Polyhedra and algorithms for problems bridging notions of connectivity and independence

Phillippe Samer

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2023

UNIVERSITY OF BERGEN

Polyhedra and algorithms for problems bridging notions of connectivity and independence

Phillippe Samer

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 21.12.2023

Year: 2023

Title: Polyhedra and algorithms for problems bridging notions of connectivity and independence

Name: Phillippe Samer

Print: Skipnes Kommunikasjon / University of Bergen

To Flaviane, my enduring definition of love

"The great revelation had never come. The great revelation perhaps never did come. Instead there were little daily miracles, illuminations, matches struck unexpectedly in the dark; here was one. This, that, and the other; [...]
In the midst of chaos there was shape; this eternal passing and flowing (she looked at clouds going and the leaves shaking) was struck into stability.
Life stand still here, Mrs. Ramsay said."

Virginia Woolf, *To the Lighthouse*

Scientific environment

This dissertation was submitted as partial fulfilment of *The PhD programme at the Faculty of Mathematics and Natural Sciences*, University of Bergen, Norway.

The author was enrolled in the Information and Communication Technology (ICT) Research School, and was supervised by Prof. Dag Haugland.

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

Abstract

We are interested in finding subgraphs that capture selected models of connectivity and independence. In short: fixed cardinality stable (or independent) sets, stable (or conflict-free) spanning trees, and matchings (or independent edge sets) inducing a connected subgraph. These are combinatorial structures that can be generalized to a number of models across network design in telecommunication and utilities, facility location, phylogenetics, among many other application domains of operations research and optimization.

We argue that the selected structures raise appealing research questions, and seek to contribute with improved mathematical understanding of the structures themselves, as well as improved algorithms to face the corresponding combinatorial optimization problems. That is, methods to identify an optimal structure, assuming the elements that form them (vertices or edges in a given graph) have a weight.

Our research spans different lines within algorithmics, combinatorics and optimization. Most of the results concern finding better descriptions of the geometric structures (namely, 0/1-polytopes) that represent all feasible solutions to each of the problems. Such improved descriptions translate to linear inequalities in integer programming formulations which, in turn, provide stronger computational results when solving benchmark instances of each problem.

We repeatedly remark the importance of sharing an open-source implementation of all algorithms and tools developed when proposing new models and solution methods in integer programming and combinatorial optimization. Our code repositories include full implementations, crafted with efficiency and modular design in mind, thus fostering reuse, further research and new applications in research and development.

Abstrakt

I denne avhandlinga interesserer vi oss for å finne delgrafer som svarer til utvalgte modeller for begrepene sammenheng og uavhengighet. I korthet betyr dette stabile (også kalt uavhengige) mengder med gitt kardinalitet, stabile (også kalt konfliktfrie) spenntrær og pardannelser (eller uavhengige kantmengder) som induserer en sammenhengende delgraf. Dette er kombinatoriske strukturer som kan generaliseres til ulike modeller for nettverksdesign innen telekommunikasjon og forsyningsvirksomhet, plassering av anlegg, fylogenetikk, og mange andre applikasjoner innen operasjonsanalyse og optimering.

Vi argumenterer for at de valgte strukturene reiser interessante forskningsspørsmål, og vi bidrar med forbedret matematisk forståelse av selve strukturene, samt forbedrede algoritmer for å takle de tilhørende kombinatoriske optimeringsproblemene. Med det mener vi metoder for å identifisere en optimal struktur, forutsatt at elementene som danner dem (hjørner eller kanter i en gitt graf) er tildelt verdier. Forskninga vår omfatter ulike områder innenfor algoritmer, kombinatorikk og optimering. De fleste resultatene omhandler det å finne bedre beskrivelser av de geometriske strukturene (nemlig 0/1-polytoper) som representerer alle mulige løsninger for hvert av problemene. Slike forbedrede beskrivelser oversettes til lineære ulikheter i heltallsprogrammeringsmodeller, noe som igjen gir mer effektive beregningsresultater når man løser referanseinstanser av hvert problem.

Vi påpeker gjentatte ganger betydninga av å dele kildekoden til implementasjonen av alle utviklede algoritmer og verktøy når det foreslås nye modeller og løsningsmetoder for heltallsprogrammering og kombinatorisk optimering. Kodearkivene våre inkluderer fullstendige implementasjoner, utformet med effektivitet og modulær design i tankene, og fremmer dermed gjenbruk, videre forskning og nye anvendelser innen forskning og utvikling.

List of publications

Part II of the present thesis includes reprints of articles 2, 4, 5 *and* 6 *below, as permitted by Elsevier B.V. and Springer-Verlag GmbH Germany.*

1. Phillippe Samer, Dag Haugland, *The unsuitable neighbourhood inequalities for the fixed cardinality stable set polytope*, chapter in **Graphs and Combinatorial Optimization: from theory to applications**, AIRO Springer Series, Volume 5, 2021. `https://doi.org/10.1007/978-3-030-63072-0_9`.

2. Phillippe Samer, Dag Haugland, *Fixed cardinality stable sets*, **Discrete Applied Mathematics**, Volume 303, 2021. `https://doi.org/10.1016/j.dam.2021.01.019`.

3. Phillippe Samer, Dag Haugland, *Towards stronger Lagrangean bounds for stable spanning trees*, peer-reviewed proceedings of **The 10th International Network Optimization Conference**, Aachen, Germany, 2022. `https://doi.org/10.48786/inoc.2022.06`.

4. Phillippe Samer, Dag Haugland, *Polyhedral results and stronger Lagrangean bounds for stable spanning trees*, **Optimization Letters**, Volume 17, Issue 6, 2023. `https://doi.org/10.1007/s11590-022-01949-8`.

5. Phillippe Samer, Phablo F.S. Moura, *Polyhedral approach to weighted connected matchings in general graphs*, **under review** for publication in Discrete Applied Mathematics; also available at `https://doi.org/10.48550/arXiv.2310.05733`.

6. Phillippe Samer, *On a class of strong valid inequalities for the connected matching polytope*, **under review** for publication in Combinatorica; also available at `https://doi.org/10.48550/arXiv.2309.14019`.

For the sake of improved typesetting and legibility, the articles included in this thesis are in their "accepted manuscript" form. That is, the papers are presented in the form they were accepted for publication, after the peer-review process, but before changing to the corresponding journal particular typeset. This is in conformity with the rights and policies pages of the corresponding publishers, and is **guaranteed** by the *rights retention policy for open access to scholarly articles* adopted by The University of Bergen from 1st December 2022. We appreciate the clear information provided by the University of Bergen Library pages, in particular by Principal Librarians Irene Eikefjord and Tormod Eismann Strømme, and congratulate all those responsible and maintaining the Bergen Open Research Archive (BORA).

List of acronyms and abbreviations

ACO Algorithms, combinatorics and optimization

CM Connected matching

KSTAB k-stable set, or, preferably, fixed cardinality stable set

ICT Information and communication technologies

IP Integer programming

IPCO Integer programming and combinatorial optimization

LP Linear programming

LD Lagrangean decomposition

MIP (MILP) Mixed-integer (linear) programming

OR Operations research

Polytime Polynomial (worst-case) time complexity

SST Stable spanning tree

SDP Semidefinite programming

UNI Unsuitable neighbourhood inequalities (valid for the KSTAB polytope)

WCM Weighted connected matching

Abbreviations from Latin:

Cf. *Confer*, meaning “compare (with)”

Et al. *Et alii/aliae/alia*, meaning “and others”

Etc. *Et cetera*, meaning “and the rest (of such things)”

E.g. *Exempli gratia*, meaning "for the sake of an example"

I.e. *Id est* , meaning "that is (to say)"

Contents

Part I

Research overview

Chapter 1

Introduction

The field of integer programming and combinatorial optimization (IPCO) concerns the search for optimal solutions to problems where there are many possible combinations or configurations of smaller choices which build up a solution. This ranges from widespread applications in logistics and industrial engineering, *e.g.* assigning post office deliveries to different vehicles and routes all over the country, to novel programs that increase the rate of successful kidney transplants by matching donors and compatible patients [Lam and Mak-Hau, 2020].

Research on IPCO deals both with fundamental questions between pure mathematics and theoretical computer science, as well as experimental work on algorithms for different problems arising from applications. By fundamental, we refer to basic research on the mathematical structures (especially *polyhedra*) defined by the set of possible solutions to a given problem. Most of the work in the present thesis concentrates on this particular discipline of IPCO, which is called *polyhedral combinatorics*. The premise of the field (*and* historical success, matter of fact) is that greater knowledge on the underlying polyhedron can have a great impact on our ability to solve the corresponding problems.

Our subject matter is the study of such mathematical structures arising from an interesting class of models; specifically, three models capturing some concept of connectivity and independence in graph theory. We defend the viewpoint that the selected problems are both general enough, as each of the three models can apply to several real world applications in areas like communication networks or utilities distribution, while somehow interconnected, which allows for a systematic approach to the study of the corresponding polyhedra. Regardless of the final success of the author, it *was* indeed rather appealing to investigate those problems simultaneously during the course of a PhD degree.

1.1 Thesis outline

From Chapter 2 onwards we make precise definitions of the specific *problems* about connectivity and independence in graphs that we investigate. Before that, it is wise to declare what we wanted to achieve, and what was deemed *beyond the scope of our project* from the start.

The broad goals of our research project leading to this thesis include the following.

(i) To extend the existing knowledge of the polyhedral structures underlying the selected problems.

(ii) To make algorithmic advances towards finding optimal solutions to the problems.

(iii) To contribute useful, high-quality, free and open-source computer implementations that increase society's ability to face the selected problems and their eventual applications.

We remark that the *reproducibility* of our materials and methods, as well as the easier comparison against future work by other researchers, are desirable consequences of the last item above.

To limit how far we would stretch our efforts before committing to the actual research questions and long path towards submitting results for publication before the PhD Programme time limit expires, some equally important aspects of research in algorithms, combinatorics and optimization (ACO) had to be left to further researchers working on the selected problems. We highlight that our contributions have the following limitations.

(i) We do not work towards making any specific real-world applications of the selected problems, but trust in the relevance and numerous decades-old success stories of the basic IPCO structures over which the selected problems are defined.

(ii) We do not attempt to write an introduction to the theory or methodologies in IPCO, polyhedral combinatorics, or discrete mathematics in this thesis.

(iii) We do not strive to propose a single, sophisticated method that tackles all aspects of problem-solving (and its applications) when investigating any of the selected problems. In some cases, we could *only* make a thorough effort to find results on theoretical questions of a problem; in other cases, we focus on finding stronger bounds to optimal values in benchmark sets. In particular, all results in this thesis contribute to either the *theoretical* understanding of a problem, or to its *exact* solution – which is not to underestimate the importance of heuristics and approximation algorithms (further comments and suggestions are collected in Section 3.2).

(iv) On a similar note, we work only in a *deterministic* computational model (we do not contemplate randomized methods), assume perfect knowledge of the problem input (stochastic or robust models are not contemplated), and we refer to computational hardness in the framework of classical complexity theory – basically P or NP-hard (not fine-grained or parameterized complexity).

With the above caveats in place, we can summarize the contents of the thesis as follows.

The remainder of Part I continues with Chapter 2, where the selected problems are finally defined, illustrated, and the state of affairs in the related literature is outlined. That is followed by Chapter 3, whose main goal is to collect connections and open questions between the selected problems and our contributions.

Part II contains the resulting scientific articles that were published or that are currently under review for publication. The papers contained in Chapters 4 and 5 concern the first two selected problems: fixed cardinality stable sets and stable spanning trees. Finally, the papers contained in Chapter 6 and Chapter 7 refer to the third selected problem: weighted connected matchings.

1.2 Background terminology and notation

As mentioned earlier in this introduction, no effort is made to overcome the most brilliant references that were fundamental to the education of the author of this thesis, trying to introduce the theory and methodologies in IPCO or polyhedral combinatorics, for example. The limited contents below include the few points where the author is aware that a choice of notation or terminology is made. Otherwise, these are actually quite standard, following the basic textbook references mentioned below.

Note that we write $[k] \stackrel{\text{def}}{=} \{1, \ldots, k\}$.

Graph theory

We refer the reader to the standard textbooks by Bondy and Murty [2007] and Diestel [2017].

All graphs in this work are finite and simple, *i.e.* they have no loops or parallel edges joining the same pair of vertices. Unless stated otherwise, they are also undirected.

We denote the set of vertices of graph G by $V(G)$, and the set of its edges by $E(G)$. Whenever graph G is clear in the context, we write $n \stackrel{\text{def}}{=} |V(G)|$ and $m \stackrel{\text{def}}{=} |E(G)|$.

Note that we call graph H a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, while the *subgraph induced* by vertices in $S \subseteq V(G)$ is $G[S] \stackrel{\text{def}}{=} (S, E(S))$, where $E(S) \subseteq E(G)$ denotes the set of edges in G with both endpoints in S.

Recall that a *matching* (or *independent edge set*) M in a graph G consists of a set of pairwise disjoint edges, that is, no two edges in M share a common vertex of G. The set of endvertices of edges in M is denoted by V_M, and we say that each such vertex is *covered* or *saturated* by M.

Polyhedral combinatorics

We do not go beyond the polyhedral theory covered by [Wolsey and Nemhauser, 1999, Chapter I.4]. Note that all polyhedra in this work are rational. For that reason, and taking the representation of numbers in a computer into consideration, we usually prefer to define systems and sets in $\mathbb{Q}^n$ instead of $\mathbb{R}^n$.

Let T be a subset of vertices (alternatively, edges) of G. We use χ^T to denote the *incidence vector*, or *characteristic vector*, of T. That is, the binary vector in $\{0,1\}^n$ (alternatively, $\{0,1\}^m$) determined by $\chi^T_u = 1$ if and only if $u \in T$.

We generally work with combinatorially defined polytopes. For instance, the spanning tree polytope of G is the convex hull of the set $\mathcal{F}_{st}$ of incidence vectors of spanning trees in G, denoted $conv(\mathcal{F}_{st})$.

Integer programming and combinatorial optimization

For integer programming and combinatorial optimization more generally, we follow Wolsey and Nemhauser [1999] and Bertsimas and Weismantel [2005]. In any case, the collection by Schrijver [2003] is our ultimate resource.

The standard measure of quality of an IP formulation is how tightly it approximates a perfect formulation [Bertsimas and Weismantel, 2005, Chapter 1]. Whenever we mention a *perfect* or *ideal* formulation of a set $S \subset \mathbb{Q}^n$, we refer to a system of linear inequalities $Ax \leq b$ such that $conv(S) = \{x \in \mathbb{Q}^n : Ax \leq b\}$.

We commit a standard abuse of language in our community, and use *convexification* to refer to the process of deriving a stronger reformulation of a set. For instance, the central theorem of Geoffrion [1974] tells us that we *convexify* the set defined by the non-dualized constraints in a particular Lagrangean relaxation scheme. Recall also that a Lagrangean relaxation scheme is said to have the *integrality property* if that convexification matches the LP relaxation of the non-dualized constraints.

Finally, we recall that the *linear programming problem* is in P, even if the simplex *algorithm* (with pivoting rules known at the time of writing) has exponential time complexity in the worst case. An important consequence throughout all of combinatorial optimization then is: the availability of a perfect formulation for some combinatorial optimization problem Π implies that $\Pi \in \mathsf{P}$.

Chapter 2

Subgraphs at the crossroads of connectivity and independence

Our goal in this chapter is to specify the questions investigated in this thesis. We also illustrate the underlying notions of connectivity and independence, although it goes beyond the mathematical definitions, and anticipating practical applications does require some degree of creativity from the reader.

We begin each section by introducing a fundamental combinatorial structure, upon which we define the subject matter of our contributions. Namely, (i) a family of polytopes determined by the convex hull of characteristic vectors of the corresponding structures in a given input problem, and (ii) the resulting combinatorial optimization problem for which we propose algorithms and bounds.

2.1 Fixed cardinality stable sets

A *stable set* (or *independent set*, or *co-clique*) in a graph G is a subset of pairwise non-adjacent vertices in G. That is, the subgraph induced by a stable set has no edges. Given $k \in [n]$ and a vertex-weighting function $w : V(G) \to \mathbb{Q}_+$, the k *stable set problem* (KSTAB) consists in finding a minimum weight stable set of cardinality *exactly* k in G, or deciding that none exists.

In what follows, we use *kstab* as an abbreviation of cardinality k stable set, while *KSTAB* is used to refer to the optimization problem. It is important to note that k is also part of the input to this problem. If we fix k to some arbitrary integer number, one could simply enumerate all kstabs to solve the KSTAB problem over stable sets of that cardinality by

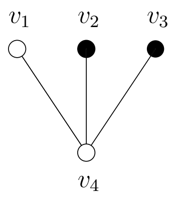


Figure 2.1: The set $\{v_2, v_3\}$ is an example inducing a 2-stab in this graph.

inspection, within time polynomially bounded by a function of n. On the other hand, having $k \in \mathbb{Z}$ be part of an input instance, we have an immediate polynomial reduction of the classical stable set problem to KSTAB, with a gadget using less than n calls to an oracle that finds optimal kstabs, implying that we have an NP-hard problem at hand.

For an example, consider the *claw graph* $K_{1,3}$, depicted in Figure 2.1. The maximum cardinality of a stable set in this graph is 3, while each of $\{v_1, v_2\}$, $\{v_1, v_3\}$, and $\{v_2, v_3\}$ is a kstab, for $k = 2$. Consider, further in the example, the objective function w defined by $w(v_1) = 1$, $w(v_2) = 2$, $w(v_3) = 3$, and $w(v_4) = 4$. Then, $\{v_1, v_2\}$ is the only optimal solution of the resulting KSTAB problem over $K_{1,3}$ and $k = 2$.

The *kstab polytope* $\mathfrak{C}(G, k)$ is the convex hull in $\mathbb{Q}^n$ of all incidence vectors of kstabs in G. The combinatorial optimization problem over $\mathfrak{C}(G, k)$ is thus formalized as follows. The optimization sense (*i.e.* whether we are seeking a kstab of minimum or maximum weight) does not play any role in this problem; we choose *minimum weight* here only for consistency with the problem defined in the next section.

Minimum Weight Fixed Cardinality Stable Set (KSTAB) Problem

Instance: a finite, simple, undirected graph G; an integer k; and a vertex weight function $w : V(G) \to \mathbb{Q}$.

Task: determine a cardinality k stable set in G of minimum weight, or decide that none exists.

Conceptual model and applications

The *independence* model captured by KSTAB (and the classical stable set problem, in this context), is that of having no interference. This suggests *conflict-free* selections of a prescribed size k, *e.g.* determining compatible channels in fiber-optic communication network, or deploying a number of sensors or surveillance agents in a map. On the other hand, the level of *connectivity* prescribed by this structure in fixed as the lowest possible: no two vertices are connected to each other.

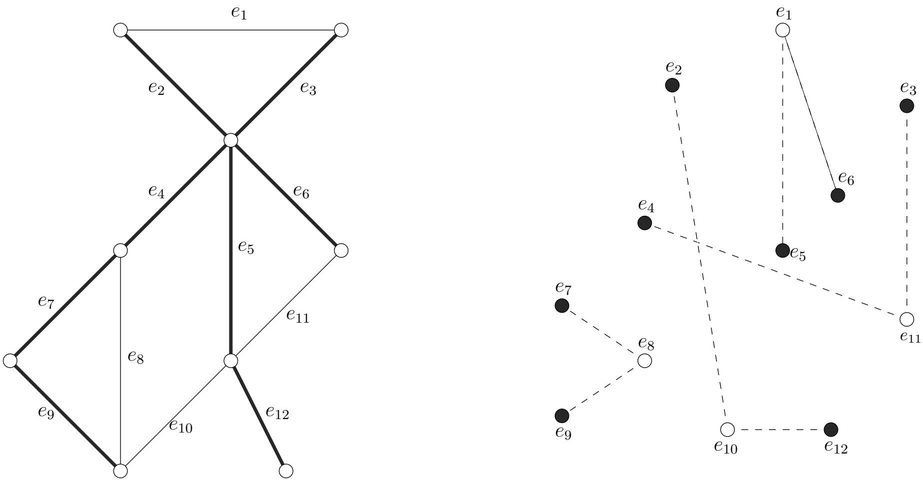
For example, we could imagine a model where vertices in a graph correspond to current warehouses of a particular grocery retailer, where two warehouses are joined by an edge if and only if both are considered critical to supply the demand of a common neighbourhood. Suppose that the company strategic board seeks advice concerning a plan to close k warehouses during a season's break. We determine vertex weights corresponding to the cost of shutting down and transferring the operations of a warehouse to close alternatives. In this scenario, solving the KSTAB problem corresponds to finding the best operational plan, so as to minimize costs while not impacting any neighbourhood to a critical extent.

Literature overview

The KSTAB problem was hardly addressed before in the literature. To the best of our knowledge, kstabs only appeared before in two settings. First, when Janssen and Kilakos [1999] studied the convex hull of stable sets of cardinality *at most* k, in the particular case of $k \in \{2, 3\}$. Later, we also find a mention to kstabs in an algorithm for a variant of the survivable network design problem [Botton, 2010, Chapter 2], where only an alternative proof of one of the original results by Janssen and Kilakos [1999] is given.

We remark, for example, that variations of fundamental combinatorial optimization problems with a fixed cardinality constraint are surveyed by Bruglieri et al. [2006]. Still, the authors do not mention stable sets, in spite of the major role played by that structure throughout the development of polyhedral combinatorics. There are a few sentences in the survey about k-cardinality cliques and k-partitioning, problems that do have connections to stable sets, but the amount of references for those problems does not exceed what we find about KSTAB. More importantly, the fact that problems transform to each other does not mean we are not interested in each of them and in the different geometric structures they define, nor that research and development in one case has a useful translation to the other.

In our research project, we first conceived studying kstabs as a relaxation of stable spanning trees, introduced in the next section. In face of such a lack of results on algorithms, combinatorics and optimization for kstabs, we decided to devote the first part of our work to claim attention to such an interesting structure.



$$\mathcal{C} = \{\{e_1, e_5\}, \{e_1, e_6\}, \{e_7, e_8\}, \{e_8, e_9\}, \{e_2, e_{10}\}, \{e_{10}, e_{12}\}, \{e_3, e_{11}\}, \{e_4, e_{11}\}\}$$

Figure 2.2: The subset given by edges in bold induce a stable spanning tree in the graph to the left. The same subset induces a kstab in the conflict graph, to the right.

2.2 Stable spanning trees

Let G be a graph and $\mathcal{C}$ be a family of unordered pairs of edges, that are regarded as being in conflict. A *stable* (or *conflict-free*) spanning tree in G is a set of edges T inducing a spanning tree in G, such that for each $\{e_i, e_j\} \in \mathcal{C}$, at most one of the edges e_i and e_j is in T.

Since its inception by Darmann et al. [2009, 2011], the study of stable spanning trees explores the associated *conflict graph* $H \stackrel{\text{def}}{=} (E(G), \mathcal{C})$, with a vertex corresponding to each edge of the original graph G, and where we represent each conflict constraint (*i.e.* an entry in $\mathcal{C}$) by an edge connecting the corresponding vertices in H. Note now that each conflict-free spanning tree in G is a subset of $E(G)$ inducing both a spanning tree in G and a stable set in H. In particular, since the number of edges in a spanning tree of G is exactly $|V(G)| - 1$, a stable spanning tree actually induces a *kstab* in H, with $k = |V| - 1$.

See Figure 2.2 for an example, where G is the graph depicted to the left, and $\mathcal{C} = \big\{\{e_1, e_5\}, \{e_1, e_6\}, \{e_7, e_8\}, \{e_8, e_9\}, \{e_2, e_{10}\}, \{e_{10}, e_{12}\}, \{e_3, e_{11}\}, \{e_4, e_{11}\}\big\}$ is the given set of edges declared to be in conflict. One stable spanning tree in G is determined by the set $\{e_2, e_3, e_4, e_5, e_6, e_7, e_9, e_{12}\}$, which induces a spanning tree in G, and a stable set in the conflict graph $H = (E(G), \mathcal{C})$ depicted to the right.

The stable spanning tree polytope of G is the convex hull in $\mathbb{Q}^{|E(G)|}$ of incidence vectors of stable spanning trees in G. Note that this polytope is contained in the polytope $\mathfrak{C}(H,k)$ of kstabs in H, with $k = |V(G)| - 1$.

Suppose now that we are also given edge weights $w : E(G) \to \mathbb{Q}$. The combinatorial optimization problem we are interested in is the *minimum spanning tree under conflict constraints* (MSTCC) problem, or simply *stable spanning tree* (SST) problem, for conciseness. Our task is to determine a stable spanning tree in G of least weight, or decide that no such tree exists in the graph.

Minimum Weight Stable Spanning Tree (SST) Problem

Instance: a finite, simple, undirected graph G; a collection $\mathcal{C}$ of unordered pairs of edges; and an edge weight function $w : E(G) \to \mathbb{Q}$.

Task: determine a minimum weight spanning tree $(V(G), T)$ of G which does not include both elements of any pair in $\mathcal{C}$, or decide that none exists.

Contrary to the classical problem of finding a minimum spanning tree, for which we have efficient algorithms available and the complete characterization of the spanning tree polytope by facet-defining inequalities (one of the cornerstones of polyhedral combinatorics settled by Edmonds [1971]), finding an SST is a remarkably hard problem, in general. Darmann et al. [2009, 2011] proved that it is strongly NP–hard, even when every connected component of the conflict graph is a path of length two. Furthermore, it cannot be approximated by a constant factor of the optimal value, assuming P $\neq$ NP.

Conceptual model and applications

Stable spanning trees give a model for problems needing both a *minimally connected* substructure, and the concept of *independence* provided by a cardinality k stable set (namely, that of *conflict-free* choices). The conflict graph where a KSTAB ought to be found isolates the independence aspect, and the original graph on which we shall find a stable tree isolates the connectivity options for a solution.

It is natural in many settings to conceive generalizations of an arbitrary (classical) spanning tree problem where pairwise incompatible edges are included. For instance, technology limitations in a heterogeneous network, multiple choices of transportation means in a freight distribution system, phylogenetic trees with an underlying model assuming two species or branches as incompatible.

The author succeeded, at least in a classroom setting, to have the audience judge as interesting the task of finding optimal SSTs in the following scenario. Suppose that the football cup in a country known for its violent associations of fans (also known as *ultras*) reaches its knockout phase, with games taking place during the following Sundays. To mitigate riots and brawls when fans travel to the stadiums, the national police and the state railway set up reduced offer in the public transport system in game days, with strengthened security. Each connection has a known cost corresponding to the number of police agents required to maintain order on that route. The national security agency provides the information of pairs of connections deemed *unstable*: the number of fans from rival teams expected in two such connections is so large that the security might be compromised. A stable spanning tree here determines a minimal selection of active connections which still allows people to travel between any two cities, while no two unstable connections are included.

Stable spanning trees also appear as a feasibility subproblem in the context of earlier network design applications modelled as a *quadratic bottleneck MST* problem, where the objective function accounts for the interaction of pairs of edges [Punnen and Zhang, 2011]. Although we could not verify their statement, Carrabs and Gaudioso [2021] go as far as to mention *practical applications* of SST, *e.g.* in the design of offshore wind farm networks, where a minimum cost layout connecting turbines should be determined, while avoiding overlap of cables.

Literature overview

Exact algorithms to find stable spanning trees had been investigated for more than a decade before we took it in this research project. The main approaches to find optimal SST explore Lagrangean relaxation strategies [Zhang et al., 2011; Carrabs and Gaudioso, 2021], as well as valid inequalities and branch-and-cut algorithms [Carrabs et al., 2021; Samer and Urrutia, 2015]. As it is relevant to understand how our work fits in this landscape, we give a broad perspective of the contributions from each of those works.

Zhang et al. [2011] introduced a number of algorithms for SST. While two Lagrangean relaxation schemes stand out as more relevant to us, they also present particular cases that can be solved in polynomial time, feasibility tests for preprocessing, and heuristics. The authors also introduced the first (and the most relevant, to this date) set of benchmark instances and discuss computational results for their algorithms. Nevertheless, considerably large optimality gaps are reported from these algorithms.

Samer and Urrutia [2015] introduced a preprocessing method and a branch-and-cut algorithm using classes of valid inequalities from the (classical) stable set polytope of the conflict graph. Odd-cycle inequalities are separated exactly, while maximal clique inequalities (which are NP-hard to separate in general) could be identified *a priori* with an efficient maximal clique enumeration algorithm. They report consistent improvements over the previously available computational results.

Carrabs et al. [2021] introduce a simple class of valid inequalities, together with the corresponding separation heuristic. They include the new separation routine in a *partial* reimplementation of the algorithm by Samer and Urrutia [2015]. While they observe minor improvements in the original benchmark instances, the authors introduce a new benchmark set, where their contribution is more expressive.

Carrabs and Gaudioso [2021] presented results from a new implementation based on the first relaxation scheme of Zhang et al. [2011] dualizing all conflict constraints. They design a combination of dual ascent and a subgradient method to compute Lagrangean bounds, and report extensive numerical results, including a comparison with their evolutionary heuristic.

An important observation upon which we base our work in the SST problem is a careful critique of previous Lagrangean approaches for this problem; see Section 3.1.

The literature of SST also includes references entirely devoted to heuristics [da Silva Barros et al., 2023], or to fine-grained complexity results [Viana et al., 2021; Barros et al., 2022].

We finally remark that conflict graphs have been used in different contexts in integer programming. General-purpose MILP solvers use a conflict graph to represent logical relations among variables, both in a preprocessing phase and across nodes of the branch-and-bound tree [Atamtürk et al., 2000]. Conflict graphs are also used to exploit SAT conflict analysis techniques towards generating cutting planes from pruned nodes in the enumeration tree [Achterberg, 2007].

2.3 Weighted connected matchings

Recall that a set of edges is a *matching* (or an *independent edge set*) in a graph if they are pairwise non-adjacent. A P-matching, in turn, consists of a matching M such that the subgraph induced by V_M (the set of vertices covered by M) has some property P. We thus refer to a *connected matching* if V_M induces a connected subgraph, an *acyclic*

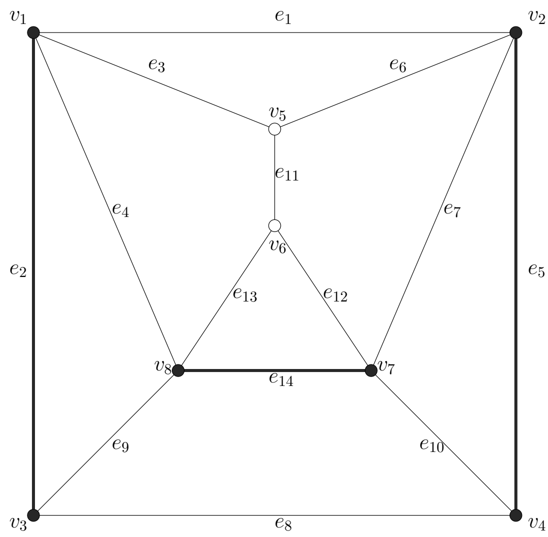


Figure 2.3: Example of a connected matching.

matching if V_M induces an acyclic subgraph, an *induced matching* if V_M induces again the matching M, *etc.*

We concentrate in the property of induced connectivity in this context, and the third, fundamental structure investigated in the present thesis is namely that of a connected matching (CM). For an example, consider the graph illustrated in Figure 2.3, and let $M = \{e_2, e_5, e_{14}\}$. The edges of M are independent, and the subgraph induced by $V_M = \{v_1, v_2, v_3, v_4, v_7, v_8\}$ is connected – note that edges $e_1, e_4, e_7, e_8, e_9, e_{10}$ are induced by V_M. So M is a connected matching in this graph.

We highlight that the name *connected matching* has also been used earlier, *e.g.* by Cameron [2003], to denote a matching M in G such that any two edges in M are actually adjacent (joined by an edge of G). We choose to adhere here to the terminology of more recent papers, concerning induced connectivity [Goddard et al., 2005; Gomes et al., 2023].

Finding a largest P-matching is a computationally hard problem for many properties P. Interestingly, Goddard et al. [2005] proved that the complexity of finding a *maximum cardinality connected matching* is the same as that of the classical matching problem – hence polynomially bounded by a function of n. See also [Gomes et al., 2023, Theorem 2] for a simpler proof and a linear time algorithm to find a maximum cardinality CM from a maximum cardinality matching.

If we now consider an edge-weighted graph, finding a CM of maximum total weight is an NP-hard problem even in very restricted graph classes, opening up a number of research directions around CM. One such direction is to study its fine-grained complexity, as done by Gomes et al. [2022] for a number of graph classes. In the remainder of the present thesis, we take the broader perspective instead of exploring a restricted input, and refer to the maximum weight connected matching (WCM) problem in a general graph.

Maximum Weight Connected Matching (WCM) Problem
Instance: a finite, simple, undirected graph G; and an edge weight function $w : E(G) \to \mathbb{Q}$.
Task: determine a maximum weight matching M of G whose covered vertices V_M induce a connected subgraph of G.

Accordingly, we refer to the *connected matching polytope* $\mathfrak{C}(G)$ of a general graph: the convex hull in $\mathbb{Q}^{|E(G)|}$ of incidence vectors of CMs in G.

Conceptual model and applications

Connected matchings give a balanced model of *independence* and *connectivity*. We now seek to find *independent edges* in a graph – links between disjoint pairs of vertices – while requiring that the covered vertices could still be navigated along a single connected component. The level of connectivity of the resulting structure should be naturally higher than in the KSTAB and SST examples, unless the particular instance at hand is excessively sparse.

As argued before for KSTAB and SST, it is natural to conceive extensions of an arbitrary (classical) matching application, where the matched pairs should not be disconnected, *e.g.* pickup and delivery points that are close enough (matchable) and induce a short vehicle route, an enterprise resource planning system for pairing software-engineers with complimentary skills or affinities, while still leading to a geographically close group.

Literature overview

As we have already indicated earlier in this section, the study of weighted P-matching problems, more generally, gains momentum in recent literature. Some examples include maximum weight induced matchings [Panda et al., 2020; Klemz and Rote, 2022] and acyclic matchings [Fürst and Rautenbach, 2019]. We remark, however, that the study of induced matchings, for example, dates back to Stockmeyer and Vazirani [1982].

Besides introducing the WCM problem, Gomes et al. [2022, 2023] establish its NP-hardness even when we assume that the input is

- a planar graph of maximum vertex degree 3 and edge weights in $\{-1, 1\}$, or
- a planar bipartite graph with edge weights in $\{0, 1\}$, or
- a star-like graph (that is, a chordal graph whose clique tree is a star) with edge weights in $\{-1, 1\}$.

On the positive side, they show that WCM is solvable in polytime when the input graph is a tree, or it has maximum degree at most 2, or if it is chordal and edge weights are non-negative.

Our work in this project seeks to explore the WCM problem from the perspective of polyhedral combinatorics. We propose the first polyhedral results on the CM polytope $\mathfrak{C}(G)$, integer programming algorithms, as well as the first computational results, hoping to define a solid foundation towards progress in the actual computation of WCM in general graphs.

Chapter 3

Scientific contributions and concluding remarks

The present chapter concludes the brief overview of our research project, as well as Part I of the thesis.

First, we review our main findings in Section 3.1. Then, we proudly compile all the imagined connections, open research questions, and suggestions for further research in Section 3.2. In our humble opinion, there are a few low hanging fruits there. If the author succeeds in the present endeavour, that section will remain as the most important one of this thesis.

An additional remark on applications and relevance for ICT

Although we already included a warning about detailed applications being out of scope in our research project, we are now in a better position to add an inspiring piece of information.

A combinatorial structure very close to SST is that of conflict-free matchings. Introduced together with the SST problem in the purely theoretical paper by Darmann et al. [2011], a practical application of that structure was reported in the thesis by Engels [2011], that investigated models for a freight car distribution problem arising in the logistics office of the German railway system (*Deutsche Bahn Schenker Rail*).

In some sense, it is reassuring to be reminded that even a *small epsilon* like our contributions in this thesis might add up to a relevant step towards new technologies and improved services to society.

3.1 A bird's-eye view of our contributions

Aiming to avoid a repetition of the summaries in our scientific articles included in Part II of this thesis, we limit ourselves here to a broader view of how the research community might perceive the contributions in each of our papers.

Paper on the KSTAB problem, in Chapter 4

This is a purely theoretical paper, although we do outline algorithms. The main goal in preparing that manuscript was to present the first results across a range of research lines in ACO concerning kstabs, and hence to claim attention to such an interesting problem, in our opinion.

We indicate that the trivial LP relaxation does not behave as nicely as the corresponding one in the classical stable set problem. On a positive direction, we propose the class of *unsuitable neighbourhood inequalities* (UNI) for the kstab polytope. Nevertheless, the author did not succeed at the time to find a separation algorithm, or to show sufficient conditions for UNI to define facets.

In terms of algorithms, we discuss an efficiently computable dual bound (via matchings in an auxiliary graph), and identify particular cases of restricted input that cast the optimization problem KSTAB in P.

Paper on the SST problem, in Chapter 5

This paper includes both theoretical and practical results. The main idea is to explore kstabs as a relaxation of the SST problem. We extend our study of the kstab polytope, and determine a lower bound on its dimension – which might be a useful ingredient in proving that new classes of valid inequalities are facet-defining.

Most of the work concentrates on the design and evaluation of a new reformulation based on Lagrangean Decomposition (LD). We make strides in arguing that the existing Lagrangean relaxation schemes have serious drawbacks, including wrong results published in related literature.

After developing the new LD scheme from scratch, exploring a kstab subproblem, and determining a careful initialization procedure with dual ascent steps, we introduce our implementation of the resulting solver. The free, open-source code of the complete algorithms is available in the LD-davol repository on GitHub:

`https://github.com/phillippesamer/stable-trees-ld-davol`

The computational results of both the first KSTAB relaxation (just a combinatorial bound) and the stronger LD bounds that we propose in this paper were rather encouraging, confirming the impression that a more judicious Lagrangean approach for the SST problem should be devised.

Paper on the WCM problem, in Chapter 6

This is a more practical paper, whose main goals are (i) to tackle the WCM problem from the perspective of polyhedral combinatorics and integer programming, and (ii) to report promising results about the WCM problem *in general graphs*.

We propose both a compact extended formulation, and an exponential one in the original space of the CM polytope $\mathfrak{C}(G)$. The latter leads to a sophisticated branch-and-cut scheme, with several separation routines to find blossom inequalities from the classical matching polytope, as well as minimal separator and indegree inequalities from the connected subgraph polytope.

We report rather encouraging numerical results using input examples from a collection of DIMACS implementation challenge benchmark instances. Again, the free, open-source implementation of all algorithms is made available:

`https://github.com/phillippesamer/wcm-branch-and-cut`

Paper on the CM polytope, in Chapter 7

This short, purely theoretical paper is dedicated to presenting a new class of $O(m^2)$ facet-defining inequalities for the CM polytope, identified by inspecting $\mathfrak{C}(G)$ with polymake [Gawrilow and Joswig, 2000; Assarf et al., 2017]. We illustrate how the new facets may dominate minimal separator inequalities, and thus strengthen formulations for WCM.

The paper contents are essentially the facet proof. We also take the opportunity there to publish our simple software tool to assist further inspection of the CM polytope with polymake, available at:

`https://github.com/phillippesamer/wcm-branch-and-cut/tree/main/polyhedra`

3.2 Connections and further research directions

We highlight next some connections between the research problems and open questions that arose along our investigation, which had to be deemed interesting for *future* work. Additional suggestions are found in the specific papers attached in Part II.

1. **On the strength and separation of UNI for KSTAB polytope**. One of earliest results in our overall project, the thorough study of unsuitable neighbourhood inequalities requires establishing conditions for them to define proper faces of the KSTAB polytope, if they might be facet-defining, as well as studying the corresponding separation problem.

   We stick to the conjecture that this separation problem is NP-hard, and hence would proceed to the design of separation heuristics. In particular, it seems promising to adapt construction and local search methods from the classical dominating set literature in light of the formulation in terms of *q-quasi dominating sets*.

   It also held in our firm belief that sticking to the methodology with which we later investigated the connected matching polytope, and adapting our polymake tool, leads to a low-hanging fruit in this research landscape.

2. **Unified result about weighted KSTAB in perfect graphs**. Most of our results on particular cases of KSTAB that are solvable in polytime concern subclasses of perfect graphs; most are even restricted to cographs. We should expect a complexity dichotomy in terms of perfect graphs.

   As a step in that direction, the author was actually able to prove later (but does not intend to publish) the more general result for chordal graphs. It suffices to note that a chordal graph can have only linearly many maximal cliques, and use the technique of bounding the size of *bags* in a tree-decomposition to derive a polytime dynamic programming algorithm.

3. **Update the implementation of the state-of-the-art solver for classical stable sets**. For a decade now, the author insists in recommending and praising the readable text of Steffen Rebennack on polyhedra and branch-and-cut algorithms for the classical stable set problem [Rebennack et al., 2012]. More than instructive, his work included the complete implementation (separation routines, branching strategies, preprocessing methods, heuristics) of the best-performing algorithm, after rounds of computational experiments over DIMACS challenge instances. We believe it would be a great contribution for the community to update that implementation, and make it widely available; something that the author regrets not being able to pursue during the study of KSTAB and SST.

4. **Characterize the dimension of the KSTAB polytope**. Our result that the dimension of the KSTAB polytope of H is at least $\alpha(H)-1$ might be helpful in proving that some inequality is facet-defining by presenting enough affinely independent points in the corresponding face. Nevertheless, it could have a considerable practical impact (besides the interest in its own right) to either identify or rule out any missing implied equalities in our formulations for the KSTAB and SST.

5. **Lagrangean heuristic for SST based on KSTAB**. Our implementation of the LD algorithm for SST, using the COIN-OR Vol framework, includes a method stub to use the fractional primal solutions (which approximate the corresponding dual optimal ones) in the design of a Lagrangean heuristic. We strongly believe in the effectiveness of extending our method with a repair heuristic, coupled with local search operators exploring neighbourhoods of solutions from both Lagrangean subproblems (KSTAB and classical minimum spanning tree). As indicated in our paper, Carrabs and Gaudioso [2021] had some success from this approach, albeit their integral relaxation scheme.

6. **Further study of our facets for the CM polytope**. For no other reason than the timeframe constraint of our project, we left the experimental evaluation of including our facet-defining inequalities in the MILP formulations to find WCM for future work. The crucial step here is devising an efficient procedure to identify which pairs of edges give a valid inequality.

   We also had partial progress in determining that the 2-connectivity condition in the corresponding theorem statement is actually necessary for the inequality to induce a facet. The interested reader should inspect the CM polytope of graph P_6 (the path on six vertices): numbering coordinates along the path edges, we find that our inequality $x_1 + x_5 - x_3 \leq 1$ is dominated by $x_1 + x_5 - x_3 + \mathbf{x_2} \leq 1$ and $x_1 + x_5 - x_3 + \mathbf{x_4} \leq 1$.

7. **Alternative formulations for WCM**. Upon closer study of the literature around the problems of maximum weight connected subgraph, Steiner trees, and induced connectivity more generally, one concludes that the practical evidence (especially, runtime of resulting MILP solvers) does not agree with the theoretical intuition given by inclusion of different polyhedral relaxations. In particular, the recent work of Rehfeldt et al. [2022] proves that the LP bound from the separators relaxation for induced connectivity (*i.e.* the formulation with vertex variables only) is weaker than earlier alternatives based on edge variables. That is in stark contrast to the success of the approach with vertex variables for maximum weight connected subgraphs [Álvarez-Miranda et al., 2013; Wang et al., 2017] and different Steiner tree problems [Fischetti et al., 2017]. In particular, the latter presented the praised solver that won most of the categories at the 11th DIMACS Implementation Challenge [DIMACS'11].

In our ILP formulation for weighted connected matchings in G using only natural design variables $x \in \{0,1\}^{|E(G)|}$, namely that on which the branch-and-cut algorithm is based, we chose the second approach above to capture induced connectivity, projecting the minimal separator inequalities onto the space of edge variables using $y_u \stackrel{\text{def}}{=} \sum_{e \in \delta(u)} x_e$. While this led to encouraging computational results, it does bring about the question of comparing the separators-based formulation with standard cutset ones, both theoretically and in practice. Again, our timeframe constraint determined that contemplating such questions should be pursued in future work.

Additionally, it is worth considering an alternative extended formulation based on *multicommodity* flows. Adjusting our compact formulation in a straightforward way, it might be possible to attain stronger LP relaxation bounds. If the computational overhead of a larger model is not excessive, an improved compact extended formulation may be readily available.

8. **Comparing the extended formulation and the separators relaxation for WCM**. It should be possible to determine if $\text{proj}_{\mathbf{x}} \mathcal{P}_{\text{ext}}(G)$, *i.e.* the projection of our extended formulation for connected matchings in the original space, is contained in the MSI relaxation $\mathcal{P}_{\text{sep}}(G)$, *i.e.* the polyhedron resulting from our exponential formulation after dropping blossom and indegree inequalities. It would not be surprising if they are actually equivalent.

   We conjecture that the first inclusion could be shown using a maxflow-mincut argument: assuming that some (a,b)-separator inequality is violated and showing that some vertex in that cutset is violating the flow balance constraint in the extended formulation. For the other inclusion, we expect that taking an arbitrary orientation of edges (*e.g.* from v_i to v_j if $i < j$) and using the equations in the extended formulation should allow one to determine a feasible flow.

9. **Primal heuristics exploring combinatorial results**. We either derived or used several combinatorial properties for each of the KSTAB, SST, and WCM problems towards proving polyhedral results, determining dual bounds, and devising exact optimization algorithms. A practitioner's perspective on our work invokes the necessity of improved methods for finding primal bounds, possibly building on such properties. In our defence, we hope that said practitioner's needs would be met by the computational power of current MILP solvers, at this stage. We thus leave all the promising research tracks of devising construction heuristics, solution neighbourhoods and their efficient exploration in local search methods, and the design of matheuristics for further research on those problems.

   The exciting results of Luteberget and Sartor [2023], who won the *MIP 2022 Computational Competition* on the design of LP-free MIP heuristics might be a promising starting point in this line, given the excellent performance of their free, open-source, reasonably small code (less than a thousand lines). Alternatively, it should be productive to design operators exploring subproblems in SST and WCM and experiment with the `alns` python package [Wouda and Lan, 2023] to quickly develop heuristics based on Adaptive Large Neighbourhood Search.

10. **Evaluate the argument for LD on WCM**. A key idea in our work on SST is the Lagrangean Decomposition technique based on (computationally tractable) NP-hard subproblems. The author is keen to learn to what extent we could explore that idea in enhanced solution methods for weighted connected matchings. In particular, a promising algorithm would be to convexify the maximum weight connected subgraph component in a Lagrangean subproblem, which would be solved by the award-winning solver from the 11th DIMACS Implementation Challenge [DIMACS'11] we mentioned earlier, as well as the more recent competitor *SCIP-Jack* [Rehfeldt and Koch, 2019, 2023].

11. **Thin formulation for stable perfect matchings using KSTAB**. Stable matchings, in the *conflict-free* sense given by a side conflict graph H such as in the SST problem, have attracted some attention in the past decade. Note that we can encode the classical degree constraints in the original graph as additional conflicts between edges in H, and thus reduce the problem to KSTAB. Different solution strategies for that problem have been discussed, concentrating on tailored branch-and-bound algorithms *e.g.* as reported by Akyüz et al. [2023]. The latter actually discusses the relationship with KSTAB, but fails to acknowledge our work. This leaves the possibility of extending our results, as well as our LD approach for SST, in the context of stable matchings.

12. **Independence systems in SST**. Stable spanning trees have a rich geometric structure: they correspond to the intersection of an independence system (stable sets in the conflict graph H) and a matroid (in particular, bases of the graphic matroid over the original graph G). Aharoni and Berger [2006] started the combinatorial study of the intersection of an independence system and a matroid, using mostly topological methods. Nevertheless, it appears that the algorithmic aspects are still not well-studied and understood, and we should be able to translate that viewpoint to new methods for SST.

13. **Parameterized extension complexity**. It should be fruitful to apply the recent and rapidly developing techniques for determining bounds on the size of extended formulations to our polytopes. An immediate question is to which extent does the negative results on the hardness of approximating the stable set polytope by an extended formulation apply to KSTAB. Bazzi et al. [2019] prove that for all n sufficiently

large, there exist graphs on n vertices such that every LP or even semidefinite programming relaxation of polynomial size for the stable set polytope on those graphs has integrality gap of $\omega(1)$.

Another interesting problem is to design (exact) extended formulations of fixed-parameter tractable (FPT) size complexity. One close example are bounds for FTP extended formulations for vertex cover polytopes parameterized by the solution size, proved by Buchanan [2016].

14. **Algebraic and nonlinear programming connections**. There is ample space to study convex relaxations and techniques in semidefinite optimization in the problems we cover. In particular, we received more than once the suggestion to apply the weighted variant of Lovász theta function (*e.g.* restricted to subgraphs or particular graph classes) to find stronger dual bounds for KSTAB and SST.

    On a different line, it was also suggested that one should explore the correspondence between maximal stable sets and local optima of a quadratic binary formulation [Bomze et al., 1999, Section 2]. It should be possible, *e.g.* by means of graph decompositions, to compute quadratic programming bounds efficiently in this context.

15. **Partition functions in analytic combinatorics**. Given a family $\mathcal{F}$ of subsets of $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$, the *partition function* of $\mathcal{F}$ is the polynomial on n real or complex variables $x_1, \ldots, x_n$ defined as $p_{\mathcal{F}}(x_1, \ldots, x_n) = \sum_{\mathcal{S} \in \mathcal{F}} \Pi_{i \in \mathcal{S}} x_i$. Famous examples are the Potts and hardcore models in statistical physics – better known as Tutte and independence polynomials in graph theory, respectively. It is typically impractical to have an explicit form of $p_{\mathcal{F}}$, but there are important cases where we may actually evaluate efficiently whether some particular set $\mathcal{S}$ is in $\mathcal{F}$, *e.g.* checking Hamiltonian cycles in a graph.

    Chudnovsky and Seymour reported that, in the class of *claw-free* graphs, all roots of the independence polynomial are real [Chudnovsky and Seymour, 2007]. Furthermore, the polynomial can be arbitrarily approximated in quasi-polynomial time, or even polynomial time when the graph has maximum degree bounded *a priori* [Patel and Regts, 2017]. Since the partition function of cardinality k stable sets in H can be evaluated using only local information from $(k-1)$-neighborhoods of the vertices [Patel and Regts, 2017], we conjecture that we can approximate the partition function of stable spanning trees over conflict graph H by the same interpolation method, and derive sufficient conditions for the existence of at least one integer point in the corresponding KSTAB and SST polytopes (as illustrated by Barvinok and Regts [2019]).

Bibliography

T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007. doi: 10.1016/j.disopt.2006.10.006. URL `https://doi.org/10.1016/j.disopt.2006.10.006`.

R. Aharoni and E. Berger. The intersection of a matroid and a simplicial complex. *Transactions of the American Mathematical Society*, 358(11):4895–4917, 2006. doi: 10.1090/S0002-9947-06-03833-5. URL `https://doi.org/10.1090/S0002-9947-06-03833-5`.

M. H. Akyüz, İ. K. Altınel, and T. Öncan. Maximum weight perfect matching problem with additional disjunctive conflict constraints. *Networks*, 81(4):465–489, 2023. doi: 10.1002/net.22139. URL `https://doi.org/10.1002/net.22139`.

E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization: Festschrift for Martin Grötschel*, pages 245–270. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-38189-8_11. URL `https://doi.org/10.1007/978-3-642-38189-8_11`.

B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. Computing convex hulls and counting integer points with polymake. *Mathematical Programming Computation*, 9:1–38, 2017. doi: 10.1007/s12532-016-0104-z. URL `https://doi.org/10.1007/s12532-016-0104-z`.

A. Atamtürk, G. L. Nemhauser, and M. W. Savelsbergh. Conflict graphs in solving integer programming problems. *European Journal of Operational Research*, 121(1):40–55, 2000. doi: 10.1016/S0377-2217(99)00015-6. URL `https://doi.org/10.1016/S0377-2217(99)00015-6`.

B. J. S. Barros, L. S. Ochi, R. G. S. Pinheiro, and U. S. Souza. On conflict-free spanning tree: algorithms and complexity. In N. Balachandran and R. Inkulu, editors, *Conference on Algorithms and Discrete Applied Mathematics, CALDAM 2022, Lecture Notes in Computer Science*, volume 13179, pages 91–102, Cham,

2022. Springer International Publishing. doi: 10.1007/978-3-030-95018-7_8. URL https://doi.org/10.1007/978-3-030-95018-7_8.

A. Barvinok and G. Regts. Weighted counting of solutions to sparse systems of equations. *Combinatorics, Probability and Computing*, 28(5):696–719, 2019. doi: 10.1017/S0963548319000105. URL https://doi.org/10.1017/S0963548319000105.

A. Bazzi, S. Fiorini, S. Pokutta, and O. Svensson. No small linear program approximates vertex cover within a factor 2 - ϵ. *Mathematics of Operations Research*, 44(1):147–172, 2019. doi: 10.1287/moor.2017.0918. URL https://doi.org/10.1287/moor.2017.0918.

D. Bertsimas and R. Weismantel. *Optimization Over Integers*. Dynamic Ideas, 2005. ISBN 978-0975914625.

I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. *Handbook of Combinatorial Optimization: Supplement Volume A. D.-Z. Du and P.M.Pardalos (Eds.)*, pages 1–74, 1999. doi: 10.1007/978-1-4757-3023-4_1. URL https://doi.org/10.1007/978-1-4757-3023-4_1.

J.-A. Bondy and U. S. R. Murty. *Graph theory*. Graduate Texts in Mathematics. Springer, New York, London, 2007. ISBN 978-1-8462-8969-9.

Q. Botton. *Survivable network design with quality of service constraints: extended formulations and Benders decomposition*. PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, September 2010. URL http://hdl.handle.net/2078.1/33300.

M. Bruglieri, M. Ehrgott, H. W. Hamacher, and F. Maffioli. An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344 – 1357, 2006. doi: 10.1016/j.dam.2005.05.036. URL https://doi.org/10.1016/j.dam.2005.05.036.

A. Buchanan. Extended formulations for vertex cover. *Operations Research Letters*, 44 (3):374 – 378, 2016. doi: 10.1016/j.orl.2016.03.008. URL https://doi.org/10.1016/j.orl.2016.03.008.

K. Cameron. Connected matchings. In M. Jünger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization – Eureka, You Shrink! Papers dedicated to Jack Edmonds, 5th International Workshop Aussois (2001), Revised Papers*, pages 34–38. Springer Berlin Heidelberg, 2003. doi: 10.1007/3-540-36478-1_5. URL https://doi.org/10.1007/3-540-36478-1_5.

F. Carrabs and M. Gaudioso. A Lagrangian approach for the minimum spanning tree problem with conflicting edge pairs. *Networks*, 78(1):32–45, 2021. doi: 10.1002/net.22009. URL `https://doi.org/10.1002/net.22009`.

F. Carrabs, R. Cerulli, R. Pentangelo, and A. Raiconi. Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach. *Annals of Operations Research*, 298(1):65–78, 2021. doi: 10.1007/s10479-018-2895-y. URL `https://doi.org/10.1007/s10479-018-2895-y`.

M. Chudnovsky and P. Seymour. The roots of the independence polynomial of a clawfree graph. *Journal of Combinatorial Theory, Series B*, 97(3):350–357, 2007. doi: 10.1016/j.jctb.2006.06.001. URL `https://doi.org/10.1016/j.jctb.2006.06.001`.

B. J. da Silva Barros, R. G. S. Pinheiro, U. S. Souza, and L. S. Ochi. Using adaptive memory in GRASP to find minimum conflict-free spanning trees. *Soft Computing*, 27(8):4699–4712, 2023. doi: 10.1007/s00500-022-07602-x. URL `https://doi.org/10.1007/s00500-022-07602-x`.

A. Darmann, U. Pferschy, and J. Schauer. Determining a minimum spanning tree with disjunctive constraints. In F. Rossi and A. Tsoukias, editors, *Algorithmic Decision Theory*, volume 5783 of *Lecture Notes in Computer Science*, pages 414–423. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-04428-1_36. URL `https://doi.org/10.1007/978-3-642-04428-1_36`.

A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726–1735, 2011. doi: 10.1016/j.dam.2010.12.016. URL `https://doi.org/10.1016/j.dam.2010.12.016`.

R. Diestel. *Graph Theory*. Springer, 5 edition, 2017. doi: 10.1007/978-3-662-53622-3. URL `https://doi.org/10.1007/978-3-662-53622-3`.

DIMACS'11. The Eleventh DIMACS Implementation Challenge, in Collaboration with ICERM: Steiner Tree Problems, 2014. URL `https://dimacs11.zib.de/downloads.html`. Organized by David S. Johnson, Thorsten Koch, Renato F. Werneck, and Martin Zachariasen.

J. Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971. doi: 10.1007/BF01584082. URL `https://doi.org/10.1007/BF01584082`.

B. Engels. *A Generalized Network Model for Freight Car Distribution*. PhD thesis, Universität zu Köln, 2011. URL `http://kups.ub.uni-koeln.de/id/eprint/4262`.

M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out Steiner trees: a node-based model for uniform edge costs. *Mathematical Programming Computation*, 9(2):203–229, 2017. doi: 10.1007/s12532-016-0111-0. URL https://doi.org/10.1007/s12532-016-0111-0.

M. Fürst and D. Rautenbach. On some hard and some tractable cases of the maximum acyclic matching problem. *Annals of Operations Research*, 279:291–300, 2019. doi: 10.1007/s10479-019-03311-1. URL https://doi.org/10.1007/s10479-019-03311-1.

E. Gawrilow and M. Joswig. Polymake: a framework for analyzing convex polytopes. In G. Kalai and G. M. Ziegler, editors, *Polytopes – combinatorics and computation*, pages 43–73. Springer, 2000. doi: 10.1007/978-3-0348-8438-9_2. URL https://doi.org/10.1007/978-3-0348-8438-9_2.

A. M. Geoffrion. Lagrangean relaxation for integer programming. In M. L. Balinski, editor, *Approaches to Integer Programming*, pages 82–114, Berlin, Heidelberg, 1974. Springer Berlin Heidelberg. doi: 10.1007/BFb0120690. URL https://doi.org/10.1007/BFb0120690.

W. Goddard, S. M. Hedetniemi, S. T. Hedetniemi, and R. Laskar. Generalized subgraph-restricted matchings in graphs. *Discrete Mathematics*, 293(1):129–138, 2005. doi: 10.1016/j.disc.2004.08.027. URL https://doi.org/10.1016/j.disc.2004.08.027.

G. C. Gomes, B. P. Masquio, P. E. Pinto, V. F. dos Santos, and J. L. Szwarcfiter. Disconnected matchings. *Theoretical Computer Science*, 956:113821, 2023. doi: 10.1016/j.tcs.2023.113821. URL https://doi.org/10.1016/j.tcs.2023.113821.

G. C. M. Gomes, B. P. Masquio, P. E. D. Pinto, V. F. d. Santos, and J. L. Szwarcfiter. Weighted connected matchings. In A. Castañeda and F. Rodríguez-Henríquez, editors, *LATIN 2022: Theoretical Informatics*, pages 54–70, Cham, 2022. Springer International Publishing. doi: 10.1007/978-3-031-20624-5_4. URL https://doi.org/10.1007/978-3-031-20624-5_4.

J. Janssen and K. Kilakos. Bounded stable sets: polytopes and colorings. *SIAM Journal on Discrete Mathematics*, 12(2):262–275, 1999. doi: 10.1137/S089548019630978X. URL https://doi.org/10.1137/S089548019630978X.

B. Klemz and G. Rote. Linear-time algorithms for maximum-weight induced matchings and minimum chain covers in convex bipartite graphs. *Algorithmica*, 84(4):1064–1080, 2022. doi: 10.1007/s00453-021-00904-w. URL https://doi.org/10.1007/s00453-021-00904-w.

E. Lam and V. Mak-Hau. Branch-and-cut-and-price for the cardinality-constrained multi-cycle problem in kidney exchange. *Computers & Operations Research*, 115: 104852, 2020. doi: 10.1016/j.cor.2019.104852. URL `https://doi.org/10.1016/j.cor.2019.104852`.

B. Luteberget and G. Sartor. Feasibility Jump: an LP-free Lagrangian MIP heuristic. *Mathematical Programming Computation*, 15(2):365–388, 2023. doi: 10.1007/s12532-023-00234-8. URL `https://doi.org/10.1007/s12532-023-00234-8`.

B. Panda, A. Pandey, J. Chaudhary, P. Dane, and M. Kashyap. Maximum weight induced matching in some subclasses of bipartite graphs. *Journal of Combinatorial Optimization*, 40(3):713–732, 2020. doi: 10.1007/s10878-020-00611-2. URL `https://doi.org/10.1007/s10878-020-00611-2`.

V. Patel and G. Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM Journal on Computing*, 46(6): 1893–1919, 2017. doi: 10.1137/16M1101003. URL `https://doi.org/10.1137/16M1101003`.

A. P. Punnen and R. Zhang. Quadratic bottleneck problems. *Naval Research Logistics*, 58(2):153–164, 2011. doi: 10.1002/nav.20446. URL `https://doi.org/10.1002/nav.20446`.

S. Rebennack, G. Reinelt, and P. M. Pardalos. A tutorial on branch and cut algorithms for the maximum stable set problem. *International Transactions in Operational Research*, 19(1-2):161–199, 2012. doi: 10.1111/j.1475-3995.2011.00805.x. URL `https://doi.org/10.1111/j.1475-3995.2011.00805.x`.

D. Rehfeldt and T. Koch. Combining NP-Hard reduction techniques and strong heuristics in an exact algorithm for the maximum-weight connected subgraph problem. *SIAM Journal on Optimization*, 29(1):369–398, 2019. doi: 10.1137/17M1145963. URL `https://doi.org/10.1137/17M1145963`.

D. Rehfeldt and T. Koch. Implications, conflicts, and reductions for Steiner Trees. *Mathematical Programming*, 197:903–966, 2023. doi: 10.1007/s10107-021-01757-5. URL `https://doi.org/10.1007/s10107-021-01757-5`.

D. Rehfeldt, H. Franz, and T. Koch. Optimal connected subgraphs: integer programming formulations and polyhedra. *Networks*, 80(3):314–332, 2022. doi: 10.1002/net.22101. URL `https://doi.org/10.1002/net.22101`.

P. Samer and S. Urrutia. A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41–55, 2015. doi: 10.1007/s11590-014-0750-x. URL `https://doi.org/10.1007/s11590-014-0750-x`.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer, 2003. ISBN 978-3-540-44389-6.

L. J. Stockmeyer and V. V. Vazirani. NP-completeness of some generalizations of the maximum matching problem. *Information Processing Letters*, 15(1):14–19, 1982. doi: 10.1016/0020-0190(82)90077-1. URL `https://doi.org/10.1016/0020-0190(82)90077-1`.

L. Viana, M. Campêlo, I. Sau, and A. Silva. A unifying model for locally constrained spanning tree problems. *Journal of Combinatorial Optimization*, 42(1): 125–150, 2021. doi: 10.1007/s10878-021-00740-2. URL `https://doi.org/10.1007/s10878-021-00740-2`.

Y. Wang, A. Buchanan, and S. Butenko. On imposing connectivity constraints in integer programs. *Mathematical Programming*, 166:241–271, 2017. doi: 10.1007/s10107-017-1117-8. URL `https://doi.org/10.1007/s10107-017-1117-8`.

L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization*, volume 55. John Wiley & Sons, 1999.

N. A. Wouda and L. Lan. ALNS: a Python implementation of the Adaptive Large Neighbourhood Search metaheuristic. *Journal of Open Source Software*, 8(81):5028, 2023. doi: 10.21105/joss.05028. URL `https://doi.org/10.21105/joss.05028`.

R. Zhang, S. N. Kabadi, and A. P. Punnen. The minimum spanning tree problem with conflict constraints and its variations. *Discrete Optimization*, 8(2):191 – 205, 2011. doi: 10.1016/j.disopt.2010.08.001. URL `https://doi.org/10.1016/j.disopt.2010.08.001`.

Part II

Scientific articles

Chapter 4

Fixed cardinality stable sets

Phillippe Samer, Dag Haugland

Fixed cardinality stable sets

The version of record of this article, published in **Discrete Applied Mathematics**, Volume 303, pages 137-148 (2021), is available online at the publisher's website: `https://doi.org/10.1016/j.dam.2021.01.019`

Fixed cardinality stable sets

Phillippe Samer and Dag Haugland

{samer@uib.no, dag.haugland@uib.no}

Universitetet i Bergen, Institutt for Informatikk, Postboks 7800, 5020, Bergen, Norway

July 21, 2020

Abstract

Given an undirected graph $G = (V, E)$ and a positive integer $k \in \{1, \ldots, |V|\}$, we initiate the combinatorial study of stable sets of cardinality exactly k in G. Our aim is to instigate the polyhedral investigation of the convex hull of fixed cardinality stable sets, inspired by the rich theory on the classical structure of stable sets. We introduce a large class of valid inequalities to the natural integer programming formulation of the problem. We also present simple combinatorial relaxations based on computing maximum weighted matchings, which yield dual bounds towards finding minimum-weight fixed cardinality stable sets, and particular cases which are solvable in polynomial time.

Note. A preliminary version of this work appears in the conference proceedings of the 18th Cologne-Twente Workshop on Graphs and Combinatorial Optimization (Samer and Haugland, 2021).

Acknowledgement. This research is partly supported by the Research Council of Norway through the research project 249994 CLASSIS.

1 Introduction

We investigate a problem that is appealing to different research directions around algorithms, combinatorics and optimization. Let $G \stackrel{\text{def}}{=} (V, E)$ be a finite, simple, undirected graph, and denote $n \stackrel{\text{def}}{=} |V|$, and $m \stackrel{\text{def}}{=} |E|$. A stable set (or independent set, or co-clique) in G consists of a subset of pairwise non-adjacent vertices. Given $k \in \{1, \dots, n\}$ and a vertex-weighting function $w : V \to \mathbb{Q}_+$, the k *stable set problem* consists in finding a minimum weight stable set of cardinality k in G, or deciding that none exists. Note that k is also part of the input to this problem; if it were an arbitrary fixed integer, the enumeration and optimization problems over stable sets of that cardinality could be solved in time polynomially bounded by a function of n.

1.1 Our contribution

The main idea of this work is to initiate the combinatorial study of fixed cardinality stable sets, introducing the first results in selected directions. We consider the problem first from the polyhedral standpoint, then we give efficiently computable dual bounds for the optimization problem, and conclude with graph classes where it can be solved in polynomial time. The different angles from which we study the problem are definitely inviting for further research. Indeed, some basic questions about our results are left open in the form of conjectures throughout the text.

To summarize, the contributions of this article include:

1. We draw attention to the fixed cardinality version of a classical structure in combinatorial optimization and graph theory, shedding light on its appeal to different research directions, besides motivating its application in the MSTCC problem.

2. We show in Section 2 that the fixed cardinality stable set polytope is not ($1/p$)-*integer*, for any integer $p > 1$. Thereafter we introduce an exponential class of valid inequalities to that polytope, whose separation problem is interesting in its own right.

3. We describe a combinatorial relaxation of the optimization problem in Section 3,

where lower bounds are calculated using maximum weighted matchings. Given the efficiency of the corresponding algorithm, this technique can be extended as a building block in different solution approaches.

4. We prove in Section 4 that the problem can be solved in polynomial time when the input is restricted to some important graph classes, including cluster, complete multipartite, split, threshold, and line graphs.

1.2 Motivation from conflict-free spanning trees

Our original motivation for considering fixed cardinality stable sets stems from the NP-hard problem of *minimum spanning trees under conflict constraints* (MSTCC). Given a graph $G \stackrel{\text{def}}{=} (V, E)$ and a set of conflicting edge pairs $C \subseteq E \times E$, a *conflict-free spanning tree* in G is a set of edges $T \subseteq E$ inducing a spanning tree in G, such that for each $(e, f) \in C$, at most one of the edges e and f is in T. The MSTCC problem, introduced by Darmann et al. (2011, 2009), asks for such a conflict-free spanning tree of minimum weight.

Different combinatorial and algorithmic results about the MSTCC problem explore the associated conflict graph $H \stackrel{\text{def}}{=} (E, C)$, which has a vertex corresponding to each edge in the original graph G, and we represent each conflict constraint by an (undirected) edge connecting the corresponding vertices in H. Note that each conflict-free spanning tree in G is a subset of E which corresponds both to a spanning tree in G and to a stable set in H. Therefore, one can equivalently search for stable sets in H of cardinality exactly $|V| - 1$ which do not induce cycles in the original graph G.

It is not hard to devise different approaches for studying the MSTCC problem exploring the connection with fixed cardinality stable sets. For the sake of illustration, consider the *relax-and-cut* approach described by Lucena (2005) for the fixed cardinality set partitioning problem. The author of that work developed a Lagrangean framework where dual bounds, heuristics and variable fixing tests are computed as a preprocessing phase, resulting in an easier problem to be handled to an integer programming (IP) solver. Note that the Lagrangean bounds are strengthened by dynamically introducing valid constraints, including those from the exponential family of clique inequalities. Now, an analogue towards conflict-free spanning trees could be described as follows. Given the original graph $G = (V, E)$, the conflict graph $H = (E, C)$, and costs $c \in \mathbb{Q}_+^{|E|}$ on edges of G, denote by $\mathfrak{C} \stackrel{\text{def}}{=} \mathfrak{C}(H, |V| - 1)$ the polytope of stable sets in H which have cardinality equal to $|V| - 1$. Using binary variables $x \in \{0, 1\}^{|E|}$, we recast the MSTCC problem as

$$\min \{cx : Ax \leq b, x \in \mathfrak{C}\}, \tag{1}$$

where the system $\{a_i x \leq b_i\}_{i=1}^m$ corresponds to the subtour elimination constraints (SEC): $\left\{\sum_{e \in E(S)} x_e \leq |S| - 1 : S \subset V, S \neq \emptyset\right\}$. Thus, the number m of inequalities is an exponential function of $|V|$. Dualizing all the SEC, with the introduction of multipliers $\lambda \in \mathbb{R}_+^m$, we have a lower bound to (1) given by the *Lagrangean Relaxation Problem*

$$LRP(\lambda) \stackrel{\text{def}}{=} \min \{(c + \lambda A)x - \lambda b : x \in \mathfrak{C}\}, \tag{2}$$

and the best-possible bound is attained by solving the *Lagrangean Dual Problem*

$$LDP \stackrel{\text{def}}{=} \max \left\{LRP(\lambda) : \lambda \in \mathbb{R}_+^m\right\}. \tag{3}$$

There are two main challenges in this approach. First, the issue of dualizing exponentially many inequalities is dealt with (in a subgradient method) by a clever selection of active constraints among those which are currently or previously violated, while arbitrarily setting to zero the subgradient vector entries corresponding to null multipliers; see (Lucena, 2005, Section 1.2). The second issue is how to optimize over $\mathfrak{C}$, to solve $LRP(\lambda)$ in (2). In order of decreasing generality, we note that:

(i) The obvious relaxation would have been to also dualize edge inequalities in H (that is, $x_u + x_v \leq 1$ for $\{u, v\} \in C$), introducing a new set of Lagrangean multipliers $\mu \in \mathbb{R}_+^{|C|}$, and solving instead the easy problem

$$LRP'(\lambda, \mu) \stackrel{\text{def}}{=} \min \left\{(c + \lambda A + \mu M)x - \lambda b - \mu : \sum_{e \in E} x_e = |V| - 1\right\}, \tag{4}$$

where M denotes the incidence matrix of the conflict graph H.

(ii) If more information on H is available (*e.g.* sparsity, structural decomposition), that could be translated as a better approximation of $\mathfrak{C}$ in the relaxed problem. For instance, if there is a natural decomposition of H into few connected components, one could design instead the special case of a Lagrangean Decomposition, with different primal variables for each component, dualizing the constraints equating the different variables (Guignard, 2003, Section 7). Alternatively, if strong valid inequalities for $\mathfrak{C}$ are known, they could be used towards a relaxation which is between (4) and (2).

(iii) If H belongs to a graph class where the *fixed* cardinality stable set problem becomes solvable in polynomial time, then we can solve problems (2) and (3) as stated above. Note that stronger bounds should follow in this case, since more problem information is embedded in the relaxation. In contrast, even if the classical stable set problem on H can be solved in polynomial time, the MSTCC problem with H as a conflict graph need not be solvable in polynomial time (the original NP-hardness proof of Darmann et al. (2011) makes the further assumption that the conflict graph is a collection of disjoint paths of length 2).

Our presentation of this first relax-and-cut approach for the MSTCC problem is limited to the above outline. We argue that results of different nature from research on the k stable set problem (*e.g.* integer programming formulations and valid inequalities, well-solved particular cases, primal and dual bounds) could provide fundamental components to advance knowledge on the MSTCC problem as well.

1.3 Further related work

It is surprising that the combinatorics and optimization literature has not addressed the *k stable set problem* problem in depth before. Note, for instance, that the thorough survey on fixed cardinality versions of combinatorial optimization problems by Bruglieri et al. (2006) does not mention stable sets, in spite of the major role played by that structure throughout the development of polyhedral combinatorics.

The convex hull of stable sets of cardinality *at most* k was studied by Janssen and Kilakos (1999), but only for $k \in \{2, 3\}$. Apart from that article, it has also appeared as part of an algorithm for a variant of the survivable network design problem (Botton, 2010, Chapter 2), where only an alternative proof of one of the original results by Janssen and Kilakos (1999) is given.

Thin graphs and frequency assignment problems The early work of Mannino et al. (2007) introduces an interesting class of graphs, as well as a cardinality-constrained stable set problem, and their application in the efficient solution of real-world instances of a frequency assignment problem. We explain next the result which is most relevant to our work, and also derive an initial fact about the problem that we study.

Given an ordering $\{v_1, \dots, v_n\}$ of the vertices of a graph $G = (V, E)$, and a partition $V = \biguplus_{i=1}^{k} V_i$, let $\overline{N}(v_j, h)_<$ denote the set of vertices in V_h of order lower than j which are non-adjacent to v_j. The ordering and the partition are called *consistent* if the only vertices in V_h of order lower than j and non-adjacent to v_j are the first $\left|\overline{N}(v_j, h)_<\right|$ ones.

A graph G is *k-thin* if there is such an ordering of the vertices and a partition of V into k classes which are consistent. The *thinness* of a graph is the smallest k such that G is k-thin. Now, if $k = 1$, this gives a characterization of *interval graphs* (those graphs for which an intersection model consisting of intervals on a straight line can be defined). Specifically, $G = (V, E)$ is an interval graph if and only if there exists an ordering $\{v_1, \dots, v_n\}$ which is consistent (with $V = V_1$, the trivial partition).

Then, the authors prove:

Theorem (2.12 in Mannino et al. (2007)). Given $G = (V, E)$, together with an ordering and a partition $V = \biguplus_{i=1}^{k} V_i$ which are consistent, and $d \in \mathbb{Z}_+^k$, a maximum (minimum) weighted stable set S, with $|S \cap V_i| = d_i$ for each $i \in \{1, \dots, k\}$, can be determined in time $O\left(|V| \cdot (\rho + 1)^{k-1} \cdot (1 + \max_{1 \le i \le k} d_i)^k\right)$, where ρ denotes the largest amount of neighbours of lower order that a vertex has in some class of the partition (thus, $\rho \le \Delta(G)$, the largest degree of a vertex in the graph).

Note that they have therefore introduced a different cardinality-constrained version of the stable set problem, more general than the one we study in this work. Thereafter, the excellent computational results on large instances of the frequency-assignment application they describe depend crucially on the efficient solution of this generalized problem on a $|H|$-thin conflict graph, where H is a special set of transmitters in the input.

Finally, we remark that setting $k = 1$, although not interesting in their application, gives an initial result for our problem of interest. For $k = 1$, the consistent ordering in the above theorem implies that G is an interval graph, and that an optimal stable set of fixed cardinality d can be found in time bounded by $O\left(n \cdot (1 + d)\right)$, which is in $O\left(n^2\right)$.

Corollary 1. *If G is an interval graph, the problem of finding a minimum-weight stable set of fixed cardinality in G is in* P.

Extension complexity The fixed cardinality stable set polytope also appears briefly in the recent and rapidly developing theory of parameterized extension complexity. This line of research aims to develop bounds on the number of inequalities necessary to describe a given polytope as the projection of a higher dimensional one. While that number can be polynomially bounded (as a function of the number of vertices in the input graph) for a few particular cases of the classical stable set and vertex cover polytopes, some striking negative results show how large that number can be in general.

There are two main categories of such results in the current literature. One is proving the hardness of approximating a polytope by an extended formulation, such as the work of Bazzi et al. (2019), who prove that for all n sufficiently large, there exist graphs on n vertices such that every linear programming (LP) or even semidefinite programming (SDP) relaxation of polynomial size for the stable set polytope on those graphs has integrality gap of $\omega(1)$.

Another category is designing (exact) extended formulations of fixed-parameter tractable (FPT) size complexity. Still on the negative side, it was recently proved by Gajarský et al. (2018) that, regardless of any computational complexity assumptions, the stable set polytope cannot have an FPT extension for all graphs (naturally parameterized by the solution size). On the positive side, the authors show that linear size

FPT extensions do exist for the class of bounded expansion graphs. Even before, it was proved by Buchanan and Butenko (2014) that, when parameterized by the treewidth of G, $\mathbf{tw}(G)$, the extension complexity of the stable set polytope on G is in $O(2^{\mathbf{tw}(G)}n)$. Afterwards, Buchanan (2016) proved bounds for FTP extended formulations for vertex cover polytopes parameterized by the solution size. Interestingly, in a lemma towards his main result, the author proves that the fixed cardinality stable set polytope for graphs of largest degree at most 2 is given by edge and odd-cycle inequalities alone.

2 Polyhedral results

For any graph G, we denote by $V(G)$ and $E(G)$ the sets of vertices and edges of G, respectively. For conciseness, we abbreviate 'stable set of cardinality k' as k-stab. The family of all k-stabs in G is denoted $\mathcal{F}(G,k)$. Recall that the incidence vector of $S \subset V(G)$ is $\chi^S \in \{0,1\}^{|V(G)|}$ defined by $\chi^S_i = 1$ if and only if $v_i \in S$; so the central object of our interest is $\mathfrak{C}(G,k) \stackrel{\text{def}}{=} \text{conv}\left\{\chi^S : S \in \mathcal{F}(G,k)\right\}$, *i.e.* the convex hull of incidence vectors of all the k-stabs in G.

The natural integer programming (IP) formulation for minimum-weight k-stabs in G is

$$z \stackrel{\text{def}}{=} \min \left\{ \sum_{v \in V(G)} w(v) x_v : \mathbf{x} \in \mathcal{P}(G,k) \cap \{0,1\}^{|V(G)|} \right\}, \tag{5}$$

where $\mathcal{P}(G,k)$ denotes the polyhedral region defined by:

$$\sum_{v \in V(G)} x_v = k, \tag{6}$$

$$x_u + x_v \leq 1 \qquad \text{for each } \{u,v\} \in E(G), \tag{7}$$

$$0 \leq x_v \leq 1 \qquad \text{for each } v \in V(G). \tag{8}$$

Constraints (7) are known as *edge inequalities*, imposing that no two adjacent vertices belong to the selection in $\mathbf{x}$. Together with bounds (8), they determine the *fractional stable set polytope* (Schrijver, 2003, Section 64.5).

Recall that a vector z is *half-integer* if $2z$ is integer (more generally, we say that z is $\frac{1}{p}$*-integer* if pz is integer). A classical result of Nemhauser and Trotter (1974) shows that the *fractional stable set polytope* is half-integer, *i.e.* all its vertices are $\{0, \frac{1}{2}, 1\}$-valued. Since that is the starting point for a series of both polyhedral and algorithmic advances, one could ask whether that result holds for $\mathcal{P}(G, k)$ as well. Unfortunately, we discovered the negative answer to an even broader question, as we show next.

Theorem 2. *For each $p \geq 2$ and each $k \geq 2$, there exists a graph G such that $\mathcal{P}(G, k)$ is not $\frac{1}{p}$-integer.*

Proof. Given $p \geq 2$ and $k \geq 2$ arbitrary, we determine $n \in \mathbb{Z}_+$, a graph G on n vertices, and a convenient point $z \in \mathbb{R}^n$ such that z is a vertex of the polyhedron $\mathcal{P}(G, k)$ which is not $\frac{1}{p}$-integer.

First, we choose $n = n(p, k)$ such that the point $z \stackrel{\text{def}}{=} (\overbrace{1/p+1\ , \ \cdots\ , 1/p+1}^{n-1 \text{ entries}}\ , \ p/p+1)$ satisfies the equality constraint (6): $\sum_{v \in V} x_v = k$. That is,

$$z_1 + \ldots + z_n = \frac{1}{p+1} + \ldots + \frac{1}{p+1} + \frac{p}{p+1} = (n-1)\frac{1}{p+1} + \frac{p}{p+1} = k, \tag{9}$$

and we therefore set $n \stackrel{\text{def}}{=} p(k-1) + k + 1$. Consider next $G \stackrel{\text{def}}{=} S_{n-1} = K_{1,n-1}$, the star on n vertices (illustrated in Figure 1).

We can show that z is a vertex of $\mathcal{P}(S_{n-1}, k) \subset \mathbb{R}^n$ using the equivalence of vertices, basic feasible solutions and extreme points of polyhedra; see *e.g.* (Bertsimas and Tsitsiklis, 1997, Section 2.2). Besides satisfying all equality constraints, a *basic solution* of a polyhedron embedded in $\mathbb{R}^n$ must have n constraints

(i) which are active (equiv. satisfied at equality) at z, and

(ii) whose corresponding vectors in $\mathbb{R}^n$ are linearly independent.

The equality constraint (6) is satisfied by construction of the point z in (9). The graph $G = S_{n-1}$ has an edge $\{v_i, v_n\} \in E(G)$ for each $i \in \{1, \ldots, n-1\}$, and the corresponding edge inequality (7) is active at z: $x_{v_i} + x_{v_n} = \frac{1}{p+1} + \frac{p}{p+1} = 1$. It remains to verify that the

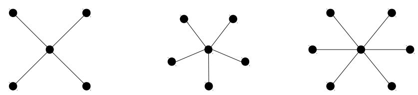


Figure 1: The star graphs S_4, S_5 and S_6.

coefficient vectors of those n constraints are linearly independent. Indeed, arranging the vectors as rows of matrix $A_{n\times n}$,

$$A_{n\times n} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n-1} & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix},$$

it follows that $\det A = (-1)^n(n-2) \neq 0$, and the row vectors are indeed linearly independent. Since z is also feasible (as the bounds $0 < {}^1\!/_{p+1} < {}^p\!/_{p+1} < 1$ are satisfied by construction), it is a basic feasible solution, and thus a vertex of $\mathcal{P}(S_{n-1}, k)$.

Finally, $p \cdot \frac{1}{p+1} = \frac{p}{p+1}$ is not an integer (but a proper fraction), and it is clear that pz is not an integer point. Therefore, the vertex z is not $\frac{1}{p}$-integer, and the result follows. $\square$

2.1 The unsuitable neighbourhood inequalities (UNI)

We introduce next a class of valid inequalities for $\mathfrak{C}(G, k)$, exploring the relationship between k, the size of the neighbourhood $N(S) \stackrel{\text{def}}{=} \{u \in V \backslash S : \exists \{u, v\} \in E \text{ for some } v \in S\}$ of any set $S \subset V$, and how many vertices from S can appear in any k-stab. First, denoting the set of neighbours of a vertex $v \in V$ by $\delta(v)$, that is $\delta(v) = N(\{v\})$, one can immediately observe that no vertex which has too many neighbours to still build a k-stab can be chosen. This gives the following simple preprocessing test.

Proposition 3. *If* $\mathbf{x}$ *is the incidence vector of any k-stab, and* $v \in V$ *is such that* $|\delta(v)| > n - k$*, then* $x_v = 0$.

In an attempt to enforce an algebraic expression that enough vertices are left upon choosing a set $S \subset V$ towards building a k-stab, we introduce a class of exponentially-many constraints, which we refer to as *unsuitable neighbourhood inequalities* (UNI).

Theorem 4. *For each $S \subset V$ such that $1 \leq |S| < k$ and $|N(S)| > n - k$, inequality $\sum_{v \in S} x_v \leq |S| - 1$ is valid for $\mathfrak{C}(G, k)$.*

Proof. From $|S| < k$, it follows that S is not a k-stab in itself. If S were a subset of any k-stab, there should be at least $k - |S|$ vertices left to choose from, while no neighbour in $N(S)$ can be selected towards building a stable set. That is

$$
\begin{aligned}
& n - |S| - |N(S)| \geq k - |S| \ , && \text{for each } S \subset V, 1 \leq |S| < k, \\
\Leftrightarrow \quad & |N(S)| \leq n - k \ , && \text{for each } S \subset V, 1 \leq |S| < k.
\end{aligned}
$$

Since $|N(S)| > n - k$ by hypothesis, S cannot be part of a k-stab. Therefore no incidence vector $\mathbf{x}$ of a k-stab induces the selection of all the vertices in S, and the result follows. □

While Proposition 3 is clearly a special case of Theorem 4, one could ask whether the UNI indeed give a stronger condition. The positive answer follows next.

Theorem 5. *For any graph G and $k > 1$, the UNI imply the condition enforced by Proposition 3 in the description of $\mathfrak{C}(G, k)$, but the converse does not hold.*

Proof. Let $\mathbf{x}$ be a vector satisfying all UNI. The inequalities in Proposition 3 are implied by the UNI with $|S| = 1$. Suppose that $S = \{u\}$ and $|N(S)| = |\delta(u)| > n - k$. Then u cannot be extended to a k-stab and the UNI include $x_u = \sum_{v \in S} x_v \leq |S| - 1 = 0$, which is the condition on the former proposition.

Now the converse does not hold, *i.e.* even if $|\delta(v)| \leq n - k$ for each $v \in V$, the UNI need not be automatically satisfied, as the following counterexample shows (see Figure 2). Consider the graph $G \stackrel{\text{def}}{=} 2P_3$, which consists of two copies of the path graph on 3 vertices put together, so that $n = 6$, and suppose that $k \stackrel{\text{def}}{=} 3$. Since all vertices have degree 1 or 2, it follows that $|\delta(u)| \leq n - k = 3$ for each vertex u. On the other hand, with a test set S consisting of the two vertices of degree 2 in the middle of the paths, we have $1 \leq |S| < k$ and $|N(S)| = 4 > n - k$, thus yielding the unsuitable neighbourhood

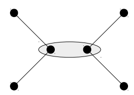


Figure 2: The graph $2P_3$ and the selection of its two central vertices.

inequality given by $\sum_{v \in S} x_v \leq |S| - 1 = 1$ which separates from the convex hull $\mathfrak{C}(G, k)$ any vector selecting those two vertices. $\square$

Proposition 6. *In either of the following two conditions, the corresponding unsuitable neighbourhood inequality is redundant in $\mathfrak{C}(G, k)$: (i) if $S \subset V$ is not independent, or (ii) if $S \subset V$ is not minimal with respect to the condition $|N(S)| > n - k$.*

Proof. If $u, v \in S$ are adjacent vertices, the edge inequality $x_u + x_v \leq 1$ implies $\sum_{v \in S} x_v \leq |S| - 1$. Otherwise, let $S \subset V$ with $1 \leq |S| < k$ and $N(S) > n - k$ be a given independent set, and suppose that $T \subsetneq S$ is such that $|N(T)| > n - k$. The UNI corresponding to T is $\sum_{v \in T} x_v \leq |T| - 1$. Combined with $x_v \leq 1$ for each $v \in S \backslash T$, it implies the UNI corresponding to S, *i.e.* $\sum_{v \in S} x_v \leq |S| - 1$, which is thus redundant in the description of $\mathfrak{C}(G, k)$. $\square$

Recall that the *domination number* $\gamma(G)$ gives the least cardinality of a dominating set in $G = (V, E)$, *i.e.* a subset $D \subset V$ such that every vertex $u \in V \backslash D$ has a neighbour in D. If a lower bound on the domination number of G is known, the following result might be useful.

Proposition 7. *If $\gamma(G) \geq k$, then there exists no UNI for $\mathfrak{C}(G, k)$.*

Proof. Suppose there were $S \subset V$ with $1 \leq |S| < k$ and $|N(S)| > n - k$, and denote $T \stackrel{\text{def}}{=} V \backslash \{S \cup N(S)\}$. Note that any vertex belongs to exactly one among S, $N(S)$, or T; then

$$|S| + |N(S)| + |T| = n \implies |S| + |T| = n - |N(S)| \implies |S| + |T| < n - [n - k] = k,$$

since $|N(S)| > n - k$. Now, $S \cup T$ would be a dominating set of cardinality strictly less than k, contradicting the hypothesis that $\gamma(G) \geq k$. $\square$

On the algorithmic side, it is in general impractical to include *a priori* all minimal UNI in an IP formulation for a black-box solver, since the number of those inequalities may grow exponentially with the size of the input (n, k). The natural approach in this case is to try to cut off successive solutions x^* to a linear programming (LP) relaxation, by finding cutting planes corresponding to UNI violated at x^*, *i.e.* separating x^* from $\mathfrak{C}(G, k)$, or deciding that none exists. Answering that question is known as the *separation problem* for a class of valid inequalities.

Definition 8 (Separation problem for UNI)**.** Given a graph $G = (V, E)$, with $n \stackrel{\text{def}}{=} |V|$, $k \in \{2, \ldots, n-1\}$, and $x^* \in [0,1]^n$ satisfying the conditions that $\sum_{v \in V} x_v^* = k$ and that $x_u^* + x_v^* \leq 1$ for each $\{u, v\} \in E$, determine

i. either a set $S \subset V$, with $1 \leq |S| \leq k-1$ and $|N(S)| \geq n - (k-1)$, such that $\sum_{v \in S} x_v^* > |S| - 1$, in which case the unsuitable neighbourhood inequality corresponding to S separates x^* from $\mathfrak{C}(G, k)$,

ii. or that no such set exists, in which case all UNI are satisfied at x^*.

We give next a slight reformulation of the separation problem which might be useful in future work. Given the input $[G, k, x^*]$ corresponding to Definition 8, define $y^* \in [0,1]^n$ such that $y_v^* \stackrel{\text{def}}{=} 1 - x_v^*$. Note now that $\sum_{v \in S} x_v^* > |S| - 1$ if and only if $\sum_{v \in S} y_v^* < 1$. We thus have the following equivalent statement of the problem.

Definition 9 (Equivalent Formulation of the Separation problem for UNI)**.** Given a graph $G = (V, E)$, with $n \stackrel{\text{def}}{=} |V|$, $k \in \{2, \ldots, n-1\}$, and $y^* \in [0,1]^n$ satisfying the conditions that $\sum_{v \in V} y_v^* = n - k$ and that $y_u^* + y_v^* \geq 1$ for each $\{u, v\} \in E$, determine

i. either a set $S \subset V$, with $|N(S)| \geq n - (k-1)$ and $\sum_{v \in S} y_v^* < 1$, in which case the unsuitable neighbourhood inequality corresponding to S separates $x^* \stackrel{\text{def}}{=} \mathbf{1} - y^*$ from $\mathfrak{C}(G, k)$,

ii. or that no such set exists, in which case all UNI are satisfied at $x^* \stackrel{\text{def}}{=} \mathbf{1} - y^*$.

We consider this statement of the problem to be particularly appealing. Note that if S has size exactly $k-1$, then $|N(S)| \geq n-(k-1)$ implies that it would be a dominating set. Given the condition that adjacent vertices have y^* values summing up to at least 1, and that we require $\sum_{v \in S} y_v^* < 1$, we would actually have an *independent dominating set* if $|S| = k-1$, *i.e.* a subset of vertices which is both dominating and independent (stable). Now, allowing $|S| \leq k-1$ means that there might be $q \in \{0, 1, \ldots, k-2\}$ vertices neither in S nor dominated by it. If we define a *q-quasi dominating set* in a graph $G = (V, E)$ to be a subset of vertices which is dominating in $G[V \backslash X]$, for some $X \subset V$, $|X| \leq q$, our separation problem corresponds to finding a $(k-2)$-quasi dominating set of weight at most 1, or deciding that none exists. (Recall that, for any graph G and $U \subset V(G)$, the *induced subgraph* $G[U]$ is a graph with vertex set U and all of the edges in $E(G)$ which have both endpoints in U.)

We leave the open question of establishing the complexity of that problem.

Conjecture 1. *The separation problem for UNI is* NP*-hard.*

2.2 UNI separation with MIP heuristics

We discuss next an alternative to actually use the UNI in a branch-and-cut solver. This part of the text is only interesting under the assumption that the above conjecture is true.

Besides the natural strategies of designing separation heuristics or including *a priori* some UNI corresponding to sets S of small cardinality, it might prove useful to explore an IP formulation of the separation problem. One can actually use good but not necessarily optimal solutions to that auxiliary IP, which give very effective cutting planes, for instance, in the context of an example of optimizing over the first Chvátal closure (Bertsimas and Weismantel, 2005, Section 5.4). Most MIP solvers include a collection of general purpose heuristics to accelerate the availability of integer feasible solutions, like local branching, feasibility pump and neighbourhood diving methods; see Hanafi and Todosijević (2017) for a recent survey.

The following is described in light of Definition 9, with input $[G, k, \mathbf{y}^*]$. We suppose further that the input is preprocessed by the reduction rules:

(i) Remove any vertex v such that $y^*_v = 1$

(ii) Remove isolated vertices

Those operations do not change the problem answer, since a UNI is automatically satisfied if it contains a vertex with $y^*_v = 1$, and since isolated vertices are not contained in a minimal set S corresponding to a UNI.

For each $v \in V(G)$, let variables $z_v \in \{0,1\}$ be such that $z_v = 1$ if and only if $v \in S$, and $w_v \in \{0,1\}$ be such that $w_v = 1$ if and only if $v \in N[S] = S \cup N(S)$, the *closed neighbourhood* of $S \subset V(G)$. Then, we have to determine

$$\rho = \min \left\{ \sum_{v \in V(G)} y^*_v \cdot z_v : (\mathbf{z}, \mathbf{w}) \in \mathcal{P}_{\mathrm{UNI}}(G, \mathbf{y}^*) \cap \{0,1\}^{2|V(G)|} \right\}, \tag{10}$$

where $\mathcal{P}_{\mathrm{UNI}}(G, \mathbf{y}^*)$ denotes the polyhedral region:

$$\sum_{v \in V(G)} (w_v - z_v) \geq n - (k-1), \tag{11}$$

$$z_u \leq w_v \qquad \text{for each } u \in V(G), \text{ and each } v \in N[u], \tag{12}$$

$$\sum_{u \in N[v]} z_u \geq w_v \qquad \text{for each } v \in V(G), \tag{13}$$

$$z_u + z_v \leq 1 \qquad \text{for each } \{u,v\} \in E(G), \tag{14}$$

$$0 \leq z_v \leq 1 \qquad \text{for each } v \in V(G), \tag{15}$$

$$0 \leq w_v \leq 1 \qquad \text{for each } v \in V(G). \tag{16}$$

The objective function in (10) accounts for the used $\mathbf{y}^*$ budget, as prescribed in Definition 9. Inequality (11) guarantees the minimum number of vertices dominated by S (excluding those which are in S). Inequalities (12) and (13) bind the binary variables $\mathbf{w}$ and $\mathbf{z}$, to enforce the domination condition that $w_v = 1$ if and only if $z_u = 1$ for some $u \in N[v]$.

Inequalities (14) are redundant, being implied at integer points in $\mathcal{P}_{\mathrm{UNI}}(G, \mathbf{y}^*)$ by (11) and the fact the input parameter satisfies $y^*_u + y^*_v \geq 1$ for each $\{u,v\} \in E$. Still, adding those inequalities is likely to tighten the LP relaxation bounds, and hence speed up the overall optimization procedure.

The exact separation problem thus reduces to deciding if $\rho < 1$. The MIP heuristic, on the other hand, consists of searching (*e.g.* allowing a MIP solver to run with a prescribed time limit) for any integer feasible solution $(\mathbf{z}', \mathbf{w}')$ with an objective value less than 1, which determines the UNI $\sum_{v \in S'} x_v \leq |S'| - 1$, with $S' = \{v \in V : z'_v = 1\}$, violated at $x^* = \mathbf{1} - y^*$.

3 Combinatorial dual bounds

We concern next the possibility to compute dual bounds to problem (5) via a *combinatorial relaxation*, *i.e.* computing a lower bound to $z \stackrel{\text{def}}{=} \min \left\{ \sum_{v \in V} w(v) x_v : \mathbf{x} \in \mathcal{P}(G, k) \cap \{0,1\}^n \right\}$ through a relaxation which is a new combinatorial optimization problem, and which is more tractable or interesting, for some reason. For instance, a key ingredient in a recent matheuristic for a class of generalized set partitioning problems (Samer et al., 2019) is an efficiently computable combinatorial bound similar to the ones introduced here, even though the actual bounds are weaker than LP relaxation ones.

In this section, we write $V(G) = \{v_1, \ldots, v_n\}$, with the vertices indexed by non-decreasing weight, so that $w(v_1) \leq w(v_2) \leq \cdots \leq w(v_n)$. Note that the most naïve lower bound corresponds to the selection of the k vertices of least weight in G. That is,

$$z \geq \sum_{i=1}^{k} w(v_i), \tag{17}$$

which corresponds to relaxing all of the edge inequalities (7) in the definition of $\mathcal{P}(G, k)$. We introduce a simple way of relaxing fewer of those inequalities.

Recall that a *matching* in a graph is a subset of pairwise non-adjacent edges, that is, a subset of edges without common vertices. While the facial structure of the matching polytope and combinatorial algorithms to find a maximum weighted matching in a graph are well-known, the following result is less frequently used.

Remark 10. Finding a minimum-weight matching of a specified cardinality in a graph is a well-solved problem. More generally, for any $l, u \in \mathbb{Z}_+$, $l \leq u$, the convex hull of incidence vectors of matchings $M \subset E(G)$ such that $l \leq |M| \leq u$ is equal to the set of those vectors in the matching polytope of G satisfying $l \leq \mathbf{1}^\top x \leq u$, that is, $l \leq \sum_{e \in E(G)} x(e) \leq u$ (Schrijver, 2003, Section 18.5f).

Theorem 11. *Suppose that $\mathcal{P}(G,k) \cap \{0,1\}^n \neq \emptyset$, so that z is well-defined in problem (5), and let $S \overset{def}{=} \{v_1, \ldots, v_k\}$.*

(i) Let $M \subset E$ be any matching in the induced subgraph $G[S]$. Then

$$b_1(M) \overset{def}{=} \sum_{i=1}^{k} w(v_i) + \sum_{\substack{\{v_i,v_j\} \in M,\\ i<j}} [w(v_{k+1}) - w(v_j)]$$

is a lower bound on z.

(ii) Let $\nu \in \left\{1, \ldots, \left\lfloor \frac{k}{2} \right\rfloor\right\}$ denote the maximum cardinality of a matching in $G[S]$. For $1 \leq q \leq \nu$, let $M_q \subset E$ be any matching in $G[S]$ such that $|M_q| = q$. Then

$$b_2(M_1, \ldots, M_\nu) \overset{def}{=} \max_{1 \leq q \leq \nu} \left\{ b_1(M_q) + \sum_{h=2}^{q} [w(v_{k+h}) - w(v_{k+1})] \right\}$$

is a lower bound on z.

Proof. Note first that S is the vertex selection giving the trivial bound (17), which is also the first summand in the definition of b_1. If there exists $\{v_i, v_j\} \in E$, for any $1 \leq i < j \leq k$, then S is not feasible and (17) is not tight. Since some (possibly all) vertices in S need to be *replaced* towards finding an optimal k-stab, an optimistic approach would be to consider disjoint pairs of vertices which are not compatible with each other in S, *i.e.* a matching in $G[S]$, and that we could form a k-stab by exchanging the vertex with larger weight in each pair with a hypothetical vertex in $G \backslash S$ with the least possible weight: $w(v_{k+1})$. Surely, there might not exist enough vertices with such weight in $G \backslash S$, and even if that is the case, the new pair could be incompatible (that is, adjacent in G). Still, to assume an additional value of $w(v_{k+1}) - w(v_j)$ to make feasible each matched pair of vertices $\{v_i, v_j\}$, $i < j$, yields a lower bound on the optimal value of z. This proves *(i)*, where only those additional values are added to the naïve bound.

The proof of *(ii)* follows from the same reasoning, while performing slightly less optimistic exchanges. Previously, we assumed the availability of enough vertices with weight $w(v_{k+1})$ in $G\backslash S$ to replace the one with larger weight on each matched pair in M. Now, we still get a lower bound on z if we use the actual weight of that many vertices among the ones with lowest weight in $G\backslash S$. More precisely: given matching M_q, to assume that replacing $\{v_j :\ \text{for each}\ \{v_i, v_j\} \in M_q, i < j\}$ by $\{v_{k+h} :\ \text{for}\ 1 \leq h \leq q\}$ would give a k-stab is still a relaxation of problem (5). We can thus increase bound $b_1(M_q)$, where we assumed $w(v_{k+1}) - w(v_j)$ would suffice to replace each v_j, by the accumulated differences $\sum_{h=2}^{q} [w(v_{k+h}) - w(v_{k+1})]$, and still get a lower bound on z. Finally, since we cannot anticipate which matching gives the greatest weight increase, we take the maximum bound among the ones attained by different matching cardinalities in $G[S]$. $\square$

Remark 12. For each edge $\{v_i, v_j\} \in E(G[S])$, with $i < j$, let $c(\{v_i, v_j\}) \stackrel{\text{def}}{=} \big[w(v_{k+1}) - w(v_j)\big]$. Then, taking M to be a maximum weighted matching in $G\,[S]$ with edge weights given by c gives the strongest bound $b_1(M)$ in Theorem 11. Analogously, taking all M_q, $q \in \{1, \ldots, \nu\}$, to be maximum weighted matchings gives the strongest bound $b_2\,(M_1, \ldots, M_\nu)$.

It is worth remarking that the graph $G\,[S]$ would no longer be a model for the pairwise compatibility of the new selection of vertices after even a single such exchange operation. Therefore, in the case of both bounds b_1 and b_2, we cannot accumulate the additional value for non-disjoint conflicting pairs of vertices and still get a lower bound on z. That is the rationale behind searching for matchings, and attaining dual bounds for z via a well-solved combinatorial problem.

We can generalize the reasoning behind the relaxations yielding bounds b_1 and b_2 in Theorem 11 by considering matchings in the whole graph G, that is, not only in a proper induced subgraph. Since each k-stab contains at most one vertex from each edge in a matching, we can simply pick the k vertices of lowest weight among: (i) the cheapest vertex in each matched edge, and (ii) the remaining vertices not covered by the matching.

So we have the following result.

Theorem 13. *Suppose that $\mathcal{P}(G,k) \cap \{0,1\}^n \neq \emptyset$, so that z is well-defined in problem* (5)*. Let $M \subset E$ be any matching in G. Define $c_e \overset{def}{=} w(v_i)$ for each edge $e = \{v_i, v_j\} \in M$, with $i < j$. Also define $c_u \overset{def}{=} w(v_u)$ for any vertex v_u not covered by the matching M. Then, the sum of the k lowest values among the $c(\cdot)$ is a lower bound on z. That is, given an order $c_1 \leq c_2 \cdots \leq c_{(n-|M|)}$ on $\{c_e\}_{e\in M} \cup \{c_u\}_{u\in V\setminus V_M}$, where V_M corresponds to the set of vertices covered by M, we have $z \geq \sum_{i=1}^{k} c_i$.*

The drawback involved in this statement is that, while the actual algorithm to compute the bounds referring to Theorem 11 is immediate (following Remark 12), the choice of a specific matching M yielding the strongest bound in Theorem 13 is not clear. A first approach would be to evaluate different greedy constructions. Alternatively, a stronger bound should follow from computing minimum-weight matchings in G with cardinality at least $l \in \{1, \ldots, k\}$, using the edge-weight function corresponding to $c(\cdot)$ in the latter theorem, that is, for each $\{v_i, v_j\} \in E(G)$, with $i < j$, define $c(\{v_i, v_j\}) \overset{\text{def}}{=} w(v_i)$. Note that we can find such a matching of least weight in polynomial time, as we note in Remark 10.

Finally, since every matching in a proper induced subgraph is also considered by Theorem 13, it follows that experimenting with the selection of the matching M yielding the latter bounds should never be weaker than bounds b_1 or b_2 from Theorem 11.

3.1 Application towards balanced branching trees

A fundamental component for the performance of branch-and-cut algorithms for the classical stable set problem is the balanced branching rule of Balas and Yu (1986); see also Rebennack et al. (2012) and Mannino and Sassano (1996). Its original motivation also applies to the fixed cardinality setting: avoiding unbalanced branch-and-bound trees when branching on a fractional variable x_v, since fixing $x_v = 1$ has the larger impact of implying $x_u = 0$ for each $u \in N(v)$, while fixing $x_v = 0$ has no impact on the neighbourhood.

The general branching scheme can be adapted to find minimum weight k-stabs with little effort. Suppose that, on a given node of the enumeration tree, $G' = (V', E')$ denotes the subgraph induced by vertices not fixed in this subproblem, and that $\overline{z}$ is the best primal bound available. Let $W \subseteq V'$ be such that we can determine *efficiently* that the minimum weight of a k-stab in the subgraph induced by W, denoted $z(W)$, is such that $z(W) \geq \overline{z}$. Note that, if $W = V'$, the subproblem is fathomed and the whole subtree rooted on this node can be pruned. Otherwise, if the search on this subtree is to eventually find that $z(V') < \overline{z}$, any bound-improving solution must intersect $V' \backslash W = \{v_1, \ldots, v_p\}$. That is, we can partition the search space into the sets

$$V_i' = \{v_i\} \bigcup V' \backslash \left(N(v_i) \cup \{v_{i+1}, \ldots, v_p\}\right)$$

for $1 \leq i \leq p$. The enumeration can therefore branch on p subproblems, each fixing $x_{v_i} = 1$, and fixing at 0 those variables corresponding to $N(v_i) \cup \{v_{i+1}, \ldots, v_p\}$.

Now, there are different strategies to determine subgraph W. The standard one is to find a collection of cliques in G', *e.g.* with as many cliques as the currently available lower bound, when searching for maximum cardinality stable sets. For minimum-weight k-stabs, the natural idea would be to greedily find k cliques, such that the combined weight of the *cheapest* vertices in each exceed $\overline{z}$.

The combinatorial bounds that we introduce give an alternative approach tailored for optimizing over k-stabs. Using the weight function corresponding to $c(\cdot)$ in Theorem 13, we can determine candidate subgraphs W by inspecting, for each $l \in \{1, \ldots, k\}$:

1. A minimum-weight matching in G' with cardinality l
2. A suitable choice of $k - l$ vertices not covered by the matching

We leave for future work the task of comparing those two strategies, whether theoretically or according to computational experience.

4 Particular cases solvable in polynomial time

A major research topic in combinatorial optimization is the study of particular cases of an NP-hard problem which admit a solution algorithm with polynomially-bounded worst case complexity. As indicated before, the rich theory on the classical stable set problem suggests that research in this direction is also promising. The work of Dabrowski et al. (2011, 2012) in parameterized complexity parallels our contributions here. Instead of NP-completeness, their work builds on the W[1]-completeness of the classical stable set problem, cf. Cygan et al. (2015, Section 13.3), to give fixed-parameter tractable algorithms for an input restricted to some graph classes which extend that of graphs of bounded clique number.

We note that the recognition problem for all classes we discuss next can be solved in polynomial time, *i.e.* given an arbitrary graph G, there exists an algorithm with polynomially-bounded worst-case time complexity which decides if G belongs to that class of graphs. We refer the interested reader to the ISGCI encyclopaedia of graph classes (de Ridder et al., 2001–2019). Throughout this section, we denote by perfect the set of all perfect graphs (*i.e.* those graphs in which the chromatic number of every induced subgraph equals its clique number), and follow similar typography for any graph class.

Remark 14. Consider the unweighted problems corresponding to the classical stable set problem and the fixed cardinality version. If $G = (V, E)$ is such that the stability number $\alpha(G)$ can be found in polynomial time (*i.e.* the classical problem over G is in complexity class P), then we also have that deciding if there exists a k-stab in G is also in P. More precisely: for $k \in \{1, \ldots, \alpha(G)\}$, the answer for the latter problem is `yes`; for $k > \alpha(G)$, the answer is `no`. Nevertheless, the same is not true regarding the weighted version of the problems. Even if a maximum-weight stable set in G can be found in polynomial time, it is not obvious how to find a k-stab in G of optimal weight, in general. In principle, there can be a number of optimal solutions for the classical problem, from which a k-stab might be retrieved or not; and, conversely, there might exist optimal-weight k-stabs in G which are not contained in any optimal solution to the classical problem.

Recall that a graph is *k-partite* (or *k-colourable*) if its vertices can be partitioned into k different stable sets. Now, a *complete k-partite* graph is a k-partite graph containing an edge between all pairs of vertices from different stable sets. A *complete multipartite graph* is complete k-partite for some k. The following result is rather straightforward.

Theorem 15. *If G is a complete multipartite graph, the problem of finding a minimum-weight k-stab in G is in* P.

Proof. Let $G = (V_1 \uplus \cdots \uplus V_c, E)$ be an arbitrary complete c-partite graph, so that each V_i induces a stable set in G, for $1 \leq i \leq c$. Clearly, no stable set in G contains vertices from more than one set in the partition. For each V_i such that $|V_i| \geq k$, then, we inspect the least-weight subset of cardinality k, *i.e.* we find $S_i \in \arg\min_{S \subset V_i, |S|=k} \sum_{v \in S} w(v)$. A minimum-weight k-stab in G is therefore one of minimum weight among all S_i. $\square$

Complete multipartite graphs are a subclass of *cographs*, or *complement-reducible graphs*: those which can be constructed from isolated vertices by disjoint union and complementation operations alone. The class cograph is equivalent to that of P_4-free graphs, and a number of other characterizations are known (McKee and McMorris, 1999, Sec. 7.9). It follows from the definition that the class of graphs which are the complement of some complete multipartite graph corresponds to another subclass of cographs. These are known as *cluster graphs*. Thus G is a cluster graph if and only if G is the disjoint union of cliques; equivalently, G is a cluster graph if and only if it is P_3-free.

Theorem 16. *If G is cluster graph, the problem of finding a minimum-weight k-stab in G is in* P.

Proof. Let $G = \biguplus_{i=1}^{q} K_{n_i}$, where each K_{n_i} induces a clique on n_i vertices. Clearly, at most one vertex from each K_{n_i} can be part of a k-stab. If $k > q$, there cannot exist a k-stab in G. Now, assuming $k \leq q$, the set of k-stabs in G corresponds to subsets of k vertices from different cliques each, since G is a *disjoint* union of the q cliques K_{n_i}, $1 \leq i \leq q$. In particular, we can restrict our attention to the set $S \stackrel{\text{def}}{=} \biguplus_{i=1}^{q} \left\{ v_i \in \arg\min_{v \in K_{n_i}} w(v) \right\}$ of least weight vertices in each clique, and a minimum-weight k-stab in G can be found by choosing k vertices of least weight in S. $\square$

Note that cograph is contained in perfect. We consider next another subclass of perfect graphs, not contained in that of cographs. We say that G is a *split graph* if there exists a partition of its vertices into two sets, one of which induces a clique in G, the other inducing a stable set. A noteworthy result is that almost all *chordal graphs* are in split.

(Recall that chordal or triangulated graphs are those in which every cycle of length at least 4 has a chord). Precisely, the probability that a chordal graph chosen uniformly at random from the set of all chordal graphs on n vertices is split goes to 1 as $n \to \infty$ (Bender et al., 1985).

Theorem 17. *If G is a split graph, the problem of finding a minimum-weight k-stab in G is in* P*.*

Proof. Suppose $V(G) = \mathrm{C} \uplus \mathrm{I}$ is such that C induces a clique and I induces a stable set in G. Note first that at most one vertex from C belongs to any k-stab. Then, if $|\mathrm{I}| < k-1$, or if $|\mathrm{I}| = k-1$ and $C = \emptyset$, there is no k-stab in G. Suppose now that $|\mathrm{I}| \geq k-1$ and $C \neq \emptyset$. For each $v_i \in \mathrm{C}$ such that $|\mathrm{I} \backslash N(v_i)| \geq k-1$, let S_i denote a subset of $k-1$ vertices in $\mathrm{I} \backslash N(v_i)$ of least weight, that is, $S_i \in \arg\min_{S \subseteq \mathrm{I} \backslash N(v_i), |S|=k-1} \sum_{v \in S} w(v)$. Now,

$$\mathcal{S} \stackrel{\text{def}}{=} \big\{\, \{v_i\} \cup S_i : \text{ for each } v_i \in \mathrm{C} \text{ such that } |\mathrm{I} \backslash N(v_i)| \geq k-1 \big\}$$

is thus an enumeration of all the k-stabs in G which include a vertex from C, and that those amount to at most $|\mathrm{C}|$ k-stabs. If $|\mathrm{I}| \geq k$, define also $S_0 \in \arg\min_{S \subseteq \mathrm{I}, |S|=k} \sum_{v \in S} w(v)$, *i.e.* a k-stab contained in I of least weight. Therefore, a minimum-weight k-stab in G can be found by inspection among those in $\mathcal{S}$ and S_0 thus defined. □

The class $\mathsf{cograph} \cap \mathsf{split}$ is equivalent to the class of *threshold graphs.* G is a threshold graph if it is possible to define a constant $t \in \mathbb{R}$ and a function $f : V(G) \to \mathbb{R}$ in such a way that $\{u, v\} \in E(G)$ if and only if $f(u) + f(v) \geq t$. An equivalent definition is that G is a threshold graph if it can be constructed from the empty graph by repeatedly adding either an isolated vertex or a universal vertex. It is therefore a consequence of Theorem 17 that our problem of interest is well-solved over threshold graphs.

Corollary 18. *If G is a threshold graph, the problem of finding a minimum-weight k-stab in G is in* P*.*

From our results, we have algorithms with polynomial worst-case time complexity to find minimum-weight k-stabs in some representative subclasses of $\mathsf{cograph}$: complete graphs, complete multipartite graphs, cluster graphs and threshold graphs. A natural question that we pose as a conjecture, then, is whether the positive results could be generalized to the whole class of cographs.

Conjecture 2. *Given an arbitrary cograph G, weights $w : V(G) \to \mathbb{R}_+$ and $k \in \mathbb{Z}_+$, the problem of finding a minimum-weight k-stab in G is in* P.

To conclude, we mention that the problem is also well-solved over a class of graphs which is not contained in perfect. An equivalent result was already shown by (Buchanan, 2016), but we include its simple proof for the sake of completeness. The *line graph* $L(G)$ of a given graph $G = (V, E)$ is the intersection graph of the edges of G, that is, the graph containing a vertex for each element in E, and where two vertices are connected if and only if the corresponding edges in G share an endpoint.

Theorem 19. *If H is a line graph, the problem of finding a minimum-weight k-stab in H is in* P.

Proof. Suppose that $H = L(G)$ is an arbitrary line graph, with G being an underlying root graph. Note that G is uniquely defined, provided $H \notin \{K_3, K_{1,3}\}$ (in which cases the result would follow immediately), as proved by Whitney (1932) cf. McKee and McMorris (1999, Example 1.4). Moreover, the original graph G can be determined from H in linear time (Lehot, 1974). Now, $S \subset V(H)$, induces a stable set in H if and only if $S \subset E(G)$ is a matching in G, and the bijection obviously preserves cardinality and weight. Therefore, a minimum-weight k-stab in H corresponds to a minimum-weight matching of cardinality k in G. The result, then, follows from the fact that finding such a matching is a well-solved problem, as described in Remark 10. □

References

E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM Journal on Computing*, 15(4):1054–1068, 1986. URL https://doi.org/10.1137/0215075.

A. Bazzi, S. Fiorini, S. Pokutta, and O. Svensson. No small linear program approximates vertex cover within a factor 2 - ϵ. *Mathematics of Operations Research*, 44(1):147–172, 2019. URL https://doi.org/10.1287/moor.2017.0918.

E. A. Bender, L. B. Richmond, and N. C. Wormald. Almost all chordal graphs split. *Journal of the Australian Mathematical Society. Series A. Pure Mathematics and Statistics*, 38(2):214–221, 1985. URL https://doi.org/10.1017/S1446788700023077.

D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997.

D. Bertsimas and R. Weismantel. *Optimization over integers*. Dynamic Ideas, Belmont, MA, USA, 2005.

Q. Botton. *Survivable network design with quality of service constraints: extended formulations and Benders decomposition*. PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, September 2010. URL http://hdl.handle.net/2078.1/33300.

M. Bruglieri, M. Ehrgott, H. W. Hamacher, and F. Maffioli. An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344 – 1357, 2006. URL https://doi.org/10.1016/j.dam.2005.05.036.

A. Buchanan. Extended formulations for vertex cover. *Operations Research Letters*, 44 (3):374 – 378, 2016. URL https://doi.org/10.1016/j.orl.2016.03.008.

A. Buchanan and S. Butenko. Tight extended formulations for independent set, 2014. URL http://www.optimization-online.org/DB_FILE/2014/09/4540.pdf. Unpublished manuscript.

M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer International Publishing, 2015. URL https://doi.org/10.1007/978-3-319-21275-3.

K. Dabrowski, V. Lozin, H. Müller, and D. Rautenbach. Parameterized algorithms for the independent set problem in some hereditary graph classes. In C. S. Iliopoulos and W. F. Smyth, editors, *International Workshop on Combinatorial Algorithms (IWOCA) 2010*, volume 6460 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 2011. URL `https://doi.org/10.1007/978-3-642-19222-7_1`.

K. Dabrowski, V. Lozin, H. Müller, and D. Rautenbach. Parameterized complexity of the weighted independent set problem beyond graphs of bounded clique number. *Journal of Discrete Algorithms*, 14:207 – 213, 2012. URL `https://doi.org/10.1016/j.jda.2011.12.012`.

A. Darmann, U. Pferschy, and J. Schauer. Determining a minimum spanning tree with disjunctive constraints. In F. Rossi and A. Tsoukias, editors, *Algorithmic Decision Theory*, volume 5783 of *Lecture Notes in Computer Science*, pages 414–423. Springer Berlin Heidelberg, 2009. URL `https://doi.org/10.1007/978-3-642-04428-1_36`.

A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726 – 1735, 2011. URL `https://doi.org/10.1016/j.dam.2010.12.016`.

H. N. de Ridder et al. Information System on Graph Classes and their Inclusions (ISGCI). `http://www.graphclasses.org`, 2001–2019.

J. Gajarský, P. Hliněný, and H. R. Tiwary. Parameterized extension complexity of independent set and related problems. *Discrete Applied Mathematics*, 248:56 – 67, 2018. URL `https://doi.org/10.1016/j.dam.2017.04.042`.

M. Guignard. Lagrangean relaxation. *Top*, 11(2):151–200, 2003. URL `https://doi.org/10.1007/BF02579036`.

S. Hanafi and R. Todosijević. Mathematical programming based heuristics for the 0–1 MIP: a survey. *Journal of Heuristics*, 23(4):165–206, 2017. URL `https://doi.org/10.1007/s10732-017-9336-y`.

J. Janssen and K. Kilakos. Bounded stable sets: Polytopes and colorings. *SIAM Journal on Discrete Mathematics*, 12(2):262–275, February 1999. URL https://doi.org/10.1137/S089548019630978X.

P. G. H. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the Association for Computing Machinery*, 21:569–575, 1974. URL https://doi.org/10.1145/321850.321853.

A. Lucena. Non delayed relax-and-cut algorithms. *Annals of Operations Research*, 140 (1):375–410, 2005. URL https://doi.org/10.1007/s10479-005-3977-1.

C. Mannino and A. Sassano. Edge projection and the maximum cardinality stable set problem. In D. S. Johnson and M. Trick, editors, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, October 11-13, 1993*, volume 26, pages 205–219. American Mathematical Society, 1996. DIMACS series in discrete mathematics and theoretical computer science.

C. Mannino, G. Oriolo, F. Ricci, and S. Chandran. The stable set problem and the thinness of a graph. *Operations Research Letters*, 35(1):1–9, 2007. URL https://doi.org/10.1016/j.orl.2006.01.009.

T. McKee and F. McMorris. *Topics in Intersection Graph Theory*. Society for Industrial and Applied Mathematics, 1999. URL https://doi.org/10.1137/1.9780898719802.

G. L. Nemhauser and L. E. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6(1):48–61, Dec 1974. URL https://doi.org/10.1007/BF01580222.

S. Rebennack, G. Reinelt, and P. M. Pardalos. A tutorial on branch and cut algorithms for the maximum stable set problem. *International Transactions in Operational Research*, 19(1-2):161–199, 2012. URL https://doi.org/10.1111/j.1475-3995.2011.00805.x.

P. Samer and D. Haugland. The unsuitable neighbourhood inequalities for the fixed cardinality stable set polytope. In C. Gentile, G. Stecca, and P. Ventura, editors, *Graphs and Combinatorial Optimization: from Theory to Applications – CTW2020 Proceedings*, volume 5 of *AIRO Springer Series*. Springer International Publishing, 2021. URL `https://doi.org/10.1007/978-3-030-63072-0`.

P. Samer, E. Cavalcante, S. Urrutia, and J. Oppen. The matching relaxation for a class of generalized set partitioning problems. *Discrete Applied Mathematics*, 253:153 – 166, 2019. URL `https://doi.org/10.1016/j.dam.2018.05.033`.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg, 2003.

H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932. URL `https://doi.org/10.2307/2371086`.

Chapter 5

Stable spanning trees

Phillippe Samer, Dag Haugland

Polyhedral results and stronger Lagrangean bounds for stable spanning trees

The version of record of this article, published in **Optimization Letters**, Volume 17, Issue 6, pages 1317-1335 (2023), is available online at the publisher's website: `https://doi.org/10.1007/s11590-022-01949-8`

NB. The proofs of Theorems 9 and 10 in the above publication are omitted, as they appear in the preliminary version published in the *open access* proceedings of INOC 2022 – the 10th International Network Optimization Conference. For completeness, we include those proofs in Sections 5.1 and 5.2 of the present thesis, after the main paper.

Polyhedral results and stronger Lagrangean bounds for stable spanning trees

Phillippe Samer and Dag Haugland

{samer@uib.no, dag@ii.uib.no}

Universitetet i Bergen, Institutt for Informatikk, Postboks 7800, 5020, Bergen, Norway

September 15, 2022

Abstract

Given a graph $G = (V, E)$ and a set C of unordered pairs of edges regarded as being in conflict, a stable spanning tree in G is a set of edges T inducing a spanning tree in G, such that for each $\{e_i, e_j\} \in C$, at most one of the edges e_i and e_j is in T. The existing work on Lagrangean algorithms to the NP-hard problem of finding minimum weight stable spanning trees is limited to relaxations with the integrality property. We exploit a new relaxation of this problem: fixed cardinality stable sets in the underlying conflict graph $H = (E, C)$. We find interesting properties of the corresponding polytope, and determine stronger dual bounds in a Lagrangean decomposition framework, optimizing over the spanning tree polytope of G and the fixed cardinality stable set polytope of H in the subproblems. This is equivalent to dualizing exponentially many subtour elimination constraints, while limiting the number of multipliers in the dual problem to $|E|$. It is also a proof of concept for combining Lagrangean relaxation with the power of integer programming solvers over strongly NP-hard subproblems. We present encouraging computational results using a dual method that comprises the Volume Algorithm, initialized with multipliers determined by Lagrangean dual-ascent. In particular, the bound is within 5.5% of the optimum in 146 out of 200 benchmark instances; it actually matches the optimum in 75 cases. All of the implementation is made available in a free, open-source repository.

Dedicated to the memory of Gerhard Woeginger, a lasting inspiration to the first author, and also one of the pioneers in the study of stable spanning trees.

Note. A preliminary version of this work, including the results in Section 3, appears in the *open access* proceedings of INOC 2022 – the 10th International Network Optimization Conference (Samer and Haugland, 2022).

Acknowledgement. This research is partly supported by the Research Council of Norway through the research project 249994 CLASSIS.

1 Introduction

Given an undirected graph $G = (V, E)$, with edge weights $w : E \to \mathbb{Q}$, and a family C of unordered pairs of edges that are regarded as being in conflict, a stable (or conflict-free) spanning tree in G is a set of edges T inducing a spanning tree in G, such that for each $\{e_i, e_j\} \in C$, at most one of the edges e_i and e_j is in T. The minimum spanning tree under conflict constraints (MSTCC) problem is to determine a stable spanning tree of least weight, or decide that none exists. It was introduced by Darmann et al. (2009, 2011), who also prove its NP-hardness.

Different combinatorial and algorithmic results about stable spanning trees explore the associated conflict graph $H = (E, C)$, which has a vertex corresponding to each edge in the original graph G, and where we represent each conflict constraint by an edge connecting the corresponding vertices in H. Note that each stable spanning tree in G is a subset of E which corresponds both to a spanning tree in G and to a stable set (or independent set, or co-clique: a subset of pairwise non-adjacent vertices) in H. Therefore, one can equivalently search for stable sets in H of cardinality exactly $|V| - 1$ which do not induce cycles in the original graph G.

We have recently initiated the combinatorial study of stable sets of cardinality exactly k in a graph (Samer and Haugland, 2021), where k is a positive integer given as part of the input. There are appealing research directions around algorithms, combinatorics and optimization for problems defined over fixed cardinality stable sets. Also from an applications perspective, conflict constraints arise naturally in operations research and management science. Stable spanning trees, in particular, model real-world settings such as communication networks with different link technologies (which might be mutually exclusive in some cases), and utilities distribution networks. In fact, the latter is a standard application of the quadratic minimum spanning tree problem (Assad and Xu, 1992), which generalizes the MSTCC one.

Exact algorithms to find stable spanning trees have been investigated for a decade now, building on branch-and-cut (Samer and Urrutia, 2015; Carrabs et al., 2021), or Lagrangean relaxation (Zhang et al., 2011; Carrabs and Gaudioso, 2021) strategies. Consider the natural integer programming (IP) formulation for the MSTCC problem:

$$
\begin{aligned}
\min & \sum_{e \in E} w_e x_e && && (1)\\
\text{s.t.} & \sum_{e \in E(S)} x_e \leq |S| - 1, && \text{for each } S \subsetneq V, S \neq \emptyset, && (2)\\
& \sum_{e \in E} x_e = |V| - 1, && && (3)\\
& x_{e_i} + x_{e_j} \leq 1, && \text{for each } \{e_i, e_j\} \in C, && (4)\\
& x_e \in \{0, 1\}, && \text{for each } e \in E. && (5)
\end{aligned}
$$

While a considerable effort in the development of branch-and-cut algorithms led to more sophisticated formulations and contributed to a better understanding of our capacity to solve MSTCC instances by judicious use of valid inequalities, the existing Lagrangean algorithms are limited to the most elementary approach. Namely, a relaxation scheme dualizing conflict constraints (4), which thus has the integrality property, as proved in the seminal work of Edmonds (1971). We review other aspects of the corresponding references in Section 3.1.

The present paper takes the standpoint that the development of a full-fledged Lagrangean strategy to find stable spanning trees is an unsolved problem. While we recognize different merits of previous work, we found it productive to investigate stronger Lagrangean bounds in this context: exploring more creative relaxation schemes, designing improved dual methods, all the while harnessing the polyhedral point of view and progress in IP computation.

The main idea of this paper is to offer an alternative starting point for this problem, building on fixed cardinality stable sets as an alluring handle to work on stable spanning trees. After presenting some elementary properties of the corresponding polytope in Section 2, we use cardinality constrained stable sets again in Section 3 to design a stronger relaxation scheme, based on Lagrangean *decomposition* (LD). We explain how classical results from the literature guarantee the superiority of such a reformulation: both with respect to the quality of dual bounds, when compared to the straightforward relaxation, and with regard to the number of multipliers, when compared to an alternative framework to determine the same bounds (relax-and-cut dualizing violated subtour elimination constraints (2) dynamically).

We see the opportunity for renewed interest in LD in light of the progress in mixed-integer linear programming (MILP) computation. Given the impressive speedup of MILP solvers over the past two decades, Dimitris Bertsimas and Jack Dunn are among a group of distinguished researchers who make a case for (exact) optimization over integers as the natural, correct model for several tasks within machine learning and towards interpretable artificial intelligence. This is the theme of their recent book (Bertsimas and Dunn, 2019); see also Bertsimas et al. (2016, 2020). We draw inspiration from this philosophy (challenging assumptions previously deemed computationally intractable) to propose less hesitation towards designing Lagrangean algorithms that exploit subproblems for which, albeit strongly NP-hard, specialized solvers attain good performance. Indeed, we present a proof of concept in the particular case of the MSTCC problem. We leverage a state-of-the-art branch-and-cut algorithm for fixed cardinality stable sets to an effective method to compute strong dual bounds for optimal stable spanning trees by means of LD.

In summary, our contributions are the following.

1. On the polyhedral combinatorics side, we present intersection properties and a bound on the dimension of the fixed cardinality stable set polytope, a relaxation of the stable spanning tree one.

2. We propose a sound analysis of different Lagrangean bounds published in the literature of the MSTCC problem, design a stronger reformulation based on LD, and justify its advantages both in theory and in a numerical evaluation. We make a case for designing new algorithms combining LD and MILP solvers exploring strongly NP-hard subproblems.

3. We present a free, open-source software package implementing the complete algorithm. It welcomes extensions and eventual collaborations, besides offering a series of useful, general-purpose algorithmic components, *e.g.* separation procedures, an LD based dual-ascent framework, an application of the Volume Algorithm framework implemented in COIN-OR.

2 Polyhedral results

As a first step towards knowledge about the polytope of stable spanning trees in a graph, we study elementary properties of the larger polytope $\mathfrak{C}(H, k)$ of fixed cardinality stable sets in the conflict graph $H = (E, C)$. The polyhedral results in this section serve their own purpose, and are not necessary for the reformulation and results presented in the remaining of the paper.

We begin with the necessary notation and terminology. For conciseness, we abbreviate "stable set of cardinality k" as *kstab* in this work. Let $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$, and let $\operatorname{conv} S$ denote the convex hull of a set S. Recall that the *incidence* (or *characteristic*) *vector* of a set $S \subset E = \{e_1, \ldots, e_m\}$ is defined as $\chi^S \in \{0,1\}^{|E|}$ such that $\chi_i^S = 1$ if and only if $e_i \in S$. The family of all incidence vectors of kstabs in H is denoted $\mathcal{F}_{\text{kstab}}(H, k)$. Hence $\mathfrak{C}(H, k) \stackrel{\text{def}}{=} \operatorname{conv} \mathcal{F}_{\text{kstab}}(H, k)$.

Also let $\mathcal{F}^{\uparrow}_{\text{kstab}}(H,k) \subset \{0,1\}^{|E|}$ denote the family of incidence vectors of stable sets of cardinality greater than or equal to k in H, and let $\mathfrak{C}^{\uparrow}(H,k) \stackrel{\text{def}}{=} \operatorname{conv} \mathcal{F}^{\uparrow}_{\text{kstab}}(H,k)$ denote their convex hull. Define $\mathcal{F}^{\downarrow}_{\text{kstab}}(H,k)$ and $\mathfrak{C}^{\downarrow}(H,k)$ analogously for stable sets of cardinality at most k. We omit the parameters H and k in such notation where it does not cause any confusion. Likewise, we occasionally omit the indices in summations over all coordinates of a point to make a passage more readable, *e.g.* $\sum \mathbf{x}$ when it clearly means $\sum_{i\in[n]} x_i$. Finally, let $\operatorname{ext} \mathcal{P}$ denote the set of extreme points of a given polyhedron $\mathcal{P}$.

In the following, we present intersection properties connecting $\mathfrak{C}$, $\mathfrak{C}^{\uparrow}$, and $\mathfrak{C}^{\downarrow}$.

Theorem 1. *Let H be an arbitrary graph on n vertices, and k be a positive integer.*

i. $\mathfrak{C}(H,k) = \mathfrak{C}^{\uparrow}(H,k) \cap \mathfrak{C}^{\downarrow}(H,k)$.

ii. $\mathfrak{C}(H,k) = \mathfrak{C}^{\uparrow}(H,k) \cap F = \mathfrak{C}^{\downarrow}(H,k) \cap F$, *where* $F \stackrel{def}{=} \left\{x \in \mathbb{Q}^n : \sum_{u\in[n]} x_u = k\right\}$.

Proof. $(i.)$ $\mathfrak{C} \subseteq \mathfrak{C}^{\uparrow} \cap \mathfrak{C}^{\downarrow}$ follows from the fact that the convex hull of the intersection of two sets is contained in the intersection of the respective convex hulls.

For the other inclusion, let $\mathbf{x}^* \in \mathfrak{C}^{\uparrow} \cap \mathfrak{C}^{\downarrow}$ be arbitrary. Without loss of generality, we write $\mathbf{x}^*$ as a convex combination of p vertices of $\mathfrak{C}^{\uparrow}$:

$$\mathbf{x}^* = \sum_{i\in[p]} \lambda_i \mathbf{y}^i, \text{ with } \lambda_i \geq 0 \text{ for each } i, \sum_{i\in[p]} \lambda_i = 1, \text{ and } \left\{\mathbf{y}^i\right\}_{i\in[p]} \subseteq \operatorname{ext} \mathfrak{C}^{\uparrow}.$$

Note that $\mathbf{y}^i \in \mathfrak{C}^{\uparrow} \implies \sum_{u\in[n]} y^i_u \geq k$ for each i. Now, if $\sum_{u\in[n]} y^i_u > k$ for some $i \in [p]$, we derive from $\lambda_i \geq 0$ and $\sum \lambda_i = 1$ that $\sum_{u\in[n]} x^*_u > k$, and $\mathbf{x} \notin \mathfrak{C}^{\downarrow}$. Hence $\sum_{u\in[n]} y^i_u = k$ for each $i \in [p]$, and $\{\mathbf{y}^i\}_{i\in[p]} \subseteq \mathfrak{C}$. By convexity of $\mathfrak{C}$, we conclude that $\mathbf{x}^* \in \mathfrak{C}$.

$(ii.)$ It is immediate that $\mathfrak{C} \subseteq \mathfrak{C}^{\uparrow} \cap F$: if $\mathbf{x}^* \in \mathfrak{C}$, we may write $\mathbf{x}^*$ as the convex combination of incidence vectors of kstabs, which is also a convex combination of vertices of $\mathfrak{C}^{\uparrow}$ within F.

For the other inclusion, observe that $\mathfrak{C}^{\uparrow} \cap F$ is the face of $\mathfrak{C}^{\uparrow}$ induced by valid inequality $\sum \mathbf{x} \geq k$. Let $\mathbf{x}^*$ denote a point in that face. Viewing the face as a polytope, $\mathbf{x}^*$ may be written as a convex combination of vertices of the face, which in turn are vertices of $\mathfrak{C}^{\uparrow}$

satisfying $\sum \mathbf{x} = k$. We thus write $\mathbf{x}^*$ as a convex combination of incidence vectors of kstabs, and $\mathbf{x}^* \in \mathfrak{C}$.

The proof is analogous for the second equality, observing that F is the face determined by inequality $\sum \mathbf{x} \leq k$, valid for $\mathfrak{C}^{\downarrow}$. □

Note that it is not necessary that a vertex of the intersection of two polytopes is a vertex of any of the polytopes. For a counterexample, consider two squares A, B in $\mathbb{Q}^2$ such that $A \cap B$ is another square; vertices of the intersection need not be vertices of A or B. The result in Theorem 3 below shows a rather favourable situation when it comes to our cardinality constrained stable set polytopes. In order to prove it, we use the following fact, which is an elementary exercise in polyhedral theory, *e.g.* Exercise 3-8 in the 2017 lecture notes *Linear programming and polyhedral combinatorics*, by Michel Goemans (`https://math.mit.edu/~goemans/18453S17/polyhedral.pdf`). We remind the reader of the equivalence of extreme points, vertices, and basic feasible solutions of a polyhedron.

Lemma 2. *Let* $\mathcal{P} = \{\mathbf{x} \in \mathbb{Q}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{C}\mathbf{x} \leq \mathbf{d}\}$*, and* $\mathcal{Q} = \{\mathbf{x} \in \mathbb{Q}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{C}\mathbf{x} = \mathbf{d}\}$*. It follows that* $\operatorname{ext} \mathcal{Q} \subseteq \operatorname{ext} \mathcal{P}$*.*

Proof. If $\mathbf{x}^* \in \operatorname{ext} \mathcal{Q}$, then $\mathbf{x}^*$ is a basic feasible solution of $\mathcal{Q}$. Let I denote the subset of indices of constraints in $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ that are active at $\mathbf{x}^*$, which is thus the unique solution of the subsystem

$$\begin{cases} \mathbf{a}_i\mathbf{x} &= b_i, \quad \text{for } i \in I, \\ \mathbf{C}\mathbf{x} &= \mathbf{d}. \end{cases} \tag{6}$$

This subsystem also corresponds to a selection of inequalities in the definition of $\mathcal{P}$ to be satisfied with equality. The same n linearly independent constraint vectors in (6) determine that $\mathbf{x}^*$ is a basic solution of $\mathcal{P}$. Since $\mathbf{x}^* \in \mathcal{P}$ as well, it follows that $\mathbf{x}^* \in \operatorname{ext} \mathcal{P}$. □

Theorem 3. $\operatorname{ext} \mathfrak{C}(H,k) = \operatorname{ext} \mathfrak{C}^{\uparrow}(H,k) \cap \operatorname{ext} \mathfrak{C}^{\downarrow}(H,k)$ *for arbitrary* H *and* k*.*

Proof. Let $\mathbf{x}^*$ denote a vertex of both $\mathfrak{C}^{\uparrow}$ and $\mathfrak{C}^{\downarrow}$. Then $\mathbf{x}^*$ is the incidence vector of a kstab in H, and $\mathbf{x}^* \in \operatorname{ext} \mathfrak{C}$. For the other inclusion, we use Lemma 2 twice: once with $\mathcal{P}$

denoting a description of $\mathfrak{C}^{\uparrow}$ (whence $\mathcal{Q}$ is identified with $\mathfrak{C}$, by item (*ii*) in Theorem 1) to show that $\operatorname{ext}\mathfrak{C} \subseteq \operatorname{ext}\mathfrak{C}^{\uparrow}$, and again with $\mathcal{P} = \mathfrak{C}^{\downarrow}$ to show that $\operatorname{ext}\mathfrak{C} \subseteq \operatorname{ext}\mathfrak{C}^{\downarrow}$. $\square$

Corollary 4. *Let H be a graph on n vertices, and k be a positive integer. Also let $\mathcal{P} = \left\{\mathbf{x} \in \mathbb{Q}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \sum_{u\in[n]} x_u \geq k\right\}$ be a formulation for stable sets of cardinality at least k in that graph, that is, $\mathcal{P} \cap \{0,1\}^n = \mathcal{F}^{\uparrow}_{kstab}(H,k)$. If $\mathcal{P}$ is actually integral ($\mathcal{P} = \mathfrak{C}^{\uparrow}$), then so is the formulation $\mathcal{P}' = \left\{\mathbf{x} \in \mathbb{Q}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \sum_{u\in[n]} x_u = k\right\} = \mathfrak{C}(H,k)$. The analogous result holds for $\mathfrak{C}^{\downarrow}(H,k)$.*

These results might be explored in future work that benefit from optimizing over kstabs with a reformulation based on stable sets of *bounded* cardinality. They may also be useful when dealing with classes of graphs for which an explicit characterization of the corresponding polytopes $\mathfrak{C}^{\uparrow}$ or $\mathfrak{C}^{\downarrow}$ is known.

Finally, we give a lower bound on the dimension of the polytope $\mathfrak{C}(H,k)$ as a function of the stability number $\alpha(H)$, that is, the size of the largest stable set in H.

Theorem 5. *Let k be a positive integer, and H be an arbitrary graph on n vertices such that $\alpha(H) \geq k+1$. Then $\alpha(H) - 1 \leq \dim \mathfrak{C}(H,k) \leq n-1$.*

Proof. The upper bound is trivial, given the presence of the cardinality constraint in the equality system of any linear inequality description of $\mathfrak{C}(H,k)$. For the lower bound, we prove by induction on $\alpha(H)$ that we can find $\alpha(H)$ linearly independent (l.i.) incidence vectors of kstabs in H. The result then follows immediately.

Suppose first that $\alpha(H) = k+1$, and let $\chi \in \mathfrak{C}$ be the incidence vector of a stable set of cardinality $k+1$ in H. Let $I \subset [n]$, $|I| = k+1$, denote the coordinates corresponding to vertices in that stable set, that is, $\chi_i = 1$ for each $i \in I$. Denoting the i-th unit vector in $\mathbb{R}^n$ by $\mathfrak{e}^i$, we have that $\{\chi - \mathfrak{e}^i\}_{i\in I}$ are $k+1$ l.i. points in $\mathfrak{C}(H,k)$.

Assume inductively that we can determine p l.i. incidence vectors of kstabs in a graph if its stability number is equal to p. Now, given H such that $\alpha(H) = p+1$, and χ the incidence vector of a maximum stable set in H, we may proceed as above to again determine $p+1$ l.i. incidence vectors of *pstabs* (cardinality p stable sets) in H. Let ϕ, ψ be two such vectors.

As the subgraph induced by ϕ has no edges, we have $\alpha(H[\phi]) = p$. The inductive hypothesis thus yields a collection $\{\chi^1, \ldots, \chi^p\} \subset \{0,1\}^p$ of l.i. incidence vectors of kstabs in the induced subgraph. Let $\{\overline{\chi}^1, \ldots, \overline{\chi}^p\}$ be the lifting of this collection to space $\mathbb{R}^n$ with zeros in the coordinates corresponding to missing vertices.

Since ϕ and ψ are l.i., we claim that it is possible to discard $p-k$ vertices from the stable set induced by ψ in such a way that the incidence vector $\overline{\psi}$ of the resulting kstab is l.i. of $\{\overline{\chi}^1, \ldots, \overline{\chi}^p\}$. Indeed, ϕ and ψ induce *different* pstabs, so that there exists a vertex in the subgraph induced by ψ that is not in the subgraph induced by ϕ. Let $u \in [n]$ be such that $\psi_u = 1$, $\phi_u = 0$, and choose $\overline{\psi}$ (kstab inducing) with $\overline{\psi}_u = 1$. In turn, note that $\overline{\chi}^j_u = 0$ for each $j \in [p]$, by construction: from $\phi_u = 0$ it follows that u is one of the coordinates padded with zero when mapping χ^j to $\overline{\chi}^j$. This means that $\overline{\psi} \notin \operatorname{span}\{\overline{\chi}^1, \ldots, \overline{\chi}^p\}$, and hence we determine $p+1$ l.i. incidence vectors of kstabs in H, completing the proof. $\square$

We remark that the down-monotone polytope $\mathfrak{C}^{\downarrow}(H,k)$ is full-dimensional for arbitrary H and k, as it contains the $|V(H)|+1$ affinely independent points corresponding to the unit vectors and zero. The problem of determining $\dim \mathfrak{C}$ may therefore be cast in terms of $\mathfrak{C}^{\uparrow}$ in future research.

3 Lagrangean relaxation and decomposition

In this section, we present the main contributions of the paper. We give special attention to justifying carefully the drawbacks of previous reformulations based on Lagrangean duality, and how a decomposition approach optimizing over the fixed cardinality stable set polytope leads to an *effective* algorithm to compute strong dual bounds for optimal stable trees.

In this section, *effectiveness* is taken from the analytical point of view: we argue that the decomposition is superior *in theory* both with respect to bound quality and tractability of the dual problem. In the next section, we discuss the practical evaluation of our (free, open-source) software implementing the resulting algorithm, and argue that it indeed contributes as an effective tool to determine tight dual bounds on a representative subset of benchmark instances of the problem.

3.1 Drawbacks of existing Lagrangean approaches for MSTCC

The work of Zhang et al. (2011) contributes in many research directions about stable spanning trees, including particular cases which are polynomially solvable, feasibility tests, several heuristics, and two exact algorithms based on Lagrangean relaxation. The first formulation is straightforward, dualizing all conflict constraints (4); they denote the corresponding dual bound L^*. The second approach relaxes a subset of inequalities (4): using an approximation to the maximum edge clique partitioning problem (Dessmark et al., 2007), this scheme dualizes a subset of conflict constraints such that the remaining conflict graph is a collection of disjoint cliques; the resulting dual bound is denoted ℓ^*. The authors argue that the latter reformulation is stronger than the former, and present extensive computational results justifying their claims.

Unfortunately, the Lagrangean dual bounds L^* and ℓ^* in Zhang et al. (2011) are in fact identical, as we show next. The first relaxation clearly has the integrality property, as the remaining constraints correspond to a description of the spanning tree polytope or, equivalently, to bases of the graphic matroid of G (Edmonds, 1971). The second relaxation scheme is designed so that the conflict constraints which remain in the subproblem of relaxation ℓ^* induce a collection of disjoint cliques in H. The subproblem thus corresponds to the intersection of two matroids: the graphic matroid of G and the partition matroid of subsets of E that intersect the enumerated cliques in H at most once. It follows that the second relaxation also has the integrality property (Nemhauser and Wolsey, 1999, Theorem III.3.5.9), and consequently, L^* and ℓ^* both equal the optimal objective function value in the continuous relaxation of $(1)-(5)$ (Nemhauser and Wolsey, 1999, Corollary II.3.6.6). In this perspective, the computational results in Tables 2–4 of Zhang et al. (2011) diverge from what Lagrangean duality theory prescribes.

Recently, Carrabs and Gaudioso (2021) presented thorough computational experiments of a new Lagrangean algorithm for the MSTCC problem. They use the same relaxation scheme dualizing all conflict constraints, and focus on a combination of dual ascent and the subgradient method to compute the Lagrangean bound, namely, L^* in Zhang et al. (2011), equal to the LP-relaxation of (1) − (5). In Table 1 of Carrabs and Gaudioso (2021), the performance of the new algorithm is compared to the results published in Zhang et al. (2011). That is, the issue we analyse above regarding the computational results of Zhang et al. (2011) is repeated as a baseline of the new numerical evaluation.

Another drawback of the new algorithm is that dual ascent steps are intertwined with subgradient optimization. While *not* incorrect, this choice undermines the advantages of a strategy to solve the dual problem in fewer iterations. A passage from a classical work of Guignard and Rosenwein (1989) is conclusive: *"An ascent procedure may also serve to initialize multipliers in a subgradient procedure. This scheme is particularly useful at the root node of an enumeration tree. However, an ascent method cannot guarantee improved bounds over bounds obtained by solving the Lagrangean dual with a subgradient procedure."*

Moreover, the ascent steps rely on a greedy heuristic, and not on *maximal ascent directions*, *i.e.* optimal step size in a direction of bound increase; see Definition 7. In the algorithm of Carrabs and Gaudioso (2021), if a conflicting pair of edges exists in a Lagrangean solution, the multiplier adjustment is derived from the observation that the dual bound shall improve by at least the increased cost of replacing one of the edges by its cheapest successor (in a list of edges ordered by current costs). The authors remedy the resulting low adjustment values by alternating subgradient optimization iterations and the ascent procedure.

We stress again that references (Carrabs and Gaudioso, 2021) and (Zhang et al., 2011) have many virtues and present concrete contributions to the MSTCC literature. Our only remark is that the first Lagrangean strategy designed to improve upon the LP-relaxation bound is matter-of-factly yet to be introduced. In the next sections, we offer an interesting approach to tackle this challenge.

3.2 Lagrangean decomposition

Renaming the variables in (4) as $\mathbf{y}$, and introducing linking constraints $x_e = y_e$ for each $e \in E$, we have the same formulation. Now, dualizing the linking constraints with Lagrangean multipliers $\lambda \in \mathbb{Q}^{|E|}$, we arrive at the Lagrangean decomposition (LD) formulation:

$$z(\lambda) \stackrel{\text{def}}{=} \min_{\mathbf{x} \in \mathcal{F}_{\text{sp.tree}}(G)} (\mathbf{w} - \lambda)^{\mathsf{T}} \mathbf{x} \; + \min_{\mathbf{y} \in \mathcal{F}_{\text{kstab}}(H, |V|-1)} \lambda^{\mathsf{T}} \mathbf{y} \tag{7}$$

where $\mathcal{F}_{\text{sp.tree}}(G)$ is given by

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \quad \text{for each } S \subsetneq V, S \neq \emptyset, \tag{8}$$

$$\sum_{e \in E} x_e = |V| - 1, \tag{9}$$

$$x_e \in \{0, 1\}, \qquad \text{for each } e \in E, \tag{10}$$

and $\mathcal{F}_{\text{kstab}}(H, |V| - 1)$ is as in Section 2, given by

$$\sum_{e \in E} y_e = |V| - 1, \tag{11}$$

$$y_{e_i} + y_{e_j} \leq 1, \quad \text{for each } \{e_i, e_j\} \in C, \tag{12}$$

$$y_e \in \{0, 1\}, \quad \text{for each } e \in E. \tag{13}$$

The Lagrangean dual problem is to determine the tightest such bound:

$$\zeta \stackrel{\text{def}}{=} \max_{\lambda \in \mathbb{Q}^{|E|}} \{z(\lambda)\}. \tag{14}$$

The first systematic study of LD as a general purpose reformulation technique was presented by Guignard and Kim (1987). They indicate earlier applications of variable splitting/layering, especially by Shepardson and Marsten (1980) and Ribeiro and Minoux (1986). See also the outstanding presentation in (Guignard, 2003, Section 7).

One of the main virtues of the decomposition principle over traditional Lagrangean relaxation schemes is that the bound from the LD dual is equal to the optimum of the primal objective function over the intersection of the convex hulls of both constraint sets (Guignard and Kim, 1987, Corollary 3.4). The decomposition bound is thus equal to the strongest of the two Lagrangean relaxation schemes corresponding to dualizing either of the constraint sets.

In our application to the MSTCC problem, we recognize the integrality of the spanning tree formulation described by (8) − (9) over $\mathbf{x} \in \mathbb{Q}^{|E|}$, following a classical result of Edmonds (1971). Hence the decomposition bound matches that of the stronger scheme where constraints (11) − (12) enforcing fixed cardinality stable sets are kept in the subproblem (which is thus *convexified*), and all subtour elimination constraints (8) are dualized. This means that we can compute stronger Lagrangean bounds, while limiting the number of multipliers in the dual problem to $|E|$, instead of dealing with exponentially many multipliers *e.g.* in a relax-and-cut approach.

We defend the advantages of breaking the original problem into two parts, exploiting their rich combinatorial and polyhedral structures, so as to derive stronger dual bounds. The price of this strategy is to solve a strongly NP-hard subproblem, which naturally leads to the design of more sophisticated dual algorithms, requiring the fewest iterations possible.

3.3 Dual algorithm

We combine two techniques to solve the problem of approximating ζ in the dual problem (14). The first is customized dual ascent, an *ad-hoc*, analytical method that integrates naturally with LD (Guignard and Kim, 1987). It guarantees monotone bound improvement, and could be employed as a stand-alone dual algorithm − though likely converging to a sub-optimal bound $z(\lambda^*) < \zeta$ due to incomplete information of ascent directions. We circumvent this by continuing the search (from the dual ascent solution λ^*) with an iterative, subgradient-based method: the Volume Algorithm (VA) of Barahona and Anbil (2000).

Proposed as an extension of subgradient optimization to attain better numerical results, VA was later characterized by Bahiense et al. (2002) as an intermediate method between classical subgradient and more robust bundle methods, using combinations of past and present subgradient vectors available at each iteration. In contrast to most bundle-type methods, which require the solution of a potentially expensive quadratic program, the computation of a new dual point in VA uses a correction factor determined by a simple recurrence relation. The revision of Bahiense et al. (2002) introduces a classification of *green/yellow/red* steps, like *serious/null* ones in bundle methods, and demonstrates the theoretical convergence of such revised VA. The combined simplicity and comparatively good computational experience reported in applications of VA make it an attractive alternative; see Briant et al. (2008) for a systematic evaluation.

Remark 6. Like many other subgradient-like methods, the Volume Algorithm also determines primal sequences of (fractional) points approximating the dual optimal solution. We do not explore this aspect in the present work. See our suggestions for further research in the discussion following our numerical results in Section 4.3.

Since VA is precisely defined, and we use it as a black-box solver, the remainder of this section is devoted to its initialization by the dual ascent procedure. In what follows, let $\mathfrak{e}_i \in \mathbb{R}^m$ denote the standard unit vector in the i-th direction, and $\mathcal{P}_{\text{sp.tree}}(G) \stackrel{\text{def}}{=} \operatorname{conv} \mathcal{F}_{\text{sp.tree}}(G)$ denote the spanning tree polytope of graph G. Note that $\mathcal{P}_{\text{sp.tree}}$ and $\mathfrak{C}$ are bounded (polytopes contained in the 0,1 hypercube), and do not contain extreme rays.

The Lagrangean dual function $z : \mathbb{Q}^{|E|} \to \mathbb{Q}$ is an implicit function of λ. It is determined by the lower envelope of $\Big\{(\mathbf{w} - \lambda)^\intercal \mathbf{x}^r + \lambda^\intercal \mathbf{y}^s : \mathbf{x}^r \in \operatorname{ext} \mathcal{P}_{\text{sp.tree}}(G), \mathbf{y}^s \in \operatorname{ext} \mathfrak{C}(H, |V| - 1)\Big\}$. Hence, it is piecewise linear concave, and differentiable almost everywhere, with breakpoints at all λ' where the optimal solution to $z(\lambda')$ is not unique.

Such breakpoints are the key ingredient in the dual ascent paradigm to solve a Lagrangean dual problem. In particular, the following kind of point deserves special attention to guide progress in this framework.

Definition 7. A **maximal ascent direction** of the Lagrangean dual function $z : \mathbb{Q}^m \to \mathbb{Q}$ at λ^r is a vector $\mathbf{u} \in \mathbb{Q}^m$ satisfying two conditions: (i) $\mathbf{u}$ determines a direction of increase from $z(\lambda^r)$, *i.e.* $z(\lambda^r + \mathbf{u}) > z(\lambda^r)$; (ii) $\lambda^r + \mathbf{u}$ is a breakpoint of z, that is, if

$(\mathbf{x}^r, \mathbf{y}^r)$ is an optimal solution to $z(\lambda^r)$, then $(\mathbf{x}^r, \mathbf{y}^r)$ also optimizes $z(\lambda^r + \mathbf{u})$, but it is not the unique solution.

A maximal ascent direction determines an optimal multiplier adjustment in a given direction of increase of the Lagrangean dual function. It need not correspond to a *steepest* ascent direction from $z(\lambda^r)$, in general.

The technique of optimizing the Lagrangean dual function by means of ascent directions uses the formulation structure to determine monotone bound improving sequences of multipliers. It was pioneered by Bilde and Krarup (1977) and Erlenkotter (1978) in the context of the facility location problem. An actual algorithm of this kind thus relies on analysing the specific problem and the information available from subproblem solutions. Although there is no pragmatic, problem-independent algorithm, we found it instructive to summarize and systematically review the following instructions in the derivation of our results.

Remark 8 (*Guiding principle of LD based dual ascent*). We may derive a maximal ascent direction by analysing the implications of updating a single multiplier λ_e, corresponding to a violation $x_e \neq y_e$. The update must improve the Lagrangean dual bound and induce an alternative optimal solution.

To avoid overloading the notation in the next two results, we omit the transposition symbol in vector products like $(\mathbf{w} - \lambda^r)^\intercal \mathbf{x}^r$.

Theorem 9. *Let $e \in E$ and let $(\mathbf{x}^r, \mathbf{y}^r)$ be an optimal solution to subproblem $z(\lambda^r)$, such that $x_e^r = 0 < 1 = y_e^r$. Define the non-negative quantities*

$$\Delta_{-e}^r \stackrel{def}{=} \min\{\lambda^r \mathbf{y} : \mathbf{y} \in \mathcal{F}_{kstab}(H, |V| - 1), y_e = 0\} - \lambda^r \mathbf{y}^r, \tag{15}$$

$$\partial_{+e}^r \stackrel{def}{=} \min\{(\mathbf{w} - \lambda^r)\mathbf{x} : \mathbf{x} \in \mathcal{F}_{sp.tree}(G), x_e = 1\} - (\mathbf{w} - \lambda^r)\,\mathbf{x}^r. \tag{16}$$

If $\min\left\{\Delta_{-e}^r, \partial_{+e}^r\right\} \neq 0$*, then* $\min\left\{\Delta_{-e}^r, \partial_{+e}^r\right\} \cdot \mathfrak{e}_e$ *is a maximal ascent direction of z at λ^r.*

Proof. See (Samer and Haugland, 2022, Theorem 4.2). □

We remark that determining a minimum spanning tree with edge $e = \{i, j\}$ fixed *a priori* in (16) can be accomplished efficiently by *contracting* that edge in G. If the contraction operator is defined so as to allow parallel edges between the new vertex ij and $k \in N(i) \cap N(j)$, where $N(u) \subset V$ denotes the neighbourhood of vertex u, we must ensure that not more than one edge between two vertices is chosen (*e.g.* in Kruskal's algorithm; this is not an issue in Prim's method). Now, if the contraction operator forbids parallel edges, we make an unambiguous choice in the original graph G by recognizing the proper edge ($\{i,k\}$ or $\{j,k\}$) yielding the correct spanning tree.

The next result is analogous, now identifying maximal ascent directions from Lagrangean solutions where $x_e^r = 1$ but $y_e^r = 0$.

Theorem 10. *Let $e \in E$ and let $(\mathbf{x}^r, \mathbf{y}^r)$ be an optimal solution to subproblem $z(\lambda^r)$, such that $x_e^r = 1 > 0 = y_e^r$. Define the non-negative quantities*

$$\Delta_{+e}^r \stackrel{def}{=} \min\{\lambda^r \mathbf{y} : \mathbf{y} \in \mathcal{F}_{kstab}(H, |V|-1), y_e = 1\} - \lambda^r \mathbf{y}^r, \tag{17}$$

$$\partial_{-e}^r \stackrel{def}{=} \min\{(\mathbf{w} - \lambda^r)\mathbf{x} : \mathbf{x} \in \mathcal{F}_{sp.tree}(G), x_e = 0\} - (\mathbf{w} - \lambda^r)\,\mathbf{x}^r. \tag{18}$$

If $\min\left\{\Delta_{+e}^r, \partial_{-e}^r\right\} \neq 0$*, then* $\min\left\{\Delta_{+e}^r, \partial_{-e}^r\right\} \cdot (-\mathfrak{e}_e)$ *is a maximal ascent direction of z at λ^r.*

Proof. See (Samer and Haugland, 2022, Theorem 4.3). □

4 Experimental evaluation

The main goal of our computational endeavour is to assess the strength of the LD bound $\zeta = \max_{\lambda \in \mathbb{Q}^{|E|}} \{z(\lambda)\}$ in (14) over benchmark instances of the MSTCC problem. This is fundamental to verify the practicality of that reformulation, as well as to understand its limitations.

A second intention of the project is to offer a careful implementation of the complete algorithm as a free, open-source software package. The code was crafted with attention to time and space efficiency, fairly tested for correctness, and is available in the LD-davol repository on GitHub (`https://github.com/phillippesamer/stable-trees-ld-davol`). It welcomes collaboration towards extensions and facilitates the direct comparison with eventual algorithms designed for the MSTCC problem in the future, besides offering useful, general-purpose algorithmic components. In the remainder of this section, we refer to our implementation of the algorithm by its repository name, LD-davol.

4.1 Implementation details

LD-davol is written in C++, with the support of two libraries integrating the COIN-OR project (Lougee-Heimer, 2003), as we describe next. We also include the preprocessing algorithm introduced by Samer and Urrutia (2015), a collection of probing tests that removes variables and identifies implied conflicts in the original input instance.

Recall that the two building blocks of the dual algorithm presented in Section 3.3 are a dual ascent initialization, followed by the Volume Algorithm. For the latter, we use the implementation in COIN-OR Vol (see `https://github.com/coin-or/Vol`, and the overview document "An implementation of the Volume Algorithm" by F. Barahona and L. Ladanyi in the same repository).

There are two Lagrangean subproblems to solve in each iteration of both the dual ascent and the volume procedures. We solve the minimum spanning tree subproblem in the original graph $G = (V, E)$ using the efficient implementation of Kruskal's algorithm in COIN-OR LEMON 1.3.1 (Dezső et al., 2011), while we solve the fixed cardinality stable set subproblem in the conflict graph $H = (E, C)$ with a branch-and-cut algorithm, implemented using the Gurobi 9.5.1 solver.

We reinforce formulation $(11) - (13)$ with two further classes of valid inequalities from the classic stable set polytope, exactly as first presented by Samer and Urrutia (2015) for the MSTCC problem. Namely, odd-cycle inequalities

$$\sum_{u \in U} y_u \leq \frac{|U| - 1}{2}, \quad \text{for each } U \subset E \text{ inducing an odd-cycle in } H, \tag{19}$$

are added dynamically using the separation algorithm of (Gerards and Schrijver, 1986, Remark 1), while maximal clique inequalities

$$\sum_{u \in Q} y_u \leq 1, \quad \text{for each } Q \subset E \text{ inducing a maximal clique in } H, \tag{20}$$

are enumerated *a priori* using the algorithm of Tomita et al. (2006), since this can be done efficiently over the MSTCC benchmark instances. The interested reader is referred to Samer and Urrutia (2015), as well as the eminently readable tutorial by Rebennack et al. (2012).

4.2 Experimental setup and benchmark instances

Our computational evaluation was performed on a desktop machine with an Intel® Core™ i5-8400 processor, with 6 CPU cores at 2.80GHz, and 16GB of RAM, runnning GNU/Linux kernel 5.4.0 under the Ubuntu 18.04.1 distribution. All the code is compiled with g++ 7.5.0, and we consider a numerical precision of 10^{-10}. We limit the execution time to 3600 seconds, allowing the dual ascent procedure to run for at most 1800 seconds, and the volume algorithm to run for the remaining time.

After preliminary experiments with the different algorithm parameters, we considered that the following combination exhibits better performance. Dual ascent follows the first maximal ascent direction available in each iteration (instead of identifying the steepest ascent). The volume algorithm implementation from COIN-OR is used with default parameters, except for screen log settings and warm-starting with the multipliers found by dual ascent. Gurobi 9.5.1 is used with default settings, except for screen log settings and switches to indicate the presence of the callback for user cuts. Odd-cycle inequalities are generated only at the root node of the enumeration tree, with the following strategy for balancing bound quality and cut pool size. When separating a relaxation solution, only the most violated cut and those close to being orthogonal to it are added; we accept hyperplanes having inner product of 0.01 or less with the most violated one.

There are two sets of benchmark instances for evaluating MSTCC algorithms. The original one was proposed by Zhang et al. (2011), and more recently Carrabs et al. (2021)

introduced a new collection. The total number of instances can be misleading, as only a small fraction correspond to interesting (*i.e.* computationally challenging) problems. Moreover, it is not possible to discriminate the hard ones by the input size, especially in the latter collection. More specifically, the available problem instances fall into three categories.

i. *Type 1* instances in Zhang et al. (2011): 23 instances, most of which are difficult; 12 still have an open optimality gap in the experiments discussed in the literature.

ii. *Type 2* instances in Zhang et al. (2011): 27 instances, all of which are trivial; the preprocessing algorithm of Samer and Urrutia (2015) solves (or reduces to a classic MST problem without conflicts) all of them in negligible time.

iii. Instances introduced by Carrabs et al. (2021): 180 instances, 107 of which (spanning each group of the collection, ordered by $|E|$) are easily solved within few seconds. The remaining 73 instances are interesting. The collection was only considered in that original work and continuing research from the same group (Carrabs and Gaudioso, 2021; Carrabs et al., 2019).

In summary, only instances in (i) and less than half of the large collection in (iii) serve the purpose of benchmarking MSTCC algorithms, in our opinion. Our discussion contemplates both benchmarks in full, but we choose to include full numerical results for the instances in (i) in the next section, while longer tables corresponding to (iii) are present in Appendix A (online supplement).

4.3 Numerical results

We present the information on bound quality and computing time for three classes of dual bounds: the combinatorial bound corresponding to the kstab relaxation (also the first subproblem solved in LD-davol), the LP relaxation bound, and the LD bound, *i.e.* the approximation of ζ by LD-davol. For a fair, unbiased comparison, note that the linear program whose bound we refer by LP is also reinforced with odd-cycle and clique inequalities in $(19) - (20)$.

Table 1: Results attained over hard instances in the original benchmark.

Instance		KSTAB		LP		LD-davol			
ID	OPT	Bound	Time (s)	Bound	Time (s)	Bound	Time (s)	% above LP	% from OPT
z50-200-199	708	612	0.0	706	0.0	705	1.2	-0.14	0.4
z50-200-398	770	652	0.0	770	0.1	770	1.4	0	0
z50-200-597	917	726	0.0	876	0.1	900	12.7	2.74	1.9
z50-200-995	1324	1164	0.3	1037	0.0	1251	315.9	20.64	5.5
z100-300-448	4041	3440	0.0	4038	0.6	4037	5.0	-0.02	0.1
z100-300-897	5658	4785	0.0	5070	0.4	5371	1402.2	5.94	5.1
z100-300-1344	6635.4*	6970	563.1	5479	0.2	6970	3602.9	27.21	**-5.0**
z100-500-1247	4275	3454	0.0	4275	0.7	4275	10.0	0.02	0
z100-500-2495	5997	5022	0.1	5363	0.4	5693	2225.9	6.15	5.1
z100-500-3741	6707.8*	6101	2.5	5830	0.3	6101	3609.2	4.65	9.0
z100-500-6237	7729.3*	**8506**†	1800.0	6789	0.3	-	-	-	-
z100-500-12474	10560.2*	10506†	1800.0	9008	1.3	-	-	-	-
z200-600-1797	13171.2*	12213	0.1	12580	5.5	12993	3603.7	3.28	1.4
z200-600-3594	17595.0*	**17785**†	1800.0	14763	2.5	-	-	-	-
z200-800-3196	20941.5*	18477	0.0	20002	5.0	20437	3609.3	2.17	2.4
z200-800-6392	26526.7*	**27124**†	1800.0	22923	3.3	-	-	-	-
z200-800-9588	30634.2*	**31132**†	1800.0	27616	2.5	-	-	-	-
z200-800-15980	36900.2*	34648†	1800.0	32050	1.6	-	-	-	-
z300-1000-4995	51398.4*	**51621**†	1800.0	45599	10.5	-	-	-	-
z300-1000-9990	61878.9*	61732†	1800.0	54593	16.4	-	-	-	-

Table 1 covers *type 1* instances in the original benchmark of Zhang et al. (2011) (apart from three that could be identified efficiently as infeasible in previous works). In this set, a problem defined on a graph (V, E) and conflict set C has identifier `z|V|-|E|-|C|`. Tables 2, 3, 4, and 5 in Appendix A (online supplement) contain the corresponding results over instances proposed by Carrabs et al. (2021). The second column in each table contains the instance optimal value, or the best dual bound reported in the literature (we mark instances with unknown optimal solution with an asterisk*).

Given the time limit that we allocate to the dual algorithms, we only report LD-davol results for instances where the kstab bound is computed within 1800 seconds. If that is not the case, we report the available dual bound for the fractional kstab relaxation and the corresponding entry appears with a mark ($z^\dagger$). Moreover, we use boldface ($\mathbf{z}^\dagger$) in case this bound is actually stronger than those previously appearing in the literature. We remark that ζ, or any Lagrangean bound, is greater than or equal to the LP bound. Nevertheless, in the seven cases where the approximation attained by LD-davol is an inferior bound, a negative number appears in the *% above LP* column. Finally, if the Lagrangean bound is better than the previously best known bound (applies only to instances with unknown optima), a negative value in bold appears in the *% from OPT* column.

We read from Table 1 that the Lagrangean bound can be up to 27.21% above the LP relaxation one. We consider it even more remarkable that LD-davol computes ζ exactly and this actually matches the optimum in 2 instances in this collection, and in 73 instances out of 180 in the remaining tables. Otherwise, the bound is within 9% of the optimum. This figure actually corresponds to one of two outliers in this table, where LD-davol does not improve on the initial kstab bound; disregarding instance `z100-500-3741`, the bound is within 5.5% of the optimum across all experiments.

Concerning the instances introduced by Carrabs et al. (2021), the bound is within

(i) 2.1% of the optimum in instances with 25 vertices ($60 \leq |E| \leq 120$, $18 \leq |C| \leq 500$);

(ii) 4.4% of the optimum in instances with 50 vertices ($245 \leq |E| \leq 490$, $299 \leq |C| \leq 8387$);

(iii) 2.6% of the optimum in instances with 75 vertices ($555 \leq |E| \leq 1110$, $1538 \leq |C| \leq 43085$);

(iv) 0.1% of the optimum in instances with 100 vertices ($990 \leq |E| \leq 1980$, $4896 \leq |C| \leq 137145$).

The initial kstab bound is the only one computed in 8 out of 20 instances in Table 1 (45 out of 180 instances in the remaining tables). Nevertheless, in 5 of these cases (respectively, in 39 of those 45) it is stronger than the previously known best bound. Note that, even though the machines and implementations cannot be compared directly, the 1800 second time limit set for this initial combinatorial relaxation is much lower than the standard (5000s) used in the literature of the MSTCC problem.

The main negative remark is as expected: the LD bound might be too expensive to compute. Even though it can be efficiently determined in a large number of instances (*e.g.* at most sixty seconds for 96 cases across all tables), the execution of LD-davol is terminated due to the time limit in 4 instances appearing in Table 1 (29 appearing in the other tables). An intuitive rule of thumb is that LD-davol yields stronger bounds in reasonable time as long as the combinatorial relaxation bound (the initial kstab problem) can be computed in reasonable time.

We avoid direct comparison of implementations/solvers altogether. As declared in the beginning of this section, our goal is to assess the strength and practicality of our ideas: exploring fixed cardinality stable sets and the reformulation by LD. It should be clear from our numerical results that the method yields high-quality dual bounds in the allotted computing time. It is probably not suited for embedding in a branch-and-bound scheme without successful work on heuristic aspects, namely: learning effective LD-davol parameters – especially setting a time limit in each node, implementing *repair heuristics* to search for primal solutions from the sequence of fractional points produced by the Volume Algorithm, as well as designing local search methods to explore neighbourhoods of the kstab and spanning tree solutions found during the Lagrangean subproblems. (Note that Carrabs and Gaudioso (2021) describe successful results from such a Lagrangean heuristic derived from the integral relaxation scheme discussed in Section 3.1.) Alternatively, one could experiment with calling LD-davol selectively in a branch-and-cut framework to strengthen dual bounds, *e.g.* when an incumbent solution is found, or when the optimality gap is not decreasing effectively.

Additional ideas that we leave for future work include improving the kstab subproblem solver, fine-tuning the Volume Algorithm to perform faster, and experimenting with different dual methods *e.g.* the sophisticated framework for subgradient optimization made available by Frangioni et al. (2017), or, more ambitiously, the approximate solution using nonsmooth optimization techniques with *inexact* function/subgradient evaluation (de Oliveira et al., 2014).

5 Concluding remarks

Stable spanning trees are not only interesting structures in combinatorial optimization, but pose a computationally challenging problem. We explore a new relaxation (fixed cardinality stable sets) to present polyhedral results and to derive stronger Lagrangean bounds. The latter builds on a careful analysis of different relaxation schemes, both old and new. Our Lagrangean decomposition (LD) bounds are also evaluated in practice, using a dual method comprising an original dual-ascent initialization followed by the Volume Algorithm. Finally, we also made great efforts to offer a high-quality, useful, open-source software in a free repository.

The LD bound actually matches the optimum in 75 out of 200 benchmark instances.

We verify that, in at least 146 of these instances (where the kstab subproblem can be solved fast enough), the LD bound is within 5.5% of the optimum or the best known bound. In 44 of the remaining instances, the initial combinatorial bound from kstabs at least improves the previously known best bounds.

We reinforce the position put forth at the end of the introduction. In light of the progress in MILP computation, it seems worthwhile to further investigate the strategy of LD based on harder subproblems, possibly replacing the common sense boundary of weakly NP-hard choices by the weaker requirement that our choice be *computationally tractable*.

References

A. Assad and W. Xu. The quadratic minimum spanning tree problem. *Naval Research Logistics (NRL)*, 39(3):399–417, 1992. ISSN 1520-6750. URL `https://doi.org/10.1002/1520-6750(199204)39:3<399::AID-NAV3220390309>3.0.CO;2-0`.

L. Bahiense, N. Maculan, and C. Sagastizábal. The volume algorithm revisited: relation with bundle methods. *Mathematical Programming*, 94(1):41–69, 2002. URL `https://doi.org/10.1007/s10107-002-0357-3`.

F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000. URL `https://doi.org/10.1007/s101070050002`.

D. Bertsimas and J. Dunn. *Machine learning under a modern optimization lens*. Dynamic Ideas LLC, Charlestown, 2019.

D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813 – 852, 2016. URL `https://doi.org/10.1214/15-AOS1388`.

D. Bertsimas, J. Pauphilet, and B. V. Parys. Sparse Regression: Scalable Algorithms and Empirical Performance. *Statistical Science*, 35(4):555 – 578, 2020. URL https://doi.org/10.1214/19-STS701.

O. Bilde and J. Krarup. Sharp lower bounds and efficient algorithms for the simple plant location problem. In P. Hammer, E. Johnson, B. Korte, and G. Nemhauser, editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 79–97. Elsevier, 1977. URL https://doi.org/10.1016/S0167-5060(08)70728-3.

O. Briant, C. Lemaréchal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical programming*, 113(2):299–344, 2008. URL https://doi.org/10.1007/s10107-006-0079-z.

F. Carrabs and M. Gaudioso. A lagrangian approach for the minimum spanning tree problem with conflicting edge pairs. *Networks*, 78(1):32–45, 2021. URL https://doi.org/10.1002/net.22009.

F. Carrabs, C. Cerrone, and R. Pentangelo. A multiethnic genetic approach for the minimum conflict weighted spanning tree problem. *Networks*, 74(2):134–147, 2019. URL https://doi.org/10.1002/net.21883.

F. Carrabs, R. Cerulli, R. Pentangelo, and A. Raiconi. Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach. *Annals of Operations Research*, 298 (1):65–78, 2021. URL https://doi.org/10.1007/s10479-018-2895-y.

A. Darmann, U. Pferschy, and J. Schauer. Determining a minimum spanning tree with disjunctive constraints. In F. Rossi and A. Tsoukias, editors, *Algorithmic Decision Theory*, volume 5783 of *Lecture Notes in Computer Science*, pages 414–423. Springer Berlin Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-642-04428-1_36.

A. Darmann, U. Pferschy, J. Schauer, and G. J. Woeginger. Paths, trees and matchings under disjunctive constraints. *Discrete Applied Mathematics*, 159(16):1726 – 1735, 2011. URL http://dx.doi.org/10.1016/j.dam.2010.12.016.

W. de Oliveira, C. Sagastizábal, and C. Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, 148(1): 241–277, 2014. URL https://doi.org/10.1007/s10107-014-0809-6.

A. Dessmark, J. Jansson, A. Lingas, E.-M. Lundell, and M. Persson. On the approximability of maximum and minimum edge clique partition problems. *International Journal of Foundations of Computer Science*, 18(02):217–226, 2007. URL https://doi.org/10.1142/S0129054107004656.

B. Dezső, A. Jüttner, and P. Kovács. LEMON – an Open Source C++ Graph Template Library. *Electronic Notes in Theoretical Computer Science*, 264(5):23 – 45, 2011. ISSN 1571-0661. URL http://dx.doi.org/10.1016/j.entcs.2011.06.003.

J. Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1(1): 127–136, 1971. URL https://doi.org/10.1007/BF01584082.

D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/170260.

A. Frangioni, B. Gendron, and E. Gorgone. On the computational efficiency of subgradient methods: a case study with lagrangian bounds. *Mathematical Programming Computation*, 9(4):573–604, 2017. URL https://doi.org/10.1007/s12532-017-0120-7.

A. Gerards and A. Schrijver. Matrices with the Edmonds–Johnson property. *Combinatorica*, 6(4):365–379, 1986. ISSN 0209-9683. URL http://dx.doi.org/10.1007/BF02579262.

M. Guignard. Lagrangean relaxation. *Top*, 11(2):151–200, 2003. URL https://doi.org/10.1007/BF02579036.

M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming*, 39:215–228, 1987. URL https://doi.org/10.1007/BF02592954.

M. Guignard and M. B. Rosenwein. An application-oriented guide for designing lagrangean dual ascent algorithms. *European Journal of Operational Research*, 43(2):197–205, 1989. ISSN 0377-2217. URL `https://doi.org/10.1016/0377-2217(89)90213-0`.

R. Lougee-Heimer. The common optimization interface for operations research: promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66, 2003. URL `https://doi.org/10.1147/rd.471.0057`.

G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, volume 55 of *Wiley-Interscience series in discrete mathematics and optimization*. John Wiley & Sons, Inc, 1999. URL `dx.doi.org/10.1002/9781118627372`.

S. Rebennack, G. Reinelt, and P. M. Pardalos. A tutorial on branch and cut algorithms for the maximum stable set problem. *International Transactions in Operational Research*, 19(1-2):161–199, 2012. URL `https://doi.org/10.1111/j.1475-3995.2011.00805.x`.

C. Ribeiro and M. Minoux. Solving hard constrained shortest path problems by lagrangean relaxation and branch-and-bound algorithms. *Methods of Operations Research*, 53: 303–316, 1986.

P. Samer and D. Haugland. Fixed cardinality stable sets. *Discrete Applied Mathematics*, 303:137–148, 2021. ISSN 0166-218X. URL `https://doi.org/10.1016/j.dam.2021.01.019`.

P. Samer and D. Haugland. Towards stronger Lagrangean bounds for stable spanning trees. In C. Büsing and A. M. C. A. Koster, editors, *Proceedings of the 10th International Network Optimization Conference, INOC 2022, Aachen, Germany, June 7-10, 2022*, pages 29–33. OpenProceedings.org, 2022. URL `https://doi.org/10.48786/inoc.2022.06`.

P. Samer and S. Urrutia. A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41–55, 2015. ISSN 1862-4472. URL `https://doi.org/10.1007/s11590-014-0750-x`.

F. Shepardson and R. E. Marsten. A lagrangean relaxation algorithm for the two duty period scheduling problem. *Management Science*, 26(3):274–281, 1980. URL `https://doi.org/10.1287/mnsc.26.3.274`.

E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363 (1):28 – 42, 2006. ISSN 0304-3975. URL `http://dx.doi.org/10.1016/j.tcs.2006.06.015`.

R. Zhang, S. N. Kabadi, and A. P. Punnen. The minimum spanning tree problem with conflict constraints and its variations. *Discrete Optimization*, 8(2):191 – 205, 2011. ISSN 1572-5286. URL `https://doi.org/10.1016/j.disopt.2010.08.001`.

A Further numerical results

Tables 2, 3, 4, and 5 in this appendix (online supplement) contain the results corresponding to instances proposed by Carrabs et al. (2021). Since this set includes five different instances of each combination of problem dimensions, those authors identify each problem by `|V|_|E|_|C|_r`, where `r` is the seed used in a random number generator.

The discussion of these additional results is contained in Section 4.3.

Table 2: Results attained over instances with 25 vertices in the second benchmark.

Instance		KSTAB		LP		LD-davol			
ID	OPT	Bound	Time (s)	Bound	Time (s)	Bound	Time (s)	% above LP	% from OPT
25_60_18_1	347	332	0.0	347	0.0	347	0.2	0	0
25_60_18_7	389	365	0.0	389	0.0	389	0.1	0	0
25_60_18_13	353	337	0.0	353	0.0	353	0.1	0	0
25_60_18_19	346	341	0.0	346	0.0	346	0.1	0	0
25_60_18_25	336	326	0.0	336	0.0	336	0.1	0	0
25_60_71_31	381	367	0.0	380	0.0	380	0.5	0	0.3
25_60_71_37	390	369	0.0	382	0.0	382	0.8	0	2.1
25_60_71_43	372	353	0.0	372	0.0	372	0.3	0	0
25_60_71_49	357	346	0.0	357	0.0	357	0.3	0	0
25_60_71_55	406	387	0.0	406	0.0	406	0.3	0	0
25_60_124_61	385	380	0.0	385	0.0	385	0.8	0	0
25_60_124_67	432	427	0.0	432	0.0	432	0.5	0	0
25_60_124_73	458	446	0.0	451	0.0	458	3.4	1.55	0
25_60_124_79	400	383	0.0	399	0.0	400	2.4	0.25	0
25_60_124_85	420	407	0.0	419	0.0	420	1.5	0.24	0
25_90_41_91	311	300	0.0	311	0.0	311	0.3	0	0
25_90_41_97	306	298	0.0	306	0.0	306	0.1	0	0
25_90_41_103	299	285	0.0	299	0.0	299	0.2	0	0
25_90_41_109	297	288	0.0	297	0.0	297	0.2	0	0
25_90_41_115	318	313	0.0	318	0.0	318	0.1	0	0
25_90_161_121	305	293	0.0	305	0.0	305	0.4	0	0
25_90_161_127	339	332	0.0	339	0.0	339	0.4	0	0
25_90_161_133	344	333	0.0	344	0.0	344	0.4	0	0
25_90_161_139	329	308	0.0	328	0.0	328	0.6	0	0.3
25_90_161_145	326	307	0.0	325	0.0	325	0.8	0	0.3
25_90_281_151	349	330	0.0	349	0.0	349	2.1	0	0
25_90_281_157	385	355	0.0	373	0.0	379	14.5	1.61	1.6
25_90_281_163	335	325	0.0	333	0.0	334	3.7	0.30	0.3
25_90_281_169	348	335	0.0	338	0.0	344	6.2	1.78	1.1
25_90_281_175	357	342	0.0	357	0.0	357	2.5	0	0
25_120_72_181	282	280	0.0	282	0.0	282	0.3	0	0
25_120_72_187	294	284	0.0	294	0.0	294	0.3	0	0
25_120_72_193	284	283	0.0	284	0.0	284	0.3	0	0
25_120_72_199	281	267	0.0	281	0.0	281	0.3	0	0
25_120_72_205	292	289	0.0	292	0.0	292	0.3	0	0
25_120_286_211	321	315	0.0	321	0.0	321	0.7	0	0
25_120_286_217	317	310	0.0	317	0.0	317	1.1	0	0
25_120_286_223	284	283	0.0	284	0.0	284	0.5	0	0
25_120_286_229	311	304	0.0	311	0.0	311	0.7	0	0
25_120_286_235	290	283	0.0	290	0.0	290	0.4	0	0
25_120_500_241	329	309	0.0	322	0.0	326	4.3	1.24	0.9
25_120_500_247	339	327	0.0	330	0.0	334	6.3	1.21	1.5
25_120_500_253	368	361	0.0	363	0.0	367	24.7	1.10	0.3
25_120_500_259	311	304	0.0	308	0.0	310	8.5	0.65	0.3
25_120_500_265	321	315	0.0	321	0.0	321	2.6	0	0

Table 3: Results attained over instances with 50 vertices in the second benchmark.

Instance		KSTAB		LP		LD-davol			
ID	OPT	Bound	Time (s)	Bound	Time (s)	Bound	Time (s)	% above LP	% from OPT
50_245_299_271	619	573	0.0	619	0.0	619	1.3	0	0
50_245_299_277	604	593	0.0	604	0.0	604	0.9	0	0
50_245_299_283	634	631	0.0	634	0.0	634	0.7	0	0
50_245_299_289	616	600	0.0	616	0.0	616	1.2	0	0
50_245_299_295	595	577	0.0	595	0.0	595	1.3	0	0
50_245_1196_301	678	663	0.0	670	0.0	674	124.9	0.60	0.6
50_245_1196_307	681	652	0.0	669	0.1	678	134.2	1.35	0.4
50_245_1196_313	709	669	0.0	685	0.0	695	184.4	1.46	2.0
50_245_1196_319	639	625	0.0	637	0.0	637	47.9	0	0.3
50_245_1196_325	681	656	0.0	663	0.0	672	125.9	1.36	1.3
50_245_2093_331	791.20*	758	1.5	714	0.0	774	3607.0	8.40	2.2
50_245_2093_337	835	788	1.2	739	0.0	803	3601.6	8.66	3.8
50_245_2093_343	773.23*	742	3.3	699	0.1	762	3609.3	9.01	1.5
50_245_2093_349	820.02*	769	1.5	721	0.0	784	3603.9	8.74	4.4
50_245_2093_355	769	739	0.8	715	0.0	758	3282.8	6.01	1.4
50_367_672_361	570	545	0.0	570	0.0	570	1.9	0	0
50_367_672_367	561	540	0.0	561	0.1	561	1.8	0	0
50_367_672_373	573	565	0.0	573	0.0	573	1.7	0	0
50_367_672_379	560	551	0.0	560	0.0	560	1.9	0	0
50_367_672_385	549	539	0.0	549	0.0	549	1.8	0	0
50_367_2687_391	612	589	0.0	601	0.1	607	228.9	1.00	0.8
50_367_2687_397	615	593	0.0	600	0.1	608	254.4	1.33	1.1
50_367_2687_403	587	566	0.0	580	0.1	585	129.7	0.86	0.3
50_367_2687_409	634	604	0.0	612	0.0	626	279.4	2.29	1.3
50_367_2687_415	643	623	0.1	638	0.1	640	108.4	0.31	0.5
50_367_4702_421	701.26*	690	7.7	647	0.1	690	3601.3	6.65	1.6
50_367_4702_427	719.45*	696	1.9	664	0.1	703	3608.4	5.87	2.3
50_367_4702_433	723.89*	721	13.4	676	0.1	721	3606.4	6.66	0.4
50_367_4702_439	669.84*	668	14.7	623	0.1	668	3601.0	7.22	0.3
50_367_4702_445	737.31*	723	5.7	687	0.1	725	3609.0	5.53	1.7
50_490_1199_451	548	532	0.0	548	0.1	548	2.2	0	0
50_490_1199_457	530	514	0.0	530	0.1	530	2.1	0	0
50_490_1199_463	549	541	0.0	549	0.0	549	2.7	0	0
50_490_1199_469	540	528	0.0	540	0.1	540	2.2	0	0
50_490_1199_475	540	527	0.0	540	0.0	540	2.4	0	0
50_490_4793_481	594	573	0.1	586	0.1	592	294.8	1.02	0.3
50_490_4793_487	579	554	0.0	564	0.1	570	323.7	1.06	1.6
50_490_4793_493	589	574	0.0	585	0.1	587	235.0	0.34	0.3
50_490_4793_499	577	562	0.1	567	0.1	571	137.7	0.71	1.0
50_490_4793_505	592	581	0.2	583	0.1	589	262.5	1.03	0.5
50_490_8387_511	631.43*	615	2.8	597	0.2	620	3604.6	3.85	1.8
50_490_8387_517	626.72*	613	10.1	589	0.2	613	3609.9	4.07	2.2
50_490_8387_523	658.38*	647	6.6	615	0.1	647	3600.8	5.20	1.7
50_490_8387_529	662.22*	655	11.7	618	0.1	655	3609.3	5.99	1.1
50_490_8387_535	641.31*	635	8.3	601	0.1	635	3607.7	5.66	1.0

Table 4: Results attained over instances with 75 vertices in the second benchmark.

Instance		KSTAB		LP		LD-davol			
ID	OPT	Bound	Time (s)	Bound	Time (s)	Bound	Time (s)	% above LP	% from OPT
75_555_1538_541	868	838	0.0	868	0.2	868	5.0	0	0
75_555_1538_547	871	858	0.0	871	0.2	871	3.8	0	0
75_555_1538_553	838	828	0.0	838	0.2	838	4.7	0	0
75_555_1538_559	855	830	0.0	855	0.2	855	3.6	0	0
75_555_1538_565	857	831	0.0	857	0.2	857	3.5	0	0
75_555_6150_571	1023.72*	1018	16.2	966	0.1	1018	3609.1	5.38	0.6
75_555_6150_577	1008.82*	997	3.7	958	0.1	997	3606.2	4.07	1.2
75_555_6150_583	987.31*	985	27.6	932	0.2	985	3603.2	5.69	0.2
75_555_6150_589	985.64*	958	3.0	937	0.3	960	3608.5	2.45	2.6
75_555_6150_595	962.55*	953	5.0	921	0.2	953	3600.4	3.47	1.0
75_555_10762_601	1054.25*	**1098**†	1800.0	1004	0.4	-	-	-	-
75_555_10762_607	1069.51*	**1107**†	1800.0	1022	0.3	-	-	-	-
75_555_10762_613	1040.97*	**1069**†	1800.0	985	0.4	-	-	-	-
75_555_10762_619	1006.30*	**1036**†	1800.0	960	0.3	-	-	-	-
75_555_10762_625	1046.43*	**1081**†	1800.0	997	0.5	-	-	-	-
75_832_3457_631	798	779	0.0	798	0.4	798	6.6	0	0
75_832_3457_637	821	801	0.0	820	0.5	820	8.6	0	0.1
75_832_3457_643	816	797	0.0	816	0.2	815	7.7	-0.12	0.1
75_832_3457_649	820	805	0.0	820	0.4	820	8.6	0	0
75_832_3457_655	815	800	0.0	815	0.4	815	8.6	0	0
75_832_13828_661	873.83*	865	5.5	839	0.3	865	3601.6	3.10	1.0
75_832_13828_667	901.81*	889	6.6	873	0.6	889	3601.5	1.83	1.4
75_832_13828_673	873.67*	858	5.1	843	0.3	858	3607.2	1.78	1.8
75_832_13828_679	885.57*	879	23.9	852	0.2	879	3605.6	3.17	0.7
75_832_13828_685	886.87*	875	5.4	856	0.2	875	3605.1	2.22	1.3
75_832_24199_691	949.55*	**965**†	1800.0	923	0.5	-	-	-	-
75_832_24199_697	907.80*	**921**†	1800.0	884	0.5	-	-	-	-
75_832_24199_703	910.00*	**925**†	1800.0	886	0.5	-	-	-	-
75_832_24199_709	943.98*	**967**†	1800.0	922	0.4	-	-	-	-
75_832_24199_715	956.31*	**974**†	1800.0	933	0.4	-	-	-	-
75_1110_6155_721	787	776	0.0	787	0.3	787	12.6	0	0
75_1110_6155_727	785	771	0.0	785	1.0	785	13.7	0	0
75_1110_6155_733	783	773	0.0	783	3.6	783	8.9	0	0
75_1110_6155_739	784	772	0.0	784	0.3	784	7.8	0	0
75_1110_6155_745	797	782	0.0	797	0.4	797	13.2	0	0
75_1110_24620_751	846.69*	838	4.9	826	0.3	838	3602.9	1.45	1.0
75_1110_24620_757	829.23*	828	19.8	805	0.4	828	3606.9	2.86	0.1
75_1110_24620_763	841.54*	847	93.8	817	0.2	847	3605.9	3.67	**-0.6**
75_1110_24620_769	841.62*	836	26.4	814	0.3	836	3603.4	2.70	0.7
75_1110_24620_775	835.04*	830	18.3	813	0.4	830	3606.3	2.09	0.6
75_1110_43085_781	868.72*	**882**†	1800.0	856	0.6	-	-	-	-
75_1110_43085_787	853.45*	**861**†	1800.0	840	0.8	-	-	-	-
75_1110_43085_793	884.67*	**890**†	1800.0	867	0.6	-	-	-	-
75_1110_43085_799	853.00*	**864**†	1800.0	841	0.7	-	-	-	-
75_1110_43085_805	853.98*	**862**†	1800.0	844	0.8	-	-	-	-

Table 5: Results attained over instances with 100 vertices in the second benchmark.

Instance		KSTAB		LP		LD-davol			
ID	OPT	Bound	Time (s)	Bound	Time (s)	Bound	Time (s)	% above LP	% from OPT
100_990_4896_811	1119	1097	0.0	1119	1.2	1118	15.1	-0.09	0.1
100_990_4896_817	1137	1115	0.0	1137	0.7	1137	20.4	0	0
100_990_4896_823	1113	1076	0.0	1113	2.1	1113	25.1	0	0
100_990_4896_829	1110	1086	0.0	1110	1.3	1110	17.4	0	0
100_990_4896_835	1090	1063	0.0	1090	1.4	1089	17.5	-0.09	0.1
100_990_19583_841	1249.38*	**1282**†	1800.0	1206	0.8	-	-	-	-
100_990_19583_847	1225.76*	**1242**†	1800.0	1171	0.9	-	-	-	-
100_990_19583_853	1215.00*	**1236**†	1800.0	1170	0.5	-	-	-	-
100_990_19583_859	1264.17*	**1284**†	1800.0	1219	0.5	-	-	-	-
100_990_19583_865	1257.27*	**1278**†	1800.1	1214	0.4	-	-	-	-
100_990_34269_871	1262.00*	**1266**†	1800.0	1233	0.9	-	-	-	-
100_990_34269_877	1290.68*	**1293**†	1800.0	1265	1.3	-	-	-	-
100_990_34269_883	1318.54*	1318†	1800.0	1295	1.3	-	-	-	-
100_990_34269_889	1282.38*	1275†	1800.0	1249	1.1	-	-	-	-
100_990_34269_895	1304.45*	**1311**†	1800.0	1273	0.6	-	-	-	-
100_1485_11019_901	1079	1054	0.1	1079	2.3	1078	41.5	-0.09	0.1
100_1485_11019_907	1056	1038	0.1	1056	1.7	1055	39.6	-0.09	0.1
100_1485_11019_913	1059	1042	0.1	1059	0.9	1059	28.8	0	0
100_1485_11019_919	1046	1030	0.1	1046	1.5	1046	32.8	0	0
100_1485_11019_925	1072	1040	0.1	1072	2.9	1072	58.7	0	0
100_1485_44075_931	1143.95*	**1152**†	1800.0	1114	1.0	-	-	-	-
100_1485_44075_937	1143.61*	**1155**†	1800.0	1109	0.6	-	-	-	-
100_1485_44075_943	1137.62*	**1144**†	1800.0	1109	2.3	-	-	-	-
100_1485_44075_949	1136.90*	**1142**†	1800.0	1113	1.7	-	-	-	-
100_1485_44075_955	1134.63*	**1145**†	1800.0	1106	0.5	-	-	-	-
100_1485_77131_961	1164.44*	**1167**†	1800.0	1155	3.5	-	-	-	-
100_1485_77131_967	1168.20*	**1170**†	1800.0	1156	2.4	-	-	-	-
100_1485_77131_973	1180.02*	**1184**†	1800.0	1171	3.1	-	-	-	-
100_1485_77131_979	1183.53*	**1185**†	1800.0	1174	3.1	-	-	-	-
100_1485_77131_985	1159.25*	1157†	1801.1	1152	2.2	-	-	-	-
100_1980_19593_991	1031	1023	0.1	1031	21.5	1031	56.4	0	0
100_1980_19593_997	1036	1028	0.1	1035	1.4	1036	53.0	0.10	0
100_1980_19593_1003	1024	1016	0.1	1024	3.2	1024	60.5	0	0
100_1980_19593_1009	1025	1018	0.1	1025	3.7	1025	45.0	0	0
100_1980_19593_1015	1028	1018	0.1	1028	2.9	1028	45.8	0	0
100_1980_78369_1021	1096.83*	**1107**†	1800.0	1082	1.5	-	-	-	-
100_1980_78369_1027	1065.64*	**1069**†	1800.1	1048	7.2	-	-	-	-
100_1980_78369_1033	1087.39*	**1096**†	1800.0	1069	1.6	-	-	-	-
100_1980_78369_1039	1081.26*	**1092**†	1800.1	1065	1.7	-	-	-	-
100_1980_78369_1045	1084.09*	**1094**†	1800.0	1068	2.4	-	-	-	-
100_1980_137145_1051	1098.61*	**1101**†	1800.0	1094	5.7	-	-	-	-
100_1980_137145_1057	1126.27*	1127†	1800.0	1121	8.0	-	-	-	-
100_1980_137145_1063	1111.27*	1112†	1800.0	1106	5.0	-	-	-	-
100_1980_137145_1069	1114.58*	**1116**†	1800.0	1111	9.6	-	-	-	-
100_1980_137145_1075	1114.07*	1114†	1800.0	1109	5.3	-	-	-	-

5.1 Proof of Theorem 9

(i.) Given that $x_e^r = 0$ and $y_e^r = 1$, increasing λ_e^r corresponds to increasing the dual bound, until alternative optimal solutions where that hypothesis fails are induced. Specifically,

$$z(\lambda^r + \epsilon \mathfrak{e}_e) > z(\lambda^r) \tag{5.1}$$

for all $\epsilon > 0$ such that

$$x^r \in \arg\min \Big\{ \Big(w - (\lambda^r + \epsilon \mathfrak{e}_e) \Big) x : x \in \mathcal{F}_{\text{sp.tree}}(G) \Big\}, \tag{5.2}$$

$$y^r \in \arg\min \Big\{ \, (\lambda^r + \epsilon \mathfrak{e}_e) \, y : y \in \mathcal{F}_{\text{kstab}}(H, |V| - 1) \Big\}. \tag{5.3}$$

As long as ϵ can be made positive, $\epsilon \mathfrak{e}_e$ is a direction of increase from $z(\lambda^r)$. The necessity of conditions (5.2) and (5.3) follows from noting that the contribution of the e-th variables x_e and y_e to z,

$$(w_e - (\lambda_e^r + \epsilon \mathfrak{e}_e)) \, x_e + (\lambda_e^r + \epsilon \mathfrak{e}_e) \, y_e,$$

remains constant as we increase ϵ after x_e joins, or y_e leaves, an optimal solution. For, if $x_e = y_e = 1$, meaning that the coefficient of edge e is attractive enough in (5.2), any further increase $+\epsilon y_e$ is cancelled by $-\epsilon x_e$. Moreover, if $x_e = y_e = 0$, which means that the coefficient of vertex e is no longer attractive enough in (5.3), further increasing ϵ in $(\lambda_e^r + \epsilon \mathfrak{e}_e) \, y_e = 0$ has no effect.

(ii.) To determine ϵ such that we find a breakpoint of z, we use the limiting conditions (5.2), (5.3).

For x^r to no longer be the unique optimum in (5.2), the cost of edge e decreases so much that an alternative solution $\tilde{x} \in \mathcal{F}_{\text{sp.tree}}(G)$ which includes e is determined. Note that $\tilde{x}$ is well-defined, as the choice of edges in a minimum spanning tree where e is fixed *a priori* does not depend on the cost of e (all other costs are kept unchanged). Also note that, since the existing solution is such that $x_e^r = 0$, the cost of $\tilde{x}$ is no less than that of x^r. The difference is precisely ∂_{+e}^r in Definition (16) in the theorem statement.

If $\partial_{+e}^r = 0$, the bound cannot be improved by adjusting λ_e^r, as an alternative minimum spanning tree including e is readily available; equivalently, we should have $\epsilon = 0$ in part (i). If $\partial_{+e}^r > 0$, it is the maximum increase in λ_e^r (*i.e.* decrease in the cost of edge e in the x subproblem) before $\tilde{x}$ becomes optimal and z starts to decrease. That is, enforcing (5.2) yields

$$\epsilon \leq \partial^r_{+e} \,. \tag{5.4}$$

(iii.) For y^r to no longer be the unique optimum in (5.3), the cost of vertex e increases so much that an alternative fixed cardinality stable set $\tilde{y} \in \mathcal{F}_{\text{kstab}}(H, |V|-1)$ which does not include e is determined.

Analogous to the situation in part (ii), $\tilde{y}$ is well-defined because the multipliers corresponding to all other vertices are kept constant: choosing $\tilde{y}$ amounts to finding a minimum cost fixed cardinality stable set in $H - e$. Also, its cost is no less than that of y^r, the existing optimal solution to the y subproblem. The difference is exactly Δ^r_{-e} in Definition (15) in the theorem statement.

If $\Delta^r_{-e} = 0$, no bound improvement by changing λ^r_e is possible, as an alternative fixed cardinality stable set of least cost not including e is readily available; *i.e.* we should have $\epsilon = 0$ in part (i). On the other hand, if $\Delta^r_{-e} > 0$, it is the maximum increase in λ^r_e before $\tilde{y}$ becomes optimal and z stops increasing. That is, enforcing (5.3) yields

$$\epsilon \leq \Delta^r_{-e} \,. \tag{5.5}$$

(iv.) In conclusion, if $\min\left\{\Delta^r_{-e}, \partial^r_{+e}\right\} = 0$, then $\epsilon = 0$ and $\epsilon\mathfrak{e}_e$ fails to be a direction of increase from $z(\lambda^r)$. Otherwise, we combine bounds (5.4) and (5.5) into (5.1):

$$\forall \epsilon > 0, \;\; z(\lambda^r + \min\left\{\Delta^r_{-e}, \partial^r_{+e}\right\} \cdot \mathfrak{e}_e) \geq z(\lambda^r + \epsilon\mathfrak{e}_e),$$

showing that $\lambda^r + \min\left\{\Delta^r_{-e}, \partial^r_{+e}\right\} \cdot \mathfrak{e}_e$ is a breakpoint of z, and $\min\left\{\Delta^r_{-e}, \partial^r_{+e}\right\} \cdot \mathfrak{e}_e$ is a maximal ascent direction. □

5.2 Proof of Theorem 10

The argument is analogous to the one in the previous section, so the proof is significantly streamlined.

Decreasing λ^r_e corresponds to increasing the dual bound, in this case. Hence, $\epsilon(-\mathfrak{e}_e)$ is a direction of increase from $z(\lambda^r)$, as long as ϵ can be made positive in

$$z\big(\lambda^r + \epsilon(-\mathfrak{e}_e)\big) > z\big(\lambda^r\big), \tag{5.6}$$

where

$$x^r \in \arg\min \Big\{ \big[w - (\lambda^r + \epsilon(-\mathfrak{e}_e))\big] x : x \in \mathcal{F}_{\text{sp.tree}}(G) \Big\}, \tag{5.7}$$

$$y^r \in \arg\min \Big\{ \big[\lambda^r + \epsilon(-\mathfrak{e}_e)\big] y : y \in \mathcal{F}_{\text{kstab}}(H, |V|-1) \Big\}. \tag{5.8}$$

For y^r to no longer be the unique optimum in (5.8), the cost of vertex e decreases enough for an alternative solution including e to be determined. Since all other multipliers are kept constant, such point $\tilde{y} \in \mathcal{F}_{\text{kstab}}(H, |V|-1)$ corresponds to a minimum cost stable set of cardinality $|V|-2$ in $H - N[e]$, that is, the conflict graph where the closed neighbourhood of vertex e is removed. As the existing solution is such that $y^r_e = 0$, the cost of $\tilde{y}$ is no less than that of y^r. The difference is precisely Δ^r_{+e} in Definition 17 in the theorem statement.

Now, for x^r to no longer be the unique optimum in (5.7), the cost of edge e increases as far as determining an alternative minimum spanning tree not including e. Let $\tilde{x} \in \mathcal{F}_{\text{sp.tree}}(G)$ denote that point, which corresponds to a minimum spanning tree in $G - e$, since all other multipliers are held constant. The cost of $\tilde{x}$ is no less than that of x^r, the existing optimal solution to the x subproblem. The difference is exactly ∂^r_{-e} in Definition 18 in the theorem statement.

If $\min\big\{\Delta^r_{+e}, \partial^r_{-e}\big\} = 0$, then $\epsilon = 0$, and $\epsilon(-\mathfrak{e}_e)$ fails to be a direction of increase from $z(\lambda^r)$. Otherwise, we have

$$\forall \epsilon > 0,\ z\big(\lambda^r + \min\big\{\Delta^r_{+e}, \partial^r_{-e}\big\} \cdot (-\mathfrak{e}_e)\big) \geq z\big(\lambda^r + \epsilon(-\mathfrak{e}_e)\big),$$

showing that $\lambda^r + \min\big\{\Delta^r_{+e}, \partial^r_{-e}\big\} \cdot (-\mathfrak{e}_e)$ is a breakpoint of z, and $\min\big\{\Delta^r_{+e}, \partial^r_{-e}\big\} \cdot (-\mathfrak{e}_e)$ is a maximal ascent direction. $\square$

Errata for

Polyhedra and algorithms for problems bridging notions of connectivity and independence

Thesis for the degree of Philosophiae Doctor (PhD)

at the University of Bergen

17.11.2023

Candidate

20.11.2023 Ingrid Solhøy

Faculty

Errata

Page v Typo: “the the” – corrected to “the”

Page 7 Grammatical error: “tell us” – corrected to “tells us”

Page 22 Typo: “Is suffices” – corrected to “It suffices”

Page 22 Misspelling: “edge-variables” – corrected to “edge variables”

Page 25 Missing word: “experiment” – corrected to “experiment with”

Page 26 Typo: “Is should” – corrected to “It should”

Page 107 Wrong preposition: “on the space” – corrected to “in the space”

Page 107 Grammatical error: “attain” – corrected to “attains”

Page 118 Misspelling: “variable” – corrected to “variables”

Page 118 Wrong preposition: “relying in” – corrected to “relying on”

Page 120 Missing word: “carried over” – corrected to “carried out over”

Page 121 Typo: “intances” – corrected to “instances”

Page 121 Wrong preposition: “in a desktop machine” – corrected to “on a desktop machine”

Page 121 Wrong preposition: “on the table” – corrected to “in the table”

Graphic design: Communication Division, UIB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230863671 (print)
9788230841396 (PDF)