Integer Programming in Parameterized Complexity: Three Miniatures

Tomáš Gavenčiak
Department of Applied Mathematics, Charles University, Prague, Czech Republic
gavento@kam.mff.cuni.cz

Dušan Knop[1]
Department of Informatics, University of Bergen, Bergen, Norway *and*
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Prague, Czech Republic
dusan.knop@uib.no

Martin Koutecký[2]
Technion - Israel Institute of Technology, Haifa, Israel *and*
Computer Science Institute, Charles University, Prague, Czech Republic
koutecky@technion.ac.il

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, and Max-q-Cut by modeling them as convex programs in fixed dimension, n-fold integer programs, bounded dual treewidth programs, and indefinite quadratic programs in fixed dimension.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms, Theory of computation → Graph algorithms analysis

Keywords and phrases graph coloring, parameterized complexity, integer linear programming, integer convex programming

Digital Object Identifier 10.4230/LIPIcs.IPEC.2018.21

[1] Author supported by the project NFR MULTIVAL.
[2] Author supported by Technion postdoctoral fellowship.

13th International Symposium on Parameterized and Exact Computation (IPEC 2018).
Editors: Christophe Paul and Michał Pilipczuk; Article No. 21; pp. 21:1–21:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version A full version of the paper is available at `https://arxiv.org/abs/1711.02032`.

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

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

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):

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

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 [22, 36, 45]). Two decades later this algorithm's potential for applications in parameterized complexity was recognized, e.g. by Niedermeier [52]:

> *[...] 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 [19, 20, 24, 44], scheduling [30, 35, 38, 51] or computational social choice [8].

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 [46], 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 [14] 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 [28] concerning nonlinear optimization.*

Interestingly, Minimum Linear Arrangement was shown to be FPT by yet another new algorithm for IP due to Lokshtanov [47].

In the last 3 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 [9, 30, 33, 38, 51], computational social choice [39, 40, 42], multichoice optimization [23], and stringology [39]. 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 resolve the parameterized complexity of three problems when parameterized by the neighborhood diversity of a graph (we defer the definitions to the relevant sections). 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 [49]), 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 $\mathrm{poly}(|G|)$ factor is in the spirit of "minding the $\mathrm{poly}(n)$" of Lokshtanov et al. [48].

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.** Capacitated Dominating Set

(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 $k^{\mathcal{O}(k^2)} \mathrm{poly}(|G|)$ and space $\mathrm{poly}(k, |G|)$.*

(c) *Can be solved in time $k^{\mathcal{O}(k)} \mathrm{poly}(|G|)$ using model a and a proximity argument.*

(d) *Has a polynomial $OPT + k^2$ approximation algorithm by rounding a relaxation of a.*

▶ **Theorem 2.** Sum Coloring

(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 $k^{\mathcal{O}(k^2)} \log |G|$.*

▶ **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.*

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 [2, 3, 6, 20, 24, 25, 50]. 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. [13]. Sum Coloring was shown to be FPT parameterized by treewidth [32]; 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 [21].

1.3 Preliminaries

For positive integers m, n with $m \le n$ we set $[m, n] = \{m, \ldots, n\}$ and $[n] = [1, n]$. We write vectors in boldface (e.g., $\mathbf{x}, \mathbf{y}$) and their entries in normal font (e.g., the i-th entry of $\mathbf{x}$ is x_i). 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 $x_i y_i \ge 0$ (that is, $\mathbf{x}$ and $\mathbf{y}$ lie in the same orthant) and $|x_i| \le |y_i|$ for all $i \in [n]$. It is well known that every subset of $\mathbb{Z}^n$ has finitely many $\sqsubseteq$-minimal elements.

▶ **Definition 4** (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.

▶ **Definition 5** (Lampis [44])**.** The *neighborhood diversity* of a graph G, denoted by $\mathrm{nd}(G)$, is the minimum 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 $\mathcal{O}\big(\log|G| \cdot k^2\big)$ space by its type graph, which is computable in linear time [44]:

▶ **Definition 6.** 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 $f_{\max} = \max_{\mathbf{x}: \mathbf{l} \le \mathbf{x} \le \mathbf{u}} |f(\mathbf{x})|$ (in general), and noting other relevant features.

Solution structure. We concur with Downey and Fellows that FPT and structure are essentially one [14]. 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 $\top$), typical use cases and strengths ($+$), limitations ($-$), and a list of references to the algorithms (♡) and their most illustrative applications (▷), both in chronological order.

2.1 Small Dimension

The following tools generally rely on results from discrete geometry.

ILP in small dimension. Problem (ILP) with small n.

$\top$ $n^{2.5n}\langle A, \mathbf{b}, \mathbf{w}\rangle$ [36, 22]

$+$ Can use large coefficients, which allows encoding logical connectives using Big-M coefficients [5]. Runs in polynomial space. Most people familiar with ILP.

$-$ Small dimension can be an obstacle in modeling polynomially many "types" of objects [7, 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 9). Big-M coefficients are impractical.

♡ [45, 36, 22]

▷ [52, 19, 35, 20, 18]

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.

$\top$ $n^{\frac{4}{3}n}\langle B\rangle$, where S is contained in a ball of radius B [12].

$+$ Strictly stronger than ILP. Representing constraints implicitely by an oracle allows better dependence on instance size (cf. Model 8).

$-$ Exponential space. Algorithms usually impractical. Proving convexity can be difficult.

♡ [26, Theorem 6.7.10] (weak separation oracle), [37] (semialgebraic sets), [27, 31] (polynomials), [11] randomized / [12] deterministic (strong separation oracle), [53] reduction to Mixed ILP subproblems (first-order oracle).

▷ [30, 8, 51, 38, 41], Model 8

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

$\top$ $g(n, \|A\|_\infty, \|Q\|_\infty)\langle \mathbf{b}\rangle$ [55]

$+$ Currently the only tractable indefinite objective.

$-$ Limiting parameterization.

♡ [47, 55]

▷ [47], Model 10

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

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

$\top$ $g(n,m)\,\mathrm{poly}(\|A, B, \mathbf{d}\|_\infty)$ [16]
$+$ Models one quantifier alternation. Useful in expressing game-like constraints (e.g., "$\forall$ moves $\exists$ a counter-move"). Allows unary big-M coefficients to model logic [42, Theorem 4.5].
$-$ Input has to be given in unary (vs. e.g. Lenstra's algorithm).
$\heartsuit$ [16, Theorem 4.2], [10, Corollary 1]
$\triangleright$ [10, 42]

2.2 Variable Dimension

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

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

where $\mathbf{b} \in \mathbb{Z}^m$ and $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$. Let $L = \langle f_{\max}, \mathbf{b}, \mathbf{l}, \mathbf{u}\rangle$. In contrast with the previous section, the following algorithms typically rely on algebraic arguments and dynamic programming.

ILP with few rows. Problem (SLinIP) with small m and a linear objective $\mathbf{wx}$ for $\mathbf{w} \in \mathbb{Z}^n$.
$\top$ $\mathcal{O}((m\|A\|_\infty)^{2m})\langle \mathbf{b}\rangle$ if $\mathbf{l} \equiv \mathbf{0}$ and $\mathbf{u} \equiv +\infty$, and $n \cdot (m\|A\|_\infty)^{\mathcal{O}(m^2)}\langle \mathbf{b}, \mathbf{l}, \mathbf{u}\rangle$ in general [34]
$+$ 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.
$\heartsuit$ [54, 17, 34]
$\triangleright$ [34]

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

$\mathbf{n}$-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 $\mathrm{td}_D(A)$.

$\top$ $(\|A\|_\infty rs)^{\mathcal{O}(r^2 s + rs^2)}(nt)^2 \log(nt)\langle L\rangle$ n-fold IP [1, 15]; $(\|A\|_\infty + 1)^{2^{\mathrm{td}_D(A)}}(nt)^2 \log(nt)\langle L\rangle$ for (SLinIP) [43].
$+$ Variable dimension useful in modeling many "types" of objects [40, 42]. 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.
$\heartsuit$ [28, 39, 9, 15, 1, 43]
$\triangleright$ [38, 40, 39, 9, 33], Model 11

2-stage and multi-stage stochastic IP, and primal treedepth. 2-stage stochastic IP is problem (SLinIP) with $A = A_{\text{stoch}}$ 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 $\text{td}_P(A)$.

⊤ $g(\text{td}_P(A), \|A\|_\infty)n^2 \log n\langle L\rangle$, g computable [43]

+ 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 explicitely 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.

♡ [29, 4, 43]

▷ N/A

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 $\mathcal{G}(A)$.

⊤ $\min\{g_\infty(A)^{\mathcal{O}(\text{tw}_P(A))}, g_1(A)^{\mathcal{O}(\text{tw}_D(A))}\}n^2 \log n\langle L\rangle$ [43]

+ Captures IPs beyond the classes defined above (cf. Section 5.3).

− Bounding $g_1(A)$ and $g_\infty(A)$ is often hard or impossible.

♡ [43]

▷ Model 14

3 Convex Constraints: Capacitated Dominating Set

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

Solution Structure. Let $<_c$ be a linear extension of ordering of V by vertex capacities, i.e., $u <_c v$ if $c(u) \le c(v)$. 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 $f_i(\ell) = \sum_{v\in V_i[2:\ell]} c(v)$; 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 7** (⋆). *There is a capacity-ordered optimal solution.*

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 8** (Capacitated Dominating Set as convex IP in fixed dimension).

Variables & notation:

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

Objective & Constraints:

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

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

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

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

Parameters & Notes:

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

Then, applying for example Dadush's algorithm [11] to Model 8 yields Theorem 1a. We can trade the non-linearity of the previous model for an increase in the number of constraints and the largest coefficient. That, combined with Lenstra's algorithm, yields Theorem 1b, where we get a larger dependence on $|G|$, but require only poly$(k, |G|)$ space.

▶ **Model 9** (Capacitated Dominating Set as ILP in fixed dimension).
Exactly as Model 8 but replace constraints (cds:cap) with the following equivalent set of $|G|$ linear constraints:

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

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

[Additive approximation] Proof of Theorem 1d. Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{k+k^2}$ be an optimal solution to the *continuous relaxation* of Model 8, i.e., we relax the requirement that $(\mathbf{x}, \mathbf{y})$ are integral; note that such $(\mathbf{x}, \mathbf{y})$ can be computed in polynomial time using the ellipsoid method [26], or by applying a polynomial LP algorithm to Model 9. 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. Ideally, we would take $\hat{\mathbf{y}} = \lceil \mathbf{y} \rceil$ and compute $\hat{\mathbf{x}}$ accordingly, i.e., set $\hat{x}_i$ to be smallest possible such that $\sum_{j \in N_{T(G)}(i)} \hat{y}_{ij} \geq f_i(\hat{x}_i)$; note that $\hat{x}_i \leq x_i + k$, since we add at most k neighbors (to be dominated) in neighborhood of V_i. However, this might result in a non-feasible solution if, for some i, $\hat{x}_i > |V_i|$. In such a case, we solve the relaxation again with an additional constraint $x_i = |V_i|$ and try rounding again, repeating this aforementioned fixing procedure if rounding fails, and so on. After at most k repetitions this rounding results in a feasible integer solution $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, in which case we have $\|\hat{\mathbf{x}} - \mathbf{x}\|_1 \leq k^2$ and thus the solution represented by $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ has value at most $OPT + k^2$; the relaxation must eventually become feasible as setting $x_i = |V_i|$ for all $i \in T(G)$ yields a feasible solution. ◀

[Speed trade-offs] Proof of Theorem 1c. Notice that on our way to proving Theorem 1d we have shown that Model 8 has *integrality gap* at most k^2, i.e., the value of the continuous optimum is at most k^2 less than the value of the integer optimum. This implies that an integer optimum $(\mathbf{x}^*, \mathbf{y}^*)$ satisfies, for each $i \in [k]$, $\max\{0, \lfloor x_i - k^2 \rfloor\} \leq x_i^* \leq \min\{|V_i|, x_i + \lceil k^2 \rceil\}$.

We can exploit this to improve Theorem 1a in terms of the parameter dependence at the cost of the dependence on $|G|$. Let us assume that we have a way to test, for a given integer vector $\hat{\mathbf{x}}$, 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 $(2k^2 + 2)^k$ choices of $\hat{\mathbf{x}}$ and choose the best. So we are left with the task of, given a vector $\hat{\mathbf{x}}$, deciding if it models a capacity-ordered solution.

But this is easy. Let $<_c$ be the assumed order and define D as above. Now, we construct an auxiliary bipartite matching problem, where we put $c(v)$ copies of each vertex from D on one side of the graph, and all vertices of $V \setminus D$ on the other side, and connect a copy of $v \in D$ to $u \in V \setminus D$ if $uv \in E(G)$. Then, D is a capacitated dominating set if and only if all vertices in $V \setminus D$ can be matched. The algorithm is then simply to compute the continuous optimum $\mathbf{x}$, and go over all integer vectors $\hat{\mathbf{x}}$ with $\|\mathbf{x} - \hat{\mathbf{x}}\|_1 \leq k^2$, verifying whether they model a solution and choosing the smallest (best) one. ◀

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., $|\{uv \in E(G) \mid u \in W_\alpha, v \in W_\beta, \alpha \neq \beta\}|$.

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 10** (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:

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

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

Parameters & Notes:

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

Applying Lokshtanov's [47] or Zemmer's [55] algorithm to Model 10 yields Theorem 3. 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 $c \colon V \to \mathbb{N}$ 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

Structure of Solution. The following observation was made by Lampis [44] for the COLORING problem, and it holds also for the SUM COLORING problem: every color $C \subseteq V(G)$ intersects each clique type in at most one vertex, and 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. We call a coloring with this structure an *essential coloring*.

▶ **Model 11** (SUM COLORING as n-fold IP).

Variables & Notation:

- $x_i^\alpha = 1$ if 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
$k\lvert G\rvert$	$k + k^2\lvert G\rvert$	1	$\lvert G\rvert$	1	$\lvert G\rvert$	k	k^2	k
- 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.

Applying the algorithm of Altmanová et al. [1] to Model 11 yields Theorem 2a. Model 11 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.

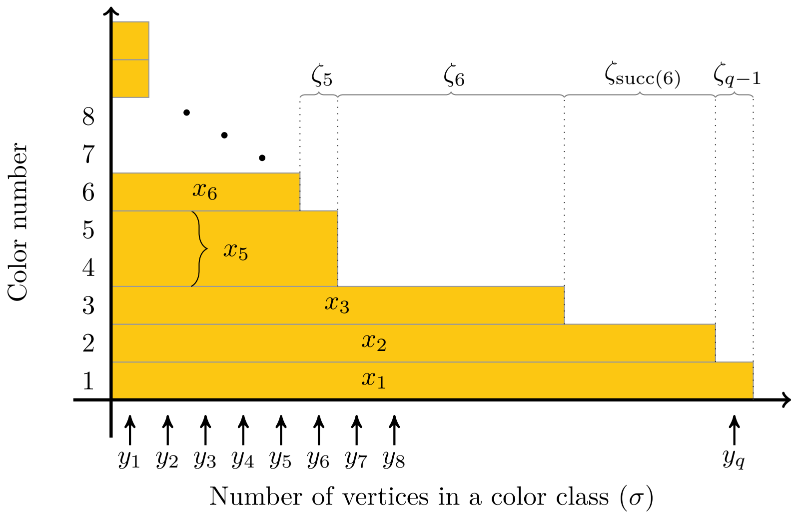

Figure 1 An illustration of the cost decomposition to the individual classes. Note that i-th row (color i) has cost i per vertex.

5.2 Sum Coloring via Convex Minimization in Fixed Dimension

Structure of Solution. 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 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 $I \in \mathcal{I}$, the number of vertices every color class of this type will contain is independent of the actual multiplicity x_I. Define the *size of a color class* $\sigma\colon \mathcal{I} \to \mathbb{N}$ as $\sigma(I) = \sum_{\text{clique } i \in I} 1 + \sum_{\text{indep. } i \in I} |V_i|$.

▶ **Lemma 12** ($\star$). *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 $p \leq q$.*

Our goal now is to show that the objective function can be expressed as a convex function in terms of the variables $\mathbf{x}$. We will get help from auxiliary variables $y_1, \ldots, y_{|G|}$ 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: $y_j = \sum_{\sigma(I) \geq j} x_I$. Then, the objective function can be expressed as $S_{\text{convex}}(\mathbf{x}) = \sum_{i=1}^{p} |i\sigma(I_i)| = \sum_{j=1}^{|G|} \binom{y_j}{2}$, where $i = 1, \ldots, p$ is the order of the color classes given by Lemma 12, every class of type I is present x_I times, where we enumerate only those I with $x_I \geq 1$. The equivalence of the two is straightforward to check.

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

1) all x_I are linear (thus affine) functions,
2) $y_i = \sum_{I : \sigma(I) \geq i} x_I$ is a sum of affine functions, thus affine,
3) $y_i(y_i - 1)/2$ 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.
4) S_{convex} is the sum of $y_i(y_i - 1)/2$, which are convex by the previous point.

▶ **Model 13** (SUM COLORING as LINIP in fixed dimension with convex objective).
Variables & Notation:
- $x_I = $ #of color class I
- $y_i = $ #of color classes I with $\sigma(I) \leq i$
- $\binom{y_i}{2}$ cost of column y_i (Figure 1)
- $S_{\text{convex}} = \sum_{i=1}^{|G|} \binom{y_i}{2} = $ cost of all columns

Objective & Constraints:

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

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

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

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).

Applying the algorithm of Dadush [11] to Model 13 yields Theorem 2b. 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.

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" $\Gamma = \{i \in [|G|] \mid \exists I \in \mathcal{I} : \sigma(I) = i\}$. For each $i \in \Gamma$ let $\text{succ}(i) = \min\{j \in \Gamma \mid j > i\}$ (and let $\text{succ}(\max \Gamma) = \max \Gamma$), define $z_i = \sum_{I \in \mathcal{I} : \sigma(I) \geq i} x_I$, 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. 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 the z_1 variable, thus increasing the dual treewidth to $k + 2^k$. To avoid this, let us define z_i equivalently as $z_i = z_{\text{succ}(i)} + \sum_{\substack{I \in \mathcal{I}: \\ \text{succ}(i) > \sigma(I) \geq i}} x_I = z_{\text{succ}(i)} + \sum_{\substack{I \in \mathcal{I}: \\ \sigma(I) = i}} x_I$.

▶ **Model 14** (SUM COLORING as LINIP with small $\text{tw}_D(A)$ and small $g_1(A)$).
Variables & Notation:
- $x_I = $ #of color class I
- $z_i = $ #of color classes I with $\sigma(I) \geq i$
- $\zeta_i = $ size difference between $I \in \mathcal{I}$ with $\sigma(I) = i$ and closest larger $J \in \mathcal{I}$
- $\zeta_i \binom{z_i}{2}$ cost of all columns between y_i and $y_{\text{succ}(i)}$ (Figure 1)

- Γ = 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:

$$\min S_{\text{sepconvex}}(\mathbf{z}) \qquad \text{(sc:graver:obj)}$$
$$z_i = z_{\text{succ}(i)} + \sum_{I \in \mathcal{I}:\sigma(I)=i} x_I \qquad \forall i \in \Gamma \qquad \text{(sc:graver:sep)}$$

Parameters & Notes:

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

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

Applying the algorithm of Koutecký et al. [43] to Model 14 yields Theorem 2c.

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 $A = \left(\begin{smallmatrix} F \\ L \end{smallmatrix}\right)$.

▶ **Lemma 15** (⋆). *In Model 14 it holds that* $\text{tw}_D(A) \leq k+1$.

Proof Idea. $G_D(F)$ is a k-clique K_k, and $G_D(L)$ is a 2^k-path P_{2^k}. Thus, $G_D(A)$ are these two graphs connected by all possible edges, and we construct a path decomposition, whose consecutive nodes contain $G_D(F)$ and consecutive vertices of $G_D(L)$. ◀

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

Proof Idea. We first simplify the structure of L by deleting duplicitous columns, and then explicitely construct a decomposition of any $\mathbf{h}$ s.t. $L\mathbf{h} = \mathbf{0}$ into conformal vectors $\mathbf{g}$ of small ℓ_1-norm. Combining with known bounds on matrices with few rows (F) and stacked matrices (A) yields the bound. ◀

References

1 Kateřina Altmanová, Dušan Knop, and Martin Kouteckỳ. Evaluating and Tuning n-fold Integer Programming. In Gianlorenzo D'Angelo, editor, *17th International Symposium on Experimental Algorithms, SEA 2018, June 27-29, 2018, L'Aquila, Italy*, volume 103 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.SEA.2018.10.

2 Sancrey Rodrigues Alves, Konrad Kazimierz Dabrowski, Luérbio Faria, Sulamita Klein, Ignasi Sau, and Uéverton 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)*, volume 10043 of *Lecture Notes in Computer Science*, pages 423–437, 2016. doi:10.1007/978-3-319-48749-6.

3 NR Aravind, Subrahmanyam Kalyanasundaram, Anjeneya Swami Kare, and Juho Lauri. Algorithms and hardness results for happy coloring problems. *arXiv preprint arXiv:1705.08282*, 2017.

4 Matthias Aschenbrenner and Raymond Hemmecke. Finiteness theorems in stochastic integer programming. *Foundations of Computational Mathematics*, 7(2):183–227, 2007.

5 Johannes Bisschop. *AIMMS Optimization Modeling*. Paragon Decision Technology BV, 2006.

6 Édouard Bonnet and Florian Sikora. The Graph Motif problem parameterized by the structure of the input graph. *Discrete Applied Mathematics*, 2016.

7 Robert Bredereck, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, and Gerhard J Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014.

8 Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. Elections with Few Candidates: Prices, Weights, and Covering Problems. In *Proc. ADT 2015*, volume 9346 of *Lecture Notes Comput. Sci.*, pages 414–431, 2015.

9 Lin Chen and Daniel 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*, pages 2801–2820. SIAM, 2018.

10 Jason Crampton, Gregory Gutin, Martin Koutecký, and Rémi Watrigant. Parameterized resiliency problems via integer linear programming. In *Proc. CIAC 2017*, volume 10236 of *Lecture Notes Comput. Sci.*, pages 164–176, 2017.

11 Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative Lattice Algorithms in any Norm Via M-ellipsoid Coverings. In Rafail Ostrovsky, editor, *FOCS*, page 580–589. IEEE, 2011. doi:10.1109/FOCS.2011.31.

12 Daniel Dadush and Santosh S Vempala. Near-optimal deterministic algorithms for volume computation via M-ellipsoids. *Proceedings of the National Academy of Sciences*, 110(48):19237–19245, 2013.

13 Michael Dom, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Capacitated domination and covering: A parameterized perspective. In *International Workshop on Parameterized and Exact Computation*, pages 78–90. Springer, 2008.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

15 Friedrich Eisenbrand, Christoph Hunkenschröder, and Kim-Manuel Klein. Faster Algorithms for Integer Programs with Block Structure. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 49:1–49:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.49.

16 Friedrich Eisenbrand and Gennady Shmonin. Parametric integer programming in fixed dimension. *Math. Oper. Res.*, 33(4), 2008.

17 Friedrich Eisenbrand and Robert 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*, pages 808–816. SIAM, 2018.

18 Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 219–225. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi:10.24963/ijcai.2018/30.

19 Michael R. Fellows, Daniel Lokshtanov, Neeldhara Misra, Frances A. Rosamond, and Saket Saurabh. Graph Layout Problems Parameterized by Vertex Cover. In Seok-Hee Hong, Hiroshi Nagamochi, and Takuro Fukunaga, editors, *ISAAC*, volume 5369 of *Lecture Notes in Computer Science*, page 294–305. Springer, 2008. doi:10.1007/978-3-540-92182-0.

20 Jiří Fiala, Tomáš Gavenčiak, Dušan Knop, Martin Kouteckỳ, and Jan Kratochvíl. Parameterized complexity of distance labeling and uniform channel assignment problems. *Discrete Applied Mathematics*, 2017.

21 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost Optimal Lower Bounds for Problems Parameterized by Clique-Width. *SIAM J. Comput.*, 43(5):1541–1563, 2014. doi:10.1137/130910932.

22 András Frank and Eva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. doi:10.1007/BF02579200.

23 Jakub Gajarský, Petr Hliněný, Martin Koutecký, and Shmuel Onn. Parameterized shifted combinatorial optimization. In *International Computing and Combinatorics Conference*, pages 224–236. Springer, 2017.

24 Robert Ganian. Using neighborhood diversity to solve hard problems. *arXiv preprint arXiv:1201.3091*, 2012.

25 Luisa Gargano and Adele A. Rescigno. Complexity of conflict-free colorings of graphs. *Theoretical Computer Science*, 566(??):39–49, February 2015. URL: http://www.sciencedirect.com/science/article/pii/S0304397514009463.

26 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

27 Sebastian Heinz. Complexity of integer quasiconvex polynomial optimization. *J. Complexity*, 21(4):543–556, 2005. doi:10.1016/j.jco.2005.04.004.

28 Raymond Hemmecke, Shmuel Onn, and Lyubov Romanchuk. n-Fold integer programming in cubic time. *Math. Program*, 137(1-2):325–341, 2013. doi:10.1007/s10107-011-0490-y.

29 Raymond Hemmecke and Rüdiger Schultz. Decomposition methods for two-stage stochastic integer programs. In *Online optimization of large scale systems*, pages 601–622. Springer, 2001.

30 Danny Hermelin, Judith-Madeleine Kubitza, Dvir Shabtay, Nimrod Talmon, and Gerhard 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.

31 Robert Hildebrand and Matthias Köppe. A new Lenstra-type algorithm for quasiconvex polynomial integer minimization with complexity $2^{O(n \log n)}$. *Discrete Optimization*, 10(1):69–84, 2013. doi:10.1016/j.disopt.2012.11.003.

32 Klaus Jansen. Complexity results for the optimum cost chromatic partition problem. Manuscript, 1997.

33 Klaus Jansen, Kim-Manuel Klein, Marten Maack, and Malin Rau. Empowering the Configuration-IP – New PTAS Results for Scheduling with Setups Times. *arXiv preprint arXiv:1801.06460*, 2018.

34 Klaus Jansen and Lars Rohwedder. On Integer Programming and Convolution. *arXiv preprint arXiv:1803.04744*, 2018.

35 Klaus Jansen and Roberto Solis-Oba. An OPT+ 1 Algorithm for the Cutting Stock Problem with Constant Number of Object Lengths. In *IPCO*, pages 438–449. Springer, 2010.

36 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

37 Leonid Khachiyan and Lorant Porkolab. Integer Optimization on Convex Semialgebraic Sets. *Discrete & Computational Geometry*, 23(2):207–224, 2000. doi:10.1007/PL00009496.

38 Dušan Knop and Martin Koutecký. Scheduling meets n-fold Integer Programming. *Journal of Scheduling*, November 2017. doi:10.1007/s10951-017-0550-0.

39 Dušan Knop, Martin Koutecký, and Matthias Mnich. Combinatorial n-fold Integer Programming and Applications. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ESA.2017.54.

40 Dušan Knop, Martin Koutecký, and Matthias Mnich. Voting and Bribing in Single-Exponential Time. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on*

Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany, volume 66 of *LIPIcs*, pages 46:1–46:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

41 Dušan Knop, Tomáš Masařík, and Tomáš Toufar. Parameterized Complexity of Fair Vertex Evaluation Problems. *CoRR*, abs/1803.06878, 2018. arXiv:1803.06878.

42 Dušan Knop, Martin Koutecký, and Matthias 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*, pages 256–264, 2018. URL: http://dl.acm.org/citation.cfm?id=3237427.

43 Martin Koutecký, Asaf Levin, and Shmuel Onn. A Parameterized Strongly Polynomial Algorithm for Block Structured Integer Programs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 85:1–85:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.85.

44 Michael Lampis. Algorithmic Meta-theorems for Restrictions of Treewidth. *Algorithmica*, 64(1):19–37, 2012.

45 Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

46 Daniel Lokshtanov. *New Methods in Parameterized Algorithms and Complexity.* PhD thesis, University of Bergen, 2009.

47 Daniel Lokshtanov. Parameterized integer quadratic programming: Variables and coefficients. *arXiv preprint arXiv:1511.00310*, 2015.

48 Daniel Lokshtanov, MS Ramanujan, and Saket Saurabh. A linear time parameterized algorithm for directed feedback vertex set. *arXiv preprint arXiv:1609.04347*, 2016.

49 Dániel Marx. What's next? Future directions in parameterized complexity. In *The Multivariate Algorithmic Revolution and Beyond*, pages 469–496. Springer, 2012.

50 Tomáš Masařík and Tomáš Toufar. Parameterized Complexity of Fair Deletion Problems. In *International Conference on Theory and Applications of Models of Computation (TAMC'16)*, pages 628–642, 2017.

51 Matthias Mnich and Andreas Wiese. Scheduling and fixed-parameter tractability. *Math. Program.*, 154(1-2, Ser. B):533–562, 2015.

52 Rolf Niedermeier. Ubiquitous Parameterization - Invitation to Fixed-Parameter Algorithms. In Jirí Fiala, Václav Koubek, and Jan Kratochvíl, editors, *MFCS*, volume 3153 of *Lecture Notes in Computer Science*, page 84–103. Springer, 2004. doi:10.1007/978-3-540-28629-5_4.

53 Timm Oertel, Christian Wagner, and Robert Weismantel. Integer convex minimization by mixed integer linear optimization. *Oper. Res. Lett*, 42(6-7):424–428, 2014. doi:10.1016/j.orl.2014.07.005.

54 Christos H Papadimitriou. On the complexity of integer programming. *Journal of the ACM (JACM)*, 28(4):765–768, 1981.

55 Kevin Zemmer. *Integer Polynomial Optimization in Fixed Dimension.* PhD thesis, ETH Zurich, 2017.