I}^{t-1}} w_i$. $\bar{x}_t$ is afterwords increased until its total weight is at most W_t. Last, observe that W_t is always achieved, since it is achieved by x^*_t. This concludes the proof of the first statement.

We now move to the statement on profits. Note that it immediately follows from the optimality of x^* and the first part of the claim if we show that $\bar{x}$ is the solution maximizing p^Tx_t for all $t\in[T]$, among all $x\in P(S,\rho)$ that satisfy $w^Tx_t = W_t$ for all $t\in[T]$. So let us prove the latter. Suppose by contradiction this is not the case, and let $\tilde{x}$ be one such solution such that $p^T\tilde{x}_t > p^T\bar{x}_t$ for some $t\in[T]$. Among all such $\tilde{x}$, take one that is lexicographically maximal, where entries are ordered $(1,1),(1,2),\ldots,(1,n),(2,1)\ldots,(T,n)$. Then there exists $\tau\in[T]$, $\ell\in[n]$ such that $\tilde{x}_{\tau,\ell} > \bar{x}_{\tau,\ell}$. Pick τ minimum such that this happens, and ℓ minimum for this τ. Using that $\bar{x}_{\tau,i} = \tilde{x}_{\tau,i}$ for $i\in\mathscr{I}^\tau\cup\mathscr{E}^\tau$ since $\bar{x},\tilde{x}\in P(S,\rho)$ and recalling $w^T\bar{x}_\tau = w^T\tilde{x}_\tau = W_\tau$ one obtains

$$\sum_{i\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau)} w_i\bar{x}_{\tau,i} = \sum_{i\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau)} w_i\tilde{x}_{\tau,i}. \tag{2.3}$$

It must be that $\bar{x}_{\tau,\ell} < 1$, since $\bar{x}_{\tau,\ell} < \tilde{x}_{\tau,\ell} \leq 1$, so step (c) of Algorithm 1 in iteration τ did not change any item $\hat{\ell} > \ell$, i.e. $\bar{x}_{\tau,\hat{\ell}} = \bar{x}_{\tau-1,\hat{\ell}}$ for each $\hat{\ell} > \ell$. Additionally, $\ell\notin\mathscr{I}^\tau$ beacuse $\bar{x}_{\tau,\ell} < 1$, and $\ell\notin\mathscr{E}^\tau$ since otherwise $\bar{x}_{\tau,\ell} = \tilde{x}_{\tau,\ell} = 0$. Hence, $\ell\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau)$. By moving the terms corresponding to $\hat{\ell} > \ell$ to the right-hand side, we rewrite (2.3) as follows

$$\sum_{\substack{\bar{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \bar{\ell}\leq\ell}} w_{\bar{\ell}}\bar{x}_{\tau,\bar{\ell}} = \sum_{\substack{\bar{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \bar{\ell}\leq\ell}} w_{\bar{\ell}}\tilde{x}_{\tau,\bar{\ell}} + \sum_{\substack{\hat{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \hat{\ell}>\ell}} w_{\hat{\ell}}(\tilde{x}_{\tau,\hat{\ell}} - \underbrace{\bar{x}_{\tau,\hat{\ell}}}_{=\bar{x}_{\tau-1,\hat{\ell}}}).$$

By minimality of τ one has $\tilde{x}_{\tau-1} \leq \bar{x}_{\tau-1}$, so $w^T\tilde{x}_{\tau-1} = W_{\tau-1} = w^T\bar{x}_{\tau-1}$ implies $\tilde{x}_{\tau-1} = \bar{x}_{\tau-1}$ and thus

$$\sum_{\substack{\bar{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \bar{\ell}\leq\ell}} w_{\bar{\ell}}\bar{x}_{\tau,\bar{\ell}} = \sum_{\substack{\bar{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \bar{\ell}\leq\ell}} w_{\bar{\ell}}\tilde{x}_{\tau,\bar{\ell}} + \overbrace{\sum_{\substack{\hat{\ell}\in[n]\setminus(\mathscr{I}^\tau\cup\mathscr{E}^\tau):\\ \hat{\ell}>\ell}} w_{\hat{\ell}}(\tilde{x}_{\tau,\hat{\ell}} - \tilde{x}_{\tau-1,\hat{\ell}})}^{\geq 0}. \tag{2.4}$$

Note that the items in $[n]$ are ordered according to monotonically decreasing profit/weight ratio. By minimality of ℓ subject to τ we have that $\bar{x}_{\tau,\bar{\ell}} \geq \tilde{x}_{\tau,\bar{\ell}}$ for $\bar{\ell} < \ell$. Thus combining $\bar{x}_{\tau,\ell} < \tilde{x}_{\tau,\ell}$ with (2.4) gives that there exists $\beta < \ell$ such that $\bar{x}_{\tau,\beta} > \tilde{x}_{\tau,\bar{\ell}}$. Then for all $\bar{\tau} \geq \tau$, one can perturb $\tilde{x}$ by increasing $\tilde{x}_{\bar{t},\beta}$ and decreasing $\tilde{x}_{\bar{t},\ell}$ while keeping $\tilde{x} \in P(S,\rho)$ and $w^T\tilde{x}_{\bar{t}} = W_{\bar{t}}$, without decreasing $p^T\tilde{x}_{\bar{t}}$. This contradicts the choice of $\tilde{x}$ being lexicographically maximal. □

For $t \in [T]$ define $\mathscr{L}_t := \{k \in [K] : |I_k \cap \mathscr{I}^t| \geq \frac{1}{\epsilon} + 1\}$ to be the set of classes with a large number of t-included items. Furthermore, for $h = 1, 2, \ldots, |S|$:

- Recall that $\mathscr{K}_h = \{k \in [K] : k_h \leq k \leq k_h + C_\varepsilon\}$ are the classes of most profitable items present in the knapsack at times $t \in [T] : t_h \leq t < t_{h-1}$, since by definition no item is taken from a class $k < k_h$ at those times. Also by definition $\rho(t_h, k_h) \geq 1$, so the largest profit item present in the knapsack at any time $t \in [T] : t_h \leq t < t_{h-1}$ is item $1(k_h)$. Denote its profit by p^h_{max}.
- Define $\bar{\mathscr{K}}_h := \{k \in [K] : k_h + C_\varepsilon < k\}$, i.e. it is the family of the other classes for which an object may be present in the knapsack at time $t \in [T] : t_h \leq t < t_{h-1}$.

Claim 6. *Fix $t \in [T]$, $t_h \leq t < t_{h-1}$. Then, $|I_k \cap \mathscr{F}_t| \leq 1$ for all $k \in [K] \cup \{\infty\}$. Moreover,* $|((\cup_{k\in\bar{\mathscr{K}}_h} I_k) \cap \mathscr{F}_t) \setminus \mathscr{F}_{t_{h-1}}| \leq 1$.

Proof. We show this by induction on t. Fix $t \geq 1$ and suppose that $|I_k \cap \mathscr{F}_t| \leq 1$ for all $k \in [K] \cup \{\infty\}$. By construction, for a class k such that $I_k \cap \mathscr{F}_t = \{i_k\}$, all items $j \in I_k$ with $\bar{x}_{t,j} = 0$ follow i_k in the profit/weight order. Hence, at time $t+1$, the algorithm will not increase $\bar{x}_{t+1,j}$ for any $j \in I_k$ until $\bar{x}_{t+1,i_k}$ is set to 1. We can repeat this argument and conclude $|I_k \cap \mathscr{F}_{t+1}| \leq 1$. Note that this also settles the basic step $t = 0$ and the case $I_k \cap \mathscr{F}_t = \emptyset$, concluding the proof of the first part. A similar argument settles the other statement. □

Claim 7. *Let $h \in [|S|]$, then: $p((\cup_{k\in\bar{\mathscr{K}}_h\setminus\mathscr{L}_t} I_k) \cap \mathscr{F}_t) \leq \epsilon \sum_{\bar{h}=h}^{|S|} p^{\bar{h}}_{\max}$, $\forall t : t_h \leq t < t_{h-1}$.*

Proof. We prove the statement by induction on h. For $h = |S|$, let t be such that $t_{|S|} \leq t < t_{|S|-1}$ and $\bar{t} = t_{|S|} - 1$. We have that $(\cup_{k\in\bar{\mathscr{K}}_{|S|}} I_k) \cap \mathscr{F}_{\bar{t}} = \emptyset$ so $((\cup_{k\in\bar{\mathscr{K}}_{|S|}} I_k) \cap \mathscr{F}_t) \setminus \mathscr{F}_{\bar{t}} = (\cup_{k\in\bar{\mathscr{K}}_{|S|}} I_k) \cap \mathscr{F}_t$. By using Claim 6 we obtain

$$|(\cup_{k\in\bar{\mathscr{K}}_{|S|}\setminus\mathscr{L}_t} I_k) \cap \mathscr{F}_t| \leq |(\cup_{k\in\bar{\mathscr{K}}_{|S|}} I_k) \cap \mathscr{F}_t| \leq 1.$$

The largest profit of an item in $\cup_{k\in\bar{\mathscr{K}}_{|S|}} I_k$ is smaller than $(1+\epsilon)^{-C_\varepsilon} p^{|S|}_{\max} \leq \epsilon p^{|S|}_{\max}$ by the definition of $\bar{\mathscr{K}}_{|S|}$ and recalling $C_\varepsilon = \log_{1+\varepsilon} \frac{1}{\varepsilon}$. The statement follows.

Assume that the statement holds for all h such that $2 \leq h \leq |S|$ and prove it for $h = 1$. Let t such that $t_1 \leq t < t_0 = T+1$ and $\bar{t} = t_1 - 1$. Observe that $\mathscr{L}_t \supseteq \mathscr{L}_{\bar{t}}$ and $(\cup_{k\in\bar{\mathscr{K}}_1} I_k) \cap \mathscr{F}_{\bar{t}} \subseteq (\cup_{k\in\mathscr{K}_2\cup\bar{\mathscr{K}}_2} I_k) \cap \mathscr{F}_{\bar{t}}$ so

$$(\cup_{k\in\bar{\mathscr{K}}_1\setminus\mathscr{L}_t} I_k) \cap \mathscr{F}_{\bar{t}} \subseteq (\cup_{k\in(\mathscr{K}_2\cup\bar{\mathscr{K}}_2)\setminus\mathscr{L}_{\bar{t}}} I_k) \cap \mathscr{F}_{\bar{t}} = (\cup_{k\in\bar{\mathscr{K}}_2\setminus\mathscr{L}_{\bar{t}}} I_k) \cap \mathscr{F}_{\bar{t}}.$$

Thus, we obtain:

$$\begin{array}{rcl} p((\cup_{k\in\bar{\mathcal{K}}_1\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t) & = & p(((\cup_{k\in\bar{\mathcal{K}}_1\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t)\setminus\mathcal{F}_{\bar{t}}) + p((\cup_{k\in\bar{\mathcal{K}}_1\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_{\bar{t}}) \\ & \le & p(((\cup_{k\in\bar{\mathcal{K}}_1\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t)\setminus\mathcal{F}_{\bar{t}}) + p((\cup_{k\in\bar{\mathcal{K}}_2\setminus\mathcal{L}_{\bar{t}}} I_k)\cap\mathcal{F}_{\bar{t}}) \\ & \le & \epsilon p^1_{\max} + \epsilon\sum_{\bar{h}=2}^{|S|} p^{\bar{h}}_{\max}, \end{array}$$

where in the last inequality we used Claim 6 and the inductive hypothesis. □

Proof of Theorem 7. We focus on showing that, $\forall\, t\in[T]$:

$$\sum_{i\in[n]\setminus I_\infty} p_i\lfloor\bar{x}_{t,i}\rfloor \ge \sum_{i\in[n]\setminus I_\infty} p_i\bar{x}_{t,i} - \sum_{i\in([n]\setminus I_\infty)\cap\mathcal{F}_t} p_i \ge (1-\epsilon)\sum_{i\in[n]\setminus I_\infty} p_i\bar{x}_{t,i}. \tag{2.5}$$

The first inequality is trivial and, if $t < t_{|S|}$, so is the second, since in this case $\bar{x}_{t,i} = 0$ for all $i\in[n]\setminus I_\infty$. Otherwise, t is such that $t_h \le t < t_{h-1}$ for some $h\in[|S|]$ with $t_0 = T+1$. Observe that:

$$\begin{array}{rcl} ([n]\setminus I_\infty)\cap\mathcal{F}_t & = & ((\cup_{k\in(\mathcal{K}_h\cup\bar{\mathcal{K}}_h)\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t)\cup((\cup_{k\in(\mathcal{K}_h\cup\bar{\mathcal{K}}_h)\cap\mathcal{L}_t} I_k)\cap\mathcal{F}_t) \\ & = & ((\cup_{k\in\bar{\mathcal{K}}_h\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t)\cup((\cup_{k\in\mathcal{L}_t} I_k)\cap\mathcal{F}_t) \end{array}$$

For $k\in[K]$ denote the profit of $i\in I_k$ with p^k. We have:

$$\begin{array}{rcl} \sum_{i\in([n]\setminus I_\infty)\cap\mathcal{F}_t} p_i\bar{x}_{t,i} & = & p((\cup_{k\in\bar{\mathcal{K}}_h\setminus\mathcal{L}_t} I_k)\cap\mathcal{F}_t) + p((\cup_{k\in\mathcal{L}_t} I_k)\cap\mathcal{F}_t) \\ \text{(By Claim 7 and Claim 6)} & \le & \epsilon\sum_{\bar{h}=h}^{|S|} p^{\bar{h}}_{\max} + \sum_{k\in\mathcal{L}_t} p^k. \end{array} \tag{2.6}$$

If $k = k_{\bar{h}}\in\mathcal{L}_t$ for $\bar{h}\in[|S|]$ then $\sum_{i\in I_k} p_i\bar{x}_{t,i} \ge (\frac{1}{\epsilon}+1)p^k = p^{\bar{h}}_{\max} + \frac{1}{\epsilon}p^k$. Together with $\rho(k_h, t_h)\ge 1\ \forall\, h\in[|S|]$ and the definition of $\mathcal{L}_t$ this gives:

$$\sum_{i\in[n]\setminus I_\infty} p_i\bar{x}_{t,i} \ge \sum_{\bar{h}=h}^{|S|} p^{\bar{h}}_{\max} + \frac{1}{\epsilon}\sum_{k\in\mathcal{L}_t} p^k. \tag{2.7}$$

Put together, (2.6) and (2.7) imply (2.5). Morever, by Claim 6, $|I_\infty\cap\mathcal{F}_t|\le 1$ for all $t\in[T]$ and since we are working with an ϵ-well-behaved instance $p_i \le \frac{\epsilon}{T} = \frac{\epsilon}{T}p^1_{\max}$ so $\sum_{t\in[T]}\sum_{i\in I_\infty\cap\mathcal{F}_t} p_i \le \epsilon p^1_{\max}$. The last fact with (2.5) and Claim 5 gives the statement of the theorem. ■

Theorem 1 now easily follows from Theorems 2, 6, and 7.

Proof of Theorem 1. Since we will need items to be sorted by profit/weight ratio, we can do this once and for all before any guessing is performed. Classical algorithms implement this in $O(n\log n)$. By Theorem 2, we know we can assume that the input instance is ε-well-behaved, and it is enough to find a solution of profit at least $(1-2\varepsilon)$ the profit of a 1-in solution of maximum profit – by Theorem 7, this is exactly vector $\lfloor\bar{x}\rfloor$. In order to produce $\lfloor\bar{x}\rfloor$, as we already sorted items by profit/weight ratio, we only need to solve the LPs associated with each choice of S and ρ, and then run Algorithm 1. The number of choices of S and ρ are $T^{f(\varepsilon)}$, and each LP has $g(\varepsilon)O(nT)$ constraints, for appropriate functions f and g (see the proof of

Theorem 6). Algorithm 1 runs in time $O(\frac{T}{\varepsilon}\log\frac{T}{\varepsilon}+n)$. The overall running time is:

$$O(n\log n + n(n+T+T^{f(\varepsilon)}(f_{LP}(g(\varepsilon)O(nT))+\frac{T}{\varepsilon}\log\frac{T}{\varepsilon}))) = O(nT^{h(\varepsilon)}f_{LP}(n)),$$

where $f_{LP}(m)$is the time required to solve an LP with $O(m)$ variables and constraints, and $h:\mathbb{R}\to\mathbb{N}_{\geq 1}$ is an appropriate function. ■

2.3 Generalizations

Following Theorem 1, one could ask for a PTAS for the general incremental knapsack (IK) problem. This is the modification of IIK (introduced in [11]) where the objective function is $p_\Delta(x) := \sum_{t\in[T]}\Delta_t\cdot p^T x_t$, where $\Delta_t\in\mathbb{Z}_{>0}$ for $t\in[T]$ can be seen as time-dependent discounts. We show here some partial results.

Corollary 8. *There exists a PTAS-preserving reduction from IK to* IIK*, assuming* $\Delta_t\leq\Delta_{t+1}$ *for* $t\in[T-1]$. *Hence, the hypothesis above, IK has a PTAS.*

We start by proving an auxiliary corollary.

Corollary 9. *There exists a strict approximation-preserving reduction from* IK *to* IIK*, assuming that the maximum discount* $\Delta_{\max}:=\|\Delta\|_\infty$ *is bounded by a polynomial*

$$g(T,n,\log\|p\|_\infty,\log\|w\|_\infty).$$

In particular, under the hypothesis above, IK *has a PTAS.*

Proof. Let $\mathscr{I}:=(n,p,w,T,b,\Delta)$ be an IK instance with $\Delta_{\max}\leq g(T,n,\log\|p\|_\infty,\log\|w\|_\infty)$. The corresponding instance $\mathscr{I}':=(n,p,w,T',b')$ of IIK is obtained by setting

$$T':=\sum_{t\in[T]}\Delta_t \quad\text{and}\quad b'_{t'}:=b_t,\ \forall t'\in[T']:\ \delta_t+1\leq t'\leq\delta_t+\Delta_t,$$

where $\delta_t:=\sum_{\bar{t}<t}\Delta_{\bar{t}}$ for $t\in[T]$. We have that $T'\leq T\cdot g(T,n,\log\|p\|_\infty,\log\|w\|_\infty)$ so the size of $\mathscr{I}'$ is polynomial in the size of $\mathscr{I}$.

Given an optimal solution $x^*\in\{0,1\}^{Tn}$ to $\mathscr{I}$, and $x'\in\{0,1\}^{T'n}$ such that $x'_{t'}=x_t$ for all $t\in[T]$ and $\delta_t+1\leq t'\leq\delta_t+\Delta_t$, one has that x' is feasible in $\mathscr{I}'$ so

$$\mathrm{OPT}(\mathscr{I})=p_\Delta(x^*)=\sum_{t\in[T]}\Delta_t\cdot p^T x^*_t=\sum_{t'\in[T']}p^T x'_{t'}\leq\mathrm{OPT}(\mathscr{I}').$$

Let $\hat{x}$ be a α-approximated solution to $\mathscr{I}'$. Define $\bar{x}\in\{0,1\}^{Tn}$ as $\bar{x}_t=\hat{x}_{\delta_t+\Delta_t}$ for $t\in[T]$. Then clearly $\bar{x}_t\leq\bar{x}_{t+1}$ for $t\in[T-1]$. Moreover,

$$w^T\bar{x}_t=w^T\hat{x}_{\delta_t+\Delta_t}\leq b'_{\delta_t+\Delta_t}=b_t,\quad\forall t\in[T].$$

Hence $\bar{x}$ is a feasible solution for $\mathscr{I}$ and

$$p_\Delta(\bar{x}) = \sum_{t\in[T]} \Delta_t \cdot p^T \bar{x}_t \geq \sum_{\bar{t}\in[T']} p^T \hat{x}_{\bar{t}}.$$

Finally, one obtains:

$$\frac{p_\Delta(\bar{x})}{\mathrm{OPT}(\mathscr{I})} \geq \frac{\sum_{\bar{t}\in[T']} p^T \hat{x}_{\bar{t}}}{\mathrm{OPT}(\mathscr{I}')} \geq \alpha. \tag{2.8}$$

□

Proof of Corollary 8. Given an instance $\mathscr{I}$ of IK with monotonically increasing discounts, and letting $p_{\max} := \|p\|_\infty$, we have that the optimal solution of $\mathscr{I}$ is at least $\Delta_{max} \cdot p_{\max}$ since $w_i \leq b_T$, $\forall i \in [n]$, otherwise an element i can be discarded from the consideration. Reduce $\mathscr{I}$ to an instance $\mathscr{I}'$ by setting $C = \frac{\varepsilon \Delta_{\max}}{Tn}$ and $\Delta'_t = \lfloor \frac{\Delta_t}{C} \rfloor$. We get that $\Delta'_{\max} \leq Tn/\varepsilon$ thus satisfying the assumption of Corollary 9 for each fixed $\varepsilon > 0$. Let x^* be an optimal solution to $\mathscr{I}$ and $\bar{x}$ a $(1-\varepsilon)$-approximated solution to $\mathscr{I}'$, one has:

$$\begin{aligned} p_\Delta(\bar{x}) &\geq C \cdot p'_\Delta(\bar{x}) \\ &\geq C \cdot (1-\varepsilon) p'_\Delta(x^*) \\ &\geq (1-\varepsilon)(p_\Delta(x^*) - C\sum_t p^T x^*_t) \\ &\geq (1-\varepsilon)(p_\Delta(x^*) - \varepsilon \Delta_{max} \cdot p_{\max}) \quad \geq (1-2\varepsilon) p_\Delta(x^*). \end{aligned}$$

■

The proof of Corollary 8 only uses the fact that an item of the maximum profit is feasible at a time with the highest discount. Thus its implications are broader.

Of independent interest is the fact that there is a PTAS for the modified version of IIK when each item can be taken multiple times. Unlike Corollary 8, this is not based on a reduction between problems, but on a modification on our algorithm.

Corollary 10. *There is a PTAS for the following modification of* IIK*: in* (2.1)*, replace* $x_t \in \{0,1\}^n$ *with:* $x_t \in \mathbb{Z}^n_{>0}$ *for* $t \in [T]$*; and* $0 \leq x_t \leq d$ *for* $t \in [T]$*, where we let* $d \in (\mathbb{Z}_{>0} \cup \{\infty\})^n$ *be part of the input.*

Proof. We detail the changes to be implemented to the algorithm and omit the analysis, since it closes follows that for IIK. Modify the definition of $P(S,\rho)$ as follows. Fix $h \in [|S|]$, $k \in \mathscr{K}_h$ and $t \in \mathscr{T}_h$. As before, items in the k-th bucket are ordered monotonically increasing according to their weight as $I_k = \{1(k), \ldots, |I_k|(k)\}$. In order to take into account item multiplicities we define $r := r(t,k) = \max\{\bar{r} : \sum_{l=1}^{\bar{r}} d_{l(k)} < \rho(t,k)\}$. Replace the third, fitfth and sixth set of constraints from $P(S,\rho)$ with the following, respectively:

(4') $0 \leq x_t \leq d$;

(5') $x_{t,i(k)} = d_{i(k)}, \ \forall i(k) : i \leq r(t,k); \ x_{t,(r+1)(k)} = \rho(t,k) - \sum_{l=1}^{r} d_{l(k)}$;

(6') $x_{t,i(r+2)} = 0, \ldots, x_{t,i(|I_k|)} = 0$ if $\rho_{t,k} \leq \frac{1}{\varepsilon}$.

For fixed S, ρ, call all items i such that $x_{t,i} = c$ appears in (5′) or in (6′) (t,c)*-fixed.* Note that items that are $(t,0)$-fixed correspond to items that were called t-excluded in IIK. Items that are (t,c)-fixed for some c are called t*-fixed.* Let $\bar{x}$ be the output of the modification of Algorithm 1 given below. Again, vector $\lfloor \bar{x} \rfloor$ gives the required $(1-2\epsilon)$-approximated integer solution. □

Algorithm 2

1: Set $\bar{x}_0 = \mathbf{0}$.

2: For $t = 1, \ldots, T$:
   (a) Set $\bar{x}_t = \bar{x}_{t-1}$.
   (b) For $i \in [n]$, if i is (t,c)-fixed for some c, set $\bar{x}_{t,i} = c$.
   (c) While $W_t - w^T \bar{x}_t > 0$:
      (i) Select the smallest $i \in [n]$ such that i is not t-fixed and $\bar{x}_{t,i} < d_i$.
      (ii) Set $\bar{x}_{t,i} = \bar{x}_{t,i} + \min\{d_i - \bar{x}_{t,i}, \frac{W_t - w^T \bar{x}_t}{w_i}\}$.

3 An improved disjunctive relaxation for the min-knapsack

By using a similar disjunctive technique as in Chapter 2, we investigate polyhedral relaxations for MINKNAP subject to a fixed objective function. In this setting one can approximately compute the optimal value with an FPTAS, and then add the objective function as a constraint, thus obtaining an integrality gap which is arbitrary close to 1. However, one is usually interested in more structured relaxations that can be applied to more general problems.

Subject to a given objective function and a constant $\varepsilon > 0$, Bienstock and McClosky [10] provide a disjunctive relaxation for MINKNAP with the integrality gap bounded by $1+\epsilon$ and size exponential in $C_\varepsilon = \Theta\left(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon})\right)$. The technique is based on grouping items according to the cost, and partitioning the set of feasible integer solutions into families, according to the number of items coming from each cost bucket. It can be shown that each family has a small linear relaxation with the integrality gap close to one.

We improve on their result by constructing an asymptotically smaller formulation, i.e. , reducing the exponent from C_ε to $\sqrt{C_\varepsilon}$, while preserving most of its properties. The reduction is achieved by exploiting the structure of vertices in the above relaxation to merge some of the cost buckets. The following theorem is our main result in this chapter.

Theorem 11. *For each $\varepsilon > 0$ and fixed objective function, there is a disjunctive relaxation for* MINKNAP *with $n^2(1/\varepsilon)^{O(\sqrt{C_\varepsilon})}$ variables and constraints, and integrality gap of at most $1+\varepsilon$.*

In Section 3.1 we recall the disjunctive relaxation from [10] and compare it to our construction, while the proof of Theorem 11 can be found in Section 3.2.

3.1 Overview of the technique

Given costs $c \in \mathbb{R}^n_+$ and profits $p \in \mathbb{R}^n_+$, the standard integer programming formulation for MINKNAP can be stated as

$$\min\{c^T x : \; p^T x \geq 1, \; x \in \{0,1\}^n\}. \tag{3.1}$$

Its natural LP relaxation is obtained by replacing $x \in \{0,1\}^n$ with $x \in [0,1]^n$. Using that the objective function is fixed, the items can be rearranged monotonically decreasing according to their cost, i.e., $1 = c_1 \geq c_2 \geq \cdots \geq c_n$.

Denote with Q the set of feasible solutions to (3.1). It is covered by a union $\cup_{j\in[n]} Q_j$, where

$$Q_j := \{x \in \{0,1\}^n : \ p^T x \geq 1, \ x_i = 0 \text{ for } i < j, \ x_j = 1\} \tag{3.2}$$

Q_j contains all the solutions where the j-th item is taken, and all the items with cost larger then c_j are explicitly set to 0. Thus, one can see that its natural relaxation

$$P_j := \{x \in [0,1]^n : \ p^T x \geq 1, \ x_i = 0 \text{ for } i < j, \ x_j = 1\}$$

has integrality gap bounded by 2. Covexity then gives that $\textsc{conv}(\cup_{j=1}^n P_j)$ is a relaxation of Q with the gap of at most 2 as well. For any constant $\varepsilon > 0$ and each j, Bienstock and McClosky [10] further strengthen P_j reducing the gap to $1+\varepsilon$. It is in fact enough to provide such a relaxation for Q_1, as the others would follow by redefining $n' = n - j + 1$, items $j, \dots, n$ as $1, \dots, n'$, and scaling costs so that $c'_j = 1$. Their relaxation is as follows.

1. Partition set $\{2, \dots, n\}$ into the following *buckets*[1]:

   $$S_k := \{i \in \{2, \dots, n\} : \ (1+\varepsilon)^{-k+1} \geq c_i > (1+\varepsilon)^{-k}\}, \quad \forall k \in [C_\varepsilon]$$

   and $S_\infty = \{i \in \{2, \dots, n\} : c_i \leq (1+\varepsilon)^{-C_\varepsilon}\}$. Note that $(1+\varepsilon)^{-C_\varepsilon} \leq \varepsilon$.

2. For $\rho \in \{0, 1, \dots, 1/\varepsilon\}^{C_\varepsilon}$ let Q_ρ be the set of all solutions in Q_1 where the number of items taken from S_k is exactly ρ_k if $\rho_k < 1/\varepsilon$, and it is at least $1/\varepsilon$ otherwise. Again $Q_1 = \cup_\rho Q_\rho$ and each Q_ρ can be relaxed to

   $$P_\rho := \left\{ x \in P_1 : \ \sum_{i\in S_k} x_i = \rho_k, \ \forall k \ : \ \rho_k < \frac{1}{\varepsilon} \text{ and } \sum_{i \in S_k} x_i \geq \rho_k, \ \forall k \ : \ \rho_k = \frac{1}{\varepsilon} \right\}.$$

The result from [10] follows from showing that each P_ρ has integrality gap $1+\varepsilon$. Then $\textsc{conv}(\cup_\rho P_\rho)$ is a relaxation of Q_1 with integrality gap $1+\varepsilon$.

Comparison to our construction. In Section 3.2, we also partition Q into Q_j's and then group items according to the cost. The main difference from the above is that we form buckets of increasing length[2]. The approximation ratio is preserved by exploiting the following observation: a vertex x^* of a polytope like P_ρ has at most two fractional components, and they lie in the same bucket (see Lemma 12).

Say those components correspond to items r and q, with $c_r \geq c_q$. The rounding of x^* is done in a standard way, and the cost variation is bounded by a function of c_r and c_q. If there is a

[1] Define $C_\varepsilon := \log_{1+\varepsilon}(1/\varepsilon)$. For simplicity, C_ε and $1/\varepsilon$ are considered to be integers.

[2] The length of a bucket stands for the cost ratio between the smallest and the largest item in the bucket.

non-empty bucket whose items have cost bigger than c_r, then this bucket contributes to the objective function at least as much as c_r. Hence, if there are many of those buckets, the ratio between c_r and c_q can be reasonably large and still the rounding induces a small change with respect to the total cost. We can then take (non-empty) buckets of increasing length, and still guarantee the integrality gap of $1+\varepsilon$ (see Lemma 15). Therefore, Q_1 can be partitioned into smaller number of sets, leading to a relaxation of smaller size (see Lemma 17).

3.2 The disjunctive relaxation

Because of the discussion from the previous section, in order to prove Theorem 11, we are left to provide a disjunctive relaxation for Q_1. We will also assume $\varepsilon \le 1/256$. Let $\mathscr{S} = \{S_1, \ldots, S_K, S_\infty\}$ be a family of pairwise disjoint subsets of $\{2, \ldots, n\}$, and $\rho \in \{1, \ldots, 1/\varepsilon\}^K$. Define:

$$
\begin{array}{llll}
P(\mathscr{S},\rho) := \{\, x \in \mathbb{R}^n : & x_1 = 1, & p^T x \ge 1, & \\
& \sum_{i \in S_k} x_i = \rho_k & \forall k \in [K] \mid \rho_k < 1/\varepsilon, & \\
& \sum_{i \in S_k} x_i \ge \rho_k & \forall k \in [K] \mid \rho_k = 1/\varepsilon, & \\
& x_i = 0 & \forall i \in \{2, \ldots, n\} \setminus \cup_{k \in [K] \cup \{\infty\}} S_k, & \\
& 0 \le x_i \le 1 & \forall i \in \cup_{k \in [K] \cup \{\infty\}} S_k & \}.
\end{array}
$$

Lemma 12. *An extreme point x^* of $P := P(\mathscr{S},\rho)$ has at most two fractional components, and if they are two, they lie in the same bucket S_h, where $h \in [K]$.*

Proof. Let x^* be an extreme point of P, and consider a set $\mathscr{C}$ of n linearly independent constraints of P at which x^* is active. Let $\mathscr{C}' \subseteq \mathscr{C}$: basic linear algebra implies that $\mathscr{C}'$ is also linearly independent, hence the number of variables that belong to the support of $\mathscr{C}'$ are at least $|\mathscr{C}'|$. By Hall's Theorem, we can then find an injective map assigning to each constraint from $\mathscr{C}$ a variables from its support. We say that the constraint is "charged" to the variable. Since $x \in \mathbb{R}^n$, the map is also surjective. Now let $0 < x_r^* < 1$. Then $r \ne 1$, i.e. $r \in S_h$ for some $h \in [k] \cup \{\infty\}$, and x_r charges either $\sum_{i \in S_k} x_i \ge \rho_k$ (or $\sum_{i \in S_k} x_i = \rho_k$), or to $p^T x \ge 1$. This implies that there are at most two fractional variables per bucket, and one if $h = \infty$. Now suppose x_r^* does not charge $p^T x \ge 1$: then, since the constraint it charges is tight at x^*, there exists $q \in S_h$, $q \ne r$, such that $x_r^* + x_q^* = 1$. In particular, x_q^* is fractional, and it must charge $p^T x \ge 1$. Hence, we showed that each time a fractional variable does not charge $p^T x \ge 1$, there is exactly one more fractional variable from the same bucket, and it charges $p^T x \ge 1$. The thesis then follows from the fact that at most one variable can charge $p^T x \ge 1$. □

The lemma above gives a new insight on the extreme points of $P(\mathscr{S},\rho)$ and it is crucial to control the decrease in the objective function when rounding. Let Γ be the set of vectors $\tau \in \mathbb{N}_0^{|\tau|}$ with the following properties:

1. $|\tau| \le 2\sqrt{C_\varepsilon}$;

2. $\tau_k + k \le \tau_{k+1}$ for $k \in [|\tau|]$;

3. $\tau_{|\tau|} \le C_\varepsilon - 1$.

and for $\tau \in \Gamma$ define $K = |\tau|$ and $\mathscr{S}(\tau)$ as follows:

(i) For $k \in [K]$, set $S_k := \{i \in \{2,\dots,n\} : (1+\varepsilon)^{-\tau_k} \ge c_i > (1+\varepsilon)^{-\min\{\tau_k+k, C_\varepsilon\}}\}$.

(ii) Set $S_\infty := \{i \in \{2,\dots,n\} : c_i < \min_{l \in S_{|\tau|}} c_l$ and $c_i \le (1+\varepsilon)^{-C_\varepsilon}\}$.

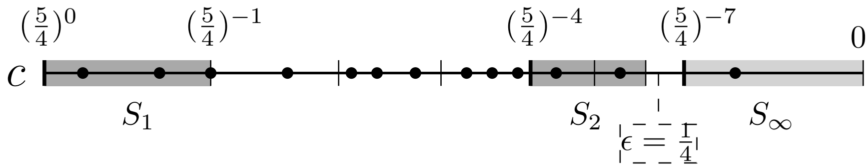

Figure 3.1 – An example of the bucketing $\mathscr{S}(\tau)$ for $\tau = (0,4)$ and $\varepsilon = 1/4$.

We start with some definitions and auxiliary lemmas.

Definition 13. *Given $\varepsilon > 0$ and $\mathscr{S}$ as above, define $c_{\min,k} := \min_{i \in S_k} c_i$ and $c_{\max,k} := \max_{i \in S_k} c_i$ for $k \in [K]$. We say that $\mathscr{S}$ is (ε, c)-ordered if:*

(a) $c_{\min,k} \ge c_{\max,k+1}$ *for $k \in [K-1]$;*

(b) $\min\{c_{\min,K},\ \varepsilon\} \ge \max_{i \in S_\infty} c_i$.

Lemma 14. *Let $\mathscr{S}$ be an (ε, c)-ordered partition and $\rho \in [1/\varepsilon]^K$. An extreme point x^* of P can be rounded to an integral vector $\bar{x}$ with cost $c(\bar{x}) \le (1+2\varepsilon)c(x^*)$ if the following condition holds. Given a fractional point of x^* in bucket $h \in [K]$ one has*

$$\frac{c_{\max,h}}{c_{\min,h}} \le (1+\varepsilon)^{d_h} \text{ for } h \in [K], \text{ with } d_h \le \min\{h, \left\lceil 2\sqrt{C_\varepsilon}\right\rceil\}.$$

Proof. Following Lemma 12, we distinguish two cases.

Case 1: x^* has exactly one fractional component, say r. Then $\bar{x}$ can be obtained by setting $\bar{x}_r = 1$ and $\bar{x}_i = x^*_i$ for $i \in [n] \setminus \{r\}$. x^* is clearly feasible. Moreover, $c(\bar{x}) - c(x^*) \le c_r$. If $h = \infty$ then by (b) one has $c_r \le \varepsilon \cdot c_1 \le \varepsilon c(x^*)$. Otherwise, $h \in [K]$ and $\sum_{i \in S_h} x^*_i$ is fractional. Hence $\rho_h = 1/\varepsilon$, otherwise x^* would not be feasible. Then one gets

$$\frac{c_r}{c(x^*)} \le \frac{c_{\max,h}}{c_j + c_{\min,h}/\varepsilon} \le \frac{(1+\varepsilon)^{d_h}}{1/\varepsilon} \le \varepsilon \frac{1-(\varepsilon d_h)^{d_h+1}}{1-\varepsilon d_h} \le 2\varepsilon,$$

for ε small enough ($\le 1/256$) using $d_h \le \left\lceil 2\sqrt{C_\varepsilon}\right\rceil$ and

$$(1+\varepsilon)^{d_h} = \sum_{l=0}^{d_h} \binom{d_h}{l} \varepsilon^l \le \sum_{l=0}^{d_h} (\varepsilon d_h)^l = \frac{1-(\varepsilon d_h)^{d_h+1}}{1-\varepsilon d_h}.$$

Case 2: x^* has exactly two fractional entries, say r and q. From Lemma 12 and its proof, we know they are exactly in the same bucket S_h with $h \in [K]$, that $\sum_{i \in S_h} x_i^* = \rho_h \in \mathbb{Z}$, and $x_r^* + x_q^* = 1$. Assume wlog $w_r \geq w_q$. Setting $\bar{x}_r = 1$, $\bar{x}_q = 0$ and $\bar{x}_i = x_i^*$ for $i \in [n] \setminus \{r, q\}$ gives an integral feasible vector $\bar{x}$ with the approximation guarantee:

$$\begin{aligned}
\frac{c(\bar{x})-c(x^*)}{c(x^*)} &\leq \frac{c_r - c_q}{c(x^*)} \overset{(*)}{\leq} \frac{c_{\max,h}-c_{\min,h}}{c_1+(d_h-1)c_{\max,h}} \overset{(\bullet)}{\leq} \frac{c_{\max,h}-c_{\min,h}}{d_h c_{\max,h}} \overset{(\circ)}{\leq} \frac{(1+\varepsilon)^{d_h}-1}{d_h(1+\varepsilon)^{d_h}} \\
&\leq \frac{\frac{1-(\varepsilon d_h)^{d_h+1}}{1-\varepsilon d_h}-1}{d_h} \leq \frac{\varepsilon}{1-\varepsilon d_h} \leq 2\varepsilon,
\end{aligned}$$

where $(*)$ follows from (a) and the fact that by construction $x_1^* = 1$, and $x_i^* = 1$ for at least one $i \in S_k$, for all $k < h$, $(\bullet)$ from $c_{max,h} \leq c_1$, $(\circ)$ from the definition of d_h, and in the last inequality we again assumed $\varepsilon \leq 1/256$ and used $d_h \leq \lceil 2\sqrt{C_\varepsilon} \rceil$. □

Incidentally, observe that the relaxation defined in points 1.-2. from Section 3.1 is induced by an (ε, c)-ordered family, by disregarding sets S_i of the partition with $\rho_i = 0$. It also trivially satisfies the condition of Lemma 14 since $d_k = 1$, $\forall k \in [K]$, and $K \leq C_\varepsilon$. Recall that Γ is the set of vectors $\tau \in \mathbb{N}_0^{|\tau|}$ with properties 1.-3. defined above, and that for $\tau \in \Gamma$ we have the family $\mathscr{S}(\tau)$ consisting of sets S_k, $k \in [K]$ and S_∞ with the specified structure.

Lemma 15. *Let $\tau \in \Gamma$ and $\mathscr{S} = \mathscr{S}(\tau)$. An extreme point of $P := P(\mathscr{S}, \rho)$ can be rounded to an integral vector $\bar{x}$ with cost $c(\bar{x}) \leq (1+2\varepsilon)c(x^*)$.*

Proof. It is enough to show that $\mathscr{S}$ satisfies the conditions from Lemma 14. One immediately checks that $\mathscr{S}$ is an (ε, c)-ordered partition. As $|\tau| \leq 2\sqrt{C_\varepsilon}$, we only need to prove that $d_h \leq h$, for each $h \in [K]$. This follows from the fact that $c_{\max,h} \leq (1+\varepsilon)^{-\tau_h}$ and $c_{\min,h} \geq (1+\varepsilon)^{-(\tau_h+h)}$, hence $d_h \leq -\tau_h + (\tau_h + h) = h$. □

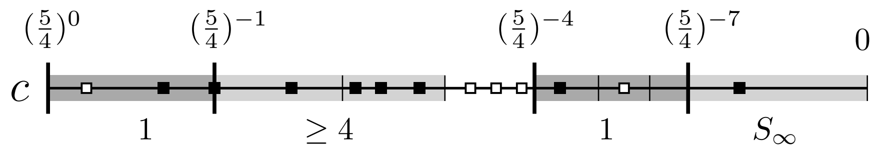

Figure 3.2 – An example of a knapsack solution $\hat{x}$ induced by the items marked with full-squares. The construction in Lemma 16 covers $\hat{x}$ with $\tau = (0,1,4)$ and $\rho = (1,4,1)$ for $\varepsilon = 1/4$.

Lemma 16. *For any solution $\hat{x} \in Q_1$ there exist $\tau \in \Gamma$ and $\rho \in [1/\varepsilon]^{|\tau|}$ such that $\hat{x} \in P(\mathscr{S}(\tau), \rho)$.*

Proof. We iteratively construct τ as follows:

1) $\tau_1 = \min\{\hat{k} \in [C_\varepsilon] : \exists \hat{\imath} > 1 \text{ s.th. } \hat{x}_{\hat{\imath}} = 1, (1+\varepsilon)^{-\hat{k}} \geq c_{\hat{\imath}} > (1+\varepsilon)^{-\hat{k}-1}\}$;

2) Given τ_k, as long as the set

$$R_{k+1} := \{\hat{k} \in [C_\varepsilon] : \hat{k} \geq \tau_k + k, \exists \hat{\imath} > 1 \text{ s.th. } \hat{x}_{\hat{\imath}} = 1, (1+\varepsilon)^{-\hat{k}} \geq c_{\hat{\imath}} > (1+\varepsilon)^{-\hat{k}-1}\}$$

is non-empty, define $\tau_{k+1} = \min\{\hat{k} \in R_{k+1}\}$.

First observe that step 2) is repeated at most $\lceil 2\sqrt{C_\varepsilon}\rceil - 1$ times, since $\sum_{k=1}^{\lceil 2\sqrt{C_\varepsilon}\rceil} k \geq C_\varepsilon$. Hence $|\tau| \leq \lceil 2\sqrt{C_\varepsilon}\rceil$. One easily concludes then that $\tau \in \Gamma$.

Now choose ρ such that $\rho_k = \min\{|\mathrm{supp}(\hat{x}) \cap S_k|, 1/\varepsilon\}$ for $k \in [|\tau|]$. Let us verify that $\hat{x} \in P(\mathscr{S}(\tau))$. Let $i \in \{2, \dots, n\}$ such that $\hat{x}_i = 1$. All we need to show is that, if $c_i > (1+\varepsilon)^{-C_\varepsilon}$, then $i \in S_k$ for some $k \in [|\tau|]$, since the feasibility of $\hat{x}$ would then follow by definition of ρ. Let $\hat{k}$ be the maximum k such that $(1+\varepsilon)^{-\tau_k} \geq c_i$. If $c_i > (1+\varepsilon)^{-(\tau_k + k)}$, then $i \in S_k$; else, the maximality of k is contradicted. □

Lemma 17. *The number of possible pairs $(\mathscr{S}(\tau), \rho)$ with $\tau \in \Gamma$ and $\rho \in [1/\varepsilon]^{|\tau|}$ are $(1/\varepsilon)^{O(\sqrt{C_\varepsilon})}$.*

Proof. $|\Gamma| = C_\varepsilon^{O(\sqrt{C_\varepsilon})}$, since $\tau_k \leq C_\varepsilon$ for $k \in [|\tau|]$ and $|\tau| \leq \lceil 2\sqrt{C_\varepsilon}\rceil$ by construction. Having that $\rho \in [1/\varepsilon]^{|\tau|}$ we get the bound:

$$C_\varepsilon^{O(\sqrt{C_\varepsilon})} \cdot \left(\frac{1}{\varepsilon}\right)^{\lceil 2\sqrt{C_\varepsilon}\rceil} \leq \left(2\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon}\right)^{O(\sqrt{C_\varepsilon})} \cdot \left(\frac{1}{\varepsilon}\right)^{\lceil 2\sqrt{C_\varepsilon}\rceil} = \left(\frac{1}{\varepsilon}\right)^{O(\sqrt{C_\varepsilon})}$$

□

Let $\hat{P}_1 := \mathrm{CONV}\left(\bigcup_{\tau\in\Gamma} \bigcup_{\rho\in[1/\varepsilon]^{|\tau|}} P(\mathscr{S}(\tau,\rho))\right)$, we can now prove Theorem 11.

Proof of Theorem 11. Let $\hat{x} \in \hat{P}_1 \cap \{0,1\}^n$. Hence $\hat{x} \in P(\mathscr{S}(\tau), \rho)$ for some $\tau \in \Gamma$ and $\rho \in [1/\varepsilon]^{|\tau|}$. Observe that constraints explicitly defining Q_1 in (3.2) are also valid for $P(\mathscr{S}(\tau), \rho)$, so $\hat{x} \in Q_1$. Conversely, if $\hat{x} \in Q_1$, $\hat{x} \in \hat{P}_1$ by Lemma 16. Hence $\hat{P}_1$ is indeed a relaxation for Q_1. Since each $P(\mathscr{S}(\tau), \rho)$ has $O(n)$ variables and constraint, $\hat{P}_1$ can be described with a system of linear inequalities of size $n(1/\varepsilon)^{O\sqrt{C_\varepsilon}}$ by Lemma 17. The thesis then follows from the fact that $Q = \cup_{j\in[n]} Q_j$ and Lemma 15.

4 On bounded pitch inequalities for the min-knapsack

In this chapter, we study structural properties and separability of bounded pitch inequalities for MINKNAP, and the strength of linear relaxations for MINKNAP when they are added. Let $\mathscr{F}$ be the set given by pitch-1, pitch-2, and inequalities from the linear relaxation of (3.1). We first show that, for any arbitrarily small precision, we can solve in polynomial time the *weak separation problem* for the set $\mathscr{F}$. Even better, our algorithm either certifies that the given point x^* violates an inequality from $\mathscr{F}$, or outputs a point that satisfies all inequalities from $\mathscr{F}$ and whose objective function value is arbitrarily close to that of x^*. We define such an algorithm as a $(1+\epsilon)$*-oracle* in Section 4.1; see Section 4.2 for the construction. A major step of our procedure is showing that non-redundant pitch-2 inequalities have a simple structure.

It is then a natural question whether bounded pitch inequalities can help to reduce the integrality gap below 2. We show that, when $p = c$, if we add to the linear relaxation of (3.1) pitch-1 and pitch-2 inequalities, the integrality gap is bounded by $3/2$; see Section 4.3.1. However, this is false in general. Indeed, we also prove that KC plus bounded pitch inequalities do not improve upon the integrality gap of 2; see Section 4.3.4. Moreover, bounded pitch alone can be much weaker than KC: we show that, for each fixed k, the integrality gap may be unbounded even if all pitch-k inequalities are added. Using the relation between bounded pitch and Chvátal-Gomory (CG) closures established in [13], this implies that, for each fixed t, the integrality gap of the t-th CG closure can be unbounded; see Section 4.3.2. For an alternative proof that having all KC inequalities bounds the integrality gap to 2 see Section 4.3.3.

4.1 Basics

A MINKNAP instance is a binary optimization problem of the form (3.1), where $p, c \in \mathbb{Q}^n$ and we assume $0 \le p_1 \le p_2 \le \cdots \le p_n \le 1$, $0 < c_i \le 1$, $\forall i \in [n]$. We will often deal with its *natural linear relaxation*

$$\min c^T x \quad \text{s.t.} \quad p^T x \ge 1,\ x \in [0,1]^n. \tag{4.1}$$

The NP-Hardness of MINKNAP immediately follows from the fact that MAXKNAP is NP-Hard [31], and that a MAXKNAP instance

$$\max v^T x \quad \text{s.t.} \quad w^T x \le 1,\ x \in \{0,1\}^n. \tag{4.2}$$

can be reduced into a MINKNAP instance (3.1) as follows: each $x \in \{0,1\}^n$ is mapped via $\pi : \mathbb{R}^n \to \mathbb{R}^n$ with $\pi(x) = \mathbf{1} - x$; $p_i = \frac{w_i}{\sum_{i=1}^n w_i - 1}$ and $c_i = v_i$ for $i \in [n]$. Note that the reduction is not approximation-preserving.

We say that an inequality $w^T x \ge \beta$ with $w \ge 0$ is *dominated* by a set of inequalities $\mathscr{F}$ if $w'^T x \ge \beta'$ can be written as a conic combination of inequalities in $\mathscr{F}$ for some $\beta' \ge \beta$ and $w' \ge w$. $w^T x \ge \beta$ is *undominated* if any set of valid inequalities dominating $w^T x \ge \beta$ contains a positive multiple of it.

Consider a family $\mathscr{F}$ of inequalities valid for (3.1). We refer to [26] for the definition of *weak separation oracle*, which is not used in this chapter. We say that $\mathscr{F}$ admits a $(1+\epsilon)$*-oracle* if, for each fixed $\epsilon > 0$, there exists an algorithm that takes as input a point $\bar{x}$ and, in time polynomial in n, either outputs an inequality from $\mathscr{F}$ that is violated by $\bar{x}$, or outputs a point $\bar{y}$, $\bar{x} \le \bar{y} \le (1+\epsilon)\bar{x}$ that satisfies all inequalities in $\mathscr{F}$. In particular, if $\mathscr{F}$ contains the linear relaxation of (3.1), $0 \le \bar{y} \le 1$.

Let $\sum_{i\in T} w_i x_i \ge \beta$ be a valid inequality for (3.1), with $w_i > 0$ for all $i \in T$. Its *pitch* is the minimum k such that, for each $I \subseteq T$ with $|I| = k$, we have $\sum_{i\in I} w_i \ge \beta$. Undominated pitch-1 inequalities are of the form $\sum_{i\in T} x_i \ge 1$. Note that the map from MAXKNAP to MINKNAP instances defined above gives a bijection between *minimal cover inequalities*

$$\sum_{i\in I} x_i \le |I| - 1$$

for MAXKNAP and undominated pitch-1 inequalities for the corresponding MINKNAP instance. Since, given a MAXKNAP instance, it is NP-Hard to separate minimal cover inequalities [33], we conclude the following.

Theorem 18. *It is NP-Hard to decide whether a given point satisfies all valid pitch-1 inequalities for a given* MINKNAP *instance.*

Given a set $S \subseteq [n]$, such that $\beta := 1 - \sum_{i\in S} p_i > 0$, the *Knapsack cover inequality* associated to S is given by

$$\sum_{i\in[n]\setminus S} \min\{p_i, \beta\} x_i \ge \beta \tag{4.3}$$

and it is valid for (3.1).

For a set $S \subseteq [n]$, we denote by χ^S its characteristic vector. An ϵ*-approximate solution* for a minimization integer programming problem is a solution $\bar{x}$ that is feasible, and whose value is at most $(1+\epsilon)$ times the value of the optimal solution. An algorithm is called a *polynomial* *time approximation scheme (PTAS)* if for each $\epsilon > 0$ and any instance of the given problem it returns an ϵ-approximate solution in time polynomial in the size of the input. If in addition the running time is polynomial in $1/\epsilon$, then the algorithm is a *fully polynomial time approximation scheme (FPTAS).*

Given a rational polyhedron $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ with $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$, the first *Chvátal-Gomory (CG) closure* [16] of P is defined as follows:

$$P^{(1)} = \{x \in \mathbb{R}^n : \lceil \lambda^\top A \rceil x \geq \lceil \lambda^\top b \rceil, \ \forall \lambda \in \mathbb{R}^m\}.$$

Equivalently, one can consider all $\lambda \in [0,1]^m$ such that $\lambda^\top A \in \mathbb{Z}^n$. For $t \in \mathbb{Z}_{\geq 2}$, the t-th CG closure of P is recursively defined as $P^{(t)} = (P^{(t-1)})^{(1)}$. The CG closure is an important tool for solving integer programs, see again [16].

4.2 A weak separation oracle for pitch-1 and pitch-2 inequalities

In this section, we show the following:

Theorem 19. *Given a* MINKNAP *instance* (3.1)*, there exists a* $(1+\epsilon)$*-oracle for the set* $\mathcal{F}$ *containing: all pitch-1 inequalities, all pitch-2 inequalities and all inequalities from the natural linear relaxation of* (3.1).

We start with a characterization of inequalities of interest for Theorem 19.

Lemma 20. *Let K be the set of feasible solutions of a* MINKNAP *instance* (3.1)*. All pitch-*2 *inequalities valid for K are implied by the set composed of:*

i) Non-negativity constraints $x_i \geq 0$ for $i \in [n]$;

*ii) All valid pitch-*1 *inequalities;*

iii) All inequalities of the form

$$\sum_{i\in I_1} x_i + 2\sum_{i\in I_2} x_i \geq 2 \tag{4.4}$$

where $I \subseteq [n]$, $|I| \geq 2$, $\beta(I) := 1 - \sum_{i\in[n]\setminus I} p_i$, $I_1 := \{i \in I : p_i < \beta(I)\} \neq \emptyset$ and $I_2 := I \setminus I_1$.

*The inequalities in iii) are pitch-*2 *and valid.*

Proofs of Lemma 22 and Theorem 19 are given in Section 4.2.1 and Section 4.2.2, respectively.

4.2.1 Restricting the set of valid pitch-2 inequalities

We will build on two auxiliary statements in order to prove Lemma 20.

Claim 8. *If $w^T x \geq \beta$ and $u^T x \geq \beta$ are distinct inequalities valid for and $u \geq w$, then the latter inequality is dominated by the former.*

Proof. $u^T x \geq \beta$ can be obtained summing nonnegative multiples of $w^T x \geq \beta$ and $x_i \geq 0$ for $i \in [n]$, which are all valid inequalities. □

Claim 9. *Let*

$$\sum_{i \in T_1} x_i + 2 \sum_{i \in T_2} x_i \geq 2 \tag{4.5}$$

be a valid inequality for MINKNAP, *with $T_1 \cap T_2 = \emptyset$ and $T_1, T_2 \subseteq [n]$. Then,* (4.5) *is dominated by the inequality in iii) with $I = T_1 \cup T_2$.*

Proof. One readily verifies that Inequality (4.4) with I as above is valid. Suppose now that $i \in T_1 \setminus I_1$. Then the integer solution that takes all elements in $([n] \setminus I) \cup \{i\}$ is feasible for MINKNAP, but it does not satisfy (4.5), a contradiction. Hence $T_1 \subseteq I_1$. Since $T_2 = I \setminus T_i \supseteq I \setminus I_1 = I_2$, (4.4) dominates (4.5) componentwise, and the thesis follows by Claim 8. □

Proof of Lemma 20. The fact that an inequality of the form (4.4) is pitch-2 and valid is immediate. Because of Claim 9, it is enough to show the thesis with (4.4) replaced by (4.5). Consider a pitch-2 inequality valid for K:

$$\sum_{i \in T} w_i x_i \geq 1, \tag{4.6}$$

where $T \subseteq [n]$ is the support of the inequality, $w \in \mathbb{R}_+^{|T|}$. Without loss of generality one can assume that $T = [h]$ for some $h \leq n$ and $w_1 \leq w_2 \leq \cdots \leq w_h$. Since (4.6) is pitch-2 we have that $w_1 + w_i \geq 1$ for all $i \in [h] \setminus \{1\}$. We can also assume $w_h \leq 1$, since otherwise $\sum_{i \in [h-1]} w_i x_i + x_h \geq 1$ is valid and dominates (4.6) by Claim 8.

Let $j \in [h]$ be the maximum index such that $w_j < 1$. Note that such j exists, since, if $w_1 \geq 1$, then (4.6) is a pitch-1 inequality. If $1 - w_1 \leq 1/2$, then, by Claim 8, (4.6) is dominated by the valid pitch-2 inequality

$$\sum_{i \in [j]} x_i + 2 \sum_{i = j+1}^{h} x_i \geq 2, \tag{4.7}$$

which again is of the type (4.5). Hence $1 - w_1 > 1/2$ and again via Claim 8, (4.6) is dominated by

$$w_1 x_1 + \sum_{i=2}^{j} (1-w_1)x_i + \sum_{i=j+1}^{h} x_i \geq 1, \tag{4.8}$$

since $w_i + w_1 \geq 1$ for all $i \neq 1$, so one has $w_i \geq 1 - w_1 > 1/2$. Thus, we can assume that (4.6) has the form (4.8). Note that inequality

$$\sum_{i=2}^{h} x_i \geq 1 \tag{4.9}$$

is a valid pitch-1 inequality, since we observed $w_1 < 1$. Therefore, (4.6) is implied by (4.7) and (4.9), taken with the coefficients w_1 and $1-2w_1$ respectively. Recalling that (4.7) is a valid pitch-2 inequality of the form (4.5) concludes the proof. ■

4.2.2 An oracle

We will prove Theorem 19 in a sequence of intermediate steps. Our argument extends the weak separation of KC inequalities in [15].

Let $\bar{x}$ be the point we want to separate. Note that it suffices to show how to separate over inequalities i)-ii)-iii) from Lemma 20. Separating over i) is trivial. We first show how to separate over iii).

Claim 10. *For $\alpha \in]0,1]$, let z^α be the optimal solution to the following IP P_α, and $v(z^\alpha)$ its value:*

$$\min \sum_{i\in[n]:\, p_i<\alpha} \bar{x}_i z_i + 2 \sum_{i\in[n]:\, p_i\geq\alpha} \bar{x}_i z_i \quad s.t. \sum_{i\in[n]} p_i(1-z_i) \leq 1-\alpha,\ z \in \{0,1\}^n. \tag{4.10}$$

If $v(z^\alpha) < 2$, then $\bar{x}$ violates Inequality (4.4) *with $I := \{i \in [n] : z_i^\alpha = 1\}$, otherwise $\bar{x}$ does not violate any Inequality* (4.4) *with $\beta(I) = \alpha$.*

Proof. Fix a feasible solution $\bar{z}$ to (4.10), and let $I := \{i \in [n] : \bar{z}_i = 1\}$. Then:

$$\beta := \beta(I) = 1 - \sum_{i\in[n]\setminus I} p_i = 1 - \sum_{i\in[n]} p_i(1-\bar{z}_i) \geq \alpha.$$

Hence:

$$\begin{aligned}\textstyle\sum_{i\in I:\, p_i<\beta} \bar{x}_i + 2\sum_{i\in I:\, p_i\geq\beta} \bar{x}_i &= \textstyle\sum_{i\in[n]:\, p_i<\beta} \bar{x}_i\bar{z}_i + 2\sum_{i\in[n]:\, p_i\geq\beta} \bar{x}_i \\ &\leq \textstyle\sum_{i\in[n]:\, p_i<\alpha} \bar{x}_i\bar{z}_i + 2\sum_{i\in[n]:\, p_i\geq\alpha} \bar{x}_i\bar{z}_i = v(\bar{z}),\end{aligned}$$

where the central inequality holds at equality if $\alpha = \beta$. Hence, if $v(z^\alpha) < 2$, the inequality with $I := \{i \in [n] : z_i^\alpha = 1\}$ from (4.10) is violated by $\bar{x}$. Else, all inequalities from (4.10) with $\beta(I) = \alpha$ are satisfied.

□

Note that P_α is a MINKNAP instance, hence we can use the appropriate FPTAS to find, for each $\epsilon > 0$, an ϵ-approximate solution for it.

Since all data are rationals, we can assume there exists $q \in \mathbb{N}_0$ such that, for each $i \in [n]$, $p_i = r_i / q$ for some $r_i \in \mathbb{N}_0$.

Claim 11. *Let $r \in \{r_i + 1 : i \in [n]\}$ and, for $\alpha = r/q$, let $\bar{z}^\alpha$ be the solution output by the FPTAS for problem P_α and $\bar{v}^\alpha$ its objective function value. If $\bar{v}^\alpha < 2$ for some α, then $\bar{x}$ violates Inequality* (4.4) *with $I = \{i \in [n] : \bar{z}^\alpha = 1\}$. Else, $(1+\epsilon)\bar{x}$ satisfies all inequalities in Lemma 20.iii).*

Proof. Let $r = r_i + 1$ for some $i \in [n]$. If $\bar{v}^\alpha < 2$ then by Claim 10 $\bar{x}$ violates the corresponding inequality (4.4). Otherwise, $v^\alpha \geq 2/(1+\epsilon)$, and $(1+\epsilon)\bar{x}$ is feasible for any pitch-2 Inequality (4.4) induced by I with $\beta(I) = \alpha$.

Now let $I^* \subseteq [n]$ with $\beta(I^*) = \frac{r^*}{q} < 1$. There exists $i^* \in [n]$ such that $r_{i^*} < r^* \leq r_{i^*+1} \leq q$ (with $r_{n+1} = q$). Let $\alpha := \frac{r_{i^*}+1}{q} \leq \alpha^* := \beta(I^*)$. The set of feasible solutions of P_α contains that of P_{α^*}, and $\{i \in [n] : p_i < \alpha\} = \{i \in [n] : p_i < \alpha^*\}$. Hence, $v^{\alpha^*} \geq v^\alpha$ and consequently $v^{\alpha^*} < 2$ implies $v^\alpha < 2$. Thus, for separating all inequalities in Lemma 20.iii), it suffices to check (4.10) for all $\alpha = \frac{r}{q}$ as in the statement of the claim.

□

The following claim follows in a similar fashion to the previous one by observing that, for $\beta(I^*) = \frac{1}{q}$, (4.10) separates over undominated pitch-1 inequalities.

Claim 12. *Let $\alpha = 1/q$, and $\bar{z}^\alpha$ be the solution output by the FPTAS for problem P_α, and $\bar{v}^\alpha$ its objective function value. If $\bar{v}^\alpha < 2$, then $\bar{x}$ violates the pitch-1 inequality with support $I = \{i \in [n] : \bar{z}^\alpha = 1\}$. Else, $(1+\epsilon)\bar{x}$ satisfies all valid pitch-1 inequalities.*

Next claim shows how to round a point in the unit cube that almost satisfies all pitch-1 and pitch-2 inequalities, to one that satisfies them and is still contained in the unit cube.

Claim 13. *Let $\bar{x} \in [0,1]^n$ be such that $(1+\epsilon)\bar{x}$ satisfies all inequalities from Lemma 20, and define $\bar{y} \in \mathbb{R}^n$ as follows: $\bar{y}_i = \min\{1, \frac{1+\epsilon}{1-\epsilon}\bar{x}_i\}$ for $i \in [n]$. Then $\bar{y} \in [0,1]^n$ and $\bar{y}$ satisfies all inequalities from Lemma 20.*

Proof. Clearly $\bar{y} \in [0,1]^n$. Let $J = \{i \in [n] : \bar{y}_i = 1\}$. If $J = \emptyset$, $(1+\epsilon)\bar{x} < \bar{y} \leq \mathbf{1}$, hence $\bar{y}$ satisfies all pitch-2 inequalities. Thus, $J \neq \emptyset$. Consider a pitch-2 inequality of the form (4.4), and note that the left-hand side of the inequality computed in $\bar{y}$ is lower bounded by $\sum_{i \in J} \alpha_i$, where α_i is the coefficient of x_i. First assume there exists $j \in J \cap I_2$. Then $\sum_{i \in J} \alpha_i \geq \alpha_j = 2$. Similarly, if

$j, j' \in J$, then $\sum_{i \in J} \alpha_i \geq \alpha_j + \alpha_{j'} \geq 2$. In both cases, $\bar{y}$ satisfies the pitch-2 inequality. Hence, we can assume $J = \{j\} \subseteq I_1$. Then:

$$\sum_{i \in I} \alpha_i \bar{x}_i \geq \frac{2}{1+\epsilon}, \text{ from which we deduce}$$

$$\sum_{i \in I \setminus \{j\}} \alpha_i \bar{x}_i \geq \frac{2}{1+\epsilon} - \bar{x}_j \geq \frac{2}{1+\epsilon} - 1 = \frac{1-\epsilon}{1+\epsilon} \text{ and}$$

$$\sum_{i \in I} \alpha_i \bar{y}_i = \sum_{i \in I \setminus \{j\}} \alpha_i \bar{y}_i + 1 = \frac{1+\epsilon}{1-\epsilon} \sum_{i \in I \setminus \{j\}} \alpha_i \bar{x}_i + 1 \geq 2,$$

as required. A similar (simpler) argument shows that $\bar{y}$ also satisfies all pitch-1 inequalities $\sum_{i \in I} x_i \geq 1$.

□

Proof of Theorem 19. We can now sum up our $(1+\epsilon)$-oracle, see Algorithm 3. Correctness and polynomiality follow from the discussion above. ■

Algorithm 3

1: Let $\epsilon' = \frac{\epsilon}{2+\epsilon}$.
2: For $r \in \{r_i + 1 : i \in [n]\}$ and for $\alpha = r/q$, run the FPTAS for P_α with approximation factor ϵ'. If any of the output solution $\bar{z}^\alpha$ has value $\bar{v}^\alpha < 2$, output inequality (4.4) with $I = \{i \in [n] : \bar{z}^\alpha = 1\}$ and stop.
3: For $\alpha = 1/q$, run the FPTAS for P_α with approximation factor ϵ'. If the output solution $\bar{z}^\alpha$ has value $\bar{v}^\alpha < 2$, output inequality $\sum_{i : \bar{z}^\alpha = 1} x_i \geq 1$ and stop.
4: Output point $\bar{y}$ constructed as in Claim 13 with ϵ' and stop. Note that $\bar{x} \leq \bar{y} \leq \frac{1+\epsilon'}{1-\epsilon'} \bar{x} = (1+\epsilon)\bar{x}$.

4.2.3 Separating inequalities of pitch-3 and larger, with fixed support

Here, we give an example showing that inequalities of pitch-3 and higher do not have the nice structure of pitch-2. Let

$$P = \{x \in [0,1]^7 : 5x_1 + 6x_2 + 11x_3 + 16x_4 + 17x_5 + 18x_6 + 21x_7 \geq 41\}. \tag{4.11}$$

Inequality $x_1 + x_3 + x_4 + 2x_5 + x_6 + 2x_7 \geq 3$ is a facet of the first CG closure $P^{(1)}$ (although not of the integer hull of P) and thus a valid pitch-3. Observe that the coefficient of x_5 is higher than x_6 in this pitch-3, while it is the opposite in (4.11). Such situations we call *inversions* and they do not occur in (relevant) pitch-2 inequalities. Inequality $x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 + 3x_6 + 4x_7 \geq 8$ is an inverted facet of both the integer hull and the first CG closure.

For later use (in Section 4.3.4), we observe here that when $I \subseteq [n]$ is fixed, we can *efficiently* and *exactly* solve the separation problem over inequalities with support I just by solving an LP.

Clearly, we are only interested in valid inequalities $\alpha^T x \geq 1$ with $\alpha \geq 0$ and points $0 \leq x^* \leq 1$. Let $\beta = 1 - p([n] \setminus I)$. We can assume $\beta > 0$, otherwise there is no valid inequality as above with support I. Call $J \subseteq I$ *massive* if $\sum_{i \in J} p_i \geq \beta$. Consider the following LP:

$$\begin{array}{lrcll} \min & \sum_{i \in I} \alpha_i x_i^* & & & \\ \text{s.t.} & & & & \\ & \sum_{i \in J} \alpha_i & \geq & 1 & \text{for all massive } J \subseteq I \\ & \alpha & \geq & 0 & \end{array} \tag{4.12}$$

Note that, for each feasible solution $\bar{\alpha}$ to the previous LP, we have that $\bar{\alpha}^T x \geq 1$ is a valid inequality for the original MINKNAP instance, and conversely that all inequalities with support I can be obtained in this way. Hence, let α^* be the optimal solution to the previous LP. If $(\alpha^*)^T x^* < 1$, we obtain an inequality whose support is contained in I, that is violated by x^*. The support of the inequality can be extended to I by setting $\alpha_i = \epsilon$ for all $i \in I$ with $\alpha_i = 0$. On the other hand, if $(\alpha^*)^T x^* \geq 1$, x^* satisfies all inequalities with support I.

4.3 Integrality gap for MINKNAP with bounded pitch inequalities

4.3.1 When p=c

Theorem 21. *Consider an instance of* MINKNAP (3.1) *with $p = c$. Denote by K the linear relaxation of* (3.1) *to which all pitch-1 and pitch-2 inequalities have been added. The integrality gap of K is at most* 3/2.

Proof. Let $p = c$, and let $\bar{x}$ be the optimal integer solution to (3.1). We can assume $p^T \bar{x} > 1.5$, else we are done.

Claim 14. *The support of $\bar{x}$ has size* 2.

Proof. Let k be the size of the support of $\bar{x}$. If $k = 1$, then $\bar{x}$ is also the optimal fractional solution. Now assume $k \geq 3$. Remove from $\bar{x}$ the cheapest item as to obtain $\bar{x}'$. We have

$$p^T x' \geq \left(1 - \frac{1}{k}\right) p^T \bar{x} > \frac{2}{3} \cdot 1.5 = 1,$$

contradicting the fact that $\bar{x}$ is the optimal integral solution.

□

Hence, we can assume that the support of $\bar{x}$ is given by $\{i, j\}$, with $0 < p_i \leq p_j \leq 1$. Since $p_i + p_j > 1.5$, we deduce $p_j > .75$. Since $p_j \leq 1$, we deduce $p_i > .5$.

Claim 15. *Let $\ell < j$ and $\ell \neq i$. Then $p_\ell < .25$.*

Proof. Recall that for $S \subseteq [n]$ we denote its characteristic vector with χ^S. If $0.25 \le p_\ell < p_i$, then $\chi^{\{\ell,j\}}$ is a feasible integral solution of cost strictly less than $\bar{x}$. Else if $0.5 < p_i \le p_\ell < p_j$, then $\chi^{\{\ell,i\}}$ is a feasible integral solution of cost strictly less than $\bar{x}$. In both cases we obtain a contradiction.

□

Because of the previous claim, we can assume w.l.o.g. $j = i+1$.

Claim 16. $p_n + \sum_{\ell=1}^{i-1} p_\ell < 1$.

Proof. Suppose $p_n + \sum_{\ell=1}^{i-1} p_\ell \ge 1$. Since $p_\ell < .25$ for all $\ell = 1, \ldots, i-1$, there exists $k \le i-1$ such that $x_n + \sum_{\ell=1}^{k} p_k \in [1, 1.25[$. Hence $x^{\{1,\ldots,k,n\}}$ is a feasible integer solution of cost at most 1.25, a contradiction.

□

Because of the previous claim, the pitch-2 inequality $\sum_{k=i}^{n} x_k \ge 2$ is valid. The fractional solution of minimum cost that satisfies this inequality is the one that sets $x_i = x_j = 1$ (since $j = i+1$) and all other variables to 0. This is exactly $\bar{x}$. ■

4.3.2 CG closures of bounded rank of the natural MINKNAP relaxation

For $t \in \mathbb{N}_0$, let K^t be the linear relaxation of the (3.1) given by: the original knapsack inequality; non-negativity constraints; all pitch-k inequalities, for $k \le t$.

Lemma 22. *For $t \ge 2$, the integrality gap of K^t is at least* $\max\{\frac{1}{2}, \frac{t-2}{t-1}\}$ *times the integrality gap of K^{t-1}.*

Proof. Fix $t \ge 2$, and let C be the cost of the optimal integral solution to (3.1). Let C/v' be the integrality gap of K^t. Since v' is the optimal value of K^t, by the strong duality theorem (and Caratheodory's theorem), there exist nonnegative multipliers $\alpha, \alpha_1, \ldots, \alpha_n, \gamma_1, \ldots, \gamma_{n+1}$ such that the inequality $c^T x \ge v'$ can be obtained as a conic combination of the original knapsack inequality (with multiplier α), non-negativity constraints (with multipliers $\alpha_1, \ldots, \alpha_n$), and at most $n+1$ inequalities of pitch at most t (with multipliers $\gamma_1, \ldots, \gamma_{n+1}$). By scaling, we can assume that the rhs of the latter inequalities is 1. Hence $v' = \alpha + \sum_{i=1}^{r} \gamma_i$.

Claim 17. *Let $d^T x \ge 1$ be a valid pitch-t inequality for* (3.1)*, and assume w.l.o.g. that $d_1 \le d_2 \le \cdots \le d_n$. Then inequality $\sum_{i=2}^{n} d_i x_i \ge$* $\max\{\frac{1}{2}, \frac{t-2}{t-1}\}$ *is a valid inequality of pitch at most $t-1$ for* (3.1).

Proof. The inequality $\sum_{i=2}^{n} d_i x_i \ge 1 - d_1$ is a valid inequality for (3.1), and by construction it is of pitch at most $t-1$. If $t \ge 3$, we obtain $\sum_{i=1}^{t-1} d_i < 1$ and consequently $d_1 < 1/(t-1)$, from which we deduce

$$1 - d_1 > 1 - \frac{1}{t-1} = \frac{t-2}{t-1} \geq \frac{1}{2}.$$

If conversely $t = 2$, by Lemma 20 we can assume w.l.o.g. that $d_1 = 1/2$, and we can conclude $1 - d_1 = \frac{1}{2} > \frac{t-2}{t-1}$. □

Now consider the conic combination with multipliers $\alpha, \alpha_1, \ldots, \alpha_n, \gamma_1, \ldots, \gamma_{n+1}$ given above, where each inequality of pitch-t is replaced with the inequality of pitch at most $t-1$ obtained using Claim 17. We obtain an inequality $(c')^T x \geq v''$, where one immediately checks that $c' \leq c$ and

$$\begin{aligned} v'' &\geq \alpha + \textstyle\sum_{i=1}^{n+1} \gamma_i \max\left\{\frac{1}{2}, \frac{t-2}{t-1}\right\} \geq \max\left\{\frac{1}{2}, \frac{t-2}{t-1}\right\}\left(\alpha + \sum_{i=1}^{n+1} \gamma_i\right) \\ &= \max\left\{\frac{1}{2}, \frac{t-2}{t-1}\right\} v'. \end{aligned}$$

Hence the integrality gap of K^t is

$$\frac{C}{v'} \geq \frac{C}{v''} \max\left\{\frac{1}{2}, \frac{t-2}{t-1}\right\}$$

and the thesis follows since the integrality gap of K^{t-1} is at most C/v''.

■

Lemma 23. *For a fixed $\epsilon > 0$ and square integers $n \geq 4$, consider the* MINKNAP *instance K defined as follows:*

$$\begin{array}{lrcrcrl} \min & \epsilon y & + & \sqrt{n} z & + & \sum_{i=1}^n x_i & \\ st & & & & & & \\ & (n - \sqrt{n}) y & + & \frac{n}{2} z & + & \sum_{i=1}^n x_i & \geq n \\ & & & & & & \\ & y, & & z, & & x & \in \{0,1\}. \end{array}$$

For every fixed $t \in \mathbb{N}_0$, the integrality gap of K^t is $\Omega(\sqrt{n})$.

Proof. Because of Lemma 22, it is enough to show that the integrality gap of K^1 is $\Omega(\sqrt{n})$. Clearly, the value of the integral optimal solution of the instance is $\sqrt{n} + \epsilon$. We claim that the fractional solution

$$(\bar{y}, \bar{x}, \bar{z}) = \left(1, \underbrace{\frac{1}{n - \sqrt{n} + 1}, \ldots, \frac{1}{n - \sqrt{n} + 1}}_{n \text{ times}}, \frac{2}{\sqrt{n}}\right)$$

is a feasible point of K^1. Since $\epsilon \bar{y} + \sqrt{n}\bar{z} + \sum_{i=1}^n \bar{x}_i = \epsilon + 2 + \frac{n}{n - \sqrt{n} + 1}$, the thesis follows.

Observe that $(n - \sqrt{n})\bar{y} + \frac{n}{2}\bar{z} + \sum_{i=1}^n \bar{x}_i = (n - \sqrt{n}) + \frac{n}{2}\frac{2}{\sqrt{n}} + \frac{n}{n - \sqrt{n} + 1} > n$, hence $(\bar{y}, \bar{x}, \bar{z})$ satisfies the original knapsack inequality.

Now consider a valid pitch-1 inequality whose support contains y. Since $\bar{y} = 1$, $(\bar{y}, \bar{x}, \bar{z})$ satisfies this inequality. Hence, the only pitch-1 inequalities of interest do not have y in the support. Note that such inequalities must have z in the support, and some of the x_i. Hence, all those inequalities are dominated by the valid pitch-1 inequalities

$$z + \sum_{i \in I} x_i \geq 1 \; \forall I \subseteq [n], |I| = n - \sqrt{n} + 1,$$

which are clearly satisfied by $(\bar{y}, \bar{x}, \bar{z})$.

■

Theorem 24. *For a fixed $q \in \mathbb{N}_0$, let $CG^q(K)$ be the $q-th$ CG closure of the* MINKNAP *instance K as defined in Lemma 23. The integrality gap of $CG^q(K)$ is $\Omega(\sqrt{n})$.*

Proof. We will use the following fact, proved (for a generic covering problem) in [13]. Let $t, q \in \mathbb{N}_0$ and suppose $(\bar{y}, \bar{z}, \bar{x}) \in K^t$. Define point (y', x', z'), where each component is the minimum between 1 and $(\frac{t+1}{t})^q$ times the corresponding component of $(\bar{y}, \bar{z}, \bar{x})$. Then $(y', x', z') \in CG^q(K)$. Now fix t, q. We have therefore that

$$\begin{array}{rcl} \epsilon y' + \sqrt{n} z' + \sum_{i=1}^n x'_i & \leq & \left(\frac{t+1}{t}\right)^q \left(\epsilon \bar{y} + \sqrt{n} \bar{z} + \sum_{i=1}^n \bar{x}_i\right) \\ & = & \left(\frac{t+1}{t}\right)^q \left(\epsilon + 2 + \frac{n}{n - \sqrt{n} + 1}\right) \end{array}$$

and the claim follows in a similar fashion to the proof of Lemma 23. ■

4.3.3 When all knapsack cover inequalities are added

In this section we consider the min-knapsack formulation with knapsack cover inequalities (we use KC to denote this LP formulation). In [15] it is shown that the integrality gap of KC is 2. In the following we provide a simpler proof.

Let $\bar{x}$ be a feasible fractional solution for KC of cost $C(\bar{x})$. Starting from $\bar{x}$, we show a simple and fast rounding procedure to obtain a feasible integral solution of cost at most $2C(\bar{x})$.

The rounding procedure: Let $S = \{i \in [n] : \bar{x}_i \geq 1/2\}$. Set $x_i = 1$ for any $i \in S$. Consider the residual variables $\bar{S} := [n] \setminus S$. The problem is to assign integral values to the the residual variables. We call this problem the *residual problem* (RP). By abusing notation, from now on, let $\bar{x}$ denote the fractional solution for KC restricted to residual variables. Consider the following *residual relaxation* (RR):

$$\min \sum_{i \in \bar{S}} C_i x_i \tag{4.13}$$

$$s.t. \sum_{i \in \bar{S}} p'_i x_i \geq b' \tag{4.14}$$

$$0 \leq x_i \leq 1/2 \tag{4.15}$$

where $p'_i = \min\{p_i, b - p(S)\}$ and $b' = b - p(S)$. Note that $\bar{x}$ satisfies (RR). So if x^* is the optimal solution of (RR) than it follows that

$$\sum_{i \in \bar{S}} C_i x_i^* \leq \sum_{i \in \bar{S}} C_i \bar{x}_i$$

Therefore, if it exists an integral solution x^{int} to (RP) of cost $C(x^{int}) \leq 2C(x^*)$ then $C(x^{int}) \leq 2C(\bar{x})$ and we are done. We can rewrite the *residual relaxation* (RR) in the following equivalent way:

$$\min \sum_{i \in \bar{S}} \frac{C_i}{2} y_i \tag{4.16}$$

$$s.t. \sum_{i \in \bar{S}} \frac{p'_i}{2} y_i \geq b' \tag{4.17}$$

$$0 \leq y_i \leq 1 \tag{4.18}$$

Clearly the optimal fractional solution to (RR) can be obtained by ordering the variables according to their densities. W.l.o.g., assume that $\frac{C_1}{p'_1} \leq \frac{C_2}{p'_2} \leq \ldots \leq \frac{C_{|\bar{S}|}}{p'_{|\bar{S}|}}$ and let $t+1 \in [|\bar{S}|]$ be the smallest integer such that $\sum_{i=1}^{t+1} \frac{p'_i}{2} \geq b'$ and therefore (recall $p'_i \leq b'$):

$$\sum_{i=1}^{t} p'_i \geq 2b' - p'_{t+1} \geq b' \tag{4.19}$$

Note that the optimal fractional solution to (RR) picks the first t variables integrally and the last $t+1$ potentially fractional (but it could be integral). It follows that:

$$\sum_{i=1}^{t} \frac{C_i}{2} \leq \sum_{i \in \bar{S}} C_i x_i^* \tag{4.20}$$

It follows that the integral solution x^{int} obtained by setting $x_1 = 1$ for $i \in [t]$ and zero otherwise is feasible by (4.19) and of cost at most twice the optimal fractional of (RR) by (4.20).

4.3.4 When all bounded pitch and knapsack cover inequalities are added

Consider the following MINKNAP instance with $\epsilon_n = \frac{1}{\sqrt{n}}$:

$$\begin{aligned} \min \quad & \sum_{i \in [n]} x_i + \frac{1}{\sqrt{n}} \sum_{j \in [n]} z_j \\ \text{s.t.} \quad & \sum_{i \in [n]} x_i + \frac{1}{n} \sum_{j \in [n]} z_j \geq 1 + \epsilon_n \\ & x, z \in \{0,1\}^n. \end{aligned} \tag{4.21}$$

Lemma 25. *For any fixed $k \in \mathbb{N}_0$ and $n \in \mathbb{N}_0$ sufficiently large, point $(\bar{x}, \bar{z}) \in \mathbb{R}^{2n}$ with $\bar{x}_i = \frac{1+\epsilon_n}{n}$, $\bar{z}_i = \frac{k}{n}$ satisfies the natural linear relaxation, all KC and all inequalities of pitch at most k valid for* (4.21)*. Observing that the optimal integral solution is* 2*, this gives an IG of* $\frac{2}{1+\frac{k}{n}} \approx 2$.

Proof. We prove the statement by induction. Fix $k \in \mathbb{N}_0$. Note that $(\bar{x}, \bar{z})$ dominates componentwise the point generated at step $k-1$, and the latter by induction hypothesis satisfies all inequalities of pitch at most $k-1$. Let

$$\sum_{i \in I} w_i x_i + \sum_{j \in J} w_j z_j \geq \beta \tag{4.22}$$

be a valid KC or pitch-k inequality with support $I \cup J$, which gives that $w_i, w_j \in \mathbb{R}_{>0}$, $\forall i \in I$, $\forall j \in J$ and $\beta > 0$. Observe the following.

Claim 18. $|I| \geq n-1$. *In addition,* $|I| = n-1$ *or* $|J| \leq n(1-\epsilon_n)$ *implies* $w_i \geq \beta$, $\forall i \in I$.

Proof. Since all coefficients in (4.22) are strictly positive and $\beta > 0$, $|I| \leq n-2$ gives that the feasible solution $(\chi^{[n] \setminus I}, \vec{0})$ for (4.21) is cut off by (4.22), a contradiction.

Furthermore, if $|I| = n-1$ and $w_{i^*} < \beta$ for some $i^* \in I$, then $(\chi^{([n] \setminus I) \cup \{i^*\}}, \vec{0})$ is cut off, again a contradiction. Finally, if $|I| = n$, $|J| \leq n(1-\epsilon_n)$ and $w_{i^*} < \beta$ for some $i^* \in I$, then $(\chi^{\{i^*\}}, \chi^{\{[n] \setminus J\}})$ does not satisfy (4.22), but it is feasible in (4.21). □

We first show the statement for (4.22) being a KC. By the definition of KC: $\beta = 1 + \epsilon_n - |[n] \setminus I| - \frac{|[n] \setminus J|}{n}$, $w_i = \min\{1, \beta\}$, $\forall i \in I$ and $w_j = \min\{\frac{1}{n}, \beta\}$, $\forall j \in J$. If $I = [n]$, then $\sum_{i \in I} w_i \bar{x}_i = \min\{1, \beta\} \cdot (1 + \epsilon_n) \geq \beta$ since $\beta \leq 1 + \epsilon_n$. Otherwise, $|[n] \setminus I| = 1$ so $w_i = \beta$, $\forall i \in I$ and $\sum_{i \in I} w_i \bar{x}_i = \beta \frac{(n-1)(1+\epsilon_n)}{n} > \beta$ for sufficiently large n.

Conversely, let (4.22) be a valid pitch-k inequality. By Claim 18, if $|I| = n-1$ or $|J| \leq n(1-\epsilon_n)$ then $w_i \geq \beta$, $\forall i \in I$ so the proof is analogous to the one for KC. Otherwise, $|I| = n$ and $|J| > n(1-\epsilon_n)$. Consider the LP (4.12) in Section 4.2 specialized for our case – that is, we want to detect if $(\bar{x}, \bar{z})$ can be separated via an inequality with support $I \cup J$. Since $(\chi^{i, \bar{i}}, \mathbf{0})$ is feasible in (4.21) for $i, \bar{i} \in I$, then

$$\alpha_i + \alpha_{\bar{i}} \geq 1. \tag{4.23}$$

Furthermore, for n large enough one has $|J| > n(1-\epsilon_n) \geq k$ so

$$\sum_{j \in K} \alpha_j \geq 1 \tag{4.24}$$

for any k-subset K of J. We claim that the minimum in (4.12) is attained at $\bar{\alpha}_i = 1/2$, $\forall i \in I$ and $\bar{\alpha}_j = 1/k$, $\forall j \in J$. Indeed, the objective function of (4.12) computed in $\bar{\alpha}$ is given by

$$|I| \cdot \frac{1}{2} \cdot \frac{1+\epsilon}{n} + |J| \cdot \frac{1}{k} \cdot \frac{k}{n} = \frac{1+\epsilon_n}{2} + \frac{|J|}{n}.$$

On the other hand, by summing (4.23) for all possible pairs with multipliers $\frac{1+\epsilon_n}{2(n-1)}$ and (4.24) for all subsets of J of size k with multipliers $\binom{|J|-1}{k-1}^{-1} \cdot \frac{k}{n}$, simple calculations lead to

$$\sum_{i \in I} \bar{x}_i \alpha_i + \sum_{j \in J} \bar{z}_j \alpha_j \geq \frac{1+\epsilon_n}{2} + \frac{|J|}{n},$$

showing the optimality of $\bar{\alpha}$. Recalling $|J| > n(1-\epsilon_n)$, we conclude that

$$\sum_{i \in I} \bar{x}_i \bar{\alpha}_i + \sum_{j \in J} \bar{z}_j \bar{\alpha}_j = \frac{1+\epsilon_n}{2} + \frac{|J|}{n} > \frac{1+\epsilon_n}{2} + 1 - \epsilon_n > 1,$$

hence $(\bar{x}, \bar{z})$ satisfies all inequalities with support $I \cup J$. ■

5 Clustered planarity testing

The notion of clustered planarity was introduced by Feng, Cohen and Eades [56, 57] under the name *c-planarity*. A similar problem, *hierarchical planarity*, was considered already by Lengauer [73]. Since then an efficient algorithm for c-planarity testing or embedding has been discovered only in some special cases. The general problem whether the c-planarity of a clustered graph (G, T) can be tested in polynomial time is wide open, already when we restrict ourselves to three pairwise disjoint clusters and the case when the combinatorial embedding of G is a part of the input!

5.1 Basic definitions and an overview of results

A *clustered graph* is a pair (G, T) where $G = (V, E)$ is a graph and T is a rooted tree whose set of leaves is the set of vertices of G. The non-leaf vertices of T represent the clusters. Let $C(T)$ be the set of non-leaf vertices of T. For each $v \in C(T)$, let T_v denote the subtree of T rooted at v. The *cluster* $V(v)$ is the set of leaves of T_v. A clustered graph (G, T) is *flat* if all non-root clusters are children of the root cluster; that is, if every root-leaf path in T has at most three vertices. When discussing flat clustered graphs, which is basically everywhere except Sections 5, 5.2 and 5.5, by "cluster" we will refer only to the non-root clusters.

A *drawing* of G is a representation of G in the plane where every vertex is represented by a unique point and every edge $e = uv$ is represented by a simple arc joining the two points that represent u and v. If it leads to no confusion, we do not distinguish between a vertex or an edge and its representation in the drawing and we use the words "vertex" and "edge" in both contexts. We assume that in a drawing no edge passes through a vertex, no two edges touch and every pair of edges cross in finitely many points. We assume that the above properties of a drawing of G are maintained during any continuous deformation of the drawing of G except for intermediate one-time events when two edges touch in a single point or an edge passes through a vertex.

A drawing of a graph is an *embedding* if no two edges cross.

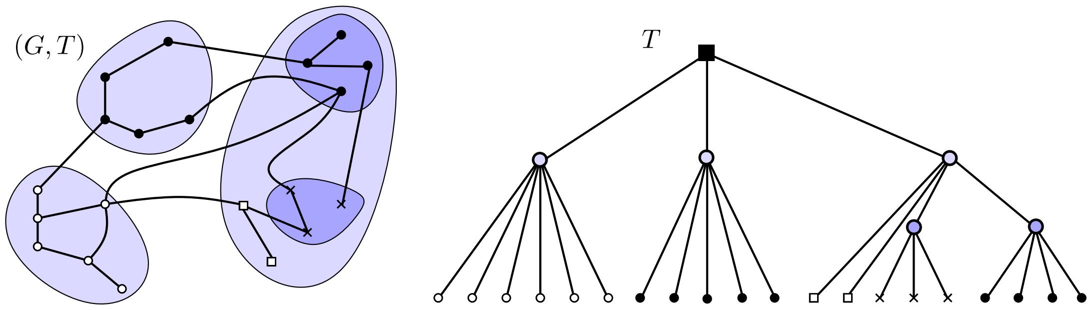

Figure 5.1 – A clustered embedding of a clustered graph (G,T) and its tree T.

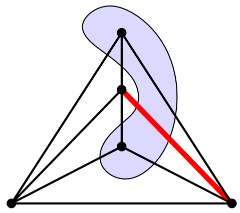

Figure 5.2 – A clustered graph with one non-root cluster, which is not c-planar.

A clustered graph (G,T) is *clustered planar* (or briefly *c-planar*) if G has an embedding in the plane such that

(i) for every $\nu \in C(T)$, there is a topological disc $\Delta(\nu)$ containing all the leaves of T_ν and no other vertices of G,

(ii) if $\mu \in T_\nu$, then $\Delta(\mu) \subseteq \Delta(\nu)$,

(iii) if μ_1 and μ_2 are children of ν in T, then $\Delta(\mu_1)$ and $\Delta(\mu_2)$ are internally disjoint, and

(iv) for every $\nu \in C(T)$, every edge of G intersects the boundary of the disc $\Delta(\nu)$ at most once.

A *clustered drawing (or embedding)* of a clustered graph (G,T) is a drawing (or embedding, respectively) of G satisfying (i)–(iv). See Figures 5.1 and 5.2 for an illustration. We will be using the word "cluster" for both the topological disc $\Delta(\nu)$ and the subset of vertices $V(\nu)$.

A clustered graph (G,T) is *c-connected* if every cluster of (G,T) induces a connected subgraph. See Figure 5.12. In order to test a c-connected clustered graph (G,T) for c-planarity, it is enough to test whether there exists an embedding of G such that for every $\nu \in C(T)$, all vertices of $V(G) \setminus V(\nu)$ are drawn in the outer face of the subgraph induced by $V(\nu)$ [57, Theorem 1]. Cortese et al. [51] gave a structural characterization of c-planarity for c-connected clustered graphs and provided a linear-time algorithm. Gutwenger et al. [64] constructed a polynomial algorithm for a more general case of *almost connected* clustered graphs, which can be also used for the case of flat clustered graphs with two clusters forming a partition of the vertex set. Biedl [47] gave the first polynomial-time algorithm for c-planarity with two clusters, including the case of straight-line or y-monotone drawings. An alternative approach to the case of two clusters was given by Hong and Nagamochi [67]. On the other hand, only very little is known in the case of three clusters, where the only clustered graphs for which a polynomial algorithm for c-planarity is known are clustered cycles [52].

Notation. In this chapter we assume that $G = (V, E)$ is a graph, and we state all our theorems for graphs. However, in some of our proofs we also use multigraphs, that is, generalized graphs that can have multiple edges and multiple loops. Most of the notions defined for graphs extend naturally to multigraphs, and thus we use them without generalizing them explicitly. We use a shorthand notation $G - v$ for $(V \setminus \{v\}, E \setminus \{vw \mid vw \in E\})$, and $G \cup E'$ for $(V, E \cup E')$. The *rotation* at a vertex v is the clockwise cyclic order of the end pieces of edges incident to v. The *rotation system* of a graph is the set of rotations at all its vertices. We say that two embeddings of a graph are the *same* if they have the same rotation system up to switching the orientations of all the rotations simultaneously. We say that a pair of edges in a graph are *independent* if they do not share a vertex. An edge in a drawing is *even* if it crosses every other edge an even number of times. A drawing of a graph is *even* if all edges are even. A drawing of a graph is *independently even* if every pair of independent edges in the drawing cross an even number of times.

Hanani–Tutte for clustered graphs. A clustered graph (G, T) is *two-clustered* if the root of T has exactly two children, A and B, and every vertex of G is a child of either A or B in T. In other words, A and B are the only non-root clusters and they form a partition of the vertex set of G. Obviously, two-clustered graphs form a subclass of flat clustered graphs. We extend both the weak and the strong variant of the Hanani–Tutte theorem to two-clustered graphs.

Theorem 26. *If a two-clustered graph (G, T) admits an even clustered drawing $\mathcal{D}$ in the plane then (G, T) is c-planar. Moreover, (G, T) has a clustered embedding with the same rotation system as $\mathcal{D}$.*

Theorem 26 has been recently generalized by the first author to the case of strip planarity [60].

Theorem 27. *If a two-clustered graph (G, T) admits an independently even clustered drawing in the plane then (G, T) is c-planar.*

We also prove a strong Hanani–Tutte theorem for c-connected clustered graphs.

Theorem 28. *If a c-connected clustered graph (G, T) admits an independently even clustered drawing in the plane then (G, T) is c-planar.*

On the other hand, we exhibit examples of clustered graphs with more than two disjoint clusters that are not c-planar, but admit an even clustered drawing. This shows that a straightforward extension of Theorem 26 and Theorem 27 to flat clustered graphs with more than two clusters is not possible.

Theorem 29. *For every $k \geq 3$ there exists a flat clustered cycle with k clusters that is not c-planar but admits an even clustered drawing in the plane.*

Gutwenger, Mutzel and Schaefer [65] recently showed that by using the reduction from [81] our counterexamples can be turned into counterexamples for [81, Conjecture 1.2][1] and for a variant of the Hanani–Tutte theorem for two simultaneously embedded planar graphs [81, Conjecture 6.20].

Embedded clustered graphs with small faces. A pair $(\mathscr{D}(G), T)$ is an *embedded clustered graph* if (G, T) is a clustered graph and $\mathscr{D}(G)$ is an embedding of G in the plane, not necessarily a clustered embedding. The embedded clustered graph $(\mathscr{D}(G), T)$ is *c-planar* if it can be extended to a clustered embedding of (G, T) by choosing a topological disc for each cluster.

We give an alternative polynomial-time algorithm for deciding c-planarity of embedded flat clustered graphs with small faces, reproving a result of Di Battista and Frati [54]. Our algorithm is based on the matroid intersection theorem. Its running time is $O(|V(G)|^{3.5})$ by [53], so it does not outperform the linear algorithm from [54]. Similarly as for our other results, we see its purpose more in mathematical foundations than in giving an efficient algorithm. We find it quite surprising that by using completely different techniques we obtained an algorithm for exactly the same case. Our approach is very similar to a technique used by Katz, Rutter and Woeginger [70] for deciding the global connectivity of switch graphs.

Theorem 30. [54] *Let $\mathscr{D}(G)$ be an embedding of a graph G in the plane such that all its faces are incident to at most five vertices. Let (G, T) be a flat clustered graph. The problem whether (G, T) admits a c-planar embedding in which G keeps its embedding $\mathscr{D}(G)$ can be solved in polynomial time.*

Organization. The rest of the chapter is organized as follows. In Section 5.2 we describe an algorithm for c-planarity testing based on Theorem 27. In Section 5.3 we prove Theorem 26. In Section 5.4 we prove Theorem 27. In Section 5.5 we prove Theorem 28. In Section 5.6 we provide a family of counterexamples to the variant of the Hanani–Tutte theorem for clustered graphs with three clusters, and discuss properties that every such counterexample, whose underlying abstract graph is a cycle, must satisfy. In Section 5.7 we prove Theorem 30. We conclude with some remarks in Section 5.8.

5.2 Algorithm

Let (G, T) be a clustered graph for which the corresponding variant of the strong Hanani–Tutte theorem holds, that is, the existence of an independently even clustered drawing of (G, T) implies that (G, T) is c-planar.

[1] For a graph G drawn in the plane the conjecture claims that we can clean (by redrawing G) of crossings a subgraph H of G consisting of independently even edges without introducing new pairs of non-adjacent edges crossing an odd number of times.

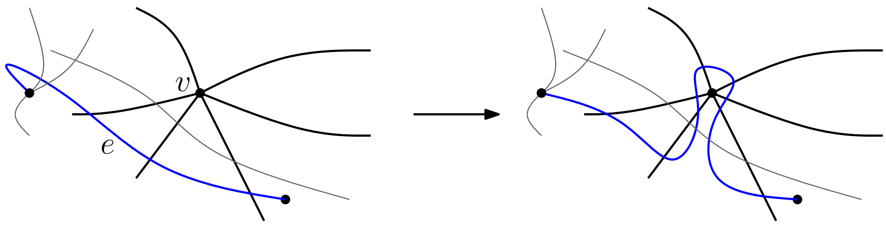

Figure 5.3 – A continuous deformation of e resulting in an edge-vertex switch (e,v).

Our algorithm for c-planarity testing is an adaptation of the algorithm for planarity testing from [80, Section 1.4.2]. The algorithm starts with an arbitrary clustered drawing $\mathscr{D}$ of (G,T). Such a drawing always exists: for example, we can traverse the tree T using depth-first search and place the vertices of G on a circle in the order encountered during the search. Then we draw every edge as a straight-line segment. Since every cluster consists of consecutive vertices on the circle, the topological discs representing the clusters can be drawn easily. The algorithm tests whether the edges of the initial drawing $\mathscr{D}$ can be continuously deformed to form an independently even clustered drawing $\mathscr{D}_0$ of (G,T). This is done by constructing and solving a system of linear equations over $\mathbb{Z}_2$. By the corresponding variant of the strong Hanani–Tutte theorem, the existence of such a drawing $\mathscr{D}_0$ is equivalent to the c-planarity of (G,T).

Now we describe the algorithm in more details. We start with the original algorithm for planarity testing and then show how to modify it for c-planarity testing.

During a "generic" continuous deformation from $\mathscr{D}$ to some other drawing $\mathscr{D}'$, the parity of the number of crossings between a pair of independent edges is affected only when an edge e passes over a vertex v that is not incident to e, in which case we change the parity of the number of crossings of e with all the edges incident to v; see Figure 5.3. We call such an event an *edge-vertex switch*. Note that every edge-vertex switch can be performed independently of others, for any initial drawing: we can always deform a given edge e to pass close to the given vertex v, while introducing new crossings with every edge "far from v" only in pairs; that is, after every event when e touches another edge, a pair of new crossings is created. For our purpose the deformation from $\mathscr{D}$ to $\mathscr{D}'$ can be represented by the set of edge-vertex switches that were performed an odd number of times during the deformation. An edge-vertex switch of an edge e with a vertex v is denoted by the ordered pair (e,v).

A drawing of (G,T) can then be represented as a vector $\mathbf{v} \in \mathbb{Z}_2^M$, where M denotes the number of unordered pairs of independent edges. The component of $\mathbf{v}$ corresponding to a pair $\{e,f\}$ is 1 if e and f cross an odd number of times and 0 otherwise. Let e be an edge of G and v a vertex of G such that $v \notin e$. Performing an edge-vertex switch (e,v) corresponds to adding the vector $\mathbf{w}_{(e,v)} \in \mathbb{Z}_2^M$ whose only components equal to 1 are those indexed by pairs $\{e,f\}$ where f is incident to v. The set of all drawings of G that can be obtained from $\mathscr{D}$ by edge-vertex switches then corresponds to an affine subspace $\mathbf{v}+W$, where W is the subspace generated by the set $\{\mathbf{w}_{(e,v)}; v \notin e\}$. The algorithm tests whether $\mathbf{0} \in \mathbf{v}+W$, which is equivalent to the solvability of a system of linear equations over $\mathbb{Z}_2$.

The difference between the original algorithm for planarity testing and our version for c-planarity testing is the following. To keep the drawing of (G,T) clustered after every deformation, for every edge $e = v_1 v_2$, we allow only those edge-vertex switches (e,v) such that v is a child of some vertex of the shortest path between v_1 and v_2 in T. Such vertices v are precisely those that are not separated from e by cluster boundaries.

We also include *edge-cluster switches* (e,C) where C is a child of some vertex of the shortest path between v_1 and v_2 in T. An edge-cluster switch (e,C) moves e over the whole topological disc representing C; see Figure 5.4. Combinatorially, this is equivalent to performing all the edge-vertex switches (e,v), $v \in C$, simultaneously. The corresponding vector $\mathbf{w}_{(e,C)}$ is the sum of all $\mathbf{w}_{(e,v)}$ for $v \in C$. Therefore, the set of allowed switches generates a subspace W_c of W. Since every allowed switch can be performed in every clustered drawing, every vector from W_c can be realized by some continuous deformation. Moreover, every clustered drawing of (G,T) can be obtained from any other clustered drawing of (G,T) by a homeomorphism of the plane and by a sequence of finitely many continuous deformations of the edges, where each of the deformations can be represented by a subset of allowed switches. Indeed, by [80, Theorem 1.18] or by the discussion of the original algorithm in previous paragraphs, the vectors $\mathbf{v}$ and $\mathbf{v}'$ corresponding to two clustered drawings $\mathscr{D}$ and $\mathscr{D}'$ of (G,T) differ by a vector $\mathbf{w} \in W$. We claim that $\mathbf{w} \in W_c$. Suppose that $\mathscr{D}$ and $\mathscr{D}'$ have the same vertices. Let e be an edge of G, let e_0 be the curve representing e in $\mathscr{D}$, and let e_1 be the curve representing e in $\mathscr{D}'$. Let γ be the closed curve obtained by joining e_0 and e_1. Let S be the set of vertices of G “inside” γ; see Subsection 5.4.1 part 2) for the definition. For every cluster C that e cannot cross, all the vertices of C belong to the same connected region of $\mathbb{R}^2 \setminus \gamma$; in particular, they are all “inside” or all “outside” γ. For every cluster C whose vertices are “inside” γ, we perform the switch (e,C) and perform the corresponding deformation on the curve e_0. Let $e_{1/2}$ be the resulting curve. The closed curve obtained by joining $e_{1/2}$ and e_1 has all the vertices of G “outside”. Therefore, if we now deform $e_{1/2}$ to e_1 arbitrarily, every vertex will be crossed an even number of times, so no changes in the parity of crossings between independent edges will occur.

Our algorithm then tests whether $\mathbf{0} \in \mathbf{v} + W_c$.

Before running the algorithm, we first remove any loops and parallel edges and check whether $|E(G')| < 3|V(G')|$ for the resulting graph G'. Then we run our algorithm on (G',T). This means solving a system of $O(|E(G')||V(G')|) = O(|V(G)|^2)$ linear equations in $O(|E(G')|^2) = O(|V(G)|^2)$ variables. This can be performed in $O(|V(G)|^{2\omega}) \le O(|V(G)|^{4.746})$ time using the algorithm by Ibarra, Moran and Hui [69].

Gutwenger, Mutzel and Schaefer [65] independently proposed a different algebraic algorithm for testing clustered planarity, based on a reduction to simultaneous planarity. It is not hard to show that their algorithm is equivalent to ours, in the sense that both algorithms accept the same instances of clustered graphs.

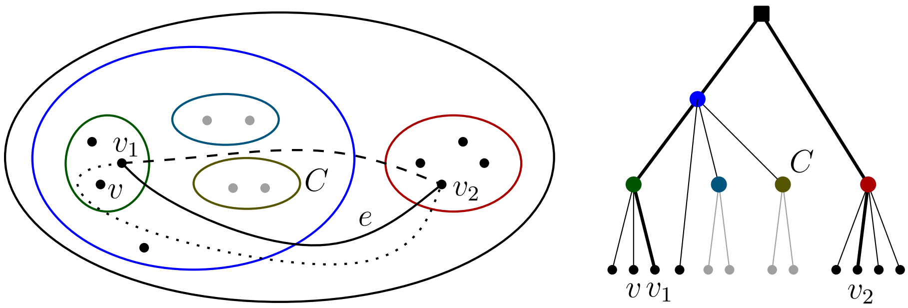

Figure 5.4 – Left: an edge-vertex switch (e, v) and an edge-cluster switch (e, C). Right: the shortest path between v_1 and v_2 in T. The four light gray vertices in the middle cannot participate in a switch with e individually.

5.3 Weak Hanani–Tutte for two-clustered graphs

First, we prove a stronger version of a special case of Theorem 26 in which G is a bipartite multigraph with the two parts corresponding to the two clusters. We note that a bipartite multigraph has no loops, but it can have multiple edges. In this stronger version, which is an easy consequence of the weak Hanani–Tutte theorem, we assume only the existence of an arbitrary even drawing of G that does not have to be a clustered drawing.

Lemma 31. *Let (G, T) be a two-clustered bipartite multigraph in which the two non-root clusters induce independent sets. If G admits an even drawing then (G, T) is c-planar. Moreover, there exists a clustered embedding of (G, T) with the same rotation system as in the given even drawing of G.*

Proof. We assume that $G = (V, E)$ is connected, since we can draw each connected component separately. Let A and B be the two clusters of (G, T) forming a partition of $V(G)$. By the weak Hanani–Tutte theorem [48, 78] we obtain an embedding $\mathcal{D}$ of G with the same rotation system as in the initial even drawing of G.

It remains to show that we can draw the discs representing clusters. This follows from a much stronger geometric result by Biedl, Kaufmann and Mutzel [46, Corollary 1]. We need only a weaker, topological, version, which has a very short proof. For each face f of $\mathcal{D}$, we may draw without crossings a set E_f of edges inside f joining one chosen vertex in A incident to f to all other vertices in A incident to f. Since the dual graph of G in $\mathcal{D}$ is connected, the multigraph $(A, \bigcup_f E_f)$ is connected as well. Let E' be a subset of $\bigcup_f E_f$ such that $T_A = (A, E')$ is a spanning tree of A. A small neighborhood of T_A is an open topological disc Δ_A containing all vertices of A, and the boundary of Δ_A crosses every edge of G at most once; see Figure 5.5. In the complement of Δ_A we can easily find a topological disc Δ_B containing all vertices of B, by drawing its boundary partially along the boundary of Δ_A and partially along the boundary of the outer face of $\mathcal{D}$. $\square$

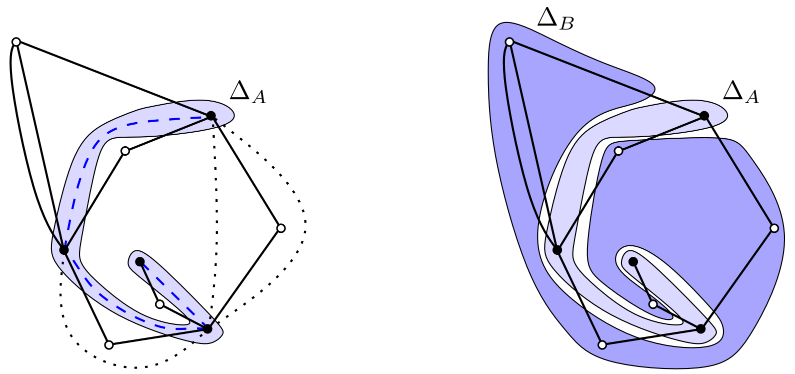

Figure 5.5 – Left: Drawing the disc Δ_A. The edges of E' are dashed, while the edges of $\bigcup_f E_f \setminus E'$ are dotted. Right: Drawing of the disc Δ_B.

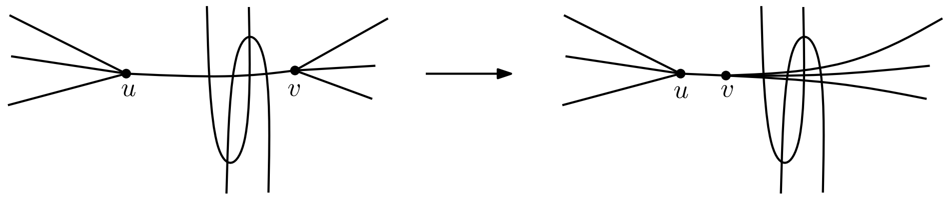

Figure 5.6 – Pulling v towards u. The evenness of the drawing is preserved.

5.3.1 Proof of Theorem 26

The proof is inspired by the proof of the weak Hanani–Tutte theorem from [78].

Let A and B be the two clusters of (G,T) forming a partition of $V(G)$. We assume that G is connected, since we can embed each component separately. Suppose that we have an even clustered drawing of (G,T). We proceed by induction on the number of vertices.

First, we discuss the inductive step. If we have an edge e between two vertices u,v in the same part (either A or B), we contract e by pulling v along e towards u while dragging all the other edges incident to v along e as well. See Figure 5.6. We keep all resulting loops and multiple edges. If some edge crosses itself during the dragging, we eliminate the self-crossing by a local redrawing. The resulting drawing is still a clustered drawing. This operation keeps the drawing even and it also preserves the rotation at each vertex. Then we apply the induction hypothesis and decontract the edge e. This can be done without introducing new crossings, since the rotation system has been preserved during the induction.

In the base step, G is a multigraph consisting of a bipartite multigraph H with parts A and B and possible additional loops at some vertices. We can embed H by Lemma 31. It remains to embed the loops. Note that after the contractions, no loop crosses the boundary of a cluster. Each loop l divides the rotation at its corresponding vertex $v(l)$ into two intervals. One of these intervals contains no end piece of an edge connecting A with B, otherwise l would cross some edge of H an odd number of times. Call such an interval a *good* interval in the rotation at $v(l)$. Observe that there are no two loops l_1 and l_2 with $v(l_1) = v(l_2) = v$ whose end-pieces have the order l_1, l_2, l_1, l_2 in the rotation at v, as otherwise the two loops would cross an odd number of times. Hence, at each vertex the good intervals of every pair of loops are either nested or disjoint.

We use induction on the number of loops to draw all the loops at a given vertex v without crossings and without changing the rotation at v. For the inductive step, we remove a loop l whose good interval in the rotation at v is inclusion minimal. Such an interval contains only the two end-pieces of l, since there exist no edges between a pair of vertices in A or B. By induction hypothesis, we can embed the rest of the loops without changing the rotation at v. Finally, we can draw l in a close neighborhood of v within the face determined by the original rotation at v. This concludes our discussion of the base step of the induction and the proof of the theorem.

5.4 Strong Hanani–Tutte for two-clustered graphs

In this section we prove Theorem 27. Let (G,T) be a two-clustered graph. Let A and B be the two clusters of (G,T) forming a partition of $V(G)$. For a subset $V' \subseteq V(G)$, let $G[V']$ denote the subgraph of G induced by V'. The following lemma gives a characterization of c-planarity for two-clustered graphs, similar to the one for c-connected clustered graphs [57, Theorem 1].

Lemma 32. *An embedding of a two-clustered graph (G,T) is a clustered embedding if and only if $G[B]$ is contained in the outer face of $G[A]$ and $G[A]$ is contained in the outer face of $G[B]$.*

Proof. The "only if" part is trivial. Let $\mathcal{D}$ be an embedding of G in which $G[B]$ is contained in the outer face of $G[A]$ and vice-versa. First we extend $\mathcal{D}$ to an embedding $\mathcal{D}_1$ of a connected two-clustered graph (G_1,T) by adding the minimum necessary number of edges between the components of $\mathcal{D}$. (If G is connected, then $G_1 = G$ and $\mathcal{D}_1 = \mathcal{D}$.) The embedding $\mathcal{D}_1$ still satisfies the assumptions of the lemma, since adding an edge between two components creates no cycle.

Next we contract each component of $G_1[A] \cup G_1[B]$ in $\mathcal{D}_1$ to a point, while keeping all the loops and multiple edges, and preserving the rotations of the vertices. Let $\mathcal{D}_2$ be the resulting embedding and (G_2,T_2) the corresponding two-clustered multigraph. The connectedness of G_2 and the assumption of the lemma imply that the interior of every loop in $\mathcal{D}_2$ is empty of vertices. We remove all the loops, and apply Lemma 31 to the resulting two-clustered multigraph (G_3,T_3). We obtain topological discs Δ_A and Δ_B certifying the c-planarity of (G_3,T_3). Finally, we reintroduce the loops and decontract the components of $G_1[A]$ and $G_1[B]$ inside the discs Δ_A and Δ_B, respectively. Finally, we delete the edges connecting the components of G. □

By the assumption of Theorem 27 and the strong Hanani–Tutte theorem, G has an embedding.

However, in this embedding, $G[B]$ does not have to be contained in a single face of $G[A]$ and vice-versa. Hence, we cannot guarantee that a clustered embedding of (G,T) exists so easily.

For an induced subgraph H of G, the *boundary* of H is the set of vertices in H that have a neighbor in $G-H$. We say that an embedding $\mathcal{D}(H)$ of H is *exposed* if all vertices on the boundary of H are incident to the outer face of $\mathcal{D}(H)$.

The following lemma is an easy consequence of the strong Hanani–Tutte theorem. It helps us to find an exposed embedding of each connected component X of $G[A]\cup G[B]$. Later in the proof of Theorem 27 this allows us to remove non-essential parts of each such component X and concentrate only on a subgraph G' of G in which both $G[A]$ and $G[B]$ are outerplanar.

Lemma 33. *Suppose that (G,T) admits an independently even clustered drawing. Then every connected component of $G[A]\cup G[B]$ admits an exposed embedding.*

Proof. Let $\mathcal{D}$ be an independently even clustered drawing of (G,T). Let Δ_A and Δ_B be the two topological discs representing the clusters A and B, respectively.

Let X be a component of $G[A]$. (For components of $G[B]$ the proof is analogous.) Let ∂X be the boundary of X. Let $E(X,B)$ be the set of edges connecting a vertex in X with a vertex in B. Observe that $E(X,B)=E(\partial X,B)$. We replace B by a single vertex v and connect it to all vertices of ∂X. We obtain a graph $X'=(V(X)\cup\{v\},E(X)\cup\{uv; u\in\partial X\}$.

We get an independently even drawing of X' from $\mathcal{D}$ by contracting Δ_B to a point and removing the vertices in $A\setminus X$ and all parallel edges. By the strong Hanani–Tutte theorem we obtain an embedding of X'. By changing this embedding so that v gets to the outer face and then removing v with all incident edges, we obtain an exposed embedding of X. □

5.4.1 Proof of Theorem 27

The proof is inspired by the proof of the strong Hanani–Tutte theorem from [78]. Its outline is as follows. First we obtain a subgraph G' of G containing the boundary of each component of $G[A]$ and $G[B]$ and such that each of $G'[A]$ and $G'[B]$ is a *cactus forest*, that is, a graph where every two cycles are edge-disjoint. Equivalently, a cactus forest is a graph with no subdivision of K_4-e. A connected component of a cactus forest is called a *cactus*. Then we apply the strong Hanani–Tutte theorem to a graph which is constructed from G' by splitting vertices common to at least two cycles in $G'[A]$ and $G'[B]$, and turning all cycles in $G'[A]$ and $G'[B]$ into wheels. The wheels guarantee that everything that has been removed from G in order to obtain G' can be inserted back. Finally we draw the clusters using Lemma 31.

Now we describe the proof in detail. Let $X_1,\dots,X_k$ be the connected components of $G[A]\cup G[B]$. By Lemma 33 we find an exposed embedding $\mathcal{D}(X_i)$ of each X_i. Let X_i' denote the subgraph of X_i obtained by deleting from X_i all the vertices and edges not incident to the outer face of $\mathcal{D}(X_i)$. Observe that X_i' is a cactus.

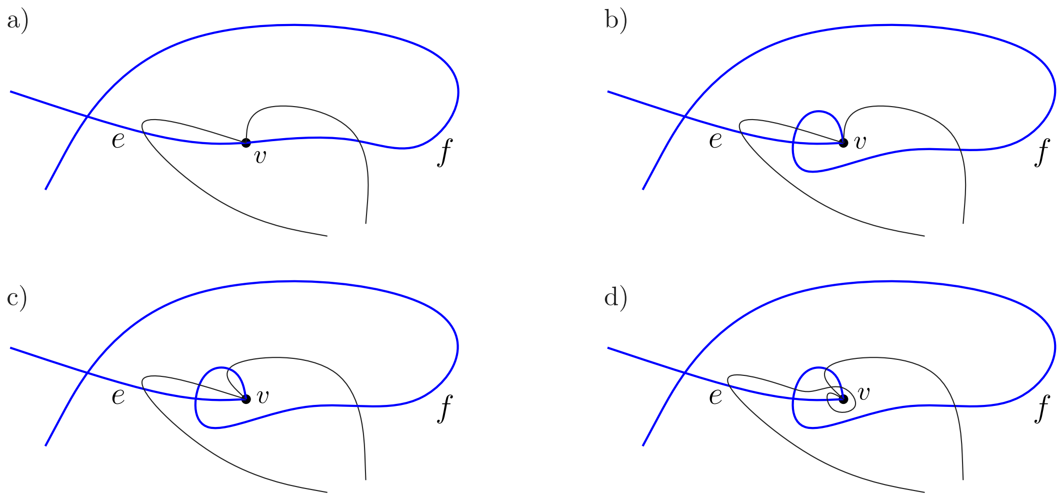

Figure 5.7 – Making e and f even by changing the drawing locally around v.

Let $G' = (\bigcup_{i=1}^{k} X_i') \cup E(A,B)$. That is, G' is a subgraph of G that consists of all the cacti X_i' and all edges between the two clusters. Let $\mathscr{D}'$ denote the drawing of G' obtained from the initial independently even clustered drawing of G by deleting the edges and vertices of G not belonging to G'. Thus, $\mathscr{D}'$ is an independently even clustered drawing of G'.

In what follows we process the cycles of $G'[A]$ and $G'[B]$ one by one. We will be modifying G' and also the drawing $\mathscr{D}'$. We will maintain the property that every processed cycle is vertex-disjoint with all other cycles in $G'[A]$ and $G'[B]$, and every edge of every processed cycle is even in $\mathscr{D}'$. Initially, the property is met as no cycle is processed. Let C denote an unprocessed cycle in $G'[A]$. For cycles in $G'[B]$, the procedure is analogous. We proceed in several steps.

1) Correcting the rotations. For every vertex v of C, we redraw the edges incident to v in a small neighborhood of v, and change the rotation at v, as follows [78]. If the two edges e, f of C incident to v cross an odd number of times, we redraw one of them, say, f, so that they cross evenly. Next, we redraw every other edge incident to v so that it crosses both e and f evenly; see Figure 5.7. After we perform these modifications at every vertex of C, all the edges of C are even. However, some pairs of edges incident to a vertex of C may cross oddly; see Figure 5.7 d). Moreover, no processed cycles have been affected since they are vertex-disjoint with C.

2) Cleaning the "inside". We two-color the connected components of $\mathbb{R}^2 \setminus C$ so that two regions sharing a nontrivial part of their boundary receive opposite colors. The existence of such a coloring is a well-known fact; for example, one can color the points of $\mathbb{R}^2 \setminus C$ using the parity of the winding number of C. We say that a point not lying on C is "outside" of C if it is contained in the region with the same color as the unbounded region. Otherwise, such a point is "inside" of C.

A C-*bridge* in G' is a "topological" connected component of $G' - E(C)$; that is, a connected

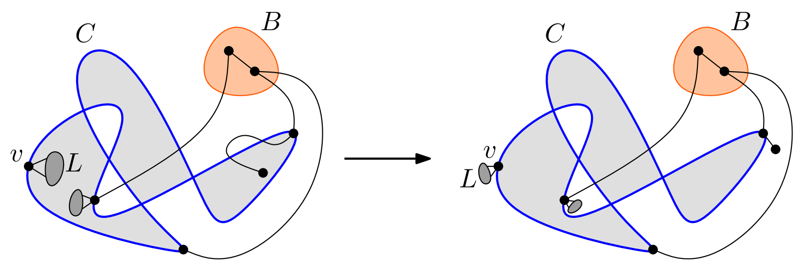

Figure 5.8 – Transforming inner C-bridges into outer C-bridges. Every nontrivial C-bridge contains a vertex in B.

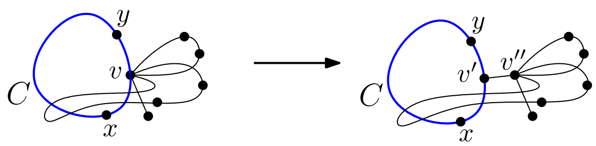

Figure 5.9 – Splitting a vertex v common to several cycles in $G'[A]$.

component K of $G' - C$ together with all the edges connecting K with C, or a chord of C in G'. We say that a C-bridge L is *outer* if all edges of L incident to C attach to the vertices of C from "outside". Similarly, we say that a C-bridge L is *inner* if all edges of L incident to C attach to the vertices of C from "inside". Since all the edges of C are even, every C-bridge is either outer or inner. A *C-bridge* is *trivial* if it attaches only to one vertex of C; otherwise it is *nontrivial*. Since C is edge-disjoint with all cycles in $G'[A]$, every nontrivial C-bridge contains a vertex of B. Since $\mathscr{D}'$ is a clustered drawing of G', all vertices of $G'[B]$ lie "outside" of C, and so every nontrivial C-bridge is outer. Therefore, every inner C-bridge is trivial. We redraw every inner C-bridge L as follows. Let v be the vertex of C to which L is attached. We select a small region in the neighborhood of v "outside" of C, and draw L in this region by continuously deforming the original drawing of L, so that L crosses no edge outside L; see Figure 5.8. After this step, nothing is attached to C from "inside".

3) Vertex splitting. Let v be a vertex of C belonging to at least one other cycle in $G'[A]$. Let x and y be the two neighbors of v in C. By the previous step, the edges xv and yv are consecutive in the rotation at v. We split the vertex v by replacing it with two new vertices v' and v'' connected by an edge, and draw them very close to v. We replace the edges xv and yv by edges xv' and yv', respectively. For every neighbor u of v that is not on C, we replace the edge uv by an edge uv''. See Figure 5.9. Clearly, this vertex-splitting introduces no pair of independent edges crossing oddly. Moreover, after all the splittings, C is vertex-disjoint with all cycles in $G'[A]$.

4) Attaching the wheels. Now we fill the cycle C with a wheel. More precisely, we add a vertex v_C into A and place it very close to an arbitrary vertex of C "inside" of C. We connect v_C with all the vertices of C by edges that closely follow the closed curve representing C either from

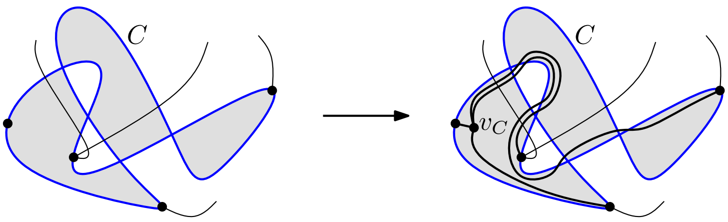

Figure 5.10 – Attaching a wheel to C.

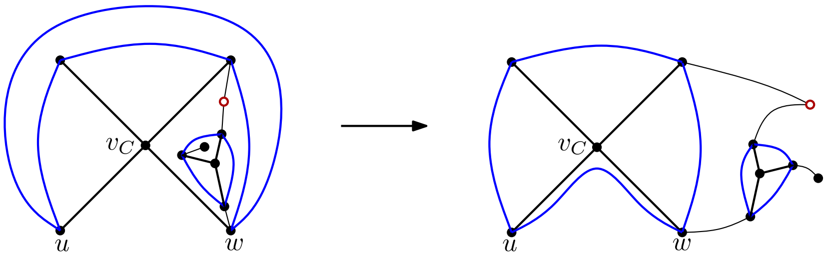

Figure 5.11 – Fixing the wheels and flipping everything else to the outer face of $G''[A]$. The circle represents a vertex in B.

the left or from the right, and attach to their endpoints on C from "inside"; see Figure 5.10. We allow portions of these new edges to lie "outside" of C only near self-crossings of C. In particular, in the neighborhoods of vertices of C, the new edges are always "inside" of C. Since no C-bridge is inner, all the new edges are even.

Let G'' denote the graph obtained after processing all the cycles of $G'[A]$ and $G'[B]$. Now we apply the strong Hanani–Tutte theorem to G''. We further modify the resulting embedding in several steps so that in the end, the only vertices and edges of G'' not incident to the outer face of $G''[A]$ or $G''[B]$ are the vertices v_C that form the centers of the wheels, and their incident edges. First, suppose that some of the wheels are embedded so that their central vertex v_C is in the outer face of the wheel. Then the outer face is a triangle, say $v_C uw$. We can then redraw the edge uw along the path $uv_C w$, without crossings, so that v_C gets inside the wheel. We fix all the wheels in this way. Next, if some of the wheels contains another part of G'' in some of its inner faces, we flip the whole part over an edge of the wheel to its outer face, without crossings. See Figure 5.11. After finitely many flips, all the inner faces of the wheels will be empty.

After the modifications, $G''[A]$ is drawn in the outer face of $G''[B]$ and vice-versa. In the resulting embedding we delete all the vertices v_C and contract the edges between the pairs of vertices v', v'' that were obtained by vertex-splits.

Thus, we obtain an embedding of G' in which for every component X_i of $G'[A] \cup G'[B]$, all vertices of $G' - X_i$ are drawn in the outer face of X_i. Now we insert the removed parts of G back to G', by copying the corresponding parts of the embeddings $\mathscr{D}(X_i)$ defined in the beginning

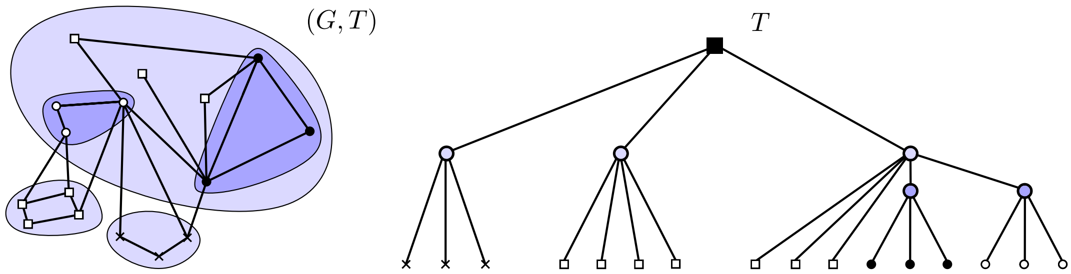

Figure 5.12 – A c-planar embedding of a c-connected clustered graph (G,T) and the corresponding tree T.

of the proof. This is possible since we are placing the removed parts of X_i inside faces bounded by simple cycles of X_i. Hence, we obtain an embedding of G in which for every component X of $G[A] \cup G[B]$, all vertices of $G - X$ are drawn in the outer face of X. Thus, Lemma 32 applies and that concludes the proof.

5.5 Strong Hanani–Tutte for c-connected clustered graphs

Here we prove Theorem 28, using the ideas from the proof of Theorem 27.

Let (G,T) be a c-connected clustered graph with an independently even clustered drawing. Our goal is to find a c-planar embedding of (G,T); see Figure 5.12. We proceed by induction on the number of clusters of (G,T). If the root cluster is the only cluster in (G,T), the theorem follows directly from the strong Hanani–Tutte theorem applied to G. For the inductive step, we assume that (G,T) has at least one non-root cluster.

A *minimal cluster* is a cluster that contains no other cluster of (G,T). Let $V(\mu)$ be a minimal cluster of (G,T). Let (G,T') be a clustered graph obtained from (G,T) by removing μ from T and attaching all its children to its parent. Note that (G,T') is still c-connected.

Starting from (G,T'), we process the connected subgraph $G[V(\mu)]$ analogously as the components of $G[A]$ in the proof of Theorem 27, where we substitute $A = V(\mu)$ and $B = V(G) - V(\mu)$. By modifying (G,T') we obtain a c-connected clustered graph (G'',T'') with an independently even clustered drawing. Now we apply the induction hypothesis and obtain a clustered embedding of (G'',T''). Again, we modify this embedding so that all vertices of $V(G'') - V(\mu)$ are in the outer face of $G''[V(\mu)]$. Then we remove the wheels, contract the new edges and insert back the removed parts of $G[V(\mu)]$. Finally we draw a topological disc $\Delta(\mu)$ around the closure of the union of all interior faces of $G[V(\mu)]$. Since $G[V(\mu)]$ is connected, this last step is straightforward and results in a clustered embedding of (G,T).

5.6 Counterexample on three clusters

In this section we construct a family of even clustered drawings of flat clustered cycles on three and more clusters that are not clustered planar. These examples imply that a straightforward generalization of the Hanani–Tutte theorem to graphs with three or more clusters is not possible.

Before giving the construction, we prove that there are no other "minimal" counterexamples to the Hanani–Tutte theorem for flat clustered cycles with three clusters, and more generally, flat clustered cycles whose clusters form a cycle structure. A reader interested only in the counterexample can immediately proceed to Subsection 5.6.1 or directly to the study of Figure 5.16.

Let $k \geq 3$. We say that a flat clustered graph (G,T) with k clusters is *cyclic-clustered* if there is a cyclic ordering of its clusters $(V_1, V_2, \ldots, V_k)$ such that for $i \neq j$, G has an edge between V_i and V_j if and only if $|i-j| \in \{1, k-1\}$; that is, if V_i and V_j are consecutive in the cyclic ordering. In this section we assume that (G,T) is a cyclic-clustered graph with k clusters. Clustered drawings of cyclic-clustered graphs with no edge-crossings outside the clusters have a simple structure.

Observation 5.6.1. *Let $\mathscr{D}$ be a clustered drawing of a cyclic-clustered graph (G,T) with k clusters on the sphere such that the edges do not cross outside the topological discs Δ_i representing the clusters V_i. Then we can draw disjoint simple curves $\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_k, \beta_k$ such that both α_i and β_i connect the boundaries of Δ_i and Δ_{i+1}, do not intersect other discs Δ_j, and the bounded region bounded by α_i, β_i and portions of the boundaries of Δ_i and Δ_{i+1} contains all portions of the edges between V_i and V_{i+1} that are outside of Δ_i and Δ_{i+1} (the indices are taken modulo k).*

Proof. The observation is obvious when there is exactly one edge between every pair of consecutive clusters. The general case follows easily by induction on the number of the inter-cluster edges. □

We note that if (G,T) has only three clusters, then the conclusion of Observation 5.6.1 holds even if (G,T) is not cyclic-clustered, that is, if there is a pair of clusters with no edge between them.

First we show that it is enough to consider clustered drawings in which the clusters are drawn as cones bounded by a pair of rays emanating from the origin. We call such drawings *radial*.

We call two clustered drawings of (G,T) *equivalent* if for every pair of independent edges e and f, the number of their crossings has the same parity in both drawings. We call a clustered drawing *weakly even* if every pair of edges between two disjoint pairs of clusters cross an even number of times. Clearly, every independently even drawing is also weakly even.

Lemma 34. *Given a weakly even clustered drawing $\mathscr{D}$ of a cyclic-clustered graph (G,T), there exists a radial clustered drawing of (G,T) equivalent to $\mathscr{D}$.*

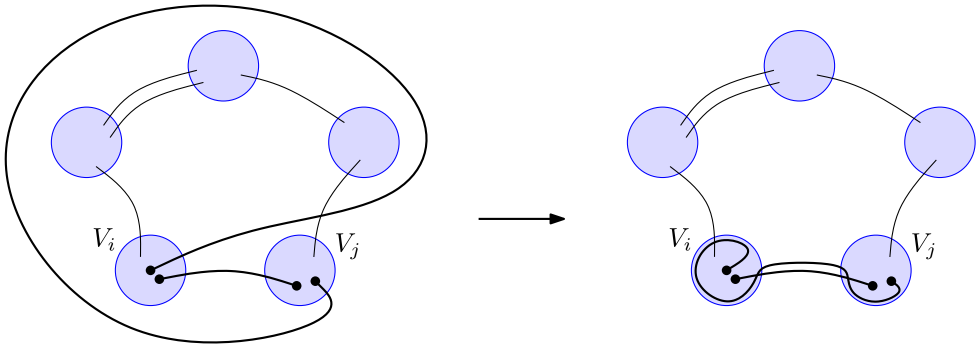

Figure 5.13 – Eliminating crossings outside clusters in a cyclic-clustered graph.

Proof. Here we refer to the topological discs representing the clusters simply by "clusters", and denote them also by V_i.

If all the crossings in $\mathcal{D}$ are inside clusters, we can easily obtain a radial drawing of (G,T) equivalent to $\mathcal{D}$ as follows. By Observation 5.6.1, we can flip some edges so that the outer face intersects all the clusters. Then the complement of the union of the discs Δ_i and the curves α_i and β_i from Observation 5.6.1 in the plane contains exactly one bounded and one unbounded component touching all the clusters. Therefore, we can continuously deform the plane and then expand the clusters to take the shape of the cones.

Suppose that there are crossings outside clusters in $\mathcal{D}$. We show how to obtain an equivalent drawing that has all crossings inside clusters, in two phases.

In the first phase, we eliminate all crossings outside clusters as follows. We continuously deform every edge of G between two consecutive clusters V_i and V_{i+1} (the indices are taken modulo k) into a narrow corridor between V_i and V_{i+1}, keeping the interiors of V_i and V_{i+1} fixed except for a small neighborhood of their boundaries. See Figure 5.13. Inside the corridor between V_i and V_{i+1}, we want the portions of the edges be noncrossing, but their order may be arbitrary. We may represent this deformation by the set $S(\mathcal{D},\mathcal{D}')$ of edge-cluster switches (see Section 5.2 for the definition) that were performed an odd number of times.

Now we again use the fact that between every two consecutive clusters of the cyclic sequence $(V_1, V_2, \dots, V_k)$, there is at least one edge of G. Since no two edges cross outside clusters in $\mathcal{D}'$, both drawings $\mathcal{D}$ and $\mathcal{D}'$ are weakly even. Hence, if $S(\mathcal{D},\mathcal{D}')$ contains an edge-cluster switch (e, V_i) with a cluster V_i that is disjoint with e, then $S(\mathcal{D},\mathcal{D}')$ contains an edge-cluster switch of e with every cluster disjoint with e. We call such an edge *switched*.

In the second phase, we further transform $\mathcal{D}'$ into a drawing $\mathcal{D}''$ by deforming the edges only inside the clusters. For every switched edge e, we perform edge-vertex switches of e with all vertices in the two clusters incident to e, except for the endpoints of e. Since performing an edge-vertex switch of e with every vertex of G not incident to e has no effect on the parity of the number of crossings of e with independent edges, the new drawing $\mathcal{D}''$ is equivalent to

$\mathcal{D}$. □

In the rest of this section we assume that G is a cycle $C_n = v_1 v_2 \dots v_n$. For technical reasons, we define v_{n+1} as v_1. For $j \in [n]$, let $\varphi(v_j)$ denote the index of the cluster containing v_j, that is, $v_j \in V_{\varphi(v_j)}$.

For every edge $v_i v_{i+1}$ of C_n we define $\mathrm{sign}(v_i v_{i+1}) \in \{-1, 0, 1\}$, as an element of $\mathbb{Z}$, so that $\mathrm{sign}(v_i v_{i+1}) \equiv \varphi(v_{i+1}) - \varphi(v_i) \pmod k$. Note that the sign is well defined since $(G, T) = (C_n, T)$ is cyclic-clustered and $k \geq 3$. We then define the *winding number* of (C_n, T) as $\frac{1}{k} \sum_{i=1}^{n} \mathrm{sign}(v_i v_{i+1})$. Note that in a radial clustered drawing of (C_n, T) where the clusters $V_1, V_2, \dots V_n$ are drawn in a counter-clockwise order, our definition of the winding number of (C_n, T) coincides with the standard winding number of the curve representing C_n with respect to the origin.

We will show that if (C_n, T) is a counterexample to the variant of the Hanani–Tutte theorem for flat cyclic-clustered graphs with k clusters, then the winding number of (C_n, T) is odd.

We say that (C_n, T) is *monotone* if $\mathrm{sign}(v_1 v_2) = \mathrm{sign}(v_2 v_3) = \dots = \mathrm{sign}(v_n v_1) \neq 0$.

In the following two lemmas we show how to reduce any even radial clustered drawing of (C_n, T) to an even radial clustered drawing of a monotone cyclic-clustered cycle $(C_{n'}, T')$, for some $n' \leq n$, that has the same winding number as (C_n, T).

We extend the notion of *edge contraction* to flat clustered cycles as follows. If (G, T) is a clustered cycle and $e = uv$ is an edge of G with both vertices u, v in the same cluster C, then $(G, T)/e$ is the clustered multigraph obtained by contracting e and keeping the vertex replacing u and v in the cluster C. The clustering of the rest of the vertices is left unchanged. If $P = uwv$ is a path of length 2 in G such that u and v are in the same cluster C, then $(G, T)/P$ is the clustered multigraph obtained by contracting the edges uw and wv and keeping the vertex replacing u and v in the cluster C. Obviously, if $G = C_n$, then the contraction of an edge yields a cycle of length $n-1$. Similarly, the contraction of a path of length 2 yields a cycle of length $n-2$.

Lemma 35. *Let $\mathcal{D}$ be an even radial clustered drawing of (C_n, T). Let e be an edge in C_n with both endpoints in the same cluster V_i. Then $(C_n, T)/e$ has an even radial clustered drawing.*

Proof. Since the edge e is completely contained inside the disc representing the cluster V_i, we can contract the curve representing e in $\mathcal{D}$ towards one of its endpoints, dragging the edges incident to the other endpoint along. Since e was even, this does not change the parity of the number of crossings between the edges of G. □

Lemma 36. *Let $\mathcal{D}$ be an even radial clustered drawing of (C_n, T). Let V_a and V_b be two adjacent clusters. Let $P_1, \dots, P_m$ be all the paths of length 2 in C_n*

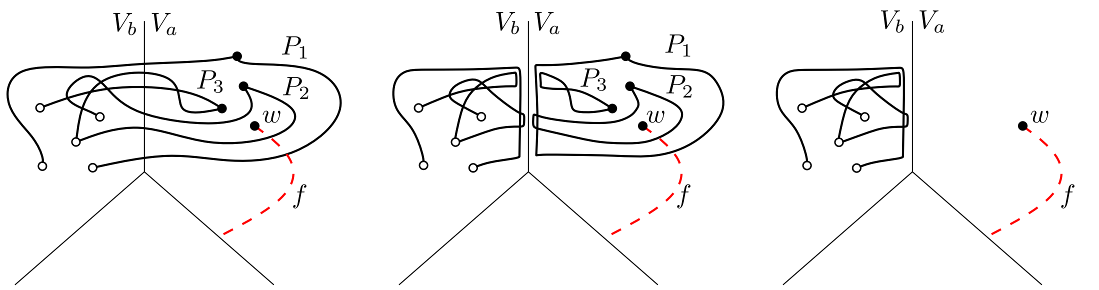

Figure 5.14 – Illustration for the proof of Lemma 36. From left to right: the successive stages of the redrawing operation eliminating paths P_1, P_2 and P_3. The edge f cannot be present in the drawing, since it would violate its evenness.

whose middle vertices belong to V_a and whose end vertices belong to V_b. Then $(C_{n'}, T') = (\dots((C_n, T)/P_1)/\dots)/P_m$ *has an even radial clustered drawing.*

Proof. Refer to Figure 5.14. By Lemma 35, we assume that no edge of C_n has both vertices in the same cluster. At the end we can recover the contracted edges by decontractions.

The proof proceeds by the following surgery performed on $\mathcal{D}$. First we cut the paths P_i at the ray r separating the clusters V_a and V_b, by removing a small neighborhood of the curves near r. Second, we reconnect the severed ends of every P_i on both sides of r, by new curves drawn close to r. This operation splits every P_i into two components. One of the components is a curve connecting the former end vertices of P_i, the other component is a closed curve containing the middle vertex of P_i. By removing the middle vertex of P_i, we replace each P_i by a single edge e_i, still represented as the union of both components of P_i. Third, we remove the closed curve of every e_i. Finally, we contract the remaining component of each e_i towards one of the end vertices, as in Lemma 35.

We claim that the resulting drawing is even. It is easy to see that during the first and the second phase, the parity of the number of crossings between each pair of edges was preserved, if we consider the edge e_i instead of each path P_i, and count the crossings on all components of every edge together. Now we show that the closed component of each e_i crosses every other edge an even number of times. This is clearly true for every edge e_j other than e_i, since only the closed component of e_j can cross the closed component of e_i. Suppose that the closed component of e_i crosses some other edge f an odd number of times. Then f intersects the region containing V_a, and so f has one endpoint, w, in V_a. Since the other endpoint of f is not in V_a, the vertex w lies "inside" the closed component of e_i (in the same sense as defined in Section 5.4). If some of the two edges incident to w had the other endpoint outside V_b, it would cross e_i, and thus P_i, an odd number of times. Therefore, both edges incident to w are incident to both clusters V_a and V_b. But every such pair was replaced by a single edge during the surgery; so there is no such f. $\square$

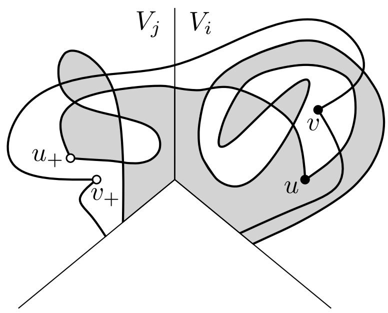

Figure 5.15 – Illustration for the proof of Theorem 37. The two pairs of vertices u, v, and u_+, v_+ are in clusters V_i and V_{i+1}, respectively. The "inside" of the curves $\gamma(u)$ and $\gamma(u_+)$ consists of the shaded regions. Thus, we have $u <_i v$ and $u_+ <_{i+1} v_+$.

Theorem 37. *Let (C_n, T) be a cyclic-clustered cycle that is not c-planar but has an even clustered drawing. Then the winding number of (C_n, T) is odd and different from 1 and -1.*

Proof. Let $k \geq 3$ be the number of clusters of (C_n, T). By Lemmas 34, 35 and 36, we may assume that (C_n, T) is monotone and that it has an even radial clustered drawing. In particular, the absolute value of the winding number of (C_n, T) is equal to n/k. Cortese et al. [52] proved that a cyclic-clustered cycle is c-planar if and only if its winding number is -1, 0 or 1. This implies that $n \geq 2k$ if (C_n, T) is not c-planar.

For every $i \in [k]$, we define a relation $<_i$ on V_i as follows. Refer to Figure 5.15. Let $u \in V_i$, and let uu_- and uu_+ denote the two edges incident to u so that $u_- \in V_{i-1}$ and $u_+ \in V_{i+1}$ (the indices are taken modulo k). Let $(uu_-)_i$ and $(uu_+)_i$ denote the parts of uu_- and uu_+, respectively, contained inside the cone representing V_i. Let $r(uu_-)$ and $r(uu_+)$ denote the endpoint of $(uu_-)_i$ and $(uu_+)_i$, respectively, different from u. That is, $r(uu_-)$ and $r(uu_+)$ are on the boundary of the cone representing V_i. Let $\gamma(u)$ denote the closed curve obtained by concatenating $(uu_-)_i$, $(uu_+)_i$, and the two line segments connecting $r(uu_-)$ and $r(uu_+)$, respectively, with the origin. We say that a pair of vertices $u, v \in V_i$ is in the relation $u <_i v$ if v is "outside" (in the same sense as defined in Section 5.4) of the curve $\gamma(u)$.

Let v_+ be the neighbor of v in V_{i+1}, and let v_- be the other neighbor of v. The relations $<_i$ and $<_{i+1}$ satisfy the following properties.

(1) the relation $<_i$ is anti-symmetric, that is, $(u <_i v) \Rightarrow \neg(v <_i u)$,

(2) $u <_i v$ if and only if $u_+ <_{i+1} v_+$.

For part (1), we observe that $(vv_-)_i$ and $(uu_+)_i$ cross an even number of times. Suppose that $u <_i v$. Then $(vv_-)_i$ and $(uu_-)_i$ cross an odd number of times if and only if $r(vv_-)$ is on $\gamma(u)$; equivalently, $r(vv_-)$ is closer to the origin than $r(uu_-)$. If also $v <_i u$, then $(vv_-)_i$ and

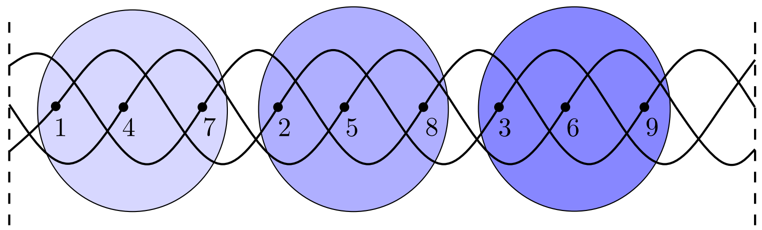

Figure 5.16 – A counterexample to the variant of the Hanani–Tutte theorem with parameters $k = 3$ and $r = 3$; the underlying graph is thus a cycle on 9 vertices. The vertices are labeled by positive integers in the order of their appearance along the cycle.

$(uu_-)_i$ cross an odd number of times if and only if $r(uu_-)$ is closer to the origin than $r(vv_-)$; a contradiction.

For part (2), let u_{++} be the neighbor of u_+ other than u. The claim follows from the fact that vv_+ crosses each of the curves $(uu_-)_i, uu_+$ and $(u_+u_{++})_{i+1}$ evenly.

Recall that $C_n = v_1v_2\dots v_n$. Let $i = \varphi(v_n)$ and suppose without loss of generality that $i+1 = \varphi(v_1)$. Suppose that n/k is even. Then both v_n and $v_{n/2}$ are in V_i. By (2), we have $v_n <_i v_{n/2} \Leftrightarrow v_1 <_{i+1} v_{n/2+1} \Leftrightarrow \cdots \Leftrightarrow v_{n/2} <_i v_n$, but this contradicts (1). Therefore, n/k is odd. □

Remark. We will see next that the relations $<_i$ are not necessarily transitive. In fact, it is not hard to see that in every counterexample to the variant of the Hanani–Tutte theorem for cyclic-clustered cycles, no relation $<_i$ is transitive.

5.6.1 Proof of Theorem 29

For simplicity of the description, we draw the graph on a cylinder, represented by a rectangle with the left and right side identified.

Let $r \geq 3$ be an odd integer and let $k \geq 3$. Our counterexample is a drawing of a monotone cyclic-clustered cycle with kr vertices and k clusters. The corresponding curve consists of $kr+1$ periods of an appropriately scaled graph of the sinus function winding r times around the cylinder, where the vertices mark the beginning of kr of the periods. We can describe the curve representing the cycle analytically as a height function $f(\alpha) = \sin\left(\frac{kr+1}{r}\alpha\right)$ on a vertical cylinder (whose axis is the z-axis) taking the angle as the parameter. The vertices of the cycle are at points $\left(i\frac{2r}{kr+1}\pi, 0\right)$, where $i = 0,\dots,kr-1$, and the clusters are separated by vertical lines at angles $\frac{2ri+1}{kr+1}\pi$, for $i = 0,\dots k-1$; see Figure 5.16. By the result of Cortese et al. [52], the cyclic-clustered cycle is not c-planar when $r > 1$.

5.7 Small faces

In this section we reprove a result of Di Battista and Frati [54] that c-planarity can be decided in polynomial time for embedded flat clustered graphs whose every face is incident to at most five vertices. In our proof, we reduce the problem to computing the largest size of a common independent set of two matroids. This can be done in polynomial time by the matroid intersection theorem [55, 72]. See e.g. [74] for further references.

In this section, we will use a shorthand notation (G,T) instead of $(\mathscr{D}(G),T)$ for an embedded clustered graph. Let (G,T) be an embedded flat clustered graph where $G=(V,E)$.

Since contracting an edge with both endpoints in the same cluster does not affect c-planarity, we will assume that (G,T) is an embedded clustered multigraph where every cluster induces an independent set. If (G,T) is c-planar and contains a loop at v, then the whole interior of the loop must belong to the same cluster as v. Hence, either there is a vertex of another cluster inside the loop, in which case (G,T) is not c-planar, or we may remove the loop and everything from its interior without affecting the c-planarity. The test and the transformation can be easily done in polynomial time. We will thus also assume that (G,T) has no loops.

A *saturator* of (G,T) is a subset S of $\binom{V}{2}\setminus E$ such that every cluster of $(G\cup S,T)$ is connected and the edges of S can be added to (G,T) without crossings.

Let S be a minimal saturator of (G,T). Then each cluster in $(G\cup S,T)$ induces a spanning tree of the cluster, and so the boundary of each cluster can be drawn easily. We have thus the following simple fact.

Observation 5.7.1 ([57])**.** *An embedded flat clustered graph (G,T) is c-planar if and only if (G,T) has a saturator.*

In order to model our problem by matroids we need to avoid two noncrossing saturating edges in one face coming from two different clusters, which might happen if the boundary of the face is not a simple cycle. To this end, we modify the multigraph further by a sequential merging of some pairs of vertices. Assuming that u and v are non-adjacent vertices incident to a common face f, *merging* of u and v in f consists in embedding a new edge uv inside f and then contracting it.

Lemma 38. *Let (G,T) be an embedded flat clustered multigraph all of whose faces are incident to at most five vertices. Suppose that G has no loops and that every cluster of (G,T) induces an independent set. Then there is an embedded flat clustered multigraph (G',T) obtained from (G,T) by merging vertices such that*

1) *(G,T) is c-planar if and only if (G',T) is c-planar, and*

2) *if (G',T) is c-planar then (G',T) has a saturator S whose edges can be embedded so that each face of G' contains at most one edge of S.*

Moreover, finding G' and verifying conditions 1) and 2) can be performed in linear time.

A *saturating pair* of a face f is a pair of vertices incident to f and belonging to the same cluster. Thus, a cluster with k vertices incident to f has $\binom{k}{2}$ saturating pairs in f. A *saturating edge* of f is a simple curve embedded in f and connecting the vertices of some saturating pair of f.

Proof of Lemma 38. Clearly, once we find that (G, T) is not c-planar we can choose $G' = G$.

A face of (G, T) is *bad* if it admits two noncrossing saturating edges, even from the same cluster. If no face of (G, T) is bad, then the choice $G' = G$ satisfies both conditions of the lemma.

Assume that (G, T) has at least one bad face f. We show that at least two vertices of f can be merged so that the resulting embedded clustered multigraph is c-planar if and only if (G, T) is c-planar. The lemma then follows by induction on the number of vertices.

Suppose that f has only two saturating pairs, $\{u, v\}$ and $\{x, y\}$. In this case, u and v belong to a different cluster than x and y. Since f is bad, the pairs $\{u, v\}$ and $\{x, y\}$ can be joined by saturating edges $e(u, v)$ and $e(x, y)$, respectively, embedded in f without crossings. Hence, we can merge u with v along $e(u, v)$ while preserving the c-planarity.

If f has more than two saturating pairs, there is a cluster C that has at least three vertices incident to f. Let $C(f)$ be the set of these vertices. If all other clusters have at most one vertex incident to f, all saturating pairs of f have vertices in $C(f)$. In this case, we can merge any pair of vertices of $C(f)$ while preserving the c-planarity.

In the remaining case, f is incident to exactly five vertices, exactly three of them, u, v and w, are in C, and the remaining two, x and y, are in another cluster D. In this case, f has four saturating pairs: $\{u, v\}$, $\{u, w\}$, $\{v, w\}$ and $\{x, y\}$. If x and y are in different components of the boundary of f, then it is possible to embed saturating edges for all the four saturating pairs without crossings. We may thus merge x with y without affecting the c-planarity. For the rest of the proof we assume that x and y are in the same component of the boundary of f. In this case, every saturating edge e joining x with y separates the face f into two components. At least one of the components is incident to at least two vertices of $C(f)$, and so at least one saturating edge of the cluster C can be embedded in f while avoiding crossings with e. If at least two saturating edges of C can be embedded in f while avoiding crossings with e, we may merge x with y along e without affecting c-planarity. Therefore, we also assume for the rest of the proof that for every saturating edge e joining x with y in f, exactly one saturating pair of C can be joined by a saturating edge embedded in f without crossings with e. This implies that for every minimal saturator of (G, T), at most two saturating pairs in total can be simultaneously joined by saturating edges embedded in f without crossings.

If for some of the saturating pairs of C in f, say, $\{u, v\}$, no saturating edge embedded in f joining u with v separates x and y, we can merge u with v without affecting c-planarity. We may thus assume that every pair of vertices in $C(f)$ can be separated by some saturating edge

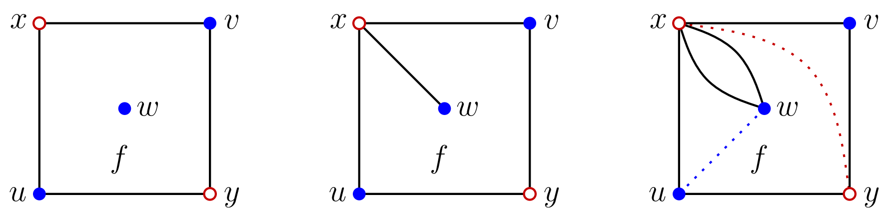

Figure 5.17 – Three cases of a bad face f whose boundary contains a 4-cycle. Saturating edges joining the pairs $\{x, y\}$ and $\{u, w\}$ are drawn in the third case. The vertices u and w can be merged without affecting the c-planarity.

joining x with y.

The boundary of f, denoted by ∂f, is a bipartite cactus forest with partitions $C(f) = \{u, v, w\}$ and $D(f) = \{x, y\}$. We call every connected component of $\mathbb{R}^2 \setminus \partial f$ other than f an *enclave*. Each enclave is bounded by a simple cycle, of length 2 or 4. Suppose that each enclave is bounded by a 2-cycle. Since each of the 2-cycles contains only one vertex of C, every saturating edge joining two vertices of $C(f)$ has to be embedded in f, and moreover, every minimal saturator of (G, T) contains exactly two of the saturating pairs $\{u, v\}$, $\{u, w\}$, $\{v, w\}$, forming a spanning tree of the triangle uvw. Similarly, every minimal saturator of (G, T) contains the pair $\{x, y\}$, and the saturating edge joining x with y must be embedded in f. By our assumptions, two of the three saturating edges in f will cross, so in this case (G, T) is not c-planar.

We are left with the case when one enclave is bounded by a 4-cycle, say, $uxvy$. Clearly, there is at most one other enclave and it is bounded by a 2-cycle. In total, there are three possibilities for the subgraph ∂f; see Figure 5.17. Every saturator of (G, T) has to contain at least one of the two saturating pairs $\{u, w\}$, $\{v, w\}$, and the corresponding saturating edge must be embedded in f. Moreover, saturating edges joining the pairs $\{x, y\}$ and $\{u, w\}$ can be simultaneously embedded without crossings. Therefore, we can merge u and w while preserving c-planarity. This finishes the proof of the lemma. □

5.7.1 Proof of Theorem 30

We start with the embedded multigraph (G', T') obtained in Lemma 38. By Observation 5.7.1 and Lemma 38, it is enough to decide whether (G', T') has a minimal saturator.

In order to test the existence of a saturator we define two matroids for which we will use the matroid intersection algorithm. The ground set of each matroid is a set $\overline{E'}$ of saturating edges of (G', T') defined as the disjoint union $\bigcup_f E_f$, over all faces of G', where E_f is a set containing one saturating edge for each saturating pair of f. By the proof of Lemma 38, no face f is bad, so every set E_f has at most two saturating edges. Moreover, if $|E_f| = 2$, then the two saturating edges in E_f cross and belong to different clusters.

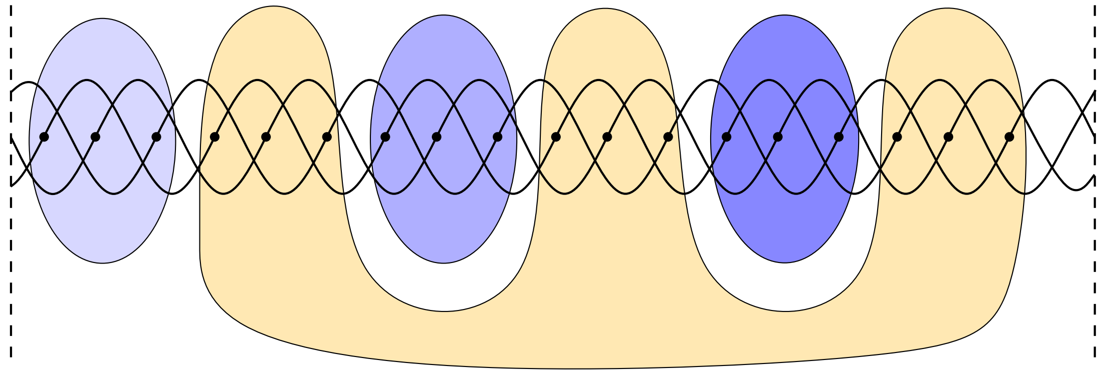

Figure 5.18 – A counterexample with $G_T = K_{1,3}$.

The first matroid, M_1, is the direct sum of graphic matroids constructed for each cluster as follows. Denote the clusters of (G', T') by C_i, $i = 1, \dots, k$. Let G_i be the multigraph induced by C_i in $\overline{G'} = (V, \overline{E'})$. The ground set of the graphic matroid $M(G_i)$ is the edge set of G_i. The rank of $M(G_i)$ is the number of vertices of G_i minus one. Since the matroids $M(G_i)$, $i = 1, \dots, k$, are pairwise disjoint, their direct sum, M_1, is also a matroid and its rank is the sum of the ranks of the matroids $M(G_i)$.

The second matroid, M_2, is a partition matroid defined as follows. A subset of $\overline{E'}$ is independent in M_2 if it has at most one edge in every face of G'.

Let M be the intersection of M_1 and M_2. If M has an independent set of size equal to the rank of M_1, then (G', T') has a saturator that has at most one edge inside each face. Thus, (G', T') is c-planar by Observation 5.7.1, and that in turn implies by Lemma 38 that (G, T) is c-planar as well. On the other hand, if (G, T), and hence (G', T'), is c-planar, then (G', T') has a minimal saturator S that has at most one edge inside each face by Lemma 38. Thus, S witnesses the fact that M has an independent set of size equal to the rank of M_1. Hence, (G', T') is c-planar if and only if M has an independent set of size equal to the rank of M_1, and this can be tested by the matroid intersection algorithm.

5.8 Concluding remarks

Let G_T be the simple graph obtained from (G, T) by contracting the clusters and deleting the loops and multiple edges. By the construction in Section 5.6 we cannot hope for the fully general variant of the Hanani–Tutte theorem when G_T contains a cycle.

A simple modification of the construction provides a counterexample also for the case when G_T is a tree with at least one vertex of degree greater than two; see Figure 5.18. This disproves our conjecture from the conference version of this work [61].

Therefore, the only open case for flat clustered graphs is the case when G_T is a collection of paths. We conjecture that the strong Hanani–Tutte theorem holds in this case.

Conjecture 1. *If G_T is a path and (G,T) admits an independently even clustered drawing then (G,T) is c-planar.*

A variant of Conjecture 1 for non-flat two-level clustered graphs in which the clusters on the bottom level form a path and one additional cluster contains all interior clusters of the path would provide a polynomial-time algorithm for c-planarity testing for strip clustered graphs, which is an open problem stated in [45].

6 Future directions

In this thesis we tackled the time-invariant incremental knapsack problem (IIK) and the min-knapsack problem (MINKNAP), as well as clustered planarity testing. While we completely settled the complexity of IIK, the status of its certain generalizations and probabilistic variants has remained unknown. Furthermore, for MINKNAP and clustered planarity we obtained negative results on bounded-pitch and Hanani–Tutte approaches, respectively. There are also evidences that those techniques can be used for giving positive results.

Incremental knapsack. A natural generalization of IIK is the incremental knapsack problem (IK) where profits are scaled by a given time-dependent factor. The construction from Section 2.2 can be extended to IK under some restrictions (see Section 2.3). A PTAS for IK with constant T is given by Della Croce et al. [18], while our construction implies an EPTAS in this case. It remains open whether IK admits a PTAS for the number of times T being logarithmic in the number of items, or for arbitrary T. The generalized assignment problem (GAP) [23] is related to IK and it cannot be approximated better then the ratio $1-1/e$. It might be worth investigating the hard instances for GAP and checking whether those could lead to an APX-hardness result for IK. For modeling industrial processes more realistically, one may consider IK\IIK in the setting where capacities are not a priori known, but revealed over time in an on-line manner. Depending on various probabilistic assumptions, the main goal in such a setting is to develop an algorithm which maximizes the competitive ratio.

Min-knapsack. We have shown that adding all bounded pitch inequalities to the natural MINKNAP relaxation does not reduce the gap in general. However, Fiorini et. al. [6, 22] have as a corollary that a quasi-polynomial extended formulation, with gap arbitrarily close to 2, can be obtained by applying pitch-1 to residual MINKNAP problems. One might like to investigate whether using higher pitch inequalities in a similar fashion can reduce the gap below 2. A major open question is whether there exists a poly-size relaxation with bounded gap.

Clustered planarity testing. We gave a counterexample for a variant of Hanani–Tutte theorem on three-clustered graphs. Nevertheless, it is plausible that the Hanani–Tutte approach can lead to a polynomial-time algorithm for c-planarity testing of strip clustered graphs.

A Appendix

A.1 Background on disjunctive programming

Introduced by Balas [3] in the 70s, disjunctive programming is based on "covering" the set by a small number of pieces which admit a relatively simple linear description. More formally, given a set $Q \subseteq \mathbb{Z}^n$ we first find a collection $\{Q_j\}_{j\in[m]}$ such that $Q = \cup_{j\in[m]} Q_j$. If there exist polyhedra P_j, $j \in [m]$ with bounded integrality gap and $P_j \cap \mathbb{Z}^n = Q_j$, then $P := \textsc{conv}(\cup_{j\in[m]} P_j)$ is a relaxation of $\textsc{conv}(Q)$ of with the same guarantee on the integrality gap. Moreover, one can describe P with (roughly) as many inequalities as the sum of the inequalities needed to describe the P_j.

A variety of benchmarks of mixed integer linear programs (MILPs) have shown the improved performances of branch-and-cut algorithms by efficiently generated disjunctive cuts [4]. Branch-and-bound algorithms for solving MILP also implicitly use disjunctive programming. The branching strategy based on thin directions that come from the Lenstra's algorithm for integer programming in fixed dimension has shown good results in practice for decomposable knapsack problems [34]. For further applications of disjunctive cuts in both linear and non-linear mixed integer settings see [8].

A.2 IIK, MKP, and UFP

A special case of GAP where profits and weights of items do not change over the set of bins is called the multiple knapsack problem (MKP). MKP is strongly NP-complete as well as IIK and has an LP-based efficient PTAS (EPTAS) [29]. Both the scheme in [29] and the one we present in Chapter 2 are based on reducing the number of possible profit classes and knapsack capacities, and then guessing the most profitable items in each class.

However, the two schemes are obtained in very different ways. The key ingredient of the approximation schemes so far developed for MKP is a "shifting trick". In rounding a fractional LP solution it redistributes and mixes together items from different buckets. Applying this technique to IIK would easily violate the monotonicity constraint, i.e. $x_{t,i} \le x_{t+1,i}$ where $x_{t,i}$ indicates whether an item i is present in the knapsack at time t. This highlights a significant difference between the problems: the ordering of the bins is irrelevant for MKP while it is crucial for IIK.

In UFP one is given a path $P = (V, E)$ with edge capacities $b : E \to \mathbb{R}_{>0}$ and a set of tasks (i.e. sub-paths) $[n]$ with profits $p : [n] \to \mathbb{R}_{>0}$ and weights $w : [n] \to \mathbb{R}_{>0}$ and, for each task $\pi \in [n]$, its starting point and ending nodes $u(\pi), v(\pi) \in V$. The goal is to select a set $S \subseteq [n]$ of maximum profit such that, for each $e \in E$, the set of tasks in S containing e has total weight at most b_e. One might like to rephrase IIK in this framework mapping times to nodes, parameters b_t to edge capacities, and the insertion of item i at time t with an appropriate path $\pi(t, i)$. However, we would need to introduce another set of constraints that for each item i at most one task $\pi = (i, t)$ is taken. This would be a more restrictive setting then UFP.

The best known approximation for UFP is $2 + \epsilon$ [1]. When all tasks share a common edge, there is a PTAS [25] based on a “sparsification” lemma introduced in [5] which, roughly speaking, considers guessing $1/\epsilon$ “locally large” tasks in the optimal solution for each $e \in E$ and by this making the computation of “locally small” tasks easier. In our approach for solving IIK we perform a kind of sparsification in Section 2.2.1 by reducing the number of times and different profits to be taken into consideration. At that point, the number of possible time/profit combinations is still too large to be able to guess a constant fraction of the highest profit items per each time. Thus, we introduce an additional pattern enumeration in Section 2.2.2 which follows the evolution of the highest-profit item in an optimal solution to an IIK instance. This pattern, – that we call ”stairway“, see Section 2.2.2 – is specific for IIK, and fundamental for describing its dynamic nature (while the set of edges for UFP is fixed). Once the stairway is fixed we can identify and distinguish between locally large and small items. This is the main difference between our approach here and the techniques used for UFP and related problems [1, 5, 25], or the techniques used in other works on IIK [11, 44].

Bibliography

[1] Aris Anagnostopoulos, Fabrizio Grandoni, Stefano Leonardi, and Andreas Wiese. A mazing 2+ϵ approximation for unsplittable flow on a path. In *Proc. of SODA '14*, pages 26–41, 2014.

[2] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. Unbounded knapsack problem: Dynamic programming revisited. *Europ. Journ. Oper. Res.*, 123(2):394 – 407, 2000.

[3] Egon Balas. Disjunctive programs: cutting planes from logical conditions. In *Nonlinear Programming 2*, pages 279 – 312. Academic Press, 1975.

[4] Egon Balas and Pierre Bonami. Generating lift-and-project cuts from the lp simplex tableau: open source implementation and testing of new variants. *Math. Prog. Comp.*, 1(2):165–199, 2009.

[5] Jatin Batra, Naveen Garg, Amit Kumar, Tobias Mömke, and Andreas Wiese. New approximation schemes for unsplittable flow on a path. In *Proc. of SODA '15*, pages 47–58, 2015.

[6] Abbas Bazzi, Samuel Fiorini, Sangxia Huang, and Ola Svensson. Small extended formulation for knapsack cover inequalities from monotone circuits. In *Proc. of SODA '17*, pages 2326–2341, 2017.

[7] Amos Beimel and Enav Weinreb. Monotone circuits for monotone weighted threshold functions. *Inf. Process. Lett.*, 97(1):12–18, 2006.

[8] Pietro Belotti, Leo Liberti, Andrea Lodi, Giacomo Nannicini, and Andrea Tramontani. Disjunctive inequalities: applications and extensions. *Wiley Encycl. of Oper. Res. and Manag. Sc.*, 2011.

[9] Daniel Bienstock. Approximate formulations for 0-1 knapsack sets. *Oper. Res. Lett.*, 36(3):317–320, May 2008.

[10] Daniel Bienstock and Benjamin McClosky. Tightening simple mixed-integer sets with guaranteed bounds. *Math. Progr.*, 133(1):337–363, 2012.

[11] Daniel Bienstock, Jay Sethuraman, and Chun Ye. Approximation algorithms for the incremental knapsack problem via disjunctive programming. *CoRR*, abs/1311.4563, 2013.

[12] Daniel Bienstock and Mark Zuckerberg. Subset algebra lift operators for 0-1 integer programming. *SIAM Journal on Optimization*, 15(1):63–95, 2004.

[13] Daniel Bienstock and Mark Zuckerberg. Approximate fixed-rank closures of covering problems. *Math. Prog.*, 105(1):9–27, Jan 2006.

[14] Daniel Bienstock and Mark Zuckerberg. Simpler derivation of bounded pitch inequalities for set covering, and minimum knapsack sets. *arXiv preprint arXiv:1806.07435*, 2018.

[15] Robert D. Carr, Lisa K. Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proc. of SODA '00*, pages 106–115, 2000.

[16] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. *Integer Programming.* Springer Publishing Company, Incorporated, 2014.

[17] George B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5(2):266–277, 1957.

[18] Federico Della Croce, Ulrich Pferschy, and Rosario Scatamacchia. Approximation results for the incremental knapsack problem. In *Proc. of IWOCA '17*, pages 75–87, 2017.

[19] Yuri Faenza and Igor Malinovic. A PTAS for the time-invariant incremental knapsack problem. In *Proc. of ISCO '18*, pages 157–169, 2018.

[20] Yuri Faenza, Igor Malinovic, Monaldo Mastrolilli, and Ola Svensson. On bounded pitch inequalities for the min-knapsack polytope. In *Proc. of ISCO '18*, pages 170–182, 2018.

[21] Yuri Faenza and Laura Sanità. On the existence of compact ϵ-approximated formulations for knapsack in the original space. *Oper. Res. Lett.*, 43(3):339–342, 2015.

[22] Samuel Fiorini, Tony Huynh, and Stefan Weltge. Strengthening convex relaxations of 0/1-sets using boolean formulas. *arXiv preprint arXiv:1711.01358*, 2017.

[23] Lisa Fleischer, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proc. of SODA '06*, pages 611–620, 2006.

[24] Radoslav Fulek, Jan Kyncl, Igor Malinovic, and Dömötör Pálvölgyi. Clustered planarity testing revisited. *Electr. J. Comb.*, 22(4):P4.24, 2015.

[25] Fabrizio Grandoni, Tobias Mömke, Andreas Wiese, and Hang Zhou. To augment or not to augment: Solving unsplittable flow on a path by creating slack. In *Proc. of SODA '17*, pages 2411–2422, 2017.

[26] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, New York, 1988.

[27] Jeffrey Hartline. Incremental optimization. *PhD thesis, Cornell University*, 2008.

[28] Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, October 1975.

[29] Klaus Jansen. A fast approximation scheme for the multiple knapsack problem. In *Proc. of SOFSEM '12*, pages 313–324, 2012.

[30] Anna R. Karlin, Claire Mathieu, and C. Thach Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. In *Proc. of IPCO '11*, pages 301–314, 2011.

[31] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

[32] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems.* Springer, 2004.

[33] Diego Klabjan, George L. Nemhauser, and Craig A. Tovey. The complexity of cover inequality separation. *Oper. Res. Lett.*, 23(1-2):35–40, August 1998.

[34] Bala Krishnamoorthy and Gábor Pataki. Column basis reduction and decomposable knapsack problems. *Discr. Opt.*, 6(3):242 – 270, 2009.

[35] Adam Kurpisz, Samuli Leppänen, and Monaldo Mastrolilli. On the hardest problem formulations for the 0/1 lasserre hierarchy. *Math. Oper. Res.*, 42(1):135–143, 2017.

[36] Eugene L. Lawler. Fast approximation algorithms for knapsack problems. *Math. Oper. Res.*, 4(4):339–356, November 1979.

[37] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Inc., New York, NY, USA, 1990.

[38] Monaldo Mastrolilli. High degree sum of squares proofs, bienstock-zuckerberg hierarchy and CG cuts. In *Proc. of IPCO '17*, pages 405–416, 2017.

[39] George B. Mathews. On the partition of numbers. *Proc- of the London Math. Soc.*, s1-28(1):486–490, 1896.

[40] Alexander Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer, 2003.

[41] Jay Sethuraman and Chun Ye. Personal communication, 2016.

[42] Alexa Sharp. Incremental algorithms: Solving problems in a changing world. *PhD thesis, Cornell University*, 2007.

[43] Martin Skutella. An introduction to network flows over time. In *Research Trends in Combinatorial Optimization*, pages 451–482, 2008.

[44] Chun Ye. On the trade-offs between modeling power and algorithmic complexity. *PhD thesis, Columbia University*, 2016.

[45] P. Angelini, G. Da Lozzo, G. Di Battista and F. Frati, Strip planarity testing, *Graph Drawing*, edited by S. Wismath and A. Wolff, vol. 8242 of *Lecture Notes in Computer Science*, 37–48, Springer (2013).

[46] T. Biedl, M. Kaufmann and P. Mutzel, Drawing planar partitions II: HH-drawings, *Graph-theoretic concepts in computer science (Smolenice Castle, 1998)*, vol. 1517 of *Lecture Notes in Computer Science*, 124–136, Springer, Berlin (1998).

[47] T. C. Biedl, Drawing planar partitions III: Two constrained embedding problems, Tech. rep., RUTCOR, Rutgers University (1998).

[48] G. Cairns and Y. Nikolayevsky, Bounds for generalized thrackles, *Discrete Comput. Geom.* **23**(2) (2000), 191–206.

[49] M. Chimani and R. Zeranski, Upward planarity testing: A computational study, *Graph Drawing*, edited by S. Wismath and A. Wolff, vol. 8242 of *Lecture Notes in Computer Science*, 13–24, Springer (2013).

[50] P. F. Cortese and G. Di Battista, Clustered planarity (invited lecture), *Twenty-first annual symposium on Computational Geometry (proc. SoCG 05)*, 30–32, ACM (2005).

[51] P. F. Cortese, G. Di Battista, F. Frati, M. Patrignani and M. Pizzonia, C-planarity of c-connected clustered graphs, *J. Graph Algorithms Appl.* **12**(2) (2008), 225–262.

[52] P. F. Cortese, G. Di Battista, M. Patrignani and M. Pizzonia, Clustering cycles into cycles of clusters, *J. Graph Algorithms Appl.* **9**(3) (2005), 391–413.

[53] W. H. Cunningham, Improved bounds for matroid partition and intersection algorithms, *SIAM J. Comput.* **15**(4) (1986), 948–957.

[54] G. Di Battista and F. Frati, Efficient c-planarity testing for embedded flat clustered graphs with small faces, *J. Graph Algorithms Appl.* **13**(3) (2009), 349–378.

[55] J. Edmonds, Submodular functions, matroids, and certain polyhedra, *Combinatorial optimization—Eureka, you shrink!*, vol. 2570 of *Lecture Notes in Computer Science*, Springer, 11–26, Berlin (2003).

[56] Q.-W. Feng, R. F. Cohen and P. Eades, How to draw a planar clustered graph, *Computing and Combinatorics*, edited by D.-Z. Du and M. Li, vol. 959 of *Lecture Notes in Computer Science*, 21–30, Springer Berlin Heidelberg (1995).

[57] Q.-W. Feng, R. F. Cohen and P. Eades, Planarity for clustered graphs, *Algorithms — ESA '95*, edited by P. Spirakis, vol. 979 of *Lecture Notes in Computer Science*, 213–226, Springer Berlin Heidelberg (1995).

[58] H. de Fraysseix, P. O. de Mendez and P. Rosenstiehl, Trémaux trees and planarity, *Internat. J. Found. Comput. Sci.* **17**(5) (2006), 1017–1029.

[59] H. de Fraysseix and P. Rosenstiehl, A characterization of planar graphs by Trémaux orders, *Combinatorica* **5**(2) (1985), 127–135.

[60] R. Fulek, Towards Hanani–Tutte theorem for clustered graphs, *Graph-Theoretic Concepts in Computer Science, Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2014)*, vol. 8747 of *Lecture Notes in Computer Science*, 176–188, Springer (2014).

[61] R. Fulek, J. Kynčl, I. Malinović and D. Pálvölgyi, Clustered planarity testing revisited, *Proceedings of the 22nd International Symposium on Graph Drawing (Graph Drawing 2014)*, vol. 8871 of *Lecture Notes in Computer Science*, 428–439, Springer, Berlin (2014).

[62] R. Fulek, M. Pelsmajer, M. Schaefer and D. Štefankovič, Hanani–Tutte, monotone drawings and level-planarity, *Thirty Essays on Geometric Graph Theory*, edited by J. Pach, 263–287, Springer New York (2013).

[63] F. L. Gall, Powers of tensors and fast matrix multiplication, arXiv:1401.7714 (2014).

[64] C. Gutwenger, M. Jünger, S. Leipert, P. Mutzel, M. Percan and R. Weiskircher, Advances in c-planarity testing of clustered graphs, *Graph Drawing*, edited by M. Goodrich and S. Kobourov, vol. 2528 of *Lecture Notes in Computer Science*, 220–236, Springer Berlin Heidelberg (2002).

[65] C. Gutwenger, P. Mutzel and M. Schaefer, Practical experience with Hanani-Tutte for testing c-planarity, *2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX)* (2014), 86–97.

[66] H. Hanani, Über wesentlich unplättbare Kurven im drei-dimensionalen Raume, *Fundamenta Mathematicae* **23** (1934), 135–142.

[67] S. Hong and H. Nagamochi, Two-page book embedding and clustered graph planarity, Tech. rep., Dept. of Applied Mathematics and Physics, University of Kyoto (2009).

[68] J. Hopcroft and R. Tarjan, Efficient planarity testing, *J. Assoc. Comput. Mach.* **21** (1974), 549–568.

[69] O. H. Ibarra, S. Moran and R. Hui, A generalization of the fast LUP matrix decomposition algorithm and applications, *J. Algorithms* **3**(1) (1982), 45–56.

[70] B. Katz, I. Rutter and G. Woeginger, An algorithmic study of switch graphs, *Acta Inform.* **49**(5) (2012), 295–312.

[71] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fundam. Math.* **15** (1930), 271–283.

[72] E. L. Lawler, Matroid intersection algorithms, *Math. Programming* **9**(1) (1975), 31–56.

[73] T. Lengauer, Hierarchical planarity testing algorithms, *J. Assoc. Comput. Mach.* **36**(3) (1989), 474–509.

[74] J. Oxley, *Matroid theory*, vol. 21 of *Oxford Graduate Texts in Mathematics*, second ed., Oxford University Press, Oxford (2011), ISBN 978-0-19-960339-8.

[75] J. Pach and G. Tóth, Which crossing number is it anyway?, *J. Combin. Theory Ser. B* **80**(2) (2000), 225–246.

[76] J. Pach and G. Tóth, Monotone drawings of planar graphs, *J. Graph Theory* **46**(1) (2004), 39–47; updated version: arXiv:1101.0967.

[77] M. J. Pelsmajer, M. Schaefer and D. Stasi, Strong Hanani–Tutte on the projective plane, *SIAM J. Discrete Math.* **23**(3) (2009), 1317–1323.

[78] M. J. Pelsmajer, M. Schaefer and D. Štefankovič, Removing even crossings, *J. Combin. Theory Ser. B* **97**(4) (2007), 489–500.

[79] M. J. Pelsmajer, M. Schaefer and D. Štefankovič, Removing even crossings on surfaces, *European J. Combin.* **30**(7) (2009), 1704–1717.

[80] M. Schaefer, Hanani-Tutte and related results, *Geometry—intuitive, discrete, and convex*, vol. 24 of *Bolyai Soc. Math. Stud.*, 259–299, János Bolyai Math. Soc., Budapest (2013).

[81] M. Schaefer, Toward a theory of planarity: Hanani-Tutte and planarity variants, *J. Graph Algorithms Appl.* **17**(4) (2013), 367–440.

[82] W. T. Tutte, Toward a theory of crossing numbers, *J. Combinatorial Theory* **8** (1970), 45–53.

[83] H. Whitney, Non-separable and planar graphs, *Trans. Amer. Math. Soc.* **34**(2) (1932), 339–362.

[84] D. H. Wiedemann, Solving sparse linear equations over finite fields, *IEEE Trans. Inform. Theory* **32**(1) (1986), 54–62.

[85] V. V. Williams, Multiplying matrices faster than Coppersmith-Winograd, *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, 887–898, ACM, New York, NY, USA (2012).

Igor Malinović

Contact information

Office:	EPFL SB MATH	Email:	igor.malinovic@epfl.ch
	MA C1 573	Tel.:	+41 21 69 32568
	1015 Lausanne	Web:	`http://disopt.epfl.ch/Malinovic`
	Switzerland	OrcID:	0000-0003-1617-9335

Research interests

Combinatorial optimization, Approximation algorithms

Education

2014-present: Ph.D. in Mathematics
EPFL, Lausanne, Switzerland
Topic: *Approximate formulations for knapsack sets*
Advisor: Prof. Friedrich Eisenbrand
Co-advisor: Prof. Yuri Faenza (Columbia University)

2014: M.Sc. in Computer Science, Minor in Mathematics
EPFL, Lausanne, Switzerland
Thesis: *Mathematical programming for SMT*
Advisor: Prof. Friedrich Eisenbrand
(GPA: 5.15/6.00)

2011: B.Sc. in Computer Science
University of Novi Sad, Novi Sad, Serbia
(GPA: 9.96/10.00)

Employment

Sept-Oct 2014: Research assistant
Invariant generation using integer programming
DISOPT group, EPFL, Lausanne, Switzerland
Advisors: Prof. Friedrich Eisenbrand,
Andrey Rybalchenko (Microsoft Research)

Aug 2013-Feb 2014: Intern
Extraction and analysis of wind farm data
ABB Corporate Research, Dätwil, Switzerland
Advisors: Carsten Franke and Yan Zhang

Mar-Jun 2013: Research assistant (part time)
SOS hierarchies for solving non-linear systems
DISOPT group, EPFL, Lausanne, Switzerland
Advisor: Prof. Friedrich Eisenbrand

Nov 2012-Jan 2013: Intern (part time)
Processes automation for financial reporting
Nestlé Capital Advisers, La Tour-de-Peilz, Switzerland
Advisor: Fabian Zepeda

Languages

Serbian (native), English (advanced), French (intermediate), German (elementary)

Student projects supervision

Joachim Moussalli : & Maurice Amendt: *Multi-objective optimization for risk treatment*, M.Sc. Spring 2018, EPFL

Augustin Prado: *Convolutional neural networks for autonomous mobile robots*, M.Sc. Fall 2017, EPFL

Jonas Racine: *Robot navigation via support vector machines*, M.Sc. Spring 2017, EPFL

Joachim Moussalli: *A PID controller for path following*, B.Sc. Fall 2016, EPFL

Pol Chapon: *Applying the Lasserre hierarchy to solving non-linear feasibility problems*, B.Sc. Spring 2016, EPFL

Teaching assistant

Spring 2015-2018: Discrete optimization (main assistant), B.Sc., EPFL

Fall 2017: Discrete mathematics, B.Sc., EPFL

Fall 2016: Analysis I, B.Sc., EPFL

Fall 2015: Linear algebra, B.Sc., EPFL

Research visits

Jan, Jul-Sept 2017: Industrial Engineering and Operations Research, Columbia University, New York, US

Apr 2014: Microsoft Research, Cambridge, UK

Organization, promotion

2015: Conference University De Suisse Occidentale (CUSO) "Career day" for Ph.D. students, EPFL, Switzerland

2010: Advances in Databases and Information Systems (ADBIS) conference, University of Novi Sad, Serbia

Prizes, awards, fellowships

2011-2013: The Swiss Government Excellence scholarship

2009, 2010: "Extraordinary award" for the results in the B.Sc. program by University of Novi Sad

2009-2011: Dr. Milan Jelić's foundation scholarship

Publications

- Y. Faenza, I. Malinović, M. Mastrolilli, and O. Svensson. *On bounded pitch inequalities for the min-knapsack polytope.* In Proc. of ISCO '18, pages 170-182, 2018
- Y. Faenza and I. Malinović. *A PTAS for the time-invariant incremental knapsack problem.* In Proc. of ISCO '18, pages 157-169, 2018
- R. Fulek, J. Kyncl, I. Malinović and D. Pálvölgyi. *Clustered planarity testing revisited.* The Electronic Journal of Combinatorics, vol. 22, num. 4, p. P4.24, 2015

Ce document a été imprimé au Centre d’impression EPFL,
imprimerie climatiquement neutre, certifiée myClimate.