ARTICLE IN PRESS

Discrete Optimization xxx (xxxx) xxx

Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

Integer programming in parameterized complexity: Five miniatures[☆]

Tomáš Gavenčiak[a], Martin Koutecký[a,1], Dušan Knop[b,*,2]

[a] Computer Science Institute of Charles University, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, Praha, Czech Republic

[b] Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Thákurova 9, Praha, Czech Republic

ARTICLE INFO

Article history:
Received 31 December 2018
Received in revised form 14 May 2020
Accepted 25 June 2020
Available online xxxx

Keywords:
Graph coloring
Sum-coloring
Parameterized complexity

ABSTRACT

Powerful results from the theory of integer programming have recently led to substantial advances in parameterized complexity. However, our perception is that, except for Lenstra's algorithm for solving integer linear programming in fixed dimension, there is still little understanding in the parameterized complexity community of the strengths and limitations of the available tools. This is understandable: it is often difficult to infer exact runtimes or even the distinction between FPT and XP algorithms, and some knowledge is simply unwritten folklore in a different community. We wish to make a step in remedying this situation.

To that end, we first provide an easy to navigate quick reference guide of integer programming algorithms from the perspective of parameterized complexity. Then, we show their applications in three case studies, obtaining FPT algorithms with runtime $f(k)\mathrm{poly}(n)$. We focus on:

- *Modeling*: since the algorithmic results follow by applying existing algorithms to new models, we shift the focus from the complexity result to the modeling result, highlighting common patterns and tricks which are used.
- *Optimality program:* after giving an FPT algorithm, we are interested in reducing the dependence on the parameter; we show which algorithms and tricks are often useful for speed-ups.
- *Minding the* $\mathrm{poly}(n)$: reducing $f(k)$ often has the unintended consequence of increasing $\mathrm{poly}(n)$; so we highlight the common trade-offs and show how to get the best of both worlds.

Specifically, we consider graphs of bounded neighborhood diversity which are in a sense the simplest of dense graphs, and we show several FPT algorithms for Capacitated Dominating Set, Sum Coloring, Max-q-Cut, and certain other coloring problems by modeling them as convex programs in fixed dimension, n-fold integer programs, bounded dual treewidth programs, indefinite quadratic programs

☆ An extended abstract presenting preliminary results appeared in proceedings of IPEC 2018 [1] (Gavenčiak et al (2018)).

* Corresponding author.
E-mail addresses: gavento@kam.mff.cuni.cz (T. Gavenčiak), koutecky@iuuk.mff.cuni.cz (M. Koutecký), dusan.knop@fit.cvut.cz (D. Knop).

1 Partially supported by Charles University project UNCE/SCI/004, by the project 19-27871X of GA ČR, and by the Israel Science Foundation grant 308/18.

2 Supported by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

https://doi.org/10.1016/j.disopt.2020.100596

in fixed dimension, parametric integer programs in fixed dimension, and 2-stage stochastic integer programs.

1. Introduction

Our focus is on modeling various problems as INTEGER PROGRAMMING (IP), and then obtaining FPT algorithms by applying known algorithms for IP. IP is the problem

$$\min\{f(\mathbf{x}) \mid \mathbf{x} \in S \cap \mathbb{Z}^n, S \subseteq \mathbb{R}^n \text{ is convex}\} \ . \tag{IP}$$

We give special attention to two restrictions of IP. First, when S is a polyhedron, we get

$$\min\{f(\mathbf{x}) \mid A\mathbf{x} \le \mathbf{b},\, \mathbf{x} \in \mathbb{Z}^n\}, \tag{LinIP}$$

where $A \in \mathbb{Z}^{m\times n}$ and $\mathbf{b} \in \mathbb{Z}^m$; we call this problem *linearly-constrained IP*, or LinIP. Further restricting f to be a linear function gives Integer Linear Programming (ILP):

$$\min\{\mathbf{w}\mathbf{x} \mid A\mathbf{x} \le \mathbf{b},\, \mathbf{x} \in \mathbb{Z}^n\}, \tag{ILP}$$

where $\mathbf{w} \in \mathbb{Z}^n$. The function $f\colon \mathbb{Z}^n \to \mathbb{Z}$ is called the *objective function*, S is the *feasible set* (defined by *constraints* or various *oracles*), and $\mathbf{x}$ is a vector of *(decision) variables*. By $\langle\cdot\rangle$ we denote the binary encoding length of numbers, vectors and matrices.

In 1983 Lenstra showed that ILP is polynomial in fixed dimension and solvable in time $n^{\mathcal{O}(n)}\langle A, \mathbf{b}, \mathbf{w}\rangle$ (including later improvements [2–4]). Two decades later this algorithm's potential for applications in parameterized complexity was recognized, e.g. by Niedermeier [5]:

> *[...] It remains to investigate further examples besides* Closest String *where the described ILP approach turns out to be applicable. More generally, it would be interesting to discover more connections between fixed-parameter algorithms and (integer) linear programming.*

This call has been answered in the following years, for example in the context of graph algorithms [6–9], scheduling [10–13] or computational social choice [14].

In the meantime, many other powerful algorithms for IP have been devised; however it seemed unclear exactly *how* could these tools be used, as Lokshtanov states in his PhD thesis [15], referring to FPT algorithms for convex IP in fixed dimension:

> *It would be interesting to see if these even more general results can be useful for showing problems fixed parameter tractable.*

Similarly, Downey and Fellows [16] highlight the FPT algorithm for so called n-fold IP:

> *Conceivably, [*Minimum Linear Arrangement*] might also be approached by the recent (and deep) FPT results of Hemmecke, Onn and Romanchuk [17] concerning nonlinear optimization.*

Interestingly, Minimum Linear Arrangement was shown to be FPT by yet another new algorithm for IP due to Lokshtanov [18].

In the last 4 years we have seen a surge of interest in, and an increased understanding of, these IP techniques beyond Lenstra's algorithm, allowing significant advances in fields such as parameterized scheduling [10,12,13,19,20], computational social choice [21–23], multichoice optimization [24], and stringology [21]. This has increased our understanding of the strengths and limitations of each tool as well as the modeling patterns and tricks which are typically applicable and used.

1.1. Our results

We start by giving a quick overview of existing techniques in Section 2, which we hope to be an accessible reference guide for parameterized complexity researchers. Then, we study the parameterized complexity of five problems when parameterized by the neighborhood diversity of a graph (we defer the definitions to the relevant sections). Fig. 1 shows that neighborhood diversity occupies an important place with respect to other relevant parameters (vertex cover number, treewidth and cliquewidth). Studying the parameterized complexity of a problem with respect to treewidth can be seen as a stepping stone to understanding its parameterized complexity with respect to cliquewidth. Since neighborhood diversity is incomparable to treewidth in this regard, it provides a different step towards cliquewidth and gives different insights. The next possible step towards cliquewidth is so-called *modular width* introduced by Gajarský et al. [25] as a direct generalization of neighborhood diversity.

However, since our complexity results follow by applying an appropriate algorithm for IP, we also highlight our modeling results. Moreover, in the spirit of the optimality program (introduced by Marx [26]), we are not content with obtaining *some* FPT algorithm, but we attempt to decrease the dependence on the parameter k as much as possible. This sometimes has the unintended consequence of increasing the polynomial dependence on the graph size $|G|$. We note this and, by combining several ideas, get the "best of both worlds". Driving down the poly($|G|$) factor is in the spirit of "minding the poly(n)" of Lokshtanov et al. [27].

We denote by $|G|$ the number of vertices of the graph G and by k its neighborhood diversity; graphs of neighborhood diversity k have a succinct representation (constructible in linear time) with $\mathcal{O}(k^2 \log |G|)$ bits and we assume to have such a representation on input.

Theorem 1. *The* Capacitated Dominating Set *problem:*

(a) has a convex IP model in $\mathcal{O}(k^2)$ variables and can be solved in time and space $k^{\mathcal{O}(k^2)} \log |G|$,
(b) has an ILP model in $\mathcal{O}(k^2)$ variables and $\mathcal{O}(|G|)$ constraints, and can be solved in time $k^{\mathcal{O}(k^2)}\operatorname{poly}(|G|)$ and space $\operatorname{poly}(k, |G|)$,
(c) can be solved in time $k^{\mathcal{O}(k)}\operatorname{poly}(|G|)$ using model (a) and a proximity argument,
(d) has a polynomial $OPT + k$ approximation algorithm by rounding a relaxation of (a).

Theorem 2. *The* Sum Coloring *problem:*

(a) has an n-fold IP model in $\mathcal{O}(k|G|)$ variables and $\mathcal{O}(k^2|G|)$ constraints, and can be solved in time $k^{\mathcal{O}(k^3)}|G|^2 \log^2 |G|$,
(b) has a LinIP *model in $\mathcal{O}(2^k)$ variables and k constraints with a non-separable convex objective, and can be solved in time $2^{2^{k^{\mathcal{O}(1)}}} \log |G|$,*
(c) has a LinIP *model in $\mathcal{O}(2^k)$ variables and $\mathcal{O}(2^k)$ constraints whose constraint matrix has dual treewidth $k+2$ and whose objective is separable convex, and can be solved in time $k^{\mathcal{O}(k^2)} \log |G|$.*

The Max-q-Cut is a natural generalization of the Max Cut problem from partitioning into 2 parts to q parts. We show that:

Theorem 3. Max-q-Cut *has a* LinIP *model with an indefinite quadratic objective and can be solved in time $g(q, k) \log |G|$ for some computable function g.*

Panagopoulu and Spirakis [28] studied a coloring game $\Gamma(G)$ on graphs in which pure Nash equilibria (PNEs) are proper colorings satisfying many good chromatic number upper bounds. Later in the paper they

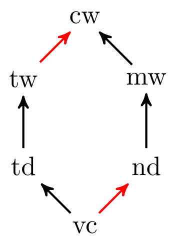

Fig. 1. A map of assumed parameters: vc is the vertex cover number, td is treedepth, tw is treewidth, cw is clique-width, nd is neighborhood diversity, and mw is modular-width. Black arrow stands for linear upper bounds, while a red arrow stands for exponential upper bounds. Note that treewidth and neighborhood diversity are incomparable because,
$\mathrm{tw}(K_n) = n-1 \quad \mathrm{nd}(K_n) = 1$
$\mathrm{tw}(P_n) = 1 \quad \mathrm{nd}(P_n) = n,$
where K_n and P_n are the complete graph and path on n vertices, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

define the problem of finding a so-called Stackelberg Strategy for this game, in which we are to precolor certain vertices, called leaders, so the subgame for the remaining vertices (the followers) has no PNEs which would require many colors (i.e., even the "worst" PNE does not require too many colors).

Theorem 4. Stackelberg Strategy for $\Gamma(G)$ *with* ζ *colors has a* $|G|$*-bounded Parametric ILP model whose representation can be described in* $\mathcal{O}\left(2^k + k\zeta\right)$ *symbols with largest coefficient* 1*, and can be solved in time* $f(k\zeta) \cdot \mathrm{poly}(k + |G|)$*, where* f *is a computable function.*

Finally, in order to demonstrate 2-stage stochastic IP, we introduce a natural precoloring problem called Stochastic Precoloring. Informally, we are given a graph G along with n possible future scenarios describing a supergraph H of G which will be formed by connecting new vertices to G, and the goal is to precolor G such that, whatever scenario happens, the chosen precoloring of G can be extended into a good coloring of H (cf. Section 6.2).

Theorem 5. Stochastic Precoloring *has a 2-stage Stochastic IP model in* $\mathcal{O}\left(2^{2^k} n\right)$ *variables and* $\mathcal{O}\left(2^k n\right)$ *constraints with a linear objective, and can be solved in time* $f(k) \cdot n^2 \log n \log |G|$*, where* f *is a computable function.*

1.2. Related work

Graphs of neighborhood diversity constitute an important stepping stone in the design of algorithms for dense graphs, because they are in a sense the simplest of dense graphs [7,8,29–33]. Studying the complexity of Capacitated Dominating Set on graphs of bounded neighborhood diversity is especially interesting because it was shown to be W[1]-hard parameterized by treewidth by Dom et al. [34]. Sum Coloring was shown to be FPT parameterized by treewidth [35]; its complexity parameterized by clique-width is open as far as we know. Max-q-Cut is FPT parameterized by q and treewidth (by reduction to CSP), but W[1]-hard parameterized by clique-width [36]. Stackelberg Strategy for $\Gamma(G)$ was introduced by Panagopoulou and Spirakis [28] in order to study properties of simple to implement approximation routine for graph coloring. Up to the best of our knowledge it has not been studied from parameterized complexity perspective until now even though graph coloring is extensively studied problem in this framework (see e.g. [37,38] and discussion therein).

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

1.3. Preliminaries

For positive integers m, n with $m \leq n$ we set $[m,n] = \{m, \dots, n\}$ and $[n] = [1,n]$. We write vectors in boldface (e.g., $\mathbf{x}, \mathbf{y}$) and their entries in normal font (e.g., the ith entry of $\mathbf{x}$ is x_i). For an integer $a \in \mathbb{Z}$, we denote by $\langle a \rangle = 1 + \log_2(|a|+1)$ the binary encoding length of a; we extend this notation to vectors, matrices and tuples of these objects. For example, $\langle A, \mathbf{b} \rangle = \langle A \rangle + \langle \mathbf{b} \rangle$, and $\langle A \rangle = \sum_{i,j} \langle a_{ij} \rangle$. For a graph G we denote by $V(G)$ its set of vertices, by $E(G)$ the set of its edges, and by $N_G(v) = \{u \in V(G) \mid uv \in E(G)\}$ the (open) neighborhood of a vertex $v \in V(G)$. For a matrix A we define

- the *primal graph* $G_P(A)$, which has a vertex for each column and two vertices are connected if there exists a row such that both columns are non-zero, and,
- the *dual graph* $G_D(A) = G_P(A^\intercal)$, which is the above with rows and columns swapped.

We call the treedepth and treewidth of $G_P(A)$ the *primal treedepth* $\mathrm{td}_P(A)$ and *primal treewidth* $\mathrm{tw}_P(A)$, and analogously for the *dual treedepth* $\mathrm{td}_D(A)$ and *dual treewidth* $\mathrm{tw}_D(A)$.

We define a partial order $\sqsubseteq$ on $\mathbb{R}^n$ as follows: for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ we write $\mathbf{x} \sqsubseteq \mathbf{y}$ and say that $\mathbf{x}$ is *conformal* to $\mathbf{y}$ if $x_i y_i \geq 0$ (that is, $\mathbf{x}$ and $\mathbf{y}$ lie in the same orthant) and $|x_i| \leq |y_i|$ for all $i \in [n]$. It is well known that every subset of $\mathbb{Z}^n$ has finitely many $\sqsubseteq$-minimal elements.

Definition 1 (*Graver Basis*)**.** The *Graver basis* of $A \in \mathbb{Z}^{m \times n}$ is the finite set $\mathcal{G}(A) \subset \mathbb{Z}^n$ of $\sqsubseteq$-minimal elements in $\{\mathbf{x} \in \mathbb{Z}^n \mid A\mathbf{x} = 0,\ \mathbf{x} \neq \mathbf{0}\}$.

Neighborhood diversity. Two vertices u, v are called *twins* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. The *twin equivalence* is the relation on vertices of a graph where two vertices are equivalent if and only if they are twins. An equivalence class is either an independent set, or a clique.

Definition 2 (*Lampis [9]*)**.** The *neighborhood diversity* of a graph G, denoted by $\mathrm{nd}(G)$, is the number k of classes (called *types*) of the twin equivalence of G.

We denote by V_i the classes of twin equivalence on G for $i \in [k]$. A graph G with $\mathrm{nd}(G) = k$ can be described in a compressed way using only $\mathcal{O}\left(\log |G| \cdot k^2\right)$ space by its type graph, which is computable in linear time [9]:

Definition 3. The *type graph* $T(G)$ of a graph G is a graph on $k = \mathrm{nd}(G)$ vertices $[k]$, where each i is assigned weight $|V_i|$, and where i, j is an edge or a loop in $T(G)$ if and only if two distinct vertices of V_i and V_j are adjacent.

Modeling. Loosely speaking, by *modeling* an optimization problem Π as a different problem Λ we mean encoding the features of Π by the features of Λ, such that the optima of Λ encode *at least some* optima of Π. Modeling differs from reduction by highlighting which features of Π are captured by which features of Λ.

In particular, when modeling Π as an integer program, the same feature of Π can often be encoded in several ways by the variables, constraints or the objective. For example, an objective of Π may be encoded as a convex objective of the IP, or as a linear objective which is lower bounded by a convex constraint; similarly a constraint of Π may be modeled as a linear constraint of IP or as minimizing a penalty objective function expressing how much is the constraint violated. Such choices greatly influence which algorithms are applicable to solve the resulting model. Specifically, in our models we focus on the parameters #variables (dimension), #constraints, the largest coefficient in the constraints $\|A\|_\infty$ (abusing the notation slightly when the constraints are not linear), the largest right hand side $\|\mathbf{b}\|_\infty$, the largest *domain* $\|\mathbf{u} - \mathbf{l}\|_\infty$, and the largest coefficient of the objective function $\|\mathbf{w}\|_\infty$ (linear objectives), $\|Q\|_\infty$ (quadratic objectives) or $f_{\max} = \max_{\mathbf{x}: \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} |f(\mathbf{x})|$ (in general), and noting other relevant features.

Solution structure. We concur with Downey and Fellows that FPT and structure are essentially one [16]. Here, it typically means restricting our attention to certain structured solutions and showing that nevertheless such structured solutions contain optima of the problem at hand. We always discuss these structural properties before formulating a model.

2. Integer programming toolbox

We give a list of the most relevant algorithms solving IP, highlighting their fastest known runtimes (marked ⊤), typical use cases and strengths (+), limitations (−), and a list of references to the algorithms (♡) and their most illustrative applications (▷), both in chronological order. We are deliberately terse here and defer a more nuanced discussion to Appendix A.

2.1. Small dimension

The following tools generally rely on results from discrete geometry. Consider for example Lenstra's theorem: it can be (hugely) simplified as follows. Let $S = \{\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b}\} \subseteq \mathbb{R}^n$; then we can decide whether $S \cap \mathbb{Z}^n$ by algorithmic versions of the following existential recursive argument:

1. Either S is "flat" in some direction; then, we can cut S up into few lower-dimensional slices and recurse into these,
2. or S is not flat in any direction, and must contain an integer point by Minkowski's first theorem.

Being able to optimize an objective then follows from testing feasibility by binary search.
ILP in small dimension. Problem (ILP) with small n.

⊤ $n^{2.5n}\langle A, \mathbf{b}, \mathbf{w}\rangle$ [2,3]
\+ Can use large coefficients, which allows encoding logical connectives using Big-M coefficients [39]. Runs in polynomial space. Most people are familiar with ILP (as compared with e.g. convex IPs etc.)
− Small dimension can be an obstacle in modeling polynomially many "types" of objects [40, Challenge #2]. Models often use exponentially many variables in the parameter, leading to double-exponential runtimes (applies to all small dimension techniques below). Encoding a convex objective or constraint requires many constraints (cf. Model 8). Big-M coefficients are impractical.
♡ Lenstra [4], Kannan [3], Frank and Tardos [2]
▷ Niedermeier (Closest String) [5] Fellows et al. (graph layout problems) [6] Jansen and Solis-Oba (scheduling; MILP column generation technique) [11], Fiala et al. (graph coloring) [7], Faliszewski et al. (computational social choice; big-M coefficients to express logical connectives) [41].

Convex IP in small dimension. Problem (IP) with f a convex function; S can be represented by polynomial inequalities, a first-order oracle, a separation oracle, or as a semialgebraic set.

⊤ $n^{\frac{4}{3}n}\langle B\rangle$, where S is contained in a ball of radius B [42].
\+ Strictly stronger than ILP. Representing constraints implicitly by an oracle allows better dependence on instance size (cf. Model 7).
− Exponential space. Algorithms usually impractical. Proving convexity can be difficult.
♡ Grötschel, Lovász, and Schrijver [43, Theorem 6.7.10] (weak separation oracle), Khachiyan and Porkolab [44] (semialgebraic sets), Heinz [45], whose algorithm is superseded by Hildebrand and Köppe [46] (polynomials), Dadush, Peikert and Vempala [47] randomized and Dadush and Vempala [42] (strong separation oracle), Oertel, Wagner, and Weismantel [48] reduction to Mixed ILP subproblems (first-order oracle).

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

▷ Hermelin et al. (multiagent scheduling; convex constraints) [10], Bredereck et al. (bribery; convex objective) [14], Mnich and Wiese, Knop and Koutecký (scheduling; convex objective) [12,13], Knop et al. (various problems; convex objectives) [49], Model 7

Indefinite quadratic IP in small dimension. Problem (LinIP) with $f(\mathbf{x}) = \mathbf{x}^\intercal Q\mathbf{x}$ indefinite (non-convex) quadratic.

⊤ $g(n, \|A\|_\infty, \|Q\|_\infty)\langle \mathbf{b}\rangle$ [50]
+ Currently the only tractable indefinite objective.
− Limiting parameterization.
♡ Lokshtanov [18], Zemmer [50]
▷ Lokshtanov (Optimal Linear Arrangement) [18], Model 9

Parametric ILP in small dimension. Given a $Q = \{\mathbf{b} \in \mathbb{R}^m \mid B\mathbf{b} \le \mathbf{d}\}$, decide

$$\forall \mathbf{b} \in Q \cap \mathbb{Z}^m \; \exists \mathbf{x} \in \mathbb{Z}^n : \; A\mathbf{x} \le \mathbf{b} \; .$$

⊤ $g(n, m)\,\mathrm{poly}(\|A, B, \mathbf{d}\|_\infty)$ [51]
+ Models one quantifier alternation. Useful in expressing game-like constraints (e.g., "∀ moves ∃ a counter-move"). Allows unary big-M coefficients to model logic [23, Theorem 4.5].
− Input has to be given in unary in order to get an FPT algorithm (vs. e.g. Lenstra's algorithm).
♡ Eisenbrand and Shmonin [51, Theorem 4.2], Crampton et al. [52, Corollary 1]
▷ Crampton et al. (resiliency) [52], Knop et al. (Dodgson bribery) [23], Model 22.

2.2. Variable dimension

In this section it will be more natural to consider the following *standard form* of (LinIP)

$$\min\{f(\mathbf{x}) \mid A\mathbf{x} = \mathbf{b},\, \mathbf{l} \le \mathbf{x} \le \mathbf{u},\, \mathbf{x} \in \mathbb{Z}^n\}, \tag{SLinIP}$$

where $\mathbf{b} \in \mathbb{Z}^m$ and $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$. Let $L = \langle f_{\max}, \|\mathbf{b}, \mathbf{l}, \mathbf{u}\|_\infty\rangle$. In contrast with the previous section, the following algorithms typically rely on algebraic arguments and dynamic programming. The large family of algorithms based on Graver bases (see below) can be described as *iterative augmentation* methods, where we start with a feasible integer solution $\mathbf{x}_0$ and iteratively find a step $\mathbf{g} \in \{\mathbf{x} \in \mathbb{Z}^n \mid A\mathbf{x} = \mathbf{0}\}$ such that $\mathbf{x}_0 + \mathbf{g}$ is still feasible and improves the objective. Under a few additional assumptions on $\mathbf{g}$ it is possible to prove quick convergence of such methods.

ILP with few rows. Problem (SLinIP) with small m and a linear objective $\mathbf{w}\mathbf{x}$ for $\mathbf{w} \in \mathbb{Z}^n$.

⊤ $\mathcal{O}((m\|A\|_\infty)^{2m})\langle \mathbf{b}\rangle$ if $\mathbf{l} \equiv \mathbf{0}$ and $\mathbf{u} \equiv +\boldsymbol{\infty}$, and $n \cdot (m\|A\|_\infty)^{\mathcal{O}(m^2)}\langle \mathbf{b}, \mathbf{l}, \mathbf{u}\rangle$ in general [53]
⊥ $m^{o(m)}\langle A, \mathbf{b}\rangle$ lower bound under ETH [54]
+ Useful for configuration IPs with small coefficients, leading to exponential speed-ups. Best runtime in the case without upper bounds. Linear dependence on n.
− Limited modeling power. Requires small coefficients.
♡ Papadimitriou [55], Eisenbrand and Weismantel [56], Jansen and Rohwedder [53]
▷ Jansen and Rohwedder (scheduling) [53]

$$A_{\text{nfold}} = \begin{pmatrix} A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_2 \end{pmatrix} \qquad A_{\text{stoch}} = \begin{pmatrix} B_1 & B_2 & 0 & \cdots & 0 \\ B_1 & 0 & B_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_1 & 0 & 0 & \cdots & B_2 \end{pmatrix}$$

n-fold IP, tree-fold IP, and dual treedepth. *n-fold IP* is problem (SLinIP) in dimension nt, with $A = A_{\text{nfold}}$ for some two blocks $A_1 \in \mathbb{Z}^{r \times t}$ and $A_2 \in \mathbb{Z}^{s \times t}$, $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^{nt}$, $\mathbf{b} \in \mathbb{Z}^{r+ns}$, and with f a separable convex function, i.e., $f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{t} f_j^i(x_j^i)$ with each $f_j^i : \mathbb{Z} \to \mathbb{Z}$ convex. *Tree-fold IP* is a generalization of n-fold IP where the block A_2 is itself replaced by an n-fold matrix, and so on, recursively, τ times. Tree-fold IP has bounded $\mathrm{td}_D(A)$.

⊤ $(\|A\|_\infty rs)^{\mathcal{O}(r^2 s + rs^2)} (nt)^2 \log(nt) L$ n-fold IP [57,58] (cf. [59] for a nearly-linear time algorithm when f is a linear function); $(\|A\|_\infty + 1)^{2^{\mathrm{td}_D(A)}} (nt)^2 \log(nt) L$ for (SLinIP) [60].
+ Variable dimension useful in modeling many “types” of objects [22,23]. Useful for obtaining exponential speed-ups (not only configuration IPs). Seemingly rigid format is in fact not problematic (blocks can be different provided coefficients and dimensions are small).
− Requires small coefficients.
♡ Hemmecke et al. [17], Knop et al. [21], Chen and Marx [19], Eisenbrand et al. [58], Altmanová et al. [57], Koutecký et al. [60]
▹ Knop and Koutecký (scheduling with many machine types) [12], Knop et al. (bribery with many voter types) [21,22], Chen and Marx (scheduling; tree-fold IP) [19], Jansen et al. (scheduling EPTAS) [20], Model 10

2-stage and multi-stage stochastic IP, and primal treedepth. 2-stage stochastic IP is problem (SLinIP) with $A = A_{\text{stoch}}$, $B_1 \in \mathbb{Z}^{t \times r}$, $B_2 \in \mathbb{Z}^{t \times s}$, and f a separable convex function; multi-stage stochastic IP is problem (SLinIP) with a multi-stage stochastic matrix, which is the transpose of a tree-fold matrix; multi-stage stochastic IP is in turn generalized by IP with small primal treedepth $\mathrm{td}_P(A)$.

⊤ $g(\mathrm{td}_P(A), \|A\|_\infty) n^2 \log nL$, g computable [60]
+ Similar to Parametric ILP in fixed dimension, but quantification $\forall \mathbf{b} \in Q \cap \mathbb{Z}^n$ is now over a polynomial sized but possibly non-convex set of explicitly given right hand sides.
− Not clear which problems are captured. Requires small coefficients. Parameter dependence g is possibly non-elementary; no upper bounds on g are known, only computability.
♡ Hemmecke and Schultz [61], Aschenbrenner and Hemmecke [62], Koutecký et al. [60], Klein [63]
▹ Model 23

Small treewidth and Graver norms. Let $g_\infty(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_\infty$ and $g_1(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_1$ be maximum norms of elements of $\mathcal{G}(A)$.

⊤ $\min\{g_\infty(A)^{\mathcal{O}(\mathrm{tw}_P(A))}, g_1(A)^{\mathcal{O}(\mathrm{tw}_D(A))}\} n^2 \log nL$ [60]
+ Captures IPs beyond the classes defined above (cf. Section 5.3).
− Bounding $g_1(A)$ and $g_\infty(A)$ is often hard or impossible.
♡ Koutecký et al. [60]
▹ Knop and Koutecký [64], Model 14

3. Convex constraints: capacitated dominating set

In this section we consider the Capacitated Dominating Set problem.

Capacitated Dominating Set	
Input:	A graph $G = (V, E)$ and a capacity function $c\colon V \to \mathbb{N}_{\geq 1}$.
Task:	Find a smallest possible set $D \subseteq V$ and a mapping $\delta\colon V \setminus D \to D$ such that for each $v \in D$, $|\delta^{-1}(v)| \leq c(v)$ and $\{u, v\} \in E$ for all $u \in \delta^{-1}(v)$.

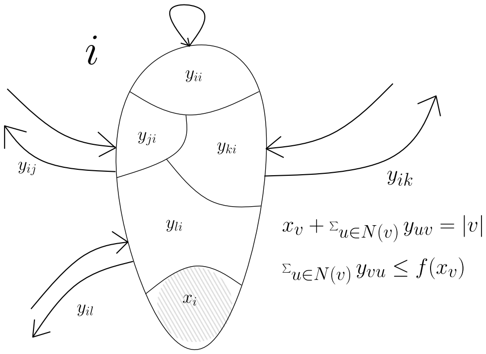

Fig. 2. Interpretation of variables of Model 7.

Solution structure. Let $<_c$ be a linear extension of ordering of V by vertex capacities, i.e., $u <_c v$ if $c(u) \leq c(v)$. For $i \in T(G)$ and $\ell \in [|V_i|]$ let $V_i[1:\ell]$ be the set of the first ℓ vertices of V_i in the ordering $<_c$ and let $f_i(\ell) = \sum_{v\in V_i[1:\ell]} c(v)$; for $\ell > |V_i|$ let $f_i(\ell) = f_i(|V_i|)$. Let D be a solution and $D_i = D \cap V_i$. We call the functions f_i the *domination capacity functions*. Intuitively, $f_i(\ell)$ is the maximum number of vertices dominated by $V_i[1:\ell]$. Observe that since $f_i(\ell)$ is a partial sum of a non-increasing sequence of numbers, it is a piece-wise linear concave function. We say that D is *capacity-ordered* if, for each $i \in T(G)$, $D_i = V_i[1:|D_i|]$. The following observation allows us to restrict our attention to such solutions; the proof goes by a simple exchange argument.

Lemma 6. *There is a capacity-ordered optimal solution.*

Proof. Consider any solution D together with a mapping $\delta : V \setminus D \to D$ witnessing that D is a solution. Our goal is to construct a capacity-ordered solution $\hat{D}$ which is at least as good as D. If D itself is capacity-ordered, we are done. Assume the contrary; thus, there exists an index $i \in [k]$ and a vertex $v \in D_i$ such that $v \notin V_i[1:|D_i|]$, and consequently there exists a vertex $u \in V_i[1:|D_i|]$ such that $u \notin D_i$.

Let $D' \subseteq V$ be defined by setting $D'_i = (D_i \cup \{u\}) \setminus \{v\}$ and $D'_j = D_j$ for each $j \neq i$. We shall define a mapping δ' witnessing that D' is again a solution. Let $\delta'(x) = y$ if and only if $\delta(x) = y$ and $x \neq u$ and $y \neq v$, let $\delta'(x) = u$ whenever $\delta(x) = v$ and let $\delta'(v) = y$ if $\delta(u) = y$. Clearly $|(\delta')^{-1}(x)| \leq c(x)$ for each $x \in D$ because $|(\delta')^{-1}(x)| = |\delta^{-1}(x)|$ when $x \notin \{u, v\}$, and $|(\delta')^{-1}(u)| = |\delta^{-1}(v)|$ and $c(u) \geq c(v)$.

If D' itself is not yet a capacity-ordered solution, we repeat the same swapping argument. Observe that 1. $\sum_{i=1}^{k} |D_i \triangle V_i[1:|D_i|] > \sum_{i=1}^{k} |D'_i \triangle V_i[1:|D'_i|]$, i.e., D' is closer than D to being capacity-ordered, and, 2. the size of D' compared to D does not increase. Finally, when $\sum_{i=1}^{k} |D'_i \triangle V_i[1:|D'_i|] = 0$, D' is our desired capacity-ordered solution $\hat{D}$. $\square$

Observe that a capacity-ordered solution is fully determined by the sizes $|D_i|$ and $<_c$ rather than the actual sets D_i, which allows modeling CDS in small dimension.

Model 7 (Capacitated Dominating Set *as Convex IP in Fixed Dimension*).
Variables & notation:

- $x_i = |D_i|$
- $y_{ij} = |\delta^{-1}(D_i) \cap D_j|$, see Fig. 2
- $f_i(x_i) =$ maximum #vertices dominated by D_i if $|D_i| = x_i$

Objective & Constraints:

$$\min \sum_{i\in T(G)} x_i \qquad\qquad \min |D| = \sum_{i\in T(G)} |D_i| \qquad \text{(cds:cds-obj)}$$

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

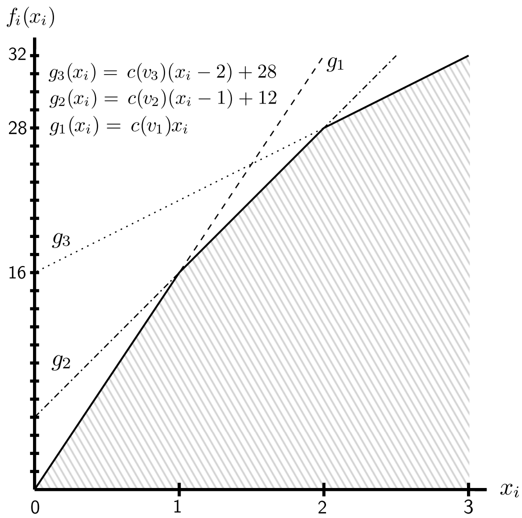

Fig. 3. The linearization (cds:cap-lin) of a piecewise linear convex constraint (cds:cap) in Model 8.

$$\sum_{j \in N_{T(G)}(i)} y_{ij} \le f_i(x_i) \qquad \forall i \in T(G) \qquad \text{respect capacities} \qquad \text{(cds:cap)}$$

$$\sum_{i \in N_{T(G)}(j)} y_{ij} \ge |V_j| - x_j \qquad \forall j \in T(G) \qquad \text{every } v \in V_j \setminus D_j \text{ dominated} \qquad \text{(cds:dom)}$$

$$0 \le x_i \le |V_i| \qquad \forall i \in T(G) \qquad \text{(cds:bounds)}$$

Parameters & Notes:

#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l},\mathbf{u}\|_\infty$	$\|\mathbf{w}\|_\infty$
$\mathcal{O}(k^2)$	$\mathcal{O}(k)$	$\|G\|$	$\|G\|$	$\|G\|$	1
- constraint (cds:cap) is convex, since it bounds the area *under* a concave function, and is piece-wise linear, see Fig. 3. □

Proof of Theorem 1(a). Apply for example Dadush's algorithm [47] to Model 7. □

Proof of Theorem 1(b). We can trade the non-linearity of the previous model for an increase in the number of constraints and the largest coefficient (Model 8). That, combined with Kannan's algorithm, yields Theorem 1(b), where we get a larger dependence on $|G|$, but require only poly$(k, |G|)$ space. □

Proof of Theorem 1(d) (*Additive Approximation*)**.** Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{k+k^2}$ be an optimal solution to the *continuous relaxation* of Model 7, i.e., we relax the requirement that $(\mathbf{x}, \mathbf{y})$ is integral; note that such $(\mathbf{x}, \mathbf{y})$ can be computed in polynomial time using the ellipsoid method [43], or by applying a polynomial LP algorithm to Model 8. We would like to round $(\mathbf{x}, \mathbf{y})$ up to an integral $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ to obtain a feasible integer solution which would be an approximation of an integer optimum. Let us take $\hat{\mathbf{x}} = \lceil \mathbf{x} \rceil$ and show how to compute a corresponding $\hat{\mathbf{y}}$ so that $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a feasible solution.

Model 8 (Capacitated Dominating Set *as ILP in Fixed Dimension*)**.**
Exactly as Model 7 but replace constraints (cds:cap) with the following equivalent set of $|G|$ linear constraints (denoted g_j in Fig. 3), where v_i^ℓ is the ℓth vertex of V_i in $<_c$:

$$\sum_{ij \in E(T(G))} y_{ij} \leq f_i(\ell - 1) + c(v_i^\ell)(x_i - \ell + 1) \quad \forall i \in T(G)\, \forall \ell \in [|V_i|] \tag{cds:cap-lin}$$

The parameters then become:	#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l}, \mathbf{u}\|_\infty$	$\|\mathbf{w}\|_\infty$
	$\mathcal{O}(k^2)$	$\mathcal{O}(k + \|G\|)$	$\|G\|$	$\|G\|$	$\|G\|$	1

□

Take a set D corresponding to $\hat{\mathbf{x}}$ by taking a union of $D_j := V_j[1 : \lceil x_j \rceil]$ for each $j \in [k]$. We construct an auxiliary bipartite graph H as follows. The first partite contains $c(v)$ copies for every $v \in D$, and the second is $V \setminus D$. We have an edge $wz \in E(H)$ if w is a copy of v and $vz \in E(G)$. Our goal is to show that H has a matching which covers the second part $V \setminus D$, which will then correspond to a mapping δ testifying to the fact that D is a CDS, and inferring $\hat{\mathbf{y}}$ from δ is straightforward.

We show the existence of such a matching by Hall's theorem: there is a matching covering $V \setminus D$ if and only if for every $A \subseteq V \setminus D$, $|A| \leq |N_H(A)|$. Consider such a set A. Without loss of generality, we assume that A is maximal with $N_H(A)$ fixed, i.e., there is no vertex in $V \setminus D$ which could be added to A without changing its neighborhood. Observe that this implies A must be a union of $V_j \setminus D$ for some subset of types $J \subseteq [k]$, i.e., $A = \bigcup_{j \in J} V_j \setminus D$. For each type j, the constraint (cds:dom) enforces $\sum_{i \in N_{T(G)}} y_{ij} \geq |V_j| - |D_j|$. Summing up these constraints over the types in J, we obtain

$$\sum_{j \in J} \sum_{i \in N_{T(G)}(i)} y_{ij} \geq \sum_{j \in J} (|V_j| - |D_j|) = |A| \ .$$

On the other hand, $N_H(A)$ contains vertices which are copies of vertices from D_i for each type $i \in N_{T(G)}(J) =: I$. Now, summing constraints (cds:cap) over I gives:

$$\sum_{i \in I} \sum_{j \in N_{T(G)}(i)} y_{ij} \leq \sum_{i \in I} f_i(x_i) \leq \sum_{i \in I} f_i(\lceil x_i \rceil) = |N_H(A)|,$$

and since we also have $J \subseteq N_{T(G)}(I)$, we have

$$\sum_{i \in I} \sum_{j \in N_{T(G)}(i)} y_{ij} \geq \sum_{j \in J} \sum_{i \in N_{T(G)}(j)} y_{ij},$$

combining the above inequalities yields $|A| \leq |N_H(A)|$ as desired. The matching can be constructed in polynomial time, e.g., $\mathcal{O}(\sqrt{n} \cdot m)$ by the Hopcroft–Karp algorithm. □

Proof of Theorem 1(c) (*Speed Trade-Offs*)**.** Notice that on our way to proving Theorem 1(d) we have shown that Model 7 has *integrality gap* at most k, i.e., the value of the continuous optimum is at most k less than the value of the integer optimum. This implies that an integer optimum $(\mathbf{x}^*, \mathbf{y}^*)$ satisfies, for each $i \in [k]$, $\max\{0, \lfloor x_i - k \rfloor\} \leq x_i^* \leq \min\{|V_i|, x_i + \lceil k \rceil\}$.

We can exploit this to improve Theorem 1(a) in terms of the parameter dependence at the cost of the dependence on $|G|$. Computing a maximum cardinality matching in the auxiliary graph H from the proof of Theorem 1(d) gives a way to test, for a given integer $\hat{\mathbf{x}}$, whether it models a capacity-ordered solution, that is, whether there exists a capacitated dominating set with $D_i = V_i[1 : \hat{x}_i]$ for each i. Then we can simply go over all possible $(2k+2)^k$ choices of $\hat{\mathbf{x}}$ which are in the ℓ_∞-ball of radius k centered around the fractional optimum $\mathbf{x}$ and choose the best. □

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

4. Indefinite quadratics: Max q-Cut

	MAX-q-CUT
Input:	A graph $G = (V, E)$.
Task:	A partition $W_1 \dot\cup \cdots \dot\cup W_q = V$ maximizing the number of edges between distinct W_α and W_β, i.e., $\lvert\{uv \in E(G) \mid u \in W_\alpha, v \in W_\beta, \alpha \neq \beta\}\rvert$.

Solution structure. As before, it is enough to describe *how many* vertices from type $i \in T(G)$ belong to W_α for $\alpha \in [q]$, and their specific choice does not matter; this gives us a small dimensional encoding of the solutions.

Model 9 (MAX-q-CUT *as* LINIP *with Indefinite Quadratic Objective*).
Variables & Notation:

- $x_{i\alpha} = |V_i \cap W_\alpha|$
- $x_{i\alpha} \cdot x_{j\beta} = \#$edges between $V_i \cap W_\alpha$ and $V_j \cap W_\beta$ if $ij \in E(T(G))$.

Objective & Constraints:

$$\max \sum_{\substack{\alpha,\beta\in[q]:\\ \alpha<\beta}} \sum_{ij\in E(T(G))} x_{i\alpha} \cdot x_{j\beta} \qquad \text{max \#edges across partites} \qquad \text{(mc:obj)}$$

$$\sum_{\alpha\in[q]} x_{i\alpha} = |V_i| \qquad \forall i \in T(G) \qquad (V_i \cap W_\alpha)_{\alpha\in[q]} \text{ partitions } V_i \qquad \text{(mc:part)}$$

Parameters & Notes:

#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l},\mathbf{u}\|_\infty$	$\|Q\|_\infty$
kq	k	1	$\lvert G\rvert$	$\lvert G\rvert$	1
- objective (mc:obj) is indefinite quadratic. □

Proof of Theorem 3. Applying Lokshtanov's [18] or Zemmer's [50] algorithm to Model 9 yields the theorem. Note that since we do not know anything about the objective except that it is quadratic, we have to make sure that $\|Q\|_\infty$ and $\|A\|_\infty$ are small. □

5. Convex Objective: SUM COLORING

	SUM COLORING
Input:	A graph $G = (V, E)$.
Task:	A proper coloring $c: V \to \mathbb{N}$ minimizing $\sum_{v\in V} c(v)$.

In the following we first give a single-exponential algorithm for SUM COLORING with a polynomial dependence on $|G|$, then a double-exponential algorithm with a logarithmic dependence on $|G|$, and finally show how to combine the two ideas together to obtain a single-exponential algorithm with a logarithmic dependence on $|G|$.

5.1. Sum coloring *via n-fold IP*

Solution structure. The following observation was made by Lampis [9] for the (ordinary) Coloring problem (where the task is to minimize the total number of colors used), and it holds also for the Sum Coloring problem. In both coloring problems there always exists an optimal coloring in which every color $C \subseteq V(G)$ intersects each clique type in at most one vertex, and (more importantly) each independent type in either none or all of its vertices. The first follows simply by the fact that it is a clique. The second by the fact that if both colors α, β with $\alpha < \beta$ are used for an independent type, then recoloring all vertices of color β to be of color α remains a valid coloring and decreases its cost (clearly, in the Coloring problem we can "merge" the color α into β as well). We call a coloring with this structure an *essential coloring*.

Model 10 (Sum Coloring *as n-fold IP*)**.**
Variables & Notation:

- $x_i^\alpha = 1$ if a color α intersects V_i
- $\alpha \cdot x_i^\alpha =$ cost of color α at a clique type i
- $\alpha|V_i| \cdot x_i^\alpha =$ cost of color α at an independent type V_i
- $S_{\text{nfold}}(\mathbf{x}) = \sum_{\alpha=1}^{|G|} \Big((\sum_{\text{clique } i \in T(G)} \alpha x_i^\alpha) + (\sum_{\text{indep. } i \in T(G)} \alpha |V_i| x_i^\alpha) \Big) =$ total cost of $\mathbf{x}$

Objective & Constraints:

$$\min \ S_{\text{nfold}}(\mathbf{x}) \qquad \text{(sc:nf:obj)}$$

$$\sum_{\alpha=1}^{|G|} x_i^\alpha = |V_i| \qquad \forall i \in T(G), V_i \text{ is clique} \qquad V_i \text{ is colored} \qquad \text{(sc:nf:cliques)}$$

$$\sum_{\alpha=1}^{|G|} x_i^\alpha = 1 \qquad \forall i \in T(G), V_i \text{ is independent} \qquad V_i \text{ is colored} \qquad \text{(sc:nf:indeps)}$$

$$x_i^\alpha + x_j^\alpha \le 1 \qquad \forall \alpha \in [|G|] \ \forall ij \in E(T(G)) \qquad \mathbf{x}^\alpha \text{ is independent set} \qquad \text{(sc:nf:xi-indep)}$$

Parameters & Notes:

#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l},\mathbf{u}\|_\infty$	$\|\mathbf{w}\|_\infty$	r	s	t	n
$k\|G\|$	$k + k^2\|G\|$	1	$\|G\|$	1	$\|G\|$	k	k^2	k	$\|G\|$
- Constraints have an n-fold format: (sc:nf:cliques) and (sc:nf:indeps) form the $(A_1 \cdots A_1)$ block and (sc:nf:xi-indep) form the A_2 blocks; see parameters r, s, t above. Observe that the matrix A_1 is the $k \times k$ identity matrix and the matrix A_2 is the incidence matrix of $T(G)$ transposed. □

Proof of Theorem 2(a). Apply the algorithm of Altmanová et al. [57] to Model 10. □

Model 10 is a typical use case of n-fold IP: we have a vector of multiplicities $\mathbf{b}$ (modeling $(|V_1|, \ldots, |V_k|)$) and we optimize over its decompositions into independent sets of $T(G)$. A clever objective function models the objective of Sum Coloring. The main drawback is large number of bricks in this model.

5.2. Sum Coloring *via convex minimization in fixed dimension*

Solution structure. The previous observations also allow us to encode a solution in a different way. Let $\mathcal{I} = \{I_1, \ldots, I_K\}$ be the set of all independent sets of $T(G)$; note that $K < 2^k$. Then we can encode an essential coloring of G by a vector of multiplicities $\mathbf{x} = (x_{I_1}, \ldots, x_{I_K})$ of elements of $\mathcal{I}$ such that there

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

are x_{I_j} colors which color exactly the types contained in I_j. The difficulty with Sum Coloring lies in the formulation of its objective function. Observe that given an $I_j \in \mathcal{I}$, the number of vertices every color class of this type will contain is independent of the actual multiplicity x_{I_j}. Define the *size of a color class* $\sigma\colon \mathcal{I} \to \mathbb{N}$ as $\sigma(I) = \sum_{\text{clique } i\in I} 1 + \sum_{\text{indep. } i \in I} |V_i|$. The following lemma justifies the fact that independent sets (colors) in an optimal solution are sorted according to their sizes.

Lemma 11. *Let $G = (V, E)$ be a graph and let $c\colon V \to \mathbb{N}$ be a proper coloring of G minimizing $\sum_{v\in V} c(v)$. Let $\mu(p)$ denote the quantity $|\{v \in V \mid c(v) = p\}|$. Then $\mu(p) \geq \mu(q)$ for every $p \leq q$.*

Proof. Suppose for contradiction that we have $p < q$ with $\mu(p) < \mu(q)$. We now construct a proper coloring c' of G as follows

$$c'(v) = \begin{cases} p & \text{if } c(v) = q, \\ q & \text{if } c(v) = p, \\ c(v) & \text{otherwise.} \end{cases}$$

Clearly c' is a proper coloring. Now we have

$$\begin{aligned} \sum_{v\in V} c(v) = & \left(\sum_{v\in V} c'(v)\right) - p\mu(q) - q\mu(p) + p\mu(p) + q\mu(q) = \\ & \left(\sum_{v\in V} c'(v)\right) - p(\mu(q) - \mu(p)) + q(\mu(q) - \mu(p)) = \\ & \left(\sum_{v\in V} c'(v)\right) + (\mu(q) - \mu(p))(q - p) > \sum_{v\in V} c'(v)\,. \end{aligned}$$

Here the last inequality holds, since both the factors following the sum are positive due to our assumptions. Thus we arrive at a contradiction that c is a coloring minimizing the first sum. $\square$

Our goal now is to show that the objective function can be expressed as a convex function in terms of the variables x_{I_j} for $j \in [K]$. Let us assume it holds that $\sigma(I_1) \leq \sigma(I_2) \leq \cdots \leq \sigma(I_K)$ and observe that without loss of generality we may assume that in the essential coloring we first use x_{I_1}-times the color I_1, then x_{I_2}-times color I_2, and so forth up to color I_K. Now it is not hard to verify that

$$\sum_{i=1}^{K} \left(\sigma(I_i) \cdot \left(\left(x_{I_i} \sum_{j<i} x_{I_j}\right) + \left(\sum_{p=1}^{x_{I_i}} p\right)\right)\right) = \sum_{i=1}^{K} \left(\sigma(I_i) \cdot \left(\left(x_{I_i} \sum_{j<i} x_{I_j}\right) + \binom{x_{I_i}}{2}\right)\right) \tag{1}$$

is the value of the objective; we stress that $\binom{0}{2} = 0$. In order to see this observe that the “cost” of every vertex in pth use of color I_i is $p + \sum_{j<i} x_{I_j}$ and that there are $\sigma(I_i)$ such vertices. We now have to argue that the above is a convex function.

In what follows we refer to Fig. 4 which demonstrates the change of perspective we are going to describe now. We will get help from auxiliary variables $y_1, \ldots, y_{\sigma(I_1)}$ which are a linear projection of variables $\mathbf{x}$; note that we do not actually introduce these variables into the model and only use them for the sake of proving convexity. Namely, y_j indicates how many color classes contain at least j vertices, i.e., $y_j = \sum_{\ell\in[K]:\sigma(I_\ell)\geq j} x_{I_\ell}$. Then, the objective function (1) can be expressed as $S_{\text{convex}}(\mathbf{x}) = \sum_{j=1}^{\sigma(I_1)} \binom{y_j}{2}$.

Lemma 12. *$S_{convex}(\mathbf{x})$ is equivalent to (the right-hand side of) (1).*

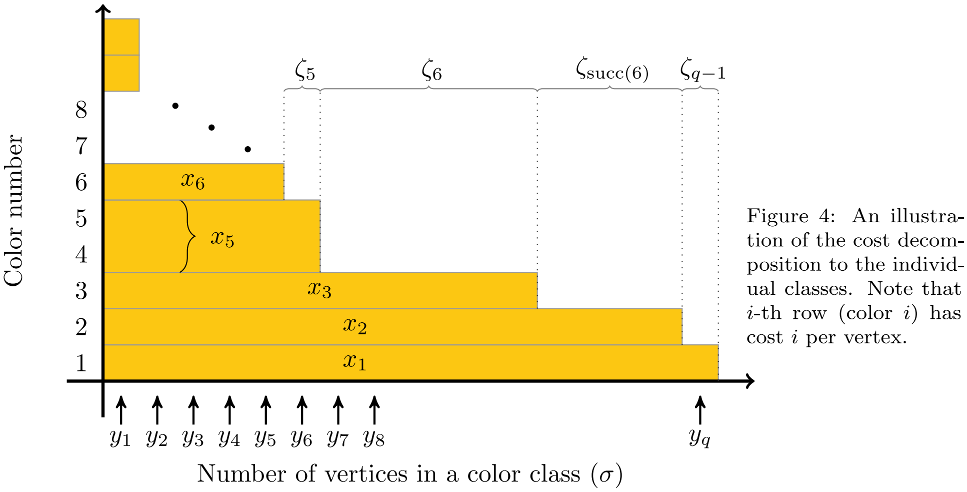

Fig. 4. An illustration of the cost decomposition to the individual classes. Note that ith row (color i) has cost i per vertex.

Proof. Let us, for the sake of this proof, define

$$y_{j,i} = \sum_{\substack{I_\ell \in \{I_1,\dots,I_i\} \\ \sigma(I_\ell)\geq j}} x_{I_\ell}.$$

Observe that we have $y_j = y_{j,K}$. Now, our goal is to prove that for $i = 1,\dots,K$ the following holds

$$\sigma(I_i)\cdot\left(\left(x_{I_i}\sum_{j<i} x_{I_j}\right)+\binom{x_{I_i}}{2}\right) = \left(\sum_{j=1}^{\sigma(I_1)}\binom{y_{j,i}}{2}\right) - \left(\sum_{j=1}^{\sigma(I_1)}\binom{y_{j,i-1}}{2}\right).$$

Notice that if this is that case, then we are done, since if we sum the left-hand sides for $i = 1,\dots,K$, we get (the right hand side of) (1) while if we sum the right-hand sides we get $\sum_{j=1}^{\sigma(I_1)}\binom{y_{j,K}}{2}$.

Note that, by the definition of $y_{j,i}$, we have that $y_{j,i} = y_{j,i-1}$ for all $\sigma(I_i) < j \leq \sigma(I_1)$. Furthermore, we have that $y_{j,i} - y_{j,i-1} = x_{I_i}$ for all $j = 1,\dots,\sigma(I_i)$, importantly for us this difference is the same. Now, we can compute

$$\begin{aligned}
&\left(\sum_{j=1}^{\sigma(I_1)}\binom{y_{j,i}}{2}\right) - \left(\sum_{j=1}^{\sigma(I_1)}\binom{y_{j,i-1}}{2}\right) \\
&\qquad = \left(\sum_{j=1}^{\sigma(I_i)}\binom{y_{j,i}}{2} + \sum_{j=\sigma(I_i)+1}^{\sigma(I_1)}\binom{y_{j,i}}{2}\right) - \left(\sum_{j=1}^{\sigma(I_i)}\binom{y_{j,i-1}}{2} + \sum_{j=\sigma(I_i)+1}^{\sigma(I_1)}\binom{y_{j,i-1}}{2}\right) \\
&\qquad = \left(\sum_{j=1}^{\sigma(I_i)}\binom{y_{j,i}}{2}\right) - \left(\sum_{j=1}^{\sigma(I_i)}\binom{y_{j,i-1}}{2}\right) \\
&\qquad = \sigma(I_i)\cdot\left(\binom{y_{1,i}}{2} - \binom{y_{1,i-1}}{2}\right).
\end{aligned}$$

Thus, we are left to verify that

$$\left(x_{I_i}\sum_{j<i} x_{I_j}\right) + \binom{x_{I_i}}{2} = \binom{y_{1,i}}{2} - \binom{y_{1,i-1}}{2}.$$

We have

$$\binom{y_{1,i}}{2} = \sum_{p=1}^{y_{1,i}} p = \sum_{p=1}^{\sum_{\ell=1}^{i} x_{I_\ell}} p \qquad \text{and} \qquad \binom{y_{1,i}}{2} - \binom{y_{1,i-1}}{2} = \sum_{p=1+\sum_{\ell=1}^{i-1} x_{I_\ell}}^{\sum_{\ell=1}^{i} x_{I_\ell}} p,$$

since the first $\sum_{\ell=1}^{i-1} x_{I_\ell}$ terms of the two sums are the same. We compute

$$\binom{y_{1,i}}{2} - \binom{y_{1,i-1}}{2} = \sum_{p=1}^{x_{I_i}} \left(p + \sum_{\ell=1}^{i-1} x_{I_\ell}\right) = \sum_{p=1}^{x_{I_i}} \left(\left(\sum_{\ell=1}^{i-1} x_{I_\ell}\right) + p\right) = x_{I_i} \cdot \left(\sum_{\ell=1}^{i-1} x_{I_\ell}\right) + \sum_{p=1}^{x_{I_i}} p$$

and the lemma follows. □

Finally, S_{convex} is convex with respect to $\mathbf{x}$ because,

- all x_{I_j} are linear (thus affine) functions,
- $y_i = \sum_{j\in[K]:\sigma(I_j)\geq i} x_{I_j}$ is a sum of affine functions, thus affine,
- $y_i(y_i - 1)/2$ is convex: it is a basic fact that $h(x) = g(f(x))$ is convex if f is affine and g is convex. Here $f = y_i$ is affine by the previous point and $g = f(f-1)/2$ is convex.
- S_{convex} is the sum of $y_i(y_i - 1)/2$, which are convex by the previous point.

Proof of Theorem 2(b). Apply the algorithm of Dadush [47] to Model 13. □

Notice that we could not apply Lokshtanov's algorithm because the objective has large coefficients. Also notice that we do not need separability of S_{convex} or any structure of A.

Model 13 (Sum Coloring *as* LinIP *in Fixed Dimension with Convex Objective*).
Variables & Notation:

- x_{I_j} = #of color class I_j
- y_i = #of color classes I_j with $\sigma(I_j) \leq i$
- $\binom{y_i}{2}$ cost of column y_i (Fig. 4)
- $S_{\text{convex}} = \sum_{i=1}^{|G|} \binom{y_i}{2}$ = cost of all columns

Objective & Constraints:

$$\min \; S_{\text{convex}}(\mathbf{x}) \qquad \text{(sc:convex:obj)}$$

$$\sum_{I_j : i \in I_j} x_{I_j} = |V_i| \qquad \forall \text{clique } i \in T(G) \qquad \text{clique } V_i \text{ gets } |V_i| \text{ colors} \qquad \text{(sc:convex:cliques)}$$

$$\sum_{I_j : i \in I_j} x_{I_j} = 1 \qquad \forall \text{indep. } i \in T(G) \qquad \text{indep. } V_i \text{ gets 1 color} \qquad \text{(sc:convex:indeps)}$$

Parameters & Notes:

-

#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l}, \mathbf{u}\|_\infty$	$f_{\max}$
2^k	k	1	$\|G\|$	$\|G\|$	$\|G\|^2$

- Objective S_{convex} is non-separable convex, and can be computed in time $2^k \log |G|$ by noticing that there are at most 2^k different y_i's (see below). □

5.3. Sum Coloring *and Graver bases*

Consider Model 13. The fact that the number of rows and the largest coefficient $\|A\|_\infty$ is small, and that we can formulate S_{convex} as a separable convex objective in terms of the y_i variables gives us some hope that Graver basis techniques would be applicable.

Since $|\mathcal{I}| \leq 2^k$, we can replace the y_i's by a smaller set of variables z_i for a set of "critical sizes" $\Gamma = \{j \in [|G|] \mid \exists i \in [K] : \sigma(I_i) = j\}$. For each $i \in \Gamma$ let $\text{succ}(i) = \min\{j \in \Gamma \mid j > i\}$ (and let $\text{succ}(\max \Gamma) = \max \Gamma$), we define $z_i = \sum_{j \in [K]: \sigma(I_j) \geq i} x_{I_j}$, and let $\zeta_i = (\text{succ}(i) - i)$ be the size difference between a color class of size i and the smallest larger color class (recall (1)). Then,

$$S_{\text{convex}}(\mathbf{x}) = \sum_{i=1}^{|G|} \binom{y_i}{2} = \sum_{i \in \Gamma} \zeta_i \binom{z_i}{2} = S_{\text{sepconvex}}(\mathbf{z}) \ .$$

Now we want to construct a system of inequalities of bounded dual treewidth $\text{tw}_D(A)$; however, adding the z_i variables as we have defined them amounts to adding many inequalities containing some variable x_{I_1} (recall that for I_i we have $\sigma(I_1) = \max \Gamma$), thus increasing the dual treewidth to $k + 2^k$. To avoid this, let us define z_i equivalently as $z_i = z_{\text{succ}(i)} + \sum_{\substack{j \in [K]: \\ \text{succ}(i) > \sigma(I_j) \geq i}} x_{I_j} = z_{\text{succ}(i)} + \sum_{\substack{j \in [K]: \\ \sigma(I_j) = i}} x_{I_j}$. The last equality follows from the definition of Γ which implies there are no independent sets with size strictly between i and $\text{succ}(i)$ in G.

Model 14 (Sum Coloring *as* LinIP *with Small* $\text{tw}_D(A)$ *and Small* $g_1(A)$)**.**
Variables & Notation:

- $x_{I_j} = \#$of color class I_j
- $z_i = \#$of color classes I_j with $\sigma(I_j) \geq i$
- $\zeta_i =$ size difference between I_j with $\sigma(I_j) = i$ and closest larger $I_\ell \in \mathcal{I}$
- $\zeta_i \binom{z_i}{2}$ cost of all columns between y_i and $y_{\text{succ}(i)}$ (Fig. 4)
- $\Gamma =$ set of critical sizes
- $S_{\text{sepconvex}}(\mathbf{z}) = \sum_{i \in \Gamma} \zeta_i \binom{z_i}{2} =$ total cost

Objective & Constraints: constraints (sc:convex:cliques) and (sc:convex:indeps), and:

$$\min \ S_{\text{sepconvex}}(\mathbf{z}) \qquad \text{(sc:graver:obj)}$$

$$z_i = z_{\text{succ}(i)} + \sum_{j \in [K]: \sigma(I_j) = i} x_{I_j} \qquad \forall i \in \Gamma \qquad \text{(sc:graver:sep)}$$

Parameters & Notes:

-

#vars	#constraints	$\|A\|_\infty$	$\|\mathbf{b}\|_\infty$	$\|\mathbf{l}, \mathbf{u}\|_\infty$	$f_{\max}$	$g_1(A)$	$\text{tw}_D(A)$
$\mathcal{O}\left(2^k\right)$	$\mathcal{O}\left(2^k\right)$	1	$\|G\|$	$\|G\|$	$\|G\|^2$	$\mathcal{O}\left(k^k\right)$	$k+2$

- Bounds on $g_1(A)$ and $\text{tw}_D(A)$ by Lemmas 16 and 15, respectively.
- Objective $S_{\text{sepconvex}}$ is separable convex. □

Proof of Theorem 2(c). Apply the algorithm of Koutecký et al. [60] to Model 14. □

Let us denote the matrix encoding the constraints (sc:convex:cliques) and (sc:convex:indeps) as $F \in \mathbb{Z}^{k \times 2 \cdot 2^k}$ (notice that we also add the empty columns for the z_i variables), and the matrix encoding the constraints (sc:graver:sep) by $L \in \mathbb{Z}^{2^k \times 2 \cdot 2^k}$; thus $A = \binom{F}{L}$.

Lemma 15. *In Model 14 it holds that* $\mathrm{tw}_D(A) \le k+1$.

Proof. We shall construct a tree decomposition of $G_D(A)$ of width $k+2$. The decomposition is a path and has $|\Gamma|-1$ nodes, one for each except the largest $i \in \Gamma$, in increasing order. We put all k rows of F in the bag of every node. In addition to that the bag of the ith node contains the ith and $(i+1)$-st row of L. It is not difficult now to check that this indeed satisfies the definition of a tree decomposition. □

Lemma 16. *In Model 14 it holds that* $g_1(A) \le k^{\mathcal{O}(k)}$.

The idea behind the proof is as follows. Since $A = \binom{F}{L}$ is a matrix obtained by stacking the two blocks F and L, the bound on $g_1(A)$, the largest coefficient in an element of the Graver basis of A, can be estimated using the following lemma for stacked matrices.

Lemma 17 (*Stacking Lemma [65, Lemma 3.7.6]*)**.** $g_1\left(\binom{F}{L}\right) = g_1(F \cdot \mathcal{G}(L)) \cdot g_1(L)$

Here, $\mathcal{G}(L)$ is a matrix whose columns are vectors from the Graver basis of L. Thus, we need to determine $g_1(L)$ and $g_1(F \cdot \mathcal{G}(L))$. For the first bound we provide the following technical lemma.

Lemma 18. $g_1(L) \le |\Gamma|+1$. *Moreover, for every vector* $\begin{pmatrix} \mathbf{g}^z \\ \mathbf{g}^x \end{pmatrix} \in \mathcal{G}(L)$ *we have* $\|\mathbf{g}^x\|_1 \le 2$.

The rest is a quite straightforward application of the stacking lemma.

Proof of Lemma 16. Consider the matrix $F \cdot \mathcal{G}(L)$: it is a matrix with k rows with entries bounded by the maximum of $\mathbf{f}^\intercal \mathbf{g}$ taking $\mathbf{f}$ to be a row of F and $\mathbf{g} \in \mathcal{G}(L)$. Trivially, $\|\mathbf{f}\|_\infty \le 1$ and Lemma 18 yields that $\|\mathbf{g}\|_1 \le |\Gamma| \le 2^k$, so we have $\|F \cdot \mathcal{G}(L)\|_\infty \le 2^k$. However, if we split $\mathbf{f}$ naturally into two parts corresponding to the two groups of variables $\mathbf{f} = \binom{\mathbf{f}^z}{\mathbf{f}^x}$, we observe that $\mathbf{f}^z = \mathbf{0}$ for every row $\mathbf{f}$ of F. By taking this and the latter part of Lemma 18 into account, we arrive at $\|F \cdot \mathcal{G}(L)\|_\infty \le 2$.

Eisenbrand et al. [58, Lemma 2] show that, for a matrix $E \in \mathbb{Z}^{m \times N}$, a bound of $g_1(E) \le (2m\|E\|_\infty + 1)^m$ holds. Plugging in, we obtain $g_1(F \cdot \mathcal{G}(L)) \le (2k \cdot 2 + 1)^k = \mathcal{O}\left(k^k\right)$, and using the stacking lemma, $g_1(A) \le \mathcal{O}\left(k^k\right) \cdot 2^k = k^{\mathcal{O}(k)}$. □

Proof of Lemma 18. We first simplify the structure of L. It is known [65, Lemma 3.7.2] that repeating columns of a matrix B does not increase $g_1(B)$; thus, it is enough to bound $g_1(L')$, where L' is obtained from L by deleting duplicitous columns. Note that the columns corresponding to variables x_{I_i}, x_{I_j} are duplicitous whenever $\sigma(I_i) = \sigma(I_j)$. So we may assume that L' has the following form, obtained by keeping only one column for $\mathbf{x}$ for every $i \in \Gamma$:

$$\alpha_1 = \beta_1 \tag{2}$$

$$\alpha_i = \alpha_{i-1} + \beta_i \qquad \forall i \in [2, |\Gamma|]. \tag{3}$$

First we are going to show that any integer vector $\mathbf{h}$ with $L'\mathbf{h} = \mathbf{0}$ can be written as a sum of integer vectors $\mathbf{g}^1, \ldots, \mathbf{g}^M$ for some $M \in \mathbb{N}$, which satisfy $L'\mathbf{g}^i = \mathbf{0}$, $\mathbf{g}^i \sqsubseteq \mathbf{h}$, and $\|\mathbf{g}^i\|_1 \le |\Gamma|+1$, for all $i \in [M]$. This is sufficient because while the $\mathbf{g}^i$'s might not be elements of $\mathcal{G}(L')$ themselves, their maximum ℓ_1-norm upper bounds $g_1(L')$. To see this, observe that each such vector can be decomposed further into a $\sqsubseteq$-sum of vectors from a Graver basis of L' and notice further that if $\mathbf{g}' \sqsubseteq \mathbf{g}$, then $\|\mathbf{g}'\|_1 \le \|\mathbf{g}\|_1$.

The rest of the proof is by induction on $\|\mathbf{h}\|_1$. If $\|\mathbf{h}\|_1 = 0$, the claim clearly follows. Otherwise let $\mathbf{h} = \begin{pmatrix} \mathbf{h}^\alpha \\ \mathbf{h}^\beta \end{pmatrix}$ with $\|\mathbf{h}\|_1 > 0$ and $L'\mathbf{h} = \mathbf{0}$. We have to find a nonzero vector $\mathbf{g} = \begin{pmatrix} \mathbf{g}^\alpha \\ \mathbf{g}^\beta \end{pmatrix}$ with $\|\mathbf{g}\|_1 \le |\Gamma|+1$ and $L'\mathbf{g} = \mathbf{0}$ such that $\mathbf{g} \sqsubseteq \mathbf{h}$ and $\|\mathbf{h} - \mathbf{g}\|_1 < \|\mathbf{h}\|_1$.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

To see this, first observe that if $\mathbf{h} \neq \mathbf{0}$, then $\mathbf{h}^\alpha \neq \mathbf{0}$. Let $i \in [|\Gamma|]$ be such that $h_1^\alpha = \cdots = h_{i-1}^\alpha = 0$ and $h_i^\alpha \neq 0$. Now, using (2) and (3), we observe the following.

Claim 1. *We have* $h_1^\beta = \cdots = h_{i-1}^\beta = 0$ *and* $\operatorname{sign}(h_i^\alpha) = \operatorname{sign}(h_i^\beta)$.

Proof. Since $h_1^\alpha = \cdots = h_{i-1}^\alpha$, we have $h_1^\beta = \cdots = h_{i-1}^\beta$. Now (3) together with $h_{i-1}^\alpha = 0$ results in $h_i^\alpha = h_i^\beta$ and the claim follows. ◁

Now there are two cases: either $\operatorname{sign}(h_{i+1}^\beta) = -\operatorname{sign}(h_i^\alpha)$ or $\operatorname{sign}(h_{i+1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$.

Suppose $\operatorname{sign}(h_{i+1}^\beta) = -\operatorname{sign}(h_i^\alpha)$. Let $\mathbf{g}^\alpha = \operatorname{sign}(h_i^\alpha) \cdot \mathbf{e}_i$ and let $\mathbf{g}^\beta = \operatorname{sign}(h_i^\alpha) \cdot (\mathbf{e}_i - \mathbf{e}_{i+1})$, where $\mathbf{e}_i$ is the ith unit vector, i.e., a vector with zeros everywhere except of the ith coordinate, which is 1. Observe that now $\mathbf{g}^\alpha$ affects solely variable α_i and thus we have to care for the only two conditions containing α_i (recall $\alpha_{i-1} = 0$):

$$\alpha_i = \beta_i \qquad \text{and} \qquad \alpha_{i+1} = \alpha_i + \beta_{i+1}.$$

This leaves us with a matrix with columns corresponding to $\alpha_i, \alpha_{i+1}, \beta_i$, and β_{i+1}

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & -1 \end{pmatrix}.$$

The vector $\mathbf{g}$, defined above, now corresponds to a vector $(1,0,1,1)^\intercal$. It is easy to see that this vector is in kernel of the matrix and, since α_i is the only affected α-variable, we get $L'\mathbf{g} = \mathbf{0}$ and we are done in this case. Notice that in this case we have $\|\mathbf{g}\|_1 = 3$.

Now suppose $\operatorname{sign}(h_{i+1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$. We observe that this affects sign of α_{i+1}.

Claim 2. *If* $\operatorname{sign}(h_{i+1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$, *then* $\operatorname{sign}(h_i^\alpha) = \operatorname{sign}(h_{i+1}^\alpha)$.

Proof. Suppose $\operatorname{sign}(h_i^\alpha) = 1$, the other case follows by a symmetric argument. Then h_{i+1}^β is nonnegative and by (3) we obtain that h_{i+1}^α is a sum of a positive and a nonnegative number, thus a positive number as claimed. ◁

We are about to design a vector $\mathbf{g}$ for which

$$\alpha_{i+1} = \alpha_i + \beta_{i+1}$$

holds. Since $\operatorname{sign}(h_{i+1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$, we cannot use g_{i+1}^β to fulfill the above condition and thus if $g_i^\alpha \neq 0$, then $\operatorname{sign}(g_{i+1}^\alpha) = \operatorname{sign}(g_i^\alpha)$. Now if we set $g_i^\alpha = g_i^\beta = g_{i+1}^\alpha = \operatorname{sign}(g_i^\alpha)$ we fulfill all conditions (3) (recall $\alpha_{i-1} = 0$). But now the condition

$$\alpha_{i+2} = \alpha_{i+1} + \beta_{i+2}$$

is not satisfied. However, we have essentially carried the difficulty from α_i to α_{i+1}. Since now either $\operatorname{sign}(h_{i+1}^\beta) = -\operatorname{sign}(h_i^\alpha)$ or $\operatorname{sign}(h_{i+1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$, we arrive at the following.

Claim 3. *Either*

1. *there exists* j *with* $i < j \leq |\Gamma|$ *such that* $\operatorname{sign}(h_i^\beta), \ldots, \operatorname{sign}(h_{j-1}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$ *and* $\operatorname{sign}(h_j^\beta) = -\operatorname{sign}(h_i^\alpha)$ *or*
2. *it holds that* $\operatorname{sign}(h_i^\beta), \ldots, \operatorname{sign}(h_{|\Gamma|}^\beta) \in \{\operatorname{sign}(h_i^\alpha), 0\}$.

Let $j = |\Gamma|$ *in the second case then we have* $\operatorname{sign}(h_i^\alpha) = \cdots = \operatorname{sign}(h_j^\alpha)$.

Proof. By repeated applications of Claim 2 we get $\operatorname{sign}(h_k^\alpha) = \operatorname{sign}(h_{k+1}^\beta)$ for all $i \le k \le j-1$. Initially the premise of Claim 2 is what we suppose for this case and each application yields the premise of Claim 2 for the next application. ◁

Let j be defined as in the above claim. Now, we finish the construction of $\mathbf{g}$ by setting $\mathbf{g}^\alpha = \sum_{k=i}^{j} \mathbf{e}_k$. In the first case of the above claim we let $\mathbf{g}^\beta = \mathbf{e}_i + \mathbf{e}_j$ while in the second we have $\mathbf{g}^\beta = \mathbf{e}_i$. It is not hard to verify that $L'\mathbf{g} = \mathbf{0}$. Indeed in the first case at index j we essentially arrive to the situation described above when we argued about $\operatorname{sign}(h_{i+1}^\beta) = -\operatorname{sign}(h_i^\alpha)$. While if $j = |\Gamma|$, there is no carry, as there are no further rows of L'. The claimed bound on $g_1(L)$ follows by observing that we have $\|\mathbf{g}\|_1 \le |\Gamma| + 1$ in both of the just described cases. As for the latter part of the Lemma, observe that in every case we have $\left\|\mathbf{g}^\beta\right\|_1 \le 2$ and notice that these variables correspond to the x_{I_i} variables of the given model. □

6. Quantifiers: Parametric ILP and 2-stage stochastic IP

In this section we shall demonstrate the use of tools related to quantified formulas of linear constraints over integer variables, namely Parametric ILP and 2-stage stochastic IP, on additional coloring problems on graphs of bounded neighborhood diversity.

6.1. Stackelberg strategy: Parametric ILP

Panagopoulu and Spirakis [28] define a coloring game on graphs and show that a pure Nash equilibrium of this game can be computed in polynomial time and satisfies many known coloring lower bounds. Eventually they pose a computational problem of computing a so-called Stackelberg strategy, which is essentially a precoloring of a subset of vertices called the *leaders* in such a way that even the worst pure Nash equilibrium of the resulting game is not too far from an optimal coloring.

For a simple undirected graph G, the *graph coloring game* $\Gamma(G)$ is defined as follows. There are $n := |V(G)|$ *players* and a set of *actions* $A = \{a_1, \dots, a_n\}$ corresponding to the n available colors. A *configuration* is a tuple $\mathbf{c} = (c_v)_{v \in V(G)} \in A^n$, where c_v is the color chosen by a vertex $v \in V(G)$. For a configuration $\mathbf{c} \in A^n$ and $a \in A$, we define $n_a(\mathbf{c}) := |\{v \in V(G) \mid c_v = a\}|$, i.e., $n_a(\mathbf{c})$ is the number of vertices which have color a in the configuration $\mathbf{c}$. The *payoff of a vertex* $v \in V(G)$ *in a configuration* $\mathbf{c} \in A^n$ is given by

$$\lambda_v(\mathbf{c}) := \begin{cases} 0 & \text{if } \exists u \in N(v) \text{ such that } c_u = c_v \\ n_{c_v}(\mathbf{c}) & \text{otherwise.} \end{cases}$$

A *pure Nash equilibrium (PNE)* is a configuration $\mathbf{c} \in A^n$ in which no vertex can increase its profit by unilaterally deviating, i.e., if for every vertex $v \in V(G)$ and each color $a \in A$ it holds that $\lambda_v(a, \mathbf{c}_{-v}) \le \lambda_v(\mathbf{c})$, where $(a, \mathbf{c}_{-v})$ is the configuration in which v chooses a instead of c_v and the remaining vertices follow the strategy $\mathbf{c}$. The *cost of* $\mathbf{c}$, denoted $\operatorname{cost}(\mathbf{c})$, is the number of distinct colors of $\mathbf{c}$, i.e., $\operatorname{cost}(\mathbf{c}) := |\{c_v \mid v \in V(G)\}|$.

Stackelberg game. Given the coloring game $\Gamma(G)$, we may define a corresponding *Stackelberg game*. In it, first at most B of vertices from a subset $V^L \subseteq V$ of *leaders* choose their color, and then the remaining *followers* $V \setminus V^L$ choose their colors. The process in which the followers choose their colors is the *subgame* induced by the choice of the leaders, and a PNE of this subgame is such a configuration $\mathbf{c}$ in which no vertex can unilaterally increase its payoff. The computational task is to compute a *strategy*, which is the choice of colors for the leaders, such that every PNE of the subgame uses at most some specified number $\chi' \in [n]$ of colors. Two interesting special cases of this game are where just B or just V^L are specified, which correspond to the tuples (V, B) and $(V^L, |V^L|)$.

	Stackelberg Strategy for $\Gamma(G)$
Input:	A graph $G=(V,E)$, a set of colors $A=[n]$ a number $\chi' \in [n]$, $V^L \subseteq V(G)$, and $B \in \mathbb{N}$.
Task:	A subset of vertices V' which satisfies $V' \subseteq V^L$ and $\lvert V'\rvert \le B$, and a proper coloring $\gamma\colon V' \to A$ such that any PNE $\mathbf{c}''$ of the subgame satisfies $\mathrm{cost}((\mathbf{c}',\mathbf{c}'')) \le \chi'$, where $\mathbf{c}'$ is the configuration corresponding to γ.

Our plan is to formulate a sentence of the form $\exists \mathbf{x} \forall \mathbf{y} : \Phi(\mathbf{x},\mathbf{y})$, where $\mathbf{x}$ represents a Stackelberg strategy, $\mathbf{y}$ represents an essential coloring, and $\Phi(\mathbf{x},\mathbf{y})$ is a formula encoding "if $\mathbf{y}$ is consistent with $\mathbf{x}$ and no selfish step exists, then $\mathbf{y}$ uses at most χ' colors". Knop et al. [23] show that sentences like these can be solved using the FPT algorithm for Parametric ILP due to Eisenbrand and Shmonin [51]. More accurately, we can phrase the main tool developed by Knop et al. as follows:

Theorem 19 *([23, Theorem 4.5]).* *Let $\Psi \equiv \exists \mathbf{x} \in \mathbb{Z}^{d_1} \forall \mathbf{y} \in \mathbb{Z}^{d_2} : \Phi(\mathbf{x},\mathbf{y})$ where $\Phi(\mathbf{x},\mathbf{y})$ is a quantifier free formula obtained by standard logical connectives from a collection of linear inequalities of the form $\mathbf{a}(\mathbf{x},\mathbf{y}) \le b$, where $\mathbf{a} \in \mathbb{Z}^{d_1+d_2}$, $b \in \mathbb{Z}$. Let L be the total number of symbols describing Ψ and let α be the largest value of $\|\mathbf{a},b\|_\infty$ over all atoms of Φ. Let β be such that $\Psi' \equiv \exists \mathbf{x} \in [-\beta,\beta]^{d_1} \forall \mathbf{y} \in [-\beta,\beta]^{d_2} : \Phi(\mathbf{x},\mathbf{y})$ is equivalent to Ψ (we say that Ψ is β-*bounded*). Then Ψ can be decided in time*[3] $f(L)\,\mathrm{poly}(\beta+\alpha)$.

Solution structure. First, Panagopoulu and Spirakis observe that every PNE of $\Gamma(G)$ is a proper coloring [28, Lemma 1], simply because in any improper coloring, there is some vertex which can selfishly increase its payoff by using a so far unused color. Clearly the same holds for PNEs of any subgame induced by some Stackelberg strategy. However, Panagopoulu and Spirakis [28] do *not* prove the existence of PNEs in the Stackelberg subgames, yet we may reuse one of their constructions to that end.

Lemma 20. *Given any configuration $\mathbf{c}'$ of a subset of vertices $V' \subseteq V$, the coloring subgame induced by $\mathbf{c}'$ has a PNE.*

Proof. Define a potential function $P(\mathbf{c}) = \frac{1}{2}\sum_{v\in V} n_{c_v}(\mathbf{c})$. Panagopoulu and Spirakis show [28, Proof of Theorem 2] that for any proper coloring $\mathbf{c}$, $P(\mathbf{c})$ is upper bounded by $\frac{n\alpha(G)}{2}$, where $\alpha(G)$ is size of the largest independent set in G, and that it is lower bounded by $\frac{n}{2}$. Moreover, any selfish step increases $P(\mathbf{c})$ by at least 1. This immediately implies that $\Gamma(G)$ has a PNE. However, it also shows that for any $\mathbf{c}'$ the corresponding subgame has a PNE: choose a proper coloring $\mathbf{c}''$ of $V \setminus V^L$ consistent with $\mathbf{c}'$ such that $P((\mathbf{c}',\mathbf{c}''))$ is maximized and note that it cannot contain any vertex which may selfishly improve its payoff, since this would increase the potential. □

Lemma 21. *In any subgame PNE and any independent type V_j, all followers in V_j have the same color.*

Proof. The proof is again by a swapping argument: assume for contradiction that V_j has a follower colored red and a follower colored blue. Without loss of generality assume there are at least as many red vertices as blue vertices. Then a blue vertex can improve its payoff by switching to red. □

Our inspiration for the following modeling approach comes from Ganian's model for the Precoloring Extension problem [8]. Notice that a partial proper coloring is a proper coloring of a subgraph of G and thus can be encoded using the previously explained approach (see Section 5.2) of counting the occurrences

[3] In fact, the time bound proven but not stated in [23] is $f(L)(\beta+\alpha)^{o(1)}$ but here we present the theorem in the same way it is presented therein.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

of independent sets of $T(G)$, denoted $\mathcal{I}$. Thus, we can encode a subgame PNE $\mathbf{c}''$, which is a proper coloring of $G[V \setminus V']$, using this approach.

However, we currently have little insight into the structure of Stackelberg strategies $\mathbf{c}'$. It would be helpful to show that, for example, they use at most one color in each independent type, or, on the other hand, that colors are distributed evenly in each independent type. Since we do not know this, we resort to expressing the Stackelberg strategy using the approach from Section 5.1, where we encode explicitly how many vertices from each type a color uses.

In order to encode a formula Φ as described above we will proceed gradually. First, we need to enforce that $\mathbf{x}$ is a coloring of $V' \subseteq V^L$ with $|V'| \leq B$, and $\mathbf{x}$ uses ζ colors. The form of $\mathbf{x}$ is $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^\zeta)$, where for each $j \in [\zeta]$, $\mathbf{x}^j = (x_1^j, \ldots, x_k^j)$ describes one color used in the precoloring of leaders. Hence, we have a predicate

$$\mathrm{color}(x_1, \ldots, x_k) \equiv \left(\bigwedge_{V_i \text{ clique}} x_i \leq 1 \right) \wedge \left(\bigwedge_{V_i \text{ independent}} x_i \leq |V_i| \right) \wedge \bigwedge_{ij \in E(T(G))} \neg(x_i \geq 1 \wedge x_j \geq 1),$$

encoding the constraint that for $(x_1, \ldots, x_k)$ to be a color, it needs to take at most 1 vertex from a clique type, at most $|V_i|$ vertices from an independent type V_i, and never a positive number of vertices from two neighboring types. Next, we express that $\mathbf{x}$ is a feasible precoloring of $G[U]$ for some $U \subseteq V$; we will reuse this formula for different $U \subseteq V$. As usual, we denote $U_j = U \cap V_j$.

$$\text{feasible-precoloring}(\mathbf{x}, U) \equiv \left(\bigwedge_{i=1}^{\zeta} \mathrm{color}(\mathbf{x}^i) \right) \wedge \left(\bigwedge_{j=1}^{k} (\sum_{\ell=1}^{\zeta} x_j^\ell \leq |U_j|) \right) \ .$$

A potential strategy is a precoloring of $G[V^L]$ using at most B vertices:

$$\text{potential-strategy}(\mathbf{x}) \equiv (\|\mathbf{x}\|_1 \leq B) \wedge \text{feasible-precoloring}(\mathbf{x}, V^L) \ .$$

Next, we will introduce variables $\bar{\mathbf{x}}$ expressing how the ζ colors of a potential strategy $\mathbf{x}$ are extended in some subgame PNE. Enforcing this is simply stating that $\bar{\mathbf{x}}$ is again a feasible-precoloring which moreover dominates $\mathbf{x}$:

$$\mathrm{extension}(\mathbf{x}, \bar{\mathbf{x}}) \equiv (\mathbf{x} \leq \bar{\mathbf{x}}) \wedge \text{feasible-precoloring}(\bar{\mathbf{x}}, V) \ .$$

Next, we will use variables $\mathbf{y}$ to describe the new colors appearing in the subgame PNE, so we need to express that $\mathbf{y}$ is an essential coloring of the vertices which remain uncolored by $\bar{\mathbf{x}}$. We will use the encoding of an essential coloring by multiplicities of independent sets of $T(G)$:

$$\begin{aligned}\text{remainder-coloring}(\bar{\mathbf{x}}, \mathbf{y}) \equiv & \left(\bigwedge_{V_i \text{ clique}} (\sum_{j=1}^{K} y_{I_j}) \leq |V_i| - (\sum_{\ell=1}^{\zeta} x_i^\ell) \right) \\ & \wedge \left(\bigwedge_{V_i \text{ independent}} (\sum_{j=1}^{K} y_{I_j}) \leq \min\{1, |V_i| - (\sum_{\ell=1}^{\zeta} x_i^\ell)\} \right) \ .\end{aligned}$$

The remaining task is to formulate that $(\bar{\mathbf{x}}, \mathbf{y})$ is a subgame PNE. To that end, we need to be able to encode that there is no vertex which can be recolored to a color of larger size. Notice that even though the number of colors can be large, it is enough to check for each color $i = 1, \ldots, \zeta$ and each $I_j \in \mathcal{I}$ with $y_{I_j} \geq 1$, and there is only parameter-many (i.e., $g(k, \zeta)$ for a computable function g) of those colors. (In other words, all colors corresponding to a set $I \in \mathcal{I}$ behave identically, so it suffices to check one of them.) With an obvious overloading of notation, we define

$$\mathrm{size}(\ell, \bar{\mathbf{x}}, \mathbf{y}) \equiv \|\bar{\mathbf{x}}^\ell\|_1$$

$$\text{size}(I, \bar{\mathbf{x}}, \mathbf{y}) \equiv \left(\sum_{i \in I: V_i \text{ clique}} 1 \right) + \left(\sum_{i \in I: V_i \text{ independent}} |V_i| - (\sum_{\ell=1}^{\zeta} x_i^\ell) \right) .$$

It is easy now to express that $\bar{\mathbf{x}}, \mathbf{y}$ is a Stackelberg strategy:

$$\begin{aligned}\text{strategy}(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{y}) \equiv & \text{potential-strategy}(\mathbf{x}) \wedge \text{extension}(\mathbf{x}, \bar{\mathbf{x}}) \wedge \text{remainder-coloring}(\bar{\mathbf{x}}, \mathbf{y}) \\ & \neg \exists_{c,c' \in [\zeta] \cup \mathcal{I}} \exists_{i \in [k]: i \in c, c' \cup \{i\} \in \mathcal{I}} \text{size}(c, \bar{\mathbf{x}}, \mathbf{y}) \le \text{size}(c', \bar{\mathbf{x}}, \mathbf{y}) \ .\end{aligned}$$

(Again, we overload notation by saying $c' \cup \{i\} \in \mathcal{I}$, which means that either i is already a part of the color c', or it can be added to it while keeping it a coloring. This is clearly also expressible, but would further complicate the formulas.)

Model 22 (Stackelberg Strategy for $\Gamma(G)$ *as a Parametric ILP in Fixed Dimension*)**.**
Variables & notation:

- x_j^ℓ = #vertices from V_j colored ℓ in $\mathbf{c}'$
- $\bar{x}_j^\ell$ #vertices from V_j colored ℓ in $(\mathbf{c}', \mathbf{c}')$
- y_{I_j} = #of color classes $I_j \in \mathcal{I}$ in the subgame PNE

Constraints:

$$\text{decide } \Psi \equiv \exists \mathbf{x} \forall (\bar{\mathbf{x}}, \mathbf{y}) : \text{strategy}(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{y}) \implies (\zeta + \sum_{j=1}^{K} y_{I_j}) \le \chi' \ . \tag{4}$$

Parameters & Notes:

L	α	β
$\mathcal{O}(2^k + \zeta \cdot k)$	1	$\lvert V(G)\rvert$
- The fact that $\beta \le |V(G)|$ follows because the largest constants in the definitions of the subformulas are the terms $|V_i|$. $\square$

Proof of Theorem 4. Apply Theorem 19 to Model 22. $\square$

6.2. Stochastic precoloring: 2-stage stochastic IP

In the following we try to demonstrate the similarities and differences of Parametric ILP and 2-stage stochastic IP. For a proper coloring γ of G, and H a supergraph of G, let $\chi_\gamma(H)$ be the smallest number such that H has a proper coloring by $\chi_\gamma(H)$ colors which is an extension of γ. Consider the following (artificial) problem:

Stochastic Precoloring	
Input:	A graph $G = (V, E)$, n pairs (p_j, H_j) where, for all $j \in [n]$, G is an induced subgraph of H_j, $p_j > 0$, and $\sum_{j=1}^n p_j = 1$.
Task:	A proper coloring γ of G minimizing $\sum_{j=1}^n p_j \chi_\gamma(H_j)$.

In plain speech, we are given a graph G and we are expecting some new vertices to arrive, eventually forming a supergraph H of G. However, there is uncertainty with respect to how will these new vertices connect to

G. The possible options are represented by n *scenarios*, each occurring with a given probability. The goal is to minimize the expected value of $\chi_\gamma(H)$.

Solution structure. Our parameter is $k := \max_j \mathrm{nd}(H_j)$. Because taking induced subgraphs does not increase the neighborhood diversity, we know that $\mathrm{nd}(G) \le k$. Denote by $\mathcal{I}$ the independent sets of $T(G)$ and by $\mathcal{I}_j$ the independent sets of $T(H_j)$. Let V_i^j be the ith type of $T(H_j)$.

We will construct a 2-stage stochastic IP with n blocks, whose variables are naturally partitioned into $n+1$ *bricks*. The 0-th brick corresponds to so-called *global* or *first stage* variables, which will encode an essential precoloring γ (i.e., an essential coloring of G) and we will denote it $\mathbf{x}$. The remaining n bricks are so called *local* or *second stage* variables and will be denoted $\mathbf{y}^j$ and $\mathbf{z}^j$. The variables $y^j_{J_\ell}$ for $J_\ell \in \mathcal{I}_j$, will encode an essential coloring of H_j, and variables x_{I_h} for $I_h \in \mathcal{I}$, will encode a precoloring of G. What remains to do is encode a certificate that $\mathbf{y}^j$ is consistent with the precoloring of G encoded by $\mathbf{x}$. This certificate is a vector of variables $\mathbf{z}^j$, where we have $z^j_{I_h,J_\ell}$, for every pair $J_\ell \in \mathcal{I}_j$, $I_h \subseteq J_\ell$ (we stress that this includes $\emptyset \subseteq J_\ell$) which encodes how many color classes whose precoloring induces I_h in $T(G)$ have been extended into a color class inducing J_ℓ in $T(H_j)$.

Specifically, for a given vector $\mathbf{x}$ encoding a precoloring of G, we say that a vector $\mathbf{y}$ encoding a coloring of G is $\mathbf{x}$-*consistent* if $\mathbf{y}$ is an extension of the coloring $\mathbf{x}$. For a precoloring $\mathbf{x}$, a coloring $\mathbf{y}$, and a certificate $\mathbf{z}$, we say that $\mathbf{z}$ *is consistent with* $\mathbf{x}$ if $\sum_{I_\ell \in \mathcal{I}_j} z_{I_h,J_\ell} = x_{I_h}$ for all $I_h \in \mathcal{I}$, and we say that $\mathbf{z}$ *is consistent with* $\mathbf{y}$ if $\sum_{I_h \in \mathcal{I}} z_{I_h,J_\ell} = y_{J_\ell}$ for all $J_\ell \in \mathcal{I}$. For a coloring $\mathbf{y}^j$ of H_j we overload the cost operator to be $\mathrm{cost}(\mathbf{y}^j) := \sum_{J_\ell \in \mathcal{I}_j} y^j_{J_\ell}$.

Model 23 (Stochastic Precoloring *as a 2-stage Stochastic IP*).
Variables & notation:

- x_{I_h} = #of color classes I_h in a coloring γ of G
- $y^j_{J_\ell}$ = #of color classes J_ℓ in an optimal extension of γ in H_j
- $z^j_{I_h,J_\ell}$ = #of color classes I_h extended into J_ℓ in H_j

Constraints:

$$\min \sum_{j=1}^{n} p_j \left(\sum_{J_\ell \in \mathcal{I}_j} y^j_{J_\ell} \right) \qquad \sum_{j=1}^{n} p_j \chi_\gamma(H_j) \qquad \text{(sp:obj)}$$

$$\sum_{I_h \in \mathcal{I} : i \in I_h} x_{I_h} = n_i \qquad \forall i \in T(G) \qquad \text{all } G \text{ colored by } \mathbf{x} \qquad \text{(sp:V)}$$

$$\sum_{J_\ell \in \mathcal{I}_j : i \in J_\ell} y^j_{J_\ell} = |V_i^j| \qquad \forall j \in [n]\, \forall i \in T(H_j) \qquad \text{all } H_j \text{ colored by } \mathbf{y}^j \qquad \text{(sp:Vj)}$$

$$\sum_{J_\ell : I_h \subseteq J_\ell} z^j_{I_h,J_\ell} = x_{I_h} \qquad \forall I_h \in \mathcal{I} \qquad \mathbf{z}^j \text{ is } \mathbf{x}\text{-consistent} \qquad \text{(sp:zx)}$$

$$\sum_{I_h : I_h \subseteq J_\ell} z^j_{I_h,J_\ell} = y^j_{J_\ell} \qquad \forall J_\ell \in \mathcal{I}_j \qquad \mathbf{z}^j \text{ is } \mathbf{y}^j\text{-consistent} \qquad \text{(sp:zy)}$$

$$\mathbf{0} \le \mathbf{x}, \mathbf{y}^1, \dots, \mathbf{y}^n, \mathbf{z}^1, \dots, \mathbf{z}^n \le |V(H)| \mathbf{1} \qquad \text{bounds} \qquad \text{(sp:bounds)}$$

Parameters & Notes:

-

#vars	#constraints	$\lVert A\rVert_\infty$	$\lVert \mathbf{b}\rVert_\infty$	$\lVert \mathbf{l}, \mathbf{u}\rVert_\infty$	$\lVert \mathbf{w}\rVert_\infty$	r	s	t	n
$\mathcal{O}(2^{2k} n)$	$\mathcal{O}(2^k n)$	1	$\lvert V(H)\rvert$	$\lvert V(H)\rvert$	$p'_{\max}$	2^k	$2^{2k}+2^k$	$\mathcal{O}(2^k)$	n

- Here and in (sp:bounds), $|V(H)| := \max_{j \in [n]} |V(H_j)|$ and $\mathbf{0}$ and $\mathbf{1}$ are the all-zeros and all-ones vectors of appropriate dimensions.

- The objective function needs to be integral, which can be achieved by scaling as follows. Let[4] $p_j = r_j/q_j$, take as q the least common multiple of all q_j, and let $p'_j := qp_j$. Let $p'_{\max} := \max_{j\in[n]} p'_j$. Then, $\mathbf{w}' := q\mathbf{w}$ is integral and clearly we have not changed the set of optima.
- The constraints have a 2-stage stochastic format with n blocks: the variables $\mathbf{x}$ are the first-stage variables, and the variables $\mathbf{y}^j, \mathbf{z}^j$ form the jth block of the second stage variables. The fact that the matrix B_2 of the jth block does not overlap with any other blocks follows from the fact that constraints (sp:Vj)–(sp:zy) only contain variables of the jth block or the first-stage variables. □

Proof of Theorem 5. Apply the algorithm of Koutecký et al. [60] to Model 23. □

Appendix A. Convex integer programming and parameterized complexity

In this section we overview existing results regarding minimization of convex (Appendix A.1), concave (Appendix A.2) and indefinite (Appendix A.3) objectives in small dimension, and them move on to the rapidly growing area of IP in variable dimension (Appendix A.4). The outline is inspired by Chapter 15 of the book *50 Years of Integer Programming* [66], omitting some parts but including many recent developments.

A.1. Convex integer minimization in small dimension

Lenstra's result from 1983 shows that solving integer linear programming (ILP) is polynomial when the integer dimension is small [4]. His result extends to the case where there are few integer variables but polynomially many continuous variables, called *mixed ILP*:

$$\min\{\mathbf{w}\mathbf{x} \mid A\mathbf{x} \le \mathbf{b},\ \mathbf{x} \in \mathbb{Z}^n \times \mathbb{R}^{n'}\}\ . \tag{MILP}$$

Lenstra's algorithm was subsequently improved by Kannan [3] and Frank and Tardos [2] in two ways. First, the required space was reduced from exponential to polynomial in the dimension, and second, running time dependency on the dimension n was reduced from $2^{2^{\mathcal{O}(n)}}$ to $n^{\mathcal{O}(n)}$. The main procedure in all of these algorithms is deciding *feasibility*, i.e., is $\{\mathbf{x} \mid A\mathbf{x} \le \mathbf{b}\} \cap (\mathbb{Z}^n \times \mathbb{R}^{n'})$ nonempty? In order to optimize one does binary search over the objective, as described by Fellows et al. [6]. We would like to point out that while Lenstra's result is old, we are aware of only a few [11,14] applications which involve *mixed* ILPs.

Theorem 24 (*Frank and Tardos [2], Fellows et al. [6]*)**.** *It is possible to solve (MILP) using $\mathcal{O}\left(n^{2.5n} \cdot \mathrm{poly}(n') \cdot \langle A, \mathbf{b}, \mathbf{w}\rangle\right)$ arithmetic operations and space polynomial in $(n + n') \cdot \langle A, \mathbf{b}, \mathbf{w}\rangle$.*

This result was later generalized to minimizing a quasiconvex function over a convex set, i.e., problem (IP) with f quasiconvex. A function $f \colon \mathbb{R}^n \to \mathbb{R}$ is called *quasiconvex* if for every $\alpha \in \mathbb{R}$, the *level set* $\{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \le \alpha\}$ is a convex subset of $\mathbb{R}^n$. The first to show this was Grötschel, Lovász and Schrijver in their famous book [43, Theorem 6.7.10]. Unlike above, all of the following results require space exponential in the dimension. Also, none of the cited results explicitly deals with the mixed integer case; however it is folklore that this is FPT as well.

The subsequent research diverged in several directions. The main difference between the papers we discuss is in the assumptions on the representation of the convex set S. Since there is, strictly speaking, no "better" or "worse" assumption, choosing one is a matter of preference with respect to the specific scenario. Another difference is in the motivation: some authors seek to achieve better time complexity while others contribute by simplifying existing proofs. Our list is categorized according to the assumptions on the representation of S.

[4] The input of the Robust Precoloring problem need not even be rational, but an equivalent rational objective can be computed using the algorithm of Frank and Tardos [2]. Thus, for simplicity we just assume that all p_j are rational.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

Semialgebraic convex set. Khachiyan and Porkolab [44] state their result for minimizing a quasiconvex function over a *semialgebraic convex set*; without going into technical details, let us say that these are closely related to spectrahedra, the solution spaces of semidefinite programs. Independently, convex sets and semialgebraic sets have been studied for a long time, but together they have been studied only in the past ten years as *Convex Algebraic Geometry*; cf. a book on the topic by Blekherman, Parillo and Thomas [67]. A drawback of this result is an exponential dependence on the number of polynomials defining the semialgebraic convex set.

Theorem 25 (*Khachiyan and Porkolab [44]*)**.** *Problem (IP) with f quasiconvex and S a semialgebraic convex set defined by k polynomials is* FPT *with respect to k and n.*

Quasiconvex polynomials. Heinz [45] studied a more specific case of minimizing a quasiconvex polynomial over a convex set given by a system of quasiconvex polynomials, that is, polynomials that are quasiconvex functions. His result improves over Khachiyan and Porkolab in terms of time complexity, dropping the exponential dependence on the number of polynomials. The dependence on the dimension n is $\mathcal{O}\left(2^{n^3}\right)$, which was further improved by Hildebrand and Köppe [46] to $n^{\mathcal{O}(n)}$. The latter result can be stated as follows. Let $\hat{F}, F_1, \ldots, F_m \in \mathbb{Z}[\mathbf{x}] = \mathbb{Z}[x_1, \ldots, x_n]$ be polynomials with integer coefficients. Then we get the Convex Polynomial IP problem

$$\min\{\hat{F}(\mathbf{x}) \mid F_i(\mathbf{x}) < 0\, \forall i \in [m],\, \mathbf{x} \in \mathbb{Z}^n\}\ . \tag{CPIP}$$

Theorem 26 (*Hildebrand and Köppe [46]*)**.** *Given a (CPIP) instance with $\mathcal{F} = \left\{\hat{F}, F_1, \ldots, F_m\right\}$. Let $d \geq 2$ be the upper bound on the degree of each $F \in \mathcal{F}$, M is the maximum number of monomials in each $F \in \mathcal{F}$ and ℓ bounds the binary length of the coefficients of F. Then it can be solved in time:*

- $n^{\mathcal{O}(n)} \cdot m(r\ell Md)^{\mathcal{O}(1)}$*, thus* FPT *with respect to the dimension n, if the feasible region is bounded such that r is the binary encoding length of that bound with $r \leq \ell d^{\mathcal{O}(p)}$,*
- $d^{\mathcal{O}(n)} n^{2n} \cdot m\ell^{\mathcal{O}(1)}$*, thus* FPT *with respect to the dimension n and the maximum degree d, if the feasible region is unbounded.*

Note that, in particular, the running time is polynomial with respect to the number of polynomials m. We also note that the quantities r and ℓ are natural and effectively appear in the $\langle A, \mathbf{b}, \mathbf{w}\rangle$ runtime term of Theorem 24.

An advantage of representing S and f by polynomials is that the representation is "explicit", in contrast to representing them by an oracle. Polynomial objectives appear for example in scheduling [12,13] where models of small dimension represent jobs by multiplicities, and an objective such as $\sum w_j C_j$ (sum of weighted completion times) becomes quadratic in this encoding. The drawback of representing S or f by polynomials is that it is sometimes unnatural, with piece-wise linear convex constraints (Model 7) or objectives [24].

Oracles. Further research lead to splitting convex IP (i.e., problem (IP) with f convex) in two independent parts to allow more focus on each of them. The first part is showing that a certain problem formulation (such as quasiconvex polynomial inequalities, semialgebraic set etc.) can be used to give a set of geometric oracles. The second part is to show that, given these oracles, solving a convex IP can be done in a certain time.

This approach is taken by Dadush, Peikert and Vempala [47] who further improve the time complexity of Hildebrand and Köppe [46] when the convex set is given by three oracles: a so-called *weak membership*, *strong separation* and *weak distance* oracles. Observe that the running time of Theorem 26 can be rewritten as $\mathcal{O}^*\left(n^{2n}\right)$; Dadush et al. improve it to $\mathcal{O}^*\left(n^{\frac{4}{3}n}\right)$. Moreover, Dadush claims in his PhD thesis [68] a randomized $\mathcal{O}^*(n^n)$ algorithm; for derandomization cf. [42]. (Here and in the following we use the $\mathcal{O}^*$ notation which suppresses polynomial factors.)

This sequence of results can be seen as a part of a race for the best running time. Dadush [68] classifies existing algorithms as *Lenstra-type* and *Kannan-type*, depending on the space decomposition they use (hyperplane and subspace, respectively). The type of algorithm determines the best possible running time — Lenstra-type algorithms depend on a so-called *flatness theorem*, which gives a lower-bound $\mathcal{O}^*(n^n)$. The best known Lenstra-type algorithm is the $\mathcal{O}^*\left(n^{\frac{4}{3}n}\right)$ algorithm of Dadush et al. [47]. Note that both Theorems 25 and 26 are Lenstra-type algorithms. On the other hand, Kannan-type algorithms could run as fast as $\mathcal{O}^*((\log n)^n)$ if a certain conjecture of Kannan and Lovász holds [68, Theorem 7.1.3]. The $\mathcal{O}^*(n^n)$ algorithm given in Dadush's thesis [68] is Kannan-type. It is also worth noting that the only known lower bound for convex IP in general is the trivial one of $\mathcal{O}^*(2^n)$ (by encoding SAT as binary ILP).

The oracle approach is also taken by Oertel, Wagner and Weismantel [48]. They show that a convex IP given by a so-called *first order evaluation oracle* can be reduced to several MILP subproblems, which are readily solved by existing solvers (implementing for example Theorem 24). In an earlier version of this paper [48] the authors take a more generic approach requiring a set of oracles to solve a minimization problem, and discuss how to construct these oracles specifically for the (CPIP) problem.

A.2. Concave integer minimization in small dimension

When we make the step from a *linear* to a general *quasiconvex* objective function, we have to distinguish carefully between convex minimization and maximization, or equivalently, between minimizing a convex and a concave function. Here we mention one result that can be applied in the concave minimization case.

Del Pia [69] has recently shown an approximation scheme for separable quadratic concave minimization when the subdeterminants of the constraint matrix and the dimension are bounded.

Vertex enumeration. Provided bounds on the encoding length and number of inequalities, there is a good bound on the number of vertices of the integer hull of a polyhedron:

Theorem 27 (*Cook et al. [70]*)**.** *Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \le \mathbf{b}\}$ be a rational polyhedron with $A \in \mathbb{Q}^{m\times n}$ and let ϕ be the largest binary encoding size of any of the rows of the system $A\mathbf{x} \le \mathbf{b}$. Let $P^I = conv(P \cap \mathbb{Z}^n)$ be the integer hull of P. Then the number of vertices of P^I is at most $2m^n(6n^2\phi)^{n-1}$.*

Since Hartmann [71] also gave an algorithm for enumerating all the vertices running in polynomial time in small dimension, it is possible to evaluate the concave objective function on each of them and pick the best. The crucial observation which makes this sufficient is that any concave objective is minimized on the boundary, which will be a vertex. Moreover, in parameterized complexity we are often dealing with combinatorial problems whose ILP description only contains numbers encoded in unary, implying that the *encoding length* ϕ is logarithmic in the size of the instance $|I|$. Since $(\log|I|)^k$ for fixed k is order $o(|I|)$ [72, Lemma 6.1], convex integer maximization is FPT in all such cases.

A.3. Indefinite optimization in small dimension

Results regarding optimizing indefinite polynomials in fixed dimension are few, indicating this area merits much attention. De Loera et al. [73] show that optimizing an indefinite non-negative polynomial over the mixed-integer points in small dimensional polytopes admits a fully-polynomial time approximation scheme (FPTAS); however, the runtime of this algorithm is XP from the perspective of parameterized complexity, and it has not yet found applications.

Hildebrand et al. [74] recently also provided an FPTAS, however, their results are incomparable to the previous one. On one hand, their results are stronger because they use a different notion of approximation, and because they do not require the non-negativity of the objective function. On the other hand, there are additional requirements on the polynomial, namely that it is quadratic and has at most one negative or at most one positive eigenvalue.

The most significant contribution from the perspective of parameterized complexity is an FPT algorithm for Quadratic Integer Programming by Lokshtanov [18], independently also discovered by Zemmer [50]:

Theorem 28 (*Lokshtanov [18], Zemmer [50]*)**.** *Let $Q \in \mathbb{Z}^{n\times n}$ and $f(\mathbf{x}) = \mathbf{x}^\intercal Q\mathbf{x}$. Then problem (LinIP) is* FPT *parameterized by n, $\|A\|_\infty$, and $\|Q\|_\infty$.*

While this parameterization may seem very restrictive, it leads to the resolution of a major open problem regarding the parameterized complexity of Minimum Linear Arrangement parameterized by the vertex cover number.

A.4. Integer linear programming in variable dimension

Two major well-known cases of linear programs (LPs) that can be solved integrally in polynomial time are LPs in small dimension (as discussed above) and LPs given by totally unimodular matrices (such as flow polytopes). Totally unimodular matrices are generalized by matrices with minors bounded by k, i.e., when every square submatrix has its determinant bounded by k in absolute value. A major conjecture states that ILP is FPT in k. Recently Paat, Schlöter, and Weismantel showed that the conjecture is true for most IPs, which means for every A and $\mathbf{c}$ and a randomly chosen right hand side $\mathbf{b}$, with probability 1 the IP $\min \mathbf{cx} :\ A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \in \mathbb{Z}^n$ is solvable in FPT time in the largest minor of A. The conjecture is also true for bimodular ILPs with $k = 2$ [75].

A large stream of research of the past 20 years has very recently converged on a result largely explaining the parameterized complexity of IP in terms of the structural complexity of the matrix A. We are interested in the parameterizations of three graphs associated to the constraint matrix A:

1. The *primal graph* $G_P(A)$, which has a vertex for every column, and two vertices share an edge if a row exists where both corresponding entries are non-zero.
2. The *dual graph* $G_D(A) = G_P(A^\intercal)$, which is the primal graph of the transpose of a matrix.
3. The *incidence graph* $G_I(A)$, which has a vertex for every row and every column, and two vertices share an edge if they correspond to a row-column coordinate which is non-zero.

Specifically, we are interested in the treedepth and treewidth of these graphs, yielding six parameters: primal/incidence/dual treedepth/treewidth, denoted $\mathrm{td}_P(A)$, $\mathrm{tw}_P(A)$, $\mathrm{td}_I(A)$, $\mathrm{tw}_I(A)$, $\mathrm{td}_D(A)$ and $\mathrm{tw}_D(A)$. The fundamental result can be phrased as follows:

Theorem 29 (*[60, Theorems 5 and 6]*)**.** *There are computable functions h_P and h_D such that problem (SLinIP) with f a separable convex function can be solved in time:*

- $h_P(\|A\|_\infty, \mathrm{td}_P(A))n^3\langle f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}\rangle$, *and*
- $h_D(\|A\|_\infty, \mathrm{td}_D(A))n^3\langle f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}\rangle$.

In the case of ILP (linear objective), these results can even be made *strongly polynomial*, i.e., not depending on the encoding lengths $\langle \mathbf{w}, \mathbf{l}, \mathbf{u}, \mathbf{b}\rangle$. Let us discuss in more detail how these results are obtained.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

Graver basis optimization. A key notion is that of *iterative augmentation*. Most readers will be familiar that the Max Flow problem can be solved by starting from a zero flow, and iteratively augmenting it with paths; when no augmenting path exists, the flow is optimal. The notion of a *Graver basis* (cf. Definition 1) lets us extend this approach to (SLinIP) as follows. Starting from some initial feasible point $\mathbf{x}_0 \in \mathbb{Z}^n$, there either exists a $\mathbf{g} \in \mathcal{G}(A)$ such that $\mathbf{x}_0 + \mathbf{g}$ is feasible (i.e., $\mathbf{l} \leq \mathbf{x}_0 + \mathbf{g} \leq \mathbf{u}$) and augmenting (i.e., $f(\mathbf{x}_0 + \mathbf{g}) < f(\mathbf{x}_0)$), or $\mathbf{x}_0$ is guaranteed to be optimal. This is not yet enough to ensure quick convergence to an optimal point $\mathbf{x}^*$, but always augmenting with a *Graver-best* step $\mathbf{g}$ then also guarantees this. Thus the question becomes in which cases it is possible to efficiently compute such Graver-best steps. This turns out to depend on the primal and dual treewidth and the norms of elements of $\mathcal{G}(A)$; recall that $g_\infty(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_\infty$ and analogously $g_1(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_1$.

Lemma 30 (*Primal and Dual Lemma [60, (roughly) Lemmas 22 and 25]*)**.** *A Graver-best step can be found in time*

- $g_\infty(A)^{\mathcal{O}(\mathrm{tw}_P(A))} \cdot (n+m)$, *and,*
- $g_1(A)^{\mathcal{O}(\mathrm{tw}_D(A))} \cdot (n+m)$.

The proof of this lemma uses two dynamic programs; the first is well known and goes back to Freuder [76,77], the second was only recently described by Ganian et al. [78].

Graver basis norms. The next obvious question is: what IPs satisfy the assumptions of Lemma 30? Hemmecke and Schultz [61] show (though not in those terms) that *2-stage stochastic* matrices have small $g_\infty(A)$, and it is not hard to see that they have small $\mathrm{td}_P(A) \leq \mathrm{tw}_P(A)$. This result was later extended by Aschenbrenner and Hemmecke [62] to *multi-stage stochastic* matrices, which are in turn generalized (and simultaneously generalize) matrices with small primal treedepth $\mathrm{td}_P(A)$, so we have that $g_\infty(A) \leq h(\|A\|_\infty, \mathrm{td}_P(A))$ for some computable function h.

Similarly, it was shown [79] that *n-fold* matrices have small $g_1(A)$ and they also have small $\mathrm{td}_D(A)$. Those are generalized by *tree-fold* matrices introduced by Chen and Marx [19] who generalize (and are generalized by) matrices with small dual treedepth $\mathrm{td}_D(A)$.

Theorem 29 (and its previous versions) has found use for example in parameterized scheduling [12,19], computational social choice and stringology [21,22], and the design of efficient polynomial time approximation schemes (EPTASes) [20].

Incidence treedepth. We note that the classification result of Theorem 29 cannot be improved in any direction: allowing unary-sized coefficients $\|A\|_\infty$ gives W[1]-hardness, and relaxing treedepth to treewidth leads to NP-hardness [60].

The complexity of parameterizing by $\mathrm{td}_I(A)$ and $\|A\|_\infty$ was open until Eiben et al. [80] showed that already for constant $\mathrm{td}_I(A)$ and $\|A\|_\infty$, ILP is NP-hard. So-called 4-block n-fold programs, which combine the structure of 2-stage stochastic and n-fold matrices, are known to be XP parameterized by the block dimensions [66], and FPT membership is an important open problem. Recently, Chen et al. [81] gave some indication that the problem might in fact be W[1]-hard. Interestingly, the hard instance of Eiben et al. [80] has so-called *topological height* 3 (see Eisenbrand et al. [82]), while all 4-block n-folds have topological height 2, so FPT membership of 4-block n-fold can be rephrased as FPT membership of ILPs parameterized by $\mathrm{td}_I(A)$ and $\|A\|_\infty$ with topological height 2.

ILP with few rows. Restricting our attention to a simpler case then the one handled by Theorem 29 leads us to considering ILPs with few rows. Papadimitriou showed that ILP is FPT parameterized by $\|A\|_\infty$ and m [55]. His algorithm was recently sped up by Eisenbrand and Weismantel [56] and in the special case without upper bounds also by Jansen and Rohwedder [53]. Many approximation algorithms (especially EPTASes) contain a subroutine using Lenstra's algorithm to solve a certain configuration IP. Provided that this IP has small coefficients, this step can be exponentially sped up by applying one of the aforementioned algorithms.

A good example is the algorithm of Lampis for Coloring on graphs of bounded neighborhood diversity [9], which can be improved from $2^{2^{k^{\mathcal{O}(1)}}} \log |G|$ to $k^{\mathcal{O}(k)} \log |G|$ simply by replacing Lenstra's algorithm.

Miscellaneous results. We highlight that we are not aware of any uses of multi-stage stochastic IP in parameterized complexity, and it would be interesting to see what kind of problems it can model. Another interesting result from this area which has not yet found applications is due to Lee et al. [83]. It states that minimizing even certain non-convex objectives is polynomial-time solvable provided the objective falls in the so-called *quadratic Graver cone*.

One way how to view the results based on Graver bases is via the parameter *fracture number*: a graph has a small fracture number if there exists a small subset of vertices whose deletion decomposes the graph into (possibly many) small components; note that the treedepth is always at most the fracture number. Dvořák et al. [84] show that ILP parameterized by the largest coefficient and the constraint or variable fracture number of the primal graph is FPT. In the case of constraint fracture number, one must delete small set of vertices corresponding only to constraints of the ILP at hand. The variable fracture number is defined accordingly. They provide an equivalent instance of either n-fold IP or 2-stage stochastic IP. These results are subsumed by Theorem 29, but the parameter mixed fracture number (allowing the deletion of both rows and columns of A) is interesting because it is equivalent to 4-block n-fold and could be useful to understand its complexity.

Jansen and Kratsch [77] studied the kernelizability of ILP and show that instances with bounded domains and bounded primal treewidth are efficiently kernelizable. Moreover, they introduce so-called *r-boundaried* ILPs which generalize totally unimodular ILPs and ILPs of bounded treewidth, and they give an FPT result regarding r-boundaried ILPs.

Finally, Ganian et al. [78] show that ILP parameterized by incidence treewidth and the largest constraint partial sum of any feasible solution is FPT. They also combine primal treewidth with Lenstra's algorithm to obtain a new structural parameter called *torso-width*, and give an FPT algorithm for this parameterization.

References

[1] T. Gavenčiak, D. Knop, M. Koutecký, Integer programming in parameterized complexity: Three miniatures, in: C. Paul, M. Pilipczuk (Eds.), 13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland, in: LIPIcs, vol. 115, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 21:1–21:16.
[2] A. Frank, E. Tardos, An application of simultaneous diophantine approximation in combinatorial optimization, Combinatorica 7 (1) (1987) 49–65.
[3] R. Kannan, Minkowski's convex body theorem and integer programming, Math. Oper. Res. 12 (3) (1987) 415–440.
[4] H.W. Lenstra Jr., Integer programming with a fixed number of variables, Math. Oper. Res. 8 (4) (1983) 538–548.
[5] R. Niedermeier, Ubiquitous parameterization - Invitation to fixed-parameter algorithms, in: V. Fiala, J. Kratochvíl (Eds.), MFCS, in: Lecture Notes in Computer Science, vol. 3153, Springer, 2004, pp. 84–103.
[6] M.R. Fellows, D. Lokshtanov, N. Misra, F.A. Rosamond, S. Saurabh, Graph layout problems parameterized by vertex cover, in: S.-H. Hong, H. Nagamochi, T. Fukunaga (Eds.), ISAAC, in: Lecture Notes in Computer Science, vol. 5369, Springer, 2008, pp. 294–305.
[7] J. Fiala, T. Gavenčiak, D. Knop, M. Kouteckỳ, J. Kratochvíl, Parameterized complexity of distance labeling and uniform channel assignment problems, Discrete Appl. Math. (2017).
[8] R. Ganian, Using neighborhood diversity to solve hard problems, 2012, arXiv preprint arXiv:1201.3091.
[9] M. Lampis, Algorithmic meta-theorems for restrictions of treewidth, Algorithmica 64 (1) (2012) 19–37.
[10] D. Hermelin, J.-M. Kubitza, D. Shabtay, N. Talmon, G. Woeginger, Scheduling two competing agents when one agent has significantly fewer jobs, in: LIPIcs-Leibniz International Proceedings in Informatics, volume 43, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
[11] K. Jansen, R. Solis-Oba, An OPT + 1 algorithm for the cutting stock problem with constant number of object lengths, in: IPCO, Springer, 2010, pp. 438–449.
[12] D. Knop, M. Koutecký, Scheduling meets n-fold integer programming, J. Sched. (2017).
[13] M. Mnich, A. Wiese, Scheduling and fixed-parameter tractability, Math. Program. 154 (1-2, Ser. B) (2015) 533–562.
[14] R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, N. Talmon, Elections with few candidates: Prices, weights, and covering problems, in: Proc. ADT 2015, in: Lecture Notes Comput. Sci., vol. 9346, 2015, pp. 414–431.
[15] D. Lokshtanov, New Methods in Parameterized Algorithms and Complexity (Ph.D. thesis), University of Bergen, 2009.
[16] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, in: Texts in Computer Science, Springer, 2013, pp. I–SSS, 3–707.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

[17] R. Hemmecke, S. Onn, L. Romanchuk, N-fold integer programming in cubic time, Math. Program 137 (1–2) (2013) 325–341.
[18] D. Lokshtanov, Parameterized integer quadratic programming: Variables and coefficients, 2015, arXiv preprint arXiv:1511.00310.
[19] L. Chen, D. Marx, Covering a tree with rooted subtrees–parameterized and approximation algorithms, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, SIAM, 2018, pp. 2801–2820.
[20] K. Jansen, K.-M. Klein, M. Maack, M. Rau, Empowering the configuration-IP – New PTAS results for scheduling with setups times, 2018, arXiv preprint arXiv:1801.06460.
[21] D. Knop, M. Koutecký, M. Mnich, Combinatorial n-fold integer programming and applications, Math. Program. (2019).
[22] D. Knop, M. Koutecký, M. Mnich, Voting and bribing in single-exponential time, in: H. Vollmer, B. Vallée (Eds.), 34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8–11, 2017, Hannover, Germany, in: LIPIcs, vol. 66, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, pp. 46:1–46:14.
[23] D. Knop, M. Koutecký, M. Mnich, A unifying framework for manipulation problems, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10–15, 2018, 2018, pp. 256–264.
[24] J. Gajarský, P. Hliněný, M. Koutecký, S. Onn, Parameterized shifted combinatorial optimization, in: International Computing and Combinatorics Conference, Springer, 2017, pp. 224–236.
[25] J. Gajarský, M. Lampis, S. Ordyniak, Parameterized algorithms for modular-width, CoRR abs/1308.2858 (2013).
[26] D. Marx, What's next? Future directions in parameterized complexity, in: The Multivariate Algorithmic Revolution and beyond, Springer, 2012, pp. 469–496.
[27] D. Lokshtanov, M. Ramanujan, S. Saurabh, A linear time parameterized algorithm for directed feedback vertex set, 2016, arXiv preprint arXiv:1609.04347.
[28] P.N. Panagopoulou, P.G. Spirakis, A game theoretic approach for efficient graph coloring, in: S.-H. Hong, H. Nagamochi, T. Fukunaga (Eds.), Algorithms and Computation, 19th International Symposium, ISAAC 2008, Gold Coast, Australia, December 15-17, 2008. Proceedings, in: Lecture Notes in Computer Science, vol. 5369, Springer, 2008, pp. 183–195.
[29] S.R. Alves, K.K. Dabrowski, L. Faria, S. Klein, I. Sau, U. dos Santos Souza, On the (parameterized) complexity of recognizing well-covered (r, l)-graphs, in: 10th International Conference on Combinatorial Optimization and Applications, COCOA'16, in: Lecture Notes in Computer Science, vol. 10043, 2016, pp. 423–437.
[30] N. Aravind, S. Kalyanasundaram, A.S. Kare, J. Lauri, Algorithms and hardness results for happy coloring problems, 2017, arXiv preprint arXiv:1705.08282.
[31] É. Bonnet, F. Sikora, The graph motif problem parameterized by the structure of the input graph, Discrete Appl. Math. (2016).
[32] L. Gargano, A.A. Rescigno, Complexity of conflict-free colorings of graphs, Theoret. Comput. Sci. 566 (??) (2015) 39–49.
[33] T. Masařík, T. Toufar, Parameterized complexity of fair deletion problems, in: International Conference on Theory and Applications of Models of Computation, TAMC'16, 2017, pp. 628–642.
[34] M. Dom, D. Lokshtanov, S. Saurabh, Y. Villanger, Capacitated domination and covering: A parameterized perspective, in: International Workshop on Parameterized and Exact Computation, Springer, 2008, pp. 78–90.
[35] K. Jansen, Complexity Results for the Optimum Cost Chromatic Partition Problem, Citeseer, 1997.
[36] F.V. Fomin, P.A. Golovach, D. Lokshtanov, S. Saurabh, Almost optimal lower bounds for problems parameterized by clique-width, SIAM J. Comput. 43 (5) (2014) 1541–1563.
[37] L. Cai, Parameterized complexity of vertex colouring, Discrete Appl. Math. 127 (3) (2003) 415–429.
[38] P.A. Golovach, D. Lokshtanov, S. Saurabh, M. Zehavi, Cliquewidth III: the odd case of graph coloring parameterized by cliquewidth, in: SODA, SIAM, 2018, pp. 262–273.
[39] J. Bisschop, AIMMS Optimization Modeling, Paragon Decision Technology BV, 2006.
[40] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, G.J. Woeginger, Parameterized algorithmics for computational social choice: Nine research challenges, Tsinghua Sci. Technol. 19 (4) (2014) 358–373.
[41] P. Faliszewski, R. Gonen, M. Koutecký, N. Talmon, Opinion diffusion and campaigning on society graphs, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, 2018, pp. 219–225, https://doi.org/10.24963/ijcai.2018/30.
[42] D. Dadush, S.S. Vempala, Near-optimal deterministic algorithms for volume computation via M-ellipsoids, Proc. Natl. Acad. Sci. 110 (48) (2013) 19237–19245.
[43] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Vol. 2, Springer Science & Business Media, 2012.
[44] L. Khachiyan, L. Porkolab, Integer optimization on convex semialgebraic sets, Discrete Comput. Geom. 23 (2) (2000) 207–224.
[45] S. Heinz, Complexity of integer quasiconvex polynomial optimization, J. Complexity 21 (4) (2005) 543–556.
[46] R. Hildebrand, M. Köppe, A new Lenstra-type algorithm for quasiconvex polynomial integer minimization with complexity $2^{O(n \log n)}$, Discrete Optim. 10 (1) (2013) 69–84.
[47] D. Dadush, C. Peikert, S. Vempala, Enumerative lattice algorithms in any norm via M-ellipsoid coverings, in: R. Ostrovsky (Ed.), FOCS, IEEE, 2011, pp. 580–589.
[48] T. Oertel, C. Wagner, R. Weismantel, Integer convex minimization by mixed integer linear optimization, Oper. Res. Lett 42 (6–7) (2014) 424–428.
[49] D. Knop, T. Masařík, T. Toufar, Parameterized complexity of fair vertex evaluation problems, in: P. Rossmanith, P. Heggernes, J. Katoen (Eds.), 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany, in: LIPIcs, vol. 138, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 33:1–33:16.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

[50] K. Zemmer, Integer Polynomial Optimization in Fixed Dimension (Ph.D. thesis), ETH Zurich, 2017.
[51] F. Eisenbrand, G. Shmonin, Parametric integer programming in fixed dimension, Math. Oper. Res. 33 (4) (2008).
[52] J. Crampton, G. Gutin, M. Koutecký, R. Watrigant, Parameterized resiliency problems via integer linear programming, in: Proc. CIAC 2017, in: Lecture Notes Comput. Sci., vol. 10236, 2017, pp. 164–176.
[53] K. Jansen, L. Rohwedder, On integer programming and convolution, 2018, arXiv preprint arXiv:1803.04744.
[54] D. Knop, M. Pilipczuk, M. Wrochna, Tight complexity lower bounds for integer linear programming with few constraints, ACM Trans. Comput. Theory 12 (3) (2020).
[55] C.H. Papadimitriou, On the complexity of integer programming, J. ACM 28 (4) (1981) 765–768.
[56] F. Eisenbrand, R. Weismantel, Proximity results and faster algorithms for integer programming using the steinitz lemma, in: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2018, pp. 808–816.
[57] K. Altmanová, D. Knop, M. Kouteckỳ, Evaluating and tuning n-fold integer programming, in: G. D'Angelo (Ed.), 17th International Symposium on Experimental Algorithms, SEA 2018, June 27–29, 2018, L'Aquila, Italy, in: LIPIcs, vol. 103, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 10:1–10:14.
[58] F. Eisenbrand, C. Hunkenschröder, K.-M. Klein, Faster algorithms for integer programs with block structure, in: I. Chatzigiannakis, C. Kaklamanis, D. Marx, D. Sannella (Eds.), 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9–13, 2018, Prague, Czech Republic, in: LIPIcs, vol. 107, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 49:1–49:13.
[59] K. Jansen, A. Lassota, L. Rohwedder, Near-linear time algorithm for n-fold ILPs via color coding, 2018, arXiv:1811.00950.
[60] M. Koutecký, A. Levin, S. Onn, A parameterized strongly polynomial algorithm for block structured integer programs, in: I. Chatzigiannakis, C. Kaklamanis, D. Marx, D. Sannella (Eds.), 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic, in: LIPIcs, vol. 107, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 85:1–85:14.
[61] R. Hemmecke, R. Schultz, Decomposition methods for two-stage stochastic integer programs, in: Online Optimization of Large Scale Systems, Springer, 2001, pp. 601–622.
[62] M. Aschenbrenner, R. Hemmecke, Finiteness theorems in stochastic integer programming, Found. Comput. Math. 7 (2) (2007) 183–227.
[63] K. Klein, About the complexity of two-stage stochastic IPs, in: D. Bienstock, G. Zambelli (Eds.), Integer Programming and Combinatorial Optimization - 21st International Conference, IPCO 2020, London, UK, June 8–10, 2020, Proceedings, in: Lecture Notes in Computer Science, vol. 12125, Springer, 2020, pp. 252–265.
[64] D. Knop, M. Koutecký, Scheduling kernels via configuration LP, CoRR abs/2003.02187 (2020).
[65] J.A. De Loera, R. Hemmecke, M. Köppe, Algebraic and Geometric Ideas in the Theory of Discrete Optimization, in: MOS-SIAM Series on Optimization, vol. 14, SIAM, 2013.
[66] R. Hemmecke, M. Köppe, J. Lee, R. Weismantel, Nonlinear integer programming, in: M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, L.A. Wolsey (Eds.), 50 Years of Integer Programming, Springer, 2010, pp. 561–618.
[67] G. Blekherman, P.A. Parrilo, R.R. Thomas, Semidefinite Optimization and Convex Algebraic Geometry, SIAM, 2012.
[68] D. Dadush, Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation (Ph.D. thesis), Georgia Institute of Technology, 2012.
[69] A.D. Pia, Subdeterminants and concave integer quadratic programming, SIAM J. Optim. 29 (4) (2019) 3154–3173.
[70] W.J. Cook, M. Hartmann, R. Kannan, C. McDiarmid, On integer points in polyhedra, Combinatorica 12 (1) (1992) 27–37.
[71] M.E. Hartmann, Cutting Planes and the Complexity of the Integer Hull (Ph.D. thesis), Cornell University, Ithaca, NY, USA, 1989, AAI8915096.
[72] R.H. Chitnis, M. Cygan, M.T. Hajiaghayi, D. Marx, Directed subset feedback vertex set is fixed-parameter tractable, ACM Trans. Algorithms 11 (4) (2015) 28:1–28:28.
[73] J.A. De Loera, R. Hemmecke, M. Köppe, R. Weismantel, FPTAS for optimizing polynomials over the mixed-integer points of polytopes in fixed dimension, Math. Program. 115 (2) (2008) 273–290.
[74] R. Hildebrand, R. Weismantel, K. Zemmer, An FPTAS for minimizing indefinite quadratic forms over integers in polyhedra, in: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2016, pp. 1715–1723.
[75] S. Artmann, R. Weismantel, R. Zenklusen, A strongly polynomial algorithm for bimodular integer linear programming, in: H. Hatami, P. McKenzie, V. King (Eds.), Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017, ACM, 2017, pp. 1206–1219.
[76] E.C. Freuder, Complexity of K-tree structured constraint satisfaction problems, in: Proc. of the 8th National Conference on Artificial Intelligence, 1990, pp. 4–9.
[77] B.M.P. Jansen, S. Kratsch, A structural approach to kernels for ILPs: Treewidth and total unimodularity, CoRR abs/1506.07729 (2015) Comment: Extended abstract in the Proceedings of the 23rd European Symposium on Algorithms (ESA 2015).
[78] R. Ganian, S. Ordyniak, M. Ramanujan, Going beyond primal treewidth for, (M) ILP, in: AAAI, 2017, pp. 815–821.
[79] S. Onn, Nonlinear discrete optimization, in: Zurich Lectures in Advanced Mathematics, European Mathematical Society, 2010.
[80] E. Eiben, R. Ganian, D. Knop, S. Ordyniak, M. Pilipczuk, M. Wrochna, Integer programming and incidence treedepth, in: Integer Programming and Combinatorial Optimization - 20th International Conference, IPCO 2019, Ann Arbor, MI, USA, May 22-24, 2019, Proceedings, 2019, pp. 194–204.

Please cite this article as: T. Gavenčiak, M. Koutecký and D. Knop, Integer programming in parameterized complexity: Five miniatures, Discrete Optimization (2020) 100596, https://doi.org/10.1016/j.disopt.2020.100596.

[81] L. Chen, M. Koutecký, L. Xu, W. Shi, New bounds on augmenting steps of block-structured integer programs, 2020, to appear in Proc. ESA 2020, arXiv preprint arXiv:1805.03741.

[82] F. Eisenbrand, C. Hunkenschröder, K. Klein, M. Koutecký, A. Levin, S. Onn, An algorithmic theory of integer programming, CoRR abs/1904.01361 (2019).

[83] J. Lee, S. Onn, L. Romanchuk, R. Weismantel, The quadratic graver cone, quadratic integer minimization, and extensions, Math. Program. 136 (2) (2012) 301–323.

[84] P. Dvořák, E. Eiben, R. Ganian, D. Knop, S. Ordyniak, Solving integer linear programs with a small number of global variables and constraints, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017, 2017, pp. 607–613.