Projection-width as a structural parameter for discrete separable optimization

Alberto Del Pia *

February 5, 2026

Abstract

While several classes of integer linear optimization problems are known to be solvable in polynomial time, far fewer tractability results exist for integer nonlinear optimization. In this work, we narrow this gap by identifying a broad class of discrete nonlinear optimization problems that admit polynomial-time algorithms. Central to our approach is the notion of *projection-width,* a structural parameter for systems of separable constraints, defined via branch decompositions of variables and constraints. We show that several fundamental discrete optimization and counting problems can be solved in polynomial time when the projection-width is polynomially bounded, including optimization, counting, top-k, and weighted constraint violation problems. Our results subsume and generalize some of the strongest known tractability results across multiple research areas: integer linear optimization, binary polynomial optimization, and Boolean satisfiability. Although these results originated independently within different communities and for seemingly distinct problem classes, our framework unifies and significantly generalizes them under a single structural perspective.

Key words: discrete separable optimization; integer nonlinear optimization; integer linear optimization; polynomial time algorithm; projection width; incidence treewidth

1 Introduction

The field of integer linear optimization has reached a high level of maturity, with a rich theoretical framework, efficient algorithms, and numerous applications. A central outcome of this development is the identification of broad and structurally rich classes of integer linear optimization problems that admit polynomial-time algorithms (see [Sch86, CCZ14] and references therein). Many optimization problems arising in applications, however, are inherently nonlinear. Despite their importance, integer nonlinear optimization problems remain far less understood from a complexity-theoretic perspective, with polynomial-time solvability known only for a few highly restrictive classes. The main goal of this paper is to narrow this gap by identifying a broad class of integer nonlinear optimization problems that can be solved in polynomial time.

Separable systems. The framework that we introduce in this paper operates over highly general systems of separable constraints. A *separable system* is a quadruple $(X, D, C^{\in}, C^{\geq})$, where X is a

*Department of Industrial and Systems Engineering & Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI, USA. E-mail: delpia@wisc.edu.

finite set of *variables*, D is a finite *domain* set, and where $C^{\in}$, $C^{\geq}$ are sets of *separable constraints* of the form

$$\sum_{x \in X} f_x^c(x) \in \Delta^c \qquad c \in C^{\in},$$
$$\sum_{x \in X} f_x^c(x) \geq \delta^c \qquad c \in C^{\geq},$$

where $f_x^c : D \to \mathbb{Z}$ for every $x \in X$ and $c \in C^{\in} \cup C^{\geq}$, where $\delta^c \in \mathbb{Z}$ for every $c \in C^{\geq}$, and where Δ^c is a finite subset of $\mathbb{Z}$ for every $c \in C^{\in}$. Throughout the paper, we assume that a separable system is given by explicitly providing f_x^c for every $x \in X$ and $c \in C$, Δ^c for every $c \in C^{\in}$, and δ^c for every $c \in C^{\geq}$. Clearly, a constraint in $C^{\geq}$ can also be expressed as a constraint in $C^{\in}$. However, we choose to handle constraints in $C^{\geq}$ separately, as doing so allows us to efficiently solve broader classes of problems. Separable constraints constitute a broad and expressive class of constraints. Notable special cases include linear and pseudo-Boolean inequalities and set constraints, as well as parity or, more generally, modulo constraints.

The problems. In this paper, we consider several fundamental discrete optimization and counting problems defined over a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, which we informally introduce below:

1. *Optimization:* Given a separable value function for each variable, find a highest-value assignment from X to D satisfying all the constraints. (See Section 4.1 and Theorem 1.)
2. *Counting:* Count the number of assignments from X to D satisfying all the constraints. (See Section 4.2 and Theorem 2.)
3. *Top-k:* Given a separable value function for each variable, find k highest-value assignments from X to D satisfying all the constraints. (See Section 4.3 and Theorem 3.)
4. *Weighted constraint violation:* Given a weight for each constraint, find an assignment from X to D that minimizes the weighted violation of the constraints. (See Section 4.4 and Theorem 4.)

Although the four problems above already form a substantial class of fundamental discrete nonlinear optimization and counting problems, our framework can be applied more broadly to other problems defined over separable systems. For instance, it can be leveraged in polyhedral theory to derive compact extended formulations using the approach of [MRC90], and in knowledge compilation to translate constraints into succinct deterministic DNNF representations, as in [Cap16].

Our results. We present exact algorithms for the four problems mentioned above. The running times of our algorithms are low-degree polynomials in $|X|$, $|D|$, $|C|$, where $C := C^{\in} \cup C^{\geq}$, and in two more parameters: w_{proj} and Λ. The first one, w_{proj}, is the *projection-width* of the separable system, which is the central concept introduced in this paper and is formally defined in Section 1.1. This concept is inspired by the *PS-width* of CNF formulas, originally developed for satisfiability problems in [STV14], and it generalizes the idea in a substantially different context and captures structural properties specific to separable systems over finite domains. The second parameter is Λ, which denotes the maximum time required to check whether an integer belongs to a set Δ^c, for some $c \in C^{\in}$. This parameter is introduced for convenience and generality, since Λ depends on how Δ^c is given. The running time that we obtain for the optimization and the counting problems is

$$O\left({w_{\text{proj}}}^3 \left(|X| + |C|\right)|C| + w_{\text{proj}} \left|C^{\in}\right| \Lambda + |X|\,|C|\,|D| \log\left(|D|\right)\right).$$

For the top-k problem, we get

$$O\left({w_{\text{proj}}}^3 \left(|X| + |C|\right)\left(|C| + k\log(k)\right) + w_{\text{proj}} \left|C^{\in}\right| \Lambda + |X|\,|C|\,|D| \log\left(|D|\right)\right),$$

and for the weighted constraint violation we get

$$O\left(w_{\text{proj}}{}^{3}\left(|X|+|C|\right)|C|+|X|\,|C|\,|D|\log\left(|D|\right)\right).$$

We refer the reader to Theorems 1 to 4 for the precise statements of our results. All running times are stated under the *arithmetic model* of computation, in which each basic arithmetic operation (addition, subtraction, multiplication, division, and comparison) is assumed to take constant time, independent of operand sizes. Moreover, in our algorithms, the sizes of all intermediate and output values are bounded by a polynomial in the input size. We note that sharper running-time bounds may be attainable through a more refined analysis, especially in specific cases. For instance, when the involved functions are linear rather than separable, when $|D| = 2$, or when branch decompositions (introduced later in this section) are linear [STV14].

Our results establish a broad class of tractable discrete nonlinear optimization and counting problems. To illustrate their generality, we show that they subsume some of the strongest known tractability results in integer linear optimization, binary polynomial optimization, and Boolean satisfiability. In integer linear optimization, it subsumes the polynomial-time solvability of instances with bounded incidence treewidth established in [GOR17]. In binary polynomial optimization, it generalizes the tractability of instances defined on hypergraphs with bounded incidence treewidth shown in [CDPDG24], while additionally allowing the presence of further constraints. In Boolean satisfiability, our framework encompasses the tractability of weighted MaxSAT and #SAT on CNF formulas with bounded PS-width [STV14], which captures nearly all known tractable cases of these problems [Cap16].

All these three results follow as direct corollaries of our framework under highly specialized and restrictive assumptions. Compared to our general setting, each of them is considerably narrower. In particular, all functions involved are linear, only inequality constraints are considered, and the variable domains are restricted to integer intervals in integer linear optimization, and to $\{0, 1\}$ in binary polynomial optimization and Boolean satisfiability. While the primary contribution of this paper lies in the nonlinear setting, the fact that these disparate results emerge as special cases highlights the broader significance of projection-width, which provides a common structural explanation for tractability across seemingly different problem classes and research communities.

In the remainder of this section, we present the definition of projection-width.

1.1 Projection-width

Translated separable system. Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, and let $C := C^{\in} \cup C^{\geq}$. Although not strictly necessary, we begin by rewriting the constraints in C. This reformulation clarifies the intuition behind projection-width and simplifies the notation used throughout the paper.

For every $c \in C$ we define

$$g_x^c := f_x^c - \min\left\{f_x^c(d) \mid d \in D\right\} \qquad \forall x \in X.$$

For every $c \in C^{\in}$, we also define

$$\Gamma^c := \Delta^c - \sum_{x \in X} \min\left\{f_x^c(d) \mid d \in D\right\},$$

and for every $c \in C^{\geq}$, we set

$$\gamma^c := \delta^c - \sum_{x \in X} \min\left\{f_x^c(d) \mid d \in D\right\}.$$

We then obtain the equivalent *translated separable system*

$$\sum_{x \in X} g_x^c(x) \in \Gamma^c \qquad c \in C^{\in},$$
$$\sum_{x \in X} g_x^c(x) \geq \gamma^c \qquad c \in C^{\geq}.$$

We now have $g_x^c : D \to \mathbb{Z}_{\geq 0}$ and $\min \{g_x^c(d) \mid d \in D\} = 0$ for every $x \in X$ and $c \in C$. As a result, we can assume $\Gamma^c \subseteq \mathbb{Z}_{\geq 0}$, for every $c \in C^{\in}$. Similarly, we can assume $\gamma^c \geq 0$ for every $c \in C^{\geq}$, since otherwise every assignment satisfies the constraint, and so the constraint can be discarded. For every $c \in C^{\in}$, we define γ^c as the largest element in Γ^c. From now on, we will mostly consider the translated separable system instead of the original separable system.

Projections. For constraint $c \in C$, $X' \subseteq X$, and $\tau : X' \to D$, we define the *constraint bound* of c by τ as

$$\mathrm{cb}^c(\tau) := \sum_{x \in X'} g_x^c(\tau(x)).$$

Note that $\mathrm{cb}^c(\tau) \in \mathbb{Z}_{\geq 0}$. To build intuition, we observe that the constraint bound provides the minimum possible left hand side of constraint c over all assignments from X to D whose restriction to X' is τ.

Given $X' \subseteq X$, $C' \subseteq C$, and $\tau : X' \to D$, we denote by C'/τ the map from C' to $\mathbb{Z}_{\geq 0}$ defined by

$$\left(C'/\tau\right)^c := \min \{\mathrm{cb}^c(\tau), \gamma^c\} \qquad \forall c \in C'.$$

Note that in this context, we use the word "map" to avoid confusion with our "assignments." Since $\gamma^c \in \mathbb{Z}_{\geq 0}$ and $\mathrm{cb}^c(\tau) \in \mathbb{Z}_{\geq 0}$, we have $\left(C'/\tau\right)^c \in \{0, 1, \ldots, \gamma^c\}$, for every $c \in C'$. Note that, if $C' = \emptyset$, then C'/τ is the empty map $\epsilon : \emptyset \to \mathbb{Z}_{\geq 0}$. On the other hand, if $X' = \emptyset$, then τ is the empty assignment $\epsilon : \emptyset \to D$, and C'/ϵ is given explicitly by

$$\left(C'/\epsilon\right)^c = 0 \qquad \forall c \in C'.$$

We observe that C'/τ captures the contribution of the assignment τ to the left hand sides of the constraints in C', up to the threshold γ^c.

Given $X' \subseteq X$ and $C' \subseteq C$, we define

$$\mathrm{proj}(C', X') := \left\{C'/\tau \mid \tau : X' \to D\right\}.$$

As a result, $\mathrm{proj}(C', X')$ encodes how all assignments from X' to D can contribute to the left hand sides of the constraints in C', before the threshold γ^c is reached.

Projection-width. Consider now a branch decomposition T of $X \cup C$, which is a rooted binary tree with a one-to-one correspondence between the leaves of T and the set $X \cup C$. We say that a vertex of T is an *inner vertex* if it has children, and is a *leaf* if it has no children. Given a vertex v of T, we denote by T_v the subtree of T rooted in v, by C_v the set of constraints in C such that the corresponding vertex appears in the leaves of T_v, and by X_v the set of variables in X that similarly appear in the leaves of T_v. We denote by $\overline{C_v} := C \setminus C_v$ and by $\overline{X_v} := X \setminus X_v$. A key role is played by the two projections

$$\mathrm{proj}(v) := \mathrm{proj}(\overline{C_v}, X_v) = \left\{\overline{C_v}/\tau \mid \tau : X_v \to D\right\},$$

$$\mathrm{proj}(\overline{v}) := \mathrm{proj}(C_v, \overline{X_v}) = \left\{C_v/\tau \mid \tau : \overline{X_v} \to D\right\}.$$

If we denote by $V(T)$ the set of vertices of T, the *projection-width* of the separable system $\left(X, D, C^{\in}, C^{\geq}\right)$ and T is defined by

$$\max_{v \in V(T)} \max\left(|\mathrm{proj}(v)|, |\mathrm{proj}(\overline{v})|\right).$$

The *projection-width* of the separable system $\left(X, D, C^{\in}, C^{\geq}\right)$ is then defined as the minimum among the projection-widths of the separable system $\left(X, D, C^{\in}, C^{\geq}\right)$ and T, over all branch decompositions T of $X \cup C$.

Organization of the paper. The rest of the paper is organized as follows. In Section 2, we study the relationship between the projections $\mathrm{proj}(v)$ and $\mathrm{proj}(\overline{v})$ in a branch decomposition, and use these insights to efficiently construct all such sets throughout the decomposition. In Section 3, we introduce the notion of *shapes*, which enables us to retain in the branch decomposition only the assignments that satisfy all the constraints, effectively filtering out the others. In Section 4, we present and analyze our algorithms for optimization, top-k, counting, and weighted constraint violation. Finally, in Section 5, we obtain some corollaries of our main theorems, and we discuss how our results subsume previously known tractability results in integer linear optimization, binary polynomial optimization, and Boolean satisfiability.

2 Projections

In the following, given X_1, X_2 disjoint subsets of X, $\tau_1 : X_1 \to D$, $\tau_2 : X_2 \to D$, we denote by $\tau_1 \cup \tau_2$ the assignment from $X_1 \cup X_2$ to D defined by

$$(\tau_1 \cup \tau_2)(x) := \begin{cases} \tau_1(x) & \text{if } x \in X_1, \\ \tau_2(x) & \text{if } x \in X_2. \end{cases}$$

Observation 1. *Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, and let $c \in C^{\in} \cup C^{\geq}$. The following properties hold:*

(i) Given $X' \subseteq X$ and $\tau : X' \to D$, $\mathrm{cb}^{c}(\tau) \in \mathbb{Z}_{\geq 0}$.

(ii) Given X_1, X_2 disjoint subsets of X, $\tau_1 : X_1 \to D$, $\tau_2 : X_2 \to D$, we have $\mathrm{cb}^{c}(\tau_1 \cup \tau_2) = \mathrm{cb}^{c}(\tau_1) + \mathrm{cb}^{c}(\tau_2)$.

Proof. (i). Follows directly from $g^{c}_{x}(\tau(x)) \in \mathbb{Z}_{\geq 0}$ for every $x \in X$.

(ii). Follows directly from the definition of constraint bound. □

We will often use the next simple result. The first few times we will employ it, we will use it with $c = 0$. However, later on we will need the more general version presented below.

Lemma 1. *For all nonnegative a, b, c, γ, the following identity holds:*

$$\min\{a + b, \gamma - c\} = \min\{a + \min\{b, \gamma\}, \gamma - c\}.$$

Proof.

$$\begin{aligned} \min\{a + \min\{b, \gamma\}, \gamma - c\} &= \min\{a + b, a + \gamma, \gamma - c\} \\ &= \min\{a + b, \gamma - c\}. \end{aligned}$$

□

2.1 Structure of proj(v)

The next result is at the heart of the relationship between sets proj(v) in a branch decomposition.

Lemma 2. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. Let* $\tau : X_v \to D$*, and let* τ_1 *and* τ_2 *denote the restrictions of* τ *to* X_{v_1} *and* X_{v_2}*, respectively. Then,*

$$\left(\overline{C_v}/\tau\right)^c = \min\left\{\left(\overline{C_{v_1}}/\tau_1\right)^c + \left(\overline{C_{v_2}}/\tau_2\right)^c, \gamma^c\right\} \qquad \forall c \in \overline{C_v}.$$

Proof. Let $c \in \overline{C_v}$ and observe that $\overline{C_v} = \overline{C_{v_1}} \cap \overline{C_{v_2}}$. We have

$$\begin{aligned}
\left(\overline{C_v}/\tau\right)^c &= \min\left\{\mathrm{cb}^c(\tau), \gamma^c\right\} \\
&= \min\left\{\mathrm{cb}^c(\tau_1) + \mathrm{cb}^c(\tau_2), \gamma^c\right\} && \text{(from Observation 1 (ii))} \\
&= \min\left\{\min\left\{\mathrm{cb}^c(\tau_1), \gamma^c\right\} + \min\left\{\mathrm{cb}^c(\tau_2), \gamma^c\right\}, \gamma^c\right\} && \text{(from Lemma 1)} \\
&= \min\left\{\left(\overline{C_{v_1}}/\tau_1\right)^c + \left(\overline{C_{v_2}}/\tau_2\right)^c, \gamma^c\right\}.
\end{aligned}$$

□

Lemma 2 suggests the following relationship between $\Phi \in \mathrm{proj}(v)$, $\Phi_1 \in \mathrm{proj}(v_1)$, and $\Phi_2 \in \mathrm{proj}(v_2)$:

$$\Phi^c = \min\left\{\Phi_1^c + \Phi_2^c, \gamma^c\right\} \qquad \forall c \in \overline{C_v}. \tag{L1}$$

Lemma 3. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. If* $\Phi \in \mathrm{proj}(v)$*, then there exist* $\Phi_1 \in \mathrm{proj}(v_1)$ *and* $\Phi_2 \in \mathrm{proj}(v_2)$ *such that* (L1) *holds. Vice versa, if* $\Phi_1 \in \mathrm{proj}(v_1)$ *and* $\Phi_2 \in \mathrm{proj}(v_2)$*, then there is a unique* $\Phi \in \mathrm{proj}(v)$ *such that* (L1) *holds.*

Proof. Let $\Phi \in \mathrm{proj}(v)$. Then, there exists $\tau : X_v \to D$ such that $\Phi = \overline{C_v}/\tau$. Observe that X_v is the disjoint union of sets X_{v_1} and X_{v_2}. Let τ_1 and τ_2 denote the restrictions of τ to X_{v_1} and X_{v_2}, respectively. Let $\Phi_1 := \overline{C_{v_1}}/\tau_1 \in \mathrm{proj}(v_1)$ and $\Phi_2 := \overline{C_{v_2}}/\tau_2 \in \mathrm{proj}(v_2)$. Then, Lemma 2 implies that (L1) holds.

Let $\Phi_1 \in \mathrm{proj}(v_1)$ and $\Phi_2 \in \mathrm{proj}(v_2)$. Then, there exist $\tau_1 : X_{v_1} \to D$ and $\tau_2 : X_{v_2} \to D$ such that $\Phi_1 = \overline{C_{v_1}}/\tau_1$ and $\Phi_2 = \overline{C_{v_2}}/\tau_2$. Let $\tau := \tau_1 \cup \tau_2 : X_v \to D$, and define $\Phi := \overline{C_v}/\tau \in \mathrm{proj}(v)$. Then, Lemma 2 implies that (L1) holds. From (L1), Φ is clearly unique. □

2.2 Structure of proj$(\overline{v})$

The relationship between sets proj$(\overline{v})$ in a branch decomposition is slightly more complex, since it also involves maps from proj(v). The next result is at the heart of this relationship.

Lemma 4. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. Let* $\tau_1 : \overline{X_{v_1}} \to D$*, and let* τ *and* τ_2 *denote the restrictions of* τ_1 *to* $\overline{X_v}$ *and* X_{v_2}*, respectively. Then,*

$$\left(C_{v_1}/\tau_1\right)^c = \min\left\{\left(C_v/\tau\right)^c + \left(\overline{C_{v_2}}/\tau_2\right)^c, \gamma^c\right\} \qquad \forall c \in C_{v_1}.$$

Symmetrically, let $\tau_2 : \overline{X_{v_2}} \to D$*, and let* τ *and* τ_1 *denote the restrictions of* τ_2 *to* $\overline{X_v}$ *and* X_{v_1}*, respectively. Then,*

$$\left(C_{v_2}/\tau_2\right)^c = \min\left\{\left(C_v/\tau\right)^c + \left(\overline{C_{v_1}}/\tau_1\right)^c, \gamma^c\right\} \qquad \forall c \in C_{v_2}.$$

Proof. We only prove the first part of the statement, since the second part is symmetric. Let $c \in C_{v_1}$ and observe that $C_{v_1} = C_v \cap \overline{C_{v_2}}$. We have

$$\begin{aligned}
(C_{v_1}/\tau_1)^c &= \min\{\mathrm{cb}^c(\tau_1), \gamma^c\} \\
&= \min\{\mathrm{cb}^c(\tau) + \mathrm{cb}^c(\tau_2), \gamma^c\} && \text{(from Observation 1 (ii))} \\
&= \min\{\min\{\mathrm{cb}^c(\tau), \gamma^c\} + \min\{\mathrm{cb}^c(\tau_2), \gamma^c\}, \gamma^c\} && \text{(from Lemma 1)} \\
&= \min\left\{(C_v/\tau)^c + \left(\overline{C_{v_2}}/\tau_2\right)^c, \gamma^c\right\}.
\end{aligned}$$

□

Lemma 4 suggests two more relationships. The first one between $\Psi \in \mathrm{proj}(\overline{v})$, $\Psi_1 \in \mathrm{proj}(\overline{v_1})$, and $\Phi_2 \in \mathrm{proj}(v_2)$:

$$\Psi_1^c = \min\{\Psi^c + \Phi_2^c, \gamma^c\} \qquad \forall c \in C_{v_1}. \tag{L2}$$

The second one between $\Psi \in \mathrm{proj}(\overline{v})$, $\Phi_1 \in \mathrm{proj}(v_1)$, and $\Psi_2 \in \mathrm{proj}(\overline{v_2})$:

$$\Psi_2^c = \min\{\Psi^c + \Phi_1^c, \gamma^c\} \qquad \forall c \in C_{v_2}. \tag{L3}$$

Lemma 5. *Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, let T be a branch decomposition of $X \cup C$, and let v be an inner vertex of T with children v_1 and v_2. If $\Psi_1 \in \mathrm{proj}(\overline{v_1})$, then there exist $\Psi \in \mathrm{proj}(\overline{v})$ and $\Phi_2 \in \mathrm{proj}(v_2)$ such that* (L2) *holds. Vice versa, if $\Psi \in \mathrm{proj}(\overline{v})$ and $\Phi_2 \in \mathrm{proj}(v_2)$, then there is a unique $\Psi_1 \in \mathrm{proj}(\overline{v_1})$ such that* (L2) *holds. Symmetrically, if $\Psi_2 \in \mathrm{proj}(\overline{v_2})$, then there exist $\Psi \in \mathrm{proj}(\overline{v})$ and $\Phi_1 \in \mathrm{proj}(v_1)$ such that* (L3) *holds. Vice versa, if $\Psi \in \mathrm{proj}(\overline{v})$ and $\Phi_1 \in \mathrm{proj}(v_1)$, then there is a unique $\Psi_2 \in \mathrm{proj}(\overline{v_2})$ such that* (L3) *holds.*

Proof. We only prove the first part of the statement, since the second part is symmetric.

Let $\Psi_1 \in \mathrm{proj}(\overline{v_1})$. Then, there exists $\tau_1 : \overline{X_{v_1}} \to D$ such that $\Psi_1 = C_{v_1}/\tau_1$. Observe that $\overline{X_{v_1}}$ is the union of disjoint sets $\overline{X_v}$ and X_{v_2}. Let τ and τ_2 denote the restrictions of τ_1 to $\overline{X_v}$ and X_{v_2}, respectively. Let $\Psi := C_v/\tau \in \mathrm{proj}(\overline{v})$ and $\Phi_2 := \overline{C_{v_2}}/\tau_2 \in \mathrm{proj}(v_2)$. Then, Lemma 4 implies that (L2) holds.

Let $\Psi \in \mathrm{proj}(\overline{v})$ and $\Phi_2 \in \mathrm{proj}(v_2)$. Then, there exist $\tau : \overline{X_v} \to D$ and $\tau_2 : X_{v_2} \to D$ such that $\Psi = C_v/\tau$ and $\Phi_2 = \overline{C_{v_2}}/\tau_2$. Let $\tau_1 := \tau \cup \tau_2 : \overline{X_{v_1}} \to D$, and define $\Psi_1 := C_{v_1}/\tau_1 \in \mathrm{proj}(\overline{v_1})$. Then, Lemma 4 implies that (L2) holds. From (L2), Ψ_1 is clearly unique. □

2.3 Constructing all projections

In this section, we use the structural results in Lemmas 3 and 5 to construct efficiently all sets $\mathrm{proj}(v)$ and $\mathrm{proj}(\overline{v})$ in a branch decomposition. We begin with some projections that have a very simple structure.

Remark 1. *Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, and let T be a branch decomposition of $X \cup C$.*

(i) <u>$\mathrm{proj}(l)$, *for a leaf l of T corresponding to a constraint $c \in C$.*</u> *We have $X_l = \emptyset$, thus there is only one assignment from X_l to D, the empty assignment ϵ. Therefore,* $\mathrm{proj}(l)$ *contains only one map. Because $\overline{C_l} = C \setminus \{c\}$, this map is given by $(C \setminus \{c\})/\epsilon$.*

(ii) <u>$\mathrm{proj}(l)$, *for a leaf l of T corresponding to a variable $x \in X$.*</u> *We have $X_l = \{x\}$, thus there are $|D|$ assignment from X_l to D. Therefore,* $\mathrm{proj}(l)$ *contains at most $|D|$ maps. Because $\overline{C_l} = C$, they are of the form C/τ, for every assignment $\tau : \{x\} \to D$.*

(iii) <u>$\mathrm{proj}(r)$, *for the root r of T.*</u> *Since $\overline{C_r} = \emptyset$,* $\mathrm{proj}(r)$ *contains only the empty map $\epsilon : \emptyset \to \mathbb{Z}_{\geq 0}$.*

(iv) $\underline{\text{proj}(\overline{r})\text{, for the root } r \text{ of } T.}$ *Since* $\overline{X_r} = \emptyset$, *proj*$(\overline{r})$ *contains only one map. Because* $C_r = C$, *this map is given by* C/ϵ.

Proposition 1. *Consider a separable system* $\left(X, D, C^{\in}, C^{\geq}\right)$*, let* $C := C^{\in} \cup C^{\geq}$*, and let* T *be a branch decomposition of* $X \cup C$ *of projection-width* w_{proj}*. There is an algorithm that computes* $\text{proj}(v)$ *and* $\text{proj}(\overline{v})$*, for every vertex* v *of* T*, in time*

$$O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \left(|X| + |C|\right) |C| + |X| \, |C| \, |D| \log\left(|D|\right)\right).$$

Proof. The construction of the sets $\text{proj}(v)$, for every vertex v of T, is performed in a bottom up manner.

Leaves corresponding to constraints. For every leaf l of T corresponding to a constraint $c \in C$, we know from Remark 1 (i) that $\text{proj}(l)$ contains only one map, and it can be constructed in time $O\left(|C|\right)$. Since there are $|C|$ leaves corresponding to constraints, they require $O(|C|^2)$ time.

Leaves corresponding to variables. For every leaf l of T corresponding to a variable $x \in X$, we know from Remark 1 (ii) that $\text{proj}(l)$ contains at most $|D|$ maps, and they can be constructed, possibly with duplicates, in time $O\left(|C| \, |D|\right)$. We sort the obtained $|D|$ maps lexicographically in time $O\left(|C| \, |D| \log\left(|D|\right)\right)$, and then we delete duplicates in time $O\left(|C| \, |D|\right)$. Therefore, for every leaf l of T, the set $\text{proj}(l)$ can be constructed in time $O\left(|C| \, |D| \log\left(|D|\right)\right)$. Since there are $|X|$ leaves corresponding to variables, they require $O\left(|X| \, |C| \, |D| \log\left(|D|\right)\right)$ time.

Inner vertices. Consider an inner vertex v of T with children v_1 and v_2. From Lemma 3, every $\Phi \in \text{proj}(v)$ can be constructed from one $\Phi_1 \in \text{proj}(v_1)$ and one $\Phi_2 \in \text{proj}(v_2)$ as in (L1). For each pair Φ_1, Φ_2, the construction requires $O\left(|C|\right)$ time. Since T is of projection-width w_{proj}, there are at most ${w_{\text{proj}}}^2$ pairs, we can construct all maps in $\text{proj}(v)$ in time $O\left({w_{\text{proj}}}^2 \, |C|\right)$. We then sort the obtained maps lexicographically in time $O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \, |C|\right)$, and then we delete duplicates in time $O\left({w_{\text{proj}}}^2 \, |C|\right)$. Therefore, for every inner vertex v of T, the set $\text{proj}(v)$ can be constructed in time $O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \, |C|\right)$. Since there are $|X| + |C| - 1$ inner vertices, they require $O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \left(|X| + |C|\right) |C|\right)$ time.

The construction of the sets $\text{proj}(\overline{v})$, for every vertex v of T, is performed in a top down manner. For the root r of T, the set $\text{proj}(\overline{r})$ can be constructed, as in Remark 1 (iv), in time $O\left(|C|\right)$. Next, consider a vertex v of T with children v_1 and v_2. From Lemma 5, every $\Psi_1 \in \text{proj}(\overline{v_1})$ can be constructed from one $\Psi \in \text{proj}(\overline{v})$ and one $\Phi_2 \in \text{proj}(v_2)$ as in (L2). For each pair Ψ, Φ_2, the construction requires $O\left(|C|\right)$ time. Since T is of projection-width w_{proj}, there are at most ${w_{\text{proj}}}^2$ pairs, and the total time required for v_1, including deleting duplicates, is $O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \, |C|\right)$. Symmetrically, we can construct $\text{proj}(\overline{v_2})$ using (L3). Since the total number of vertices of T is $2\left(|X| + |C|\right) - 1$, the construction of the sets $\text{proj}(\overline{v})$, for every vertex v of T, requires $O\left({w_{\text{proj}}}^2 \log(w_{\text{proj}}) \left(|X| + |C|\right) |C|\right)$ time. ∎

3 Shapes

Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, let T be a branch decomposition of $X \cup C$, and let v be a vertex of T. A *shape* (for v, with respect to T) is a pair

$$(\Phi, \Psi)$$

such that $\Phi \in \text{proj}(v)$ and $\Psi \in \text{proj}(\overline{v})$. In other words, (Φ, Ψ) is a shape if there exists $\tau : X_v \to D$ such that $\Phi = \overline{C_v}/\tau$, and there exists $\tau' : \overline{X_v} \to D$ such that $\Psi = C_v/\tau'$. Observe that if T is of projection-width w_{proj}, then $\text{proj}(v)$ and $\text{proj}(\overline{v})$ have cardinality at most w_{proj}, thus there are at most ${w_{\text{proj}}}^2$ different shapes for v.

Linked shapes. We now define the concept of "linked shapes", which allows us to relate the shapes for an inner vertex v to the shapes for its children v_1 and v_2. Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, let T be a branch decomposition of $X \cup C$, and let v be an inner vertex of T with children v_1 and v_2. We say that (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) are *linked shapes* for v, v_1, v_2, if they can be constructed as follows: First, let $\Psi \in \text{proj}(\overline{v})$, $\Phi_1 \in \text{proj}(v_1)$, and $\Phi_2 \in \text{proj}(v_2)$ such that

$$\Phi_1^c + \Phi_2^c \leq \gamma^c \qquad \forall c \in \overline{C_v} \cap C^{\in}, \tag{L1*}$$
$$\Psi^c + \Phi_2^c \leq \gamma^c \qquad \forall c \in C_{v_1} \cap C^{\in}, \tag{L2*}$$
$$\Psi^c + \Phi_1^c \leq \gamma^c \qquad \forall c \in C_{v_2} \cap C^{\in}. \tag{L3*}$$

Then define Φ, Ψ_1, and Ψ_2 according to (L1), (L2), and (L3), respectively. We also say that the above linked shapes *originated from* Ψ, Φ_1, Φ_2. Since $\Phi^c, \Psi_1^c, \Psi_2^c, \in \{0, 1, \ldots, \gamma^c\}$, (L1), (L1*), (L2), (L2*), (L3), (L3*) are equivalent to

$$\Phi^c = \Phi_1^c + \Phi_2^c \qquad \forall c \in \overline{C_v} \cap C^{\in}, \tag{L1$^{\in}$}$$
$$\Psi_1^c = \Psi^c + \Phi_2^c \qquad \forall c \in C_{v_1} \cap C^{\in}, \tag{L2$^{\in}$}$$
$$\Psi_2^c = \Psi^c + \Phi_1^c \qquad \forall c \in C_{v_2} \cap C^{\in}, \tag{L3$^{\in}$}$$
$$\Phi^c = \min\left\{\Phi_1^c + \Phi_2^c, \gamma^c\right\} \qquad \forall c \in \overline{C_v} \cap C^{\geq}, \tag{L1$^{\geq}$}$$
$$\Psi_1^c = \min\left\{\Psi^c + \Phi_2^c, \gamma^c\right\} \qquad \forall c \in C_{v_1} \cap C^{\geq}, \tag{L2$^{\geq}$}$$
$$\Psi_2^c = \min\left\{\Psi^c + \Phi_1^c, \gamma^c\right\} \qquad \forall c \in C_{v_2} \cap C^{\geq}. \tag{L3$^{\geq}$}$$

Note that linked shapes are indeed shapes. In fact, Lemma 3 implies that $\Phi \in \text{proj}(v)$, thus (Φ, Ψ) is a shape for v. On the other hand, Lemma 5 implies $\Psi_1 \in \text{proj}(\overline{v_1})$ and $\Psi_2 \in \text{proj}(\overline{v_2})$, so (Φ_1, Ψ_1) is a shape for v_1 and (Φ_2, Ψ_2) is a shape for v_2.

Assignments of shape. Consider a separable system $\left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, let T be a branch decomposition of $X \cup C$, and let v be a vertex of T. Given a shape (Φ, Ψ) for v, we say that $\tau : X_v \to D$ has *shape* (Φ, Ψ) if

$$\Phi^c = \min\left\{\text{cb}^c(\tau), \gamma^c\right\} \qquad \forall c \in \overline{C_v}, \tag{S1}$$
$$\text{cb}^c(\tau) \leq \gamma^c \qquad \forall c \in \overline{C_v} \cap C^{\in}, \tag{S1*}$$
$$\Psi^c + \text{cb}^c(\tau) \in \Gamma^c \qquad \forall c \in C_v \cap C^{\in}, \tag{S2$^{\in}$}$$
$$\Psi^c + \text{cb}^c(\tau) \geq \gamma^c \qquad \forall c \in C_v \cap C^{\geq}. \tag{S2$^{\geq}$}$$

Note that (S1) can be written in the form $\Phi = \overline{C_v}/\tau$. Also, since $\Phi^c \leq \gamma^c$, (S1) and (S1*) are equivalent to

$$\Phi^c = \text{cb}^c(\tau) \qquad \forall c \in \overline{C_v} \cap C^{\in}, \tag{S1$^{\in}$}$$
$$\Phi^c = \min\left\{\text{cb}^c(\tau), \gamma^c\right\} \qquad \forall c \in \overline{C_v} \cap C^{\geq}. \tag{S1$^{\geq}$}$$

Note that an assignment $\tau : X_v \to D$ can have more than one shape. If τ has shape (Φ_1, Ψ_1) and shape (Φ_2, Ψ_2), then it only implies $\Phi_1 = \Phi_2 = \overline{C_v}/\tau$.

The intuition behind the notion of "assignment of shape" is that if $\tau : X_v \to D$ has shape (Φ, Ψ), then it can be extended to an assignment satisfying all constraints in C_v by combining it with $\tau' : \overline{X_v} \to D$ such that $C_v/\tau' \geq \Psi$. The next result details how the notion of shape allows us to obtain the assignments that satisfy all the constraints in C.

Remark 2. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* r *be the root of* T*. Consider the shape* $(\epsilon, C/\epsilon)$ *for* r *(see Remark 1 (iii) and (iv)), and note that* $X_r = X$*,* $C_r = C$*, and* $\overline{C_r} = \emptyset$*. Therefore,* (S1$^{\in}$)*,* (S1$^{\geq}$)*,* (S2$^{\in}$)*,* (S2$^{\geq}$) *imply that an assignment* $\tau : X \to D$ *has shape* $(\epsilon, C/\epsilon)$ *if and only if*

$$\begin{aligned} &\text{cb}^{c}(\tau) \in \Gamma^{c} && \forall c \in C^{\in}, \\ &\text{cb}^{c}(\tau) \geq \gamma^{c} && \forall c \in C^{\geq}. \end{aligned}$$

Since

$$\text{cb}^{c}(\tau) = \sum_{x \in X} g_{x}^{c}(\tau(x)),$$

this happens if and only if τ *satisfies the constraints in* C*.*

In particular, Remark 2 implies that there can be shapes (Φ, Ψ) for v such that there is no $\tau : X_v \to D$ of shape (Φ, Ψ).

3.1 From children to parent

Our next goal is to relate the "assignments of shape" for an inner vertex v to the "assignments of shape" for its children v_1 and v_2. First, we study this connection when traversing the tree in a bottom up manner.

Lemma 6. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. Let* (Φ, Ψ)*,* (Φ_1, Ψ_1)*,* (Φ_2, Ψ_2) *be linked shapes for* v, v_1, v_2*. If* $\tau_1 : X_{v_1} \to D$ *has shape* (Φ_1, Ψ_1) *and* $\tau_2 : X_{v_2} \to D$ *has shape* (Φ_2, Ψ_2)*, then* $\tau := \tau_1 \cup \tau_2 : X_v \to D$ *has shape* (Φ, Ψ)*.*

Proof. Let $\tau_1 : X_{v_1} \to D$ of shape (Φ_1, Ψ_1) and $\tau_2 : X_{v_2} \to D$ of shape (Φ_2, Ψ_2). To prove that τ has shape (Φ, Ψ), we need to show that τ satisfies conditions (S1$^{\in}$), (S1$^{\geq}$), (S2$^{\in}$), (S2$^{\geq}$).

Condition (S1$^{\in}$)**.** Let $c \in \overline{C_v} \cap C^{\in}$, and note that $j \in \overline{C_{v_1}} \cap \overline{C_{v_2}}$. We have

$$\begin{aligned} \Phi^{c} &= \Phi_1^{c} + \Phi_2^{c} && \text{(from (L1}^{\in}\text{))} \\ &= \text{cb}^{c}(\tau_1) + \text{cb}^{c}(\tau_2) && \text{(from (S1}^{\in}\text{) for } \tau_1 \text{ and } \tau_2\text{)} \\ &= \text{cb}^{c}(\tau) && \text{(from Observation 1 (ii)).} \end{aligned}$$

Therefore, τ satisfies condition (S1$^{\in}$).

Condition (S1$^{\geq}$)**.** Let $c \in \overline{C_v} \cap C^{\geq}$, and note that $j \in \overline{C_{v_1}} \cap \overline{C_{v_2}}$. We have

$$\begin{aligned} \Phi^{c} &= \min\{\Phi_1^{c} + \Phi_2^{c}, \gamma^{c}\} && \text{(from (L1}^{\geq}\text{))} \\ &= \min\{\min\{\text{cb}^{c}(\tau_1), \gamma^{c}\} + \min\{\text{cb}^{c}(\tau_2), \gamma^{c}\}, \gamma^{c}\} && \text{(from (S1}^{\geq}\text{) for } \tau_1 \text{ and } \tau_2\text{)} \\ &= \min\{\text{cb}^{c}(\tau), \gamma^{c}\} && \text{(from Lemma 2).} \end{aligned}$$

Therefore, τ satisfies condition (S1$^{\geq}$).

Condition (S2$^{\in}$)**.** Let $c \in C_v \cap C^{\in}$. Then $c \in C_{v_1} \cap \overline{C_{v_2}}$ or $c \in C_{v_2} \cap \overline{C_{v_1}}$. We assume, without loss of generality, that $c \in C_{v_1} \cap \overline{C_{v_2}}$, since the other case is symmetric. We get

$$\begin{aligned} \Psi^{c} + \text{cb}^{c}(\tau) &= \Psi^{c} + \text{cb}^{c}(\tau_1) + \text{cb}^{c}(\tau_2) && \text{(from Observation 1 (ii))} \\ &= \Psi^{c} + \text{cb}^{c}(\tau_1) + \Phi_2^{c} && \text{(from (S1}^{\in}\text{) for } \tau_2\text{)} \end{aligned}$$

$$
\begin{aligned}
&= \Psi_1^c + \mathrm{cb}^c(\tau_1) && \text{(from (L2}^{\in}\text{))} \\
&\in \Gamma^c && \text{(from (S2}^{\in}\text{) for } \tau_1\text{).}
\end{aligned}
$$

Therefore, τ satisfies condition (S2$^{\in}$).

Condition (S2$^{\geq}$)**.** Let $c \in C_v \cap C^{\geq}$. Then $c \in C_{v_1} \cap \overline{C_{v_2}}$ or $c \in C_{v_2} \cap \overline{C_{v_1}}$. We assume, without loss of generality, that $c \in C_{v_1} \cap \overline{C_{v_2}}$, since the other case is symmetric. We get

$$
\begin{aligned}
\Psi^c + \mathrm{cb}^c(\tau) &= \Psi^c + \mathrm{cb}^c(\tau_1) + \mathrm{cb}^c(\tau_2) && \text{(from Observation 1 (ii))} \\
&\geq \Psi^c + \mathrm{cb}^c(\tau_1) + \min\{\mathrm{cb}^c(\tau_2), \gamma^c\} \\
&= \Psi^c + \mathrm{cb}^c(\tau_1) + \Phi_2^c && \text{(from (S1}^{\geq}\text{) for } \tau_2\text{)} \\
&\geq \min\{\Psi^c + \Phi_2^c, \gamma^c\} + \mathrm{cb}^c(\tau_1) \\
&= \Psi_1^c + \mathrm{cb}^c(\tau_1) && \text{(from (L2}^{\geq}\text{))} \\
&\geq \gamma^c && \text{(from (S2}^{\geq}\text{) for } \tau_1\text{).}
\end{aligned}
$$

Therefore, τ satisfies condition (S2$^{\geq}$). □

3.2 From parent to children

Next, we study this connection among "assignments of shape" when traversing the tree in a top down manner.

Lemma 7. *Consider a separable system* $(X, D, C^{\in}, C^{\geq})$*, let* $C := C^{\in} \cup C^{\geq}$*, let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. Let* (Φ, Ψ) *be a shape for* v*, and let* $\tau : X_v \to D$ *of shape* (Φ, Ψ)*. Let* τ_1 *and* τ_2 *denote the restrictions of* τ *to* X_{v_1} *and* X_{v_2}*, respectively. There is a unique pair of shapes* (Φ_1, Ψ_1) *for* v_1 *and* (Φ_2, Ψ_2) *for* v_2 *such that* (Φ, Ψ)*,* (Φ_1, Ψ_1)*,* (Φ_2, Ψ_2) *are linked shapes,* τ_1 *has shape* (Φ_1, Ψ_1)*, and* τ_2 *has shape* (Φ_2, Ψ_2)*.*

Proof. Since $\tau_1 : X_{v_1} \to D$ must have shape (Φ_1, Ψ_1), and $\tau_2 : X_{v_2} \to D$ must have shape (Φ_2, Ψ_2), according to (S1) we need to set $\Phi_1 := \overline{C_{v_1}}/\tau_1 \in \mathrm{proj}(v_1)$ and $\Phi_2 := \overline{C_{v_2}}/\tau_2 \in \mathrm{proj}(v_2)$. Then, Lemma 2 implies that (L1) holds. Therefore, the only triple of linked shapes that we can consider is the one originated from Ψ, Φ_1, Φ_2, which we denote by (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2). Note that, according to Lemma 2 and (L1), the linked shape (Φ, Ψ) that we obtained is indeed the shape for v we started from.

To complete the proof, we only need to show that τ_1 has shape (Φ_1, Ψ_1) and τ_2 has shape (Φ_2, Ψ_2). We only show it for τ_1, as the other one is symmetric. We already know that (S1) holds for τ_1, thus to prove that τ_1 has shape (Φ_1, Ψ_1), it suffices to show that τ_1 satisfies (S1*), (S2$^{\in}$), (S2$^{\geq}$).

Condition (S1*)**.** Let $j \in \overline{C_{v_1}} \cap C^{\in}$. We consider separately two cases. If $c \in \overline{C_v}$, we have

$$
\begin{aligned}
\mathrm{cb}^c(\tau_1) &= \mathrm{cb}^c(\tau) - \mathrm{cb}^c(\tau_2) && \text{(from Observation 1 (ii))} \\
&\leq \mathrm{cb}^c(\tau) && \text{(from Observation 1 (i))} \\
&\leq \gamma^c && \text{(from (S1}^*\text{) for } \tau\text{).}
\end{aligned}
$$

If $c \in C_v$, we have

$$
\begin{aligned}
\mathrm{cb}^c(\tau_1) &= \mathrm{cb}^c(\tau) - \mathrm{cb}^c(\tau_2) && \text{(from Observation 1 (ii))} \\
&\leq \mathrm{cb}^c(\tau) && \text{(from Observation 1 (i))} \\
&\in \Gamma^c - \Psi^c && \text{(from (S2}^{\in}\text{) for } \tau\text{)}
\end{aligned}
$$

$$\leq \gamma^c.$$

Therefore, τ satisfies condition (S1*).

Condition (S2$^{\in}$)**.** Let $c \in C_{v_1} \cap C^{\in}$. Then $c \in C_v \cap \overline{C_{v_2}}$. We have

$$\begin{aligned}
\Psi_1^c + \mathrm{cb}^c(\tau_1) &= \min\{\Psi^c + \Phi_2^c, \gamma^c\} + \mathrm{cb}^c(\tau_1) && \text{(from (L2))}\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau_2), \gamma^c\} + \mathrm{cb}^c(\tau_1) && \text{(from (S1}^{\in}\text{) for } \tau_2)\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau_1) + \mathrm{cb}^c(\tau_2), \gamma^c + \mathrm{cb}^c(\tau_1)\} &&\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau), \gamma^c + \mathrm{cb}^c(\tau_1)\} && \text{(from Observation 1 (ii))}\\
&= \Psi^c + \mathrm{cb}^c(\tau) && \text{(from (S2}^{\in}\text{) for } \tau)\\
&\in \Gamma^c && \text{(from (S2}^{\in}\text{) for } \tau).
\end{aligned}$$

Therefore, τ satisfies condition (S2$^{\in}$).

Condition (S2$^{\geq}$)**.** Let $c \in C_{v_1} \cap C^{\geq}$. Then $c \in C_v \cap \overline{C_{v_2}}$. We have

$$\begin{aligned}
\Psi_1^c + \mathrm{cb}^c(\tau_1) &= \min\{\Psi^c + \Phi_2^c, \gamma^c\} + \mathrm{cb}^c(\tau_1) && \text{(from (L2))}\\
&= \min\{\Psi^c + \min\{\mathrm{cb}^c(\tau_2), \gamma^c\}, \gamma^c\} + \mathrm{cb}^c(\tau_1) && \text{(from (S1}^{\geq}\text{) for } \tau_2)\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau_2), \gamma^c\} + \mathrm{cb}^c(\tau_1) && \text{(from Lemma 1)}\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau_1) + \mathrm{cb}^c(\tau_2), \mathrm{cb}^c(\tau_1) + \gamma^c\} &&\\
&= \min\{\Psi^c + \mathrm{cb}^c(\tau), \mathrm{cb}^c(\tau_1) + \gamma^c\} && \text{(from Observation 1 (ii))}\\
&\geq \gamma^c && \text{(from (S2}^{\geq}\text{) for } \tau).
\end{aligned}$$

Therefore, τ satisfies condition (S2$^{\geq}$). ☐

3.3 Structure of shapes

The next key result is a direct consequence of Lemmas 6 and 7, and will play a major role in our main algorithms. To state it, we define

$$\hat{X}_v(\Phi, \Psi) := \{\tau : X_v \to D \mid \tau \text{ has shape } (\Phi, \Psi)\}.$$

Proposition 2. *Consider a separable system $(X, D, C^{\in}, C^{\geq})$, let $C := C^{\in} \cup C^{\geq}$, let T be a branch decomposition of $X \cup C$, and let v be an inner vertex of T with children v_1 and v_2. Let (Φ, Ψ) be a shape for v, and let $\mathcal{P}$ denote the set of pairs of shapes (Φ_1, Ψ_1) for v_1 and (Φ_2, Ψ_2) for v_2 such that (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) are linked shapes. Then $\hat{X}_v(\Phi, \Psi)$ is the union of disjoint sets*

$$\left\{\tau_1 \cup \tau_2 \mid \tau_1 \in \hat{X}_{v_1}(\Phi_1, \Psi_1), \tau_2 \in \hat{X}_{v_2}(\Phi_2, \Psi_2)\right\}, \qquad \forall\, ((\Phi_1, \Psi_1), (\Phi_2, \Psi_2)) \in \mathcal{P}.$$

Proof. Containment $\supseteq$ follows from Lemma 6, while containment $\subseteq$ follows from Lemma 7. The fact that the union is disjoint follows from the uniqueness in Lemma 7. ☐

4 Algorithms

4.1 Optimization

In this section, we consider our first problem defined over a separable system, and we show how shapes can be used to solve it. In the *optimization problem,* we are given a separable system

$\left(X, D, C^{\in}, C^{\geq}\right)$ and $\nu_x : D \to \mathbb{R}$ for every $x \in X$. For every assignment $\tau : X \to D$, we define its *value*

$$\nu(\tau) := \sum_{x \in X} \nu_x\left(\tau(x)\right). \tag{1}$$

The goal is to find a highest-value assignment from X to D satisfying the constraints in $C^{\in} \cup C^{\geq}$, or prove that no such assignment exists. In the next result, we show that we can solve efficiently the optimization problem on separable systems with bounded projection-width.

Theorem 1 (Optimization)**.** *Consider a separable system* $\left(X, D, C^{\in}, C^{\geq}\right)$*, let* $C := C^{\in} \cup C^{\geq}$*, and let* T *be a branch decomposition of* $X \cup C$ *of projection-width* w_{proj}*. Let* $\nu_x : D \to \mathbb{R}$ *for every* $x \in X$*. Then, the optimization problem can be solved in time*

$$O\left({w_{\mathrm{proj}}}^3\left(|X| + |C|\right)|C| + w_{\mathrm{proj}}\left|C^{\in}\right|\Lambda + |X|\,|C|\,|D|\log\left(|D|\right)\right). \tag{2}$$

Proof. First, we apply Proposition 1 and compute $\mathrm{proj}(v)$ and $\mathrm{proj}(\overline{v})$, for every vertex v of T, in time $O\left({w_{\mathrm{proj}}}^2\log(w_{\mathrm{proj}})\left(|X| + |C|\right)|C| + |X|\,|C|\,|D|\log\left(|D|\right)\right)$.

Table. Next, our algorithm will construct, for each vertex v of T, a table Tab_v indexed by the shapes (Φ, Ψ) for v. For every assignment $\tau : X_v \to D$, we define its *value*

$$\nu_v(\tau) := \sum_{x \in X_v} \nu_x\left(\tau(x)\right).$$

For a shape (Φ, Ψ), the content of the table Tab_v at this index, which we denote by $\mathrm{Tab}_v(\Phi, \Psi)$, should be a pair $(\tau, \nu_v(\tau))$, where τ is a highest-value assignment from X_v to D of shape (Φ, Ψ). If no such assignment exists, we should have $\mathrm{Tab}_v(\Phi, \Psi) = \mathrm{NA}$. We now explain how we can compute the table Tab_v, for every vertex v of T. This is done in a bottom up manner.

Leaves corresponding to constraints. Consider a leaf l of T corresponding to a constraint $c \in C$. From Remark 1 (i), there is only one assignment from X_l to D, the empty assignment ϵ, which has value 0, and that can be constructed in time $O(1)$. We now find all shapes (Φ, Ψ) for l such that ϵ has shape (Φ, Ψ); We then set $\mathrm{Tab}_l(\Phi, \Psi) := (\epsilon, 0)$ if ϵ has shape (Φ, Ψ), and $\mathrm{Tab}_l(\Phi, \Psi) := \mathrm{NA}$ otherwise. Clearly, there is only one $\Phi \in \mathrm{proj}(l)$, and it satisfies $(\mathrm{S1}^{\in})$ and $(\mathrm{S1}^{\geq})$. For every $\Phi \in \mathrm{proj}(\overline{l})$, we need to check $(\mathrm{S2}^{\in})$ or $(\mathrm{S2}^{\geq})$, depending on whether c is in $C^{\in}$ or $C^{\geq}$, and this can be done in time $O(\Lambda)$ and $O(1)$ respectively. Therefore, we can compute the table Tab_l in time $O(w_{\mathrm{proj}}\Lambda)$ if $c \in C^{\in}$, and in time $O(w_{\mathrm{proj}})$ if $c \in C^{\geq}$. Since there are $|C^{\in}|$ leaves corresponding to constraints in $C^{\in}$, and $\left|C^{\geq}\right|$ leaves corresponding to constraints in $C^{\geq}$, in total they require $O\left(w_{\mathrm{proj}}|C^{\in}|\Lambda + w_{\mathrm{proj}}\left|C^{\geq}\right|\right)$ time.

Leaves corresponding to variables. Consider a leaf l of T corresponding to a variable $x \in X$. From Remark 1 (ii), there are $|D|$ assignment from X_l to D, and they can be constructed, with their values, in time $O(|D|)$. We now fix one such assignment τ, and find all shapes (Φ, Ψ) for l such that τ has shape (Φ, Ψ). There is at most one $\Phi \in \mathrm{proj}(l)$ satisfying $(\mathrm{S1}^{\in})$ and $(\mathrm{S1}^{\geq})$, and it can be constructed in time $O(C)$. We fix such Φ, and observe that τ has shape (Φ, Ψ), for every $\Psi \in \mathrm{proj}(\overline{l})$, which are at most w_{proj}. This is because conditions $(\mathrm{S2}^{\in})$, $(\mathrm{S2}^{\geq})$ are always satisfied since $C_v = \emptyset$. Once we have considered all $|D|$ assignments, for every shape (Φ, Ψ) for l, which are at most ${w_{\mathrm{proj}}}^2$, we set $\mathrm{Tab}_l(\Phi, \Psi) := (\tau, \nu_v(\tau))$, where τ is the highest-value assignment of shape (Φ, Ψ), or $\mathrm{Tab}_l(\Phi, \Psi) := \mathrm{NA}$ if no assignment has shape (Φ, Ψ). Therefore, we can compute the table Tab_l in time $O\left(|C|\,|D| + {w_{\mathrm{proj}}}^2\right)$. Since there are $|X|$ leaves corresponding to variables, in total they require $O\left(|X|\,|C|\,|D| + {w_{\mathrm{proj}}}^2\,|X|\right)$ time.

Inner vertices. Consider now an inner vertex v of T, with children v_1, v_2. We loop over all triples (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) of linked shapes for v, v_1, v_2. To do so, we pick $\Psi \in \text{proj}(\overline{v})$, $\Phi_1 \in \text{proj}(v_1)$, and $\Phi_2 \in \text{proj}(v_2)$, we check (L1*), (L2*), (L1*), and if they are satisfied, we define Φ, Ψ_1, and Ψ_2 according to (L1), (L2), and (L3), respectively. Note that we have at most ${w_{\text{proj}}}^3$ such triples, and for each the above check and construction requires $O(|C|)$ time. For each triple of linked shapes with $\text{Tab}_{v_1}(\Phi_1, \Psi_1) \neq \text{NA}$ and $\text{Tab}_{v_2}(\Phi_2, \Psi_2) \neq \text{NA}$, let $(\tau_1, \nu_{v_1}(\tau_1)) := \text{Tab}_{v_1}(\Phi_1, \Psi_1)$ and $(\tau_2, \nu_{v_2}(\tau_2)) := \text{Tab}_{v_2}(\Phi_2, \Psi_2)$. We then construct the value of the "candidate assignment for (Φ, Ψ)" given by $\tau_1 \cup \tau_2$ in time $O(1)$ by summing $\nu_{v_1}(\tau_1)$ and $\nu_{v_2}(\tau_2)$.

For each shape (Φ, Ψ) for v, we then set the highest-value assignment $\tau_1 \cup \tau_2$, among all candidate assignments for (Φ, Ψ), as the content of $\text{Tab}_v(\Phi, \Psi)$, together with its value. If there are no candidate assignments for (Φ, Ψ), we set $\text{Tab}_l(\Phi, \Psi) := \text{NA}$. To improve runtime, here we do not explicitly construct $\tau_1 \cup \tau_2$; instead, we store pointers to (Φ_1, Ψ_1) and (Φ_2, Ψ_2) giving the highest-value, so that this is done in time $O(1)$ instead of $O(|X|)$. Proposition 2 implies that we set Tab_v correctly, for every inner vertex v of T. Therefore, for each inner vertex v of T, the table Tab_v can be computed (partially implicitly) in time $O({w_{\text{proj}}}^3 |C|)$. Since there are $|X| + |C| - 1$ inner vertices of T, in total they require $O({w_{\text{proj}}}^3 (|X| + |C|) |C|)$ time.

Root. Once the table Tab_v is computed, for every vertex v of T, we consider the root r of T, and the shape $(\epsilon, C/\epsilon)$ for r (see Remark 1 (iii) and (iv)). From Remark 2, the assignments $\tau : X \to D$ that have this shape are precisely those that satisfy the constraints in C. The table Tab_r, indexed by $(\epsilon, C/\epsilon)$, implicitly contains a highest-value assignment from X to D satisfying the constraints in C. Following our pointers, we can construct it explicitly in time $O(|X| + |C|)$. ☐

4.2 Counting

In this section, we consider the *counting problem.* In this problem, we are given a separable system $(X, D, C^{\in}, C^{\geq})$, and our goal is to return the number of assignments $\tau : X \to D$ satisfying the constraints in $C^{\in} \cup C^{\geq}$. Next, in Theorem 2, we show how we can solve efficiently the counting problem on separable systems with bounded projection-width.

Theorem 2 (Counting)**.** *Consider a separable system $(X, D, C^{\in}, C^{\geq})$, let $C := C^{\in} \cup C^{\geq}$, and let T be a branch decomposition of $X \cup C$ of projection-width w_{proj}. Then, the counting problem can be solved in time* (2)*.*

Proof. The proof follows the same structure as that of Theorem 1, and we only highlight the differences here.

Table. For a shape (Φ, Ψ), the content of the table Tab_v at this index should be the number of assignments $\tau : X_v \to D$ of shape (Φ, Ψ).

Leaves corresponding to constraints. Here, we set $\text{Tab}_l(\Phi, \Psi) := 1$ if ϵ has shape (Φ, Ψ), and $\text{Tab}_l(\Phi, \Psi) := 0$ otherwise.

Leaves corresponding to variables. Here, we set $\text{Tab}_l(\Phi, \Psi)$ to be the number of assignments that have shape (Φ, Ψ), among the $|D|$ that we constructed.

Inner vertices. Here, we first initialize $\text{Tab}_v(\Phi, \Psi) := 0$ for every shape (Φ, Ψ) for v. Then, for each triple (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) of linked shapes for v, v_1, v_2, we let $\nu_1 := \text{Tab}_{v_1}(\Phi_1, \Psi_1)$ and $\nu_2 := \text{Tab}_{v_2}(\Phi_2, \Psi_2)$, and add $\nu_1 \nu_2$ to $\text{Tab}_v(\Phi, \Psi)$ in time $O(1)$.

Root. From Remark 2, the content of Tab_r, indexed by $(\epsilon, C/\epsilon)$, is the number of assignments $\tau : X \to D$ that satisfy the constraints in C. ☐

4.3 Top-k

In this section, we consider the *top-k problem*. In this problem, we are given a separable system $(X, D, C^{\in}, C^{\geq})$ and $\nu_x : D \to \mathbb{R}$ for every $x \in X$. The value of an assignment is defined by (1), like for the optimization problem. The goal is to return a sorted list of k highest-value assignments from X to D that satisfy the constraints in C. More formally, the output should be a list of k assignments $\tau_1, \tau_2, \ldots, \tau_k$ from X to D satisfying the constraints in C and such that

$$\begin{aligned}&\nu(\tau_k) \leq \nu(\tau_{k-1}) \leq \cdots \leq \nu(\tau_1)\,,\\ &\nu(\tau') \leq \nu(\tau_k) \text{ for every other } \tau' : X \to D \text{ satisfying the constraints in } C,\end{aligned}$$

with the understanding that, in case there are only $h < k$ assignments from X to D satisfying the constraints in C, the output should be a sorted list of only those h assignments. Next, in Theorem 3, we show how we can solve efficiently the top-k problem on separable systems with bounded projection-width.

Theorem 3 (Top-k)**.** *Consider a separable system $(X, D, C^{\in}, C^{\geq})$, let $C := C^{\in} \cup C^{\geq}$, and let T be a branch decomposition of $X \cup C$ of projection-width w_{proj}. Let $\nu_x : D \to \mathbb{R}$ for every $x \in X$. Then, the top-k problem can be solved in time*

$$O\left({w_{\text{proj}}}^3\left(|X| + |C|\right)\left(|C| + k\log(k)\right) + w_{\text{proj}}\left|C^{\in}\right|\Lambda + |X|\,|C|\,|D|\log\left(|D|\right)\right).$$

Proof. The proof follows the same structure as that of Theorem 1, and we only highlight the differences here.

Table. For a shape (Φ, Ψ), the content of the table Tab_v at this index should be a sorted list of k highest-value assignments from X_v to D of shape (Φ, Ψ), with their respective values.

Leaves corresponding to constraints. Here, we set $\mathrm{Tab}_l(\Phi, \Psi) := (\epsilon, 0)$ if ϵ has shape (Φ, Ψ), and we set $\mathrm{Tab}_l(\Phi, \Psi)$ as an empty list otherwise.

Leaves corresponding to variables. Here, we set $\mathrm{Tab}_l(\Phi, \Psi)$ to be a sorted list of k-highest value assignments of shape (Φ, Ψ), with their respective values, among the $|D|$ that we constructed. The only extra step required, after constructing the $|D|$ assignments, is to order them according to their value, which can be done in time $O(|D| \log(|D|))$.

Inner vertices. Fix a triple (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) of linked shapes for v, v_1, v_2. It is well known that we can find k largest values, in sorted order, in the Cartesian sum of two sorted arrays in time $O(k \log(k))$, using a best-first search strategy with a max-heap, and that only k largest elements in each array need to be considered. We then construct the values of the k highest-value assignments, in sorted order, among all assignments of the form $\tau_1 \cup \tau_2$ with τ_1 of shape (Φ_1, Ψ_1) and τ_2 of shape (Φ_2, Ψ_2), by only considering those with τ_1 in $\mathrm{Tab}_{v_1}(\Phi_1, \Psi_1)$ and τ_2 in $\mathrm{Tab}_{v_2}(\Phi_2, \Psi_2)$. We call these k highest-value assignments a "candidate top-k for (Φ, Ψ)." Since we store the corresponding assignments implicitly, the total time for one triple is $O(k \log(k))$. This is done for each triple of linked shapes.

Now fix one shape (Φ, Ψ) for v, consider all candidate top-k for (Φ, Ψ), and denote by $N_{(\Phi,\Psi)} \leq {w_{\text{proj}}}^3$ their number. It then follows from Proposition 2 that we can set the content of $\mathrm{Tab}_v(\Phi, \Psi)$ by finding the k highest-value assignments, in sorted order, among all $N_{(\Phi,\Psi)}$ candidate top-k for (Φ, Ψ). It is well known that we can find k largest values, in sorted order, in N sorted arrays of k elements each in time $O(N + k\log(N))$, using a k-way merge with a max-heap. Since we store the assignments implicitly, the total time for the merge corresponding to (Φ, Ψ) is $O(N_{(\Phi,\Psi)} + k\log(N_{(\Phi,\Psi)}))$. Using $\sum_{(\Phi,\,\Psi)\text{ shape of } v} N_{(\Phi,\Psi)} \leq {w_{\text{proj}}}^3$, we obtain that the total time for the merges corresponding to the shapes of v is $O({w_{\text{proj}}}^3 k)$. Therefore, for each inner vertex v of T, the table Tab_v can be computed in time $O\left({w_{\text{proj}}}^3 |C| + {w_{\text{proj}}}^3 k \log(k)\right)$. Since there are $|X| + |C| - 1$ inner vertices of T, in total they require $O({w_{\text{proj}}}^3 (|X| + |C|)(|C| + k\log(k)))$ time.

Root. From Remark 2, the content of Tab_r, indexed by $(\epsilon, C/\epsilon)$, implicitly contains a sorted list of k highest-value assignments from X to D that satisfy the constraints in C. Following our pointers, we can construct them explicitly in time $O(k(|X| + |C|))$. □

4.4 Weighted constraint violation

In this section we define a problem that significantly extends the weighted MaxSAT problem. This problem is inherently defined on a separable system with $C^{\in} = \emptyset$. We remark that this problem can also be extended to general separable systems, but such an extension appears to offer little value and practical relevance. In the *weighted constraint violation problem,* we are given a separable system of the form $(X, D, \emptyset, C)$, and $\omega^c \in \mathbb{R}$ for every $c \in C$. For every assignment $\tau : X \to D$, we define its *weight*

$$\begin{aligned}\omega(\tau) &:= \sum_{c \in C} \omega^c \min\left\{\sum_{x \in X} g_x^c(\tau(x)), \gamma^c\right\} \\ &= \sum_{c \in C} \omega^c \min\left\{\text{cb}^c(\tau), \gamma^c\right\} \\ &= \sum_{c \in C} \omega^c (C/\tau)^c.\end{aligned}$$

The goal is to find a highest-weight assignment from X to D.

The name of the problem arises by considering the special case $\omega^c \geq 1$, where for all $c \in C$, we have $\omega(\tau) \leq \sum_{c \in C} \omega^c \gamma^c$, for every $\tau : X \to D$, and $\omega(\tau) = \sum_{c \in C} \omega^c \gamma^c$ if and only if τ satisfies all the constraints in C. Compared to the problems considered in Sections 4.1 and 4.2, the weighted constraint violation problem might seem more exotic. However, it contains as special cases MaxSAT and weighted MaxSAT, and it will allows us to show how our techniques can be used to significantly extend the known tractability of these problems for formulas with bounded PS-width [STV14]. To write the weighted MaxSAT problem as a separable constraint problem, it suffices to observe that each clause can be written as a constraint c with $D := \{0, 1\}$, $\gamma^c := 1$, and where ω^c is the (non-negative) weight of c in the weighted MaxSAT problem.

While the approach that we use to solve the weighted constraint violation problem is still based on the theory of shapes that we developed in Section 2, we will not be using the concept of *assignments of shape,* which we introduced in Section 3, and that played a key role in Sections 4.1 and 4.2. Instead, we rely on the "weaker" notion of *assignments of configuration,* which we define next.

Let T be a branch decomposition of $X \cup C$, let v be a vertex of T, and let (Φ, Ψ) be a shape for v. We say that $\tau : X_v \to D$ has *configuration* Φ if condition (S1$^{\geq}$) holds, that is,

$$\Phi = \overline{C_v}/\tau.$$

Furthermore, for $\tau : X_v \to D$ we define its Ψ*-weight*

$$\omega^{\Psi}(\tau) := \sum_{c \in C_v} \omega^c \min\left\{\text{cb}^c(\tau), \gamma^c - \Psi^c\right\} \in \mathbb{Z}.$$

In our algorithm, for the shape (Φ, Ψ), we will compute a highest-Ψ-weight assignment from X_v to D of configuration Φ. Such an assignment will be simple to compute in the leaves, and will yield the solution to the problem in the root, as discussed below.

Remark 3. *Consider a separable system of the form* $(X, D, \emptyset, C)$*, let* T *be a branch decomposition of* $X \cup C$*, and let* r *be the root of* T*. Consider the shape* $(\epsilon, C/\epsilon)$ *for* r *(see Remark 1 (iii) and (iv)), and note that* $X_r = X$*. Note that every assignment from* X *to* D *has configuration* ϵ*. Furthermore,* C/ϵ *is given explicitly by*

$$(C/\epsilon)^c = 0 \qquad \forall c \in C,$$

thus $\omega_r^{C/\epsilon}(\tau) = \omega(\tau)$ *for every assignment* τ *from* X *to* D*. Therefore, the set of highest-*(C/ϵ)*-weight assignments from* X *to* D *of configuration* ϵ *coincides with the set of highest-weight assignments from* X *to* D*.*

The next result shows how $\omega^{\Psi}(\tau_1 \cup \tau_2)$ can be easily computed from $\omega^{\Psi_1}(\tau_1)$ and $\omega^{\Psi_2}(\tau_2)$; It will be the key in computing the highest-Ψ-weight assignments, traversing T in a bottom up manner.

Lemma 8. *Consider a separable system of the form* $(X, D, \emptyset, C)$*, and let* $\omega^c \in \mathbb{R}$ *for every* $c \in C$*. Let* T *be a branch decomposition of* $X \cup C$*, and let* v *be an inner vertex of* T *with children* v_1 *and* v_2*. Let* (Φ, Ψ)*,* (Φ_1, Ψ_1)*,* (Φ_2, Ψ_2) *be linked shapes for* v, v_1, v_2*. Let* $\tau_1, \tau_1' : X_{v_1} \to D$ *of configuration* Φ_1 *and* $\tau_2, \tau_2' : X_{v_2} \to D$ *of configuration* Φ_2*. Then,*

$$\omega^{\Psi}(\tau_1 \cup \tau_2) = \omega^{\Psi_1}(\tau_1) + \omega^{\Psi_2}(\tau_2) + \sum_{c \in C_{v_1}} \omega^c \min\{\Phi_2^c, \gamma^c - \Psi^c\} + \sum_{c \in C_{v_2}} \omega^c \min\{\Phi_1^c, \gamma^c - \Psi^c\}.$$

Proof. It suffices to prove that, for every $c \in C_{v_1}$,

$$\min\{\mathrm{cb}^c(\tau_1 \cup \tau_2), \gamma^c - \Psi^c\} = \min\{\mathrm{cb}^c(\tau_1), \gamma^c - \Psi_1^c\} + \min\{\Phi_2^c, \gamma^c - \Psi^c\}. \tag{3}$$

In fact, from (3) and the symmetric identity for v_2, we obtain

$$\begin{aligned}
\omega^{\Psi}(\tau_1 \cup \tau_2) &= \sum_{c \in C_v} \omega^c \min\{\mathrm{cb}^c(\tau_1 \cup \tau_2), \gamma^c - \Psi^c\} \\
&= \sum_{c \in C_{v_1}} \omega^c \min\{\mathrm{cb}^c(\tau_1 \cup \tau_2), \gamma^c - \Psi^c\} + \sum_{c \in C_{v_2}} \omega^c \min\{\mathrm{cb}^c(\tau_1 \cup \tau_2), \gamma^c - \Psi^c\} \\
&= \sum_{c \in C_{v_1}} \omega^c \min\{\mathrm{cb}^c(\tau_1), \gamma^c - \Psi_1^c\} + \sum_{c \in C_{v_1}} \omega^c \min\{\Phi_2^c, \gamma^c - \Psi^c\} \\
&\quad + \sum_{c \in C_{v_2}} \omega^c \min\{\mathrm{cb}^c(\tau_2), \gamma^c - \Psi_2^c\} + \sum_{c \in C_{v_2}} \omega^c \min\{\Phi_1^c, \gamma^c - \Psi^c\} \\
&= \omega^{\Psi_1}(\tau_1) + \omega^{\Psi_2}(\tau_2) + \sum_{c \in C_{v_1}} \omega^c \min\{\Phi_2^c, \gamma^c - \Psi^c\} + \sum_{c \in C_{v_2}} \omega^c \min\{\Phi_1^c, \gamma^c - \Psi^c\}.
\end{aligned}$$

In the remainder of the proof, we show (3). First, we simplify the left hand side.

$$\begin{aligned}
\min\{\mathrm{cb}^c(\tau_1 \cup \tau_2), \gamma^c - \Psi^c\} &= \min\{\mathrm{cb}^c(\tau_1) + \mathrm{cb}^c(\tau_2), \gamma^c - \Psi^c\} && \text{(from Observation 1 (ii))} \\
&= \min\{\mathrm{cb}^c(\tau_1) + \min\{\mathrm{cb}^c(\tau_2), \gamma^c\}, \gamma^c - \Psi^c\} && \text{(from Lemma 1)} \\
&= \min\left\{\mathrm{cb}^c(\tau_1) + \left(\overline{C_{v_2}}/\tau_2\right)^c, \gamma^c - \Psi^c\right\} \\
&= \min\{\mathrm{cb}^c(\tau_1) + \Phi_2^c, \gamma^c - \Psi^c\} && (\tau_2 \text{ of configuration } \Phi_2).
\end{aligned}$$

Next, we rewrite the first minimum on the right hand side using (L2).

$$\min\{\mathrm{cb}^c(\tau_1), \gamma^c - \Psi_1^c\} = \min\{\mathrm{cb}^c(\tau_1), \gamma^c - \min\{\Psi^c + \Phi_2^c, \gamma^c\}\}.$$

We can then rewrite (3) as follows:

$$\min\left\{\mathrm{cb}^{c}\left(\tau_1\right)+\Phi_2^c,\gamma^c-\Psi^c\right\}-\min\left\{\mathrm{cb}^{c}\left(\tau_1\right),\gamma^c-\min\left\{\Psi^c+\Phi_2^c,\gamma^c\right\}\right\}=\min\left\{\Phi_2^c,\gamma^c-\Psi^c\right\}. \quad (4)$$

To check the identity (4), we first consider the case $\Phi_2^c \geq \gamma^c - \Psi^c$. In this case, the right hand side of (4) equals $\gamma^c - \Psi^c$, and the left hand side equals

$$\left(\gamma^c-\Psi^c\right)-\min\left\{\mathrm{cb}^{c}\left(\tau_1\right),\gamma^c-\gamma^c\right\}=\gamma^c-\Psi^c.$$

Next, we consider the case $\Phi_2^c < \gamma^c - \Psi^c$. In this case, the right hand side of (4) equals Φ_2^c and the left hand side equals

$$\min\left\{\mathrm{cb}^{c}\left(\tau_1\right)+\Phi_2^c,\gamma^c-\Psi^c\right\}-\min\left\{\mathrm{cb}^{c}\left(\tau_1\right),\gamma^c-\Psi^c-\Phi_2^c\right\}.$$

We add and subtract Φ_2^c, and bring $-\Phi_2^c$ inside the first minimum.

$$\Phi_2^c+\min\left\{\mathrm{cb}^{c}\left(\tau_1\right),\gamma^c-\Psi^c-\Phi_2^c\right\}-\min\left\{\mathrm{cb}^{c}\left(\tau_1\right),\gamma^c-\Psi^c-\Phi_2^c\right\}=\Phi_2^c.$$

□

The following result is a direct consequence of Lemma 8.

Lemma 9. *Consider a separable system of the form $(X, D, \emptyset, C)$, and let $\omega^c \in \mathbb{R}$ for every $c \in C$. Let T be a branch decomposition of $X \cup C$, and let v be an inner vertex of T with children v_1 and v_2. Let (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) be linked shapes for v, v_1, v_2. Let $\tau_1, \tau_1' : X_{v_1} \to D$ of configuration Φ_1 and $\tau_2, \tau_2' : X_{v_2} \to D$ of configuration Φ_2. If $\omega^{\Psi_1}(\tau_1') \leq \omega^{\Psi_1}(\tau_1)$ and $\omega^{\Psi_2}(\tau_2') \leq \omega^{\Psi_2}(\tau_2)$, then $\omega^{\Psi}(\tau_1' \cup \tau_2') \leq \omega^{\Psi}(\tau_1 \cup \tau_2)$.*

Proof. From Lemma 8,

$$\begin{aligned}
\omega^{\Psi}\left(\tau_1'\cup\tau_2'\right) &= \omega^{\Psi_1}\left(\tau_1'\right)+\omega^{\Psi_2}\left(\tau_2'\right)+\sum_{c\in C_{v_1}}\omega^c\min\left\{\Phi_2^c,\gamma^c-\Psi^c\right\}+\sum_{c\in C_{v_2}}\omega^c\min\left\{\Phi_1^c,\gamma^c-\Psi^c\right\}\\
&\leq \omega^{\Psi_1}\left(\tau_1\right)+\omega^{\Psi_2}\left(\tau_2\right)+\sum_{c\in C_{v_1}}\omega^c\min\left\{\Phi_2^c,\gamma^c-\Psi^c\right\}+\sum_{c\in C_{v_2}}\omega^c\min\left\{\Phi_1^c,\gamma^c-\Psi^c\right\}\\
&= \omega^{\Psi}\left(\tau_1\cup\tau_2\right).
\end{aligned}$$

□

We are now ready to present our algorithm for weighted constraint violation.

Theorem 4 (Weighted constraint violation)**.** *Consider a separable system of the form $(X, D, \emptyset, C)$, let $\omega^c \in \mathbb{R}$ for every $c \in C$, and let T be a branch decomposition of $X \cup C$ of projection-width w_{proj}. Then, the weighted constraint violation problem can be solved in time*

$$O\left({w_{\mathrm{proj}}}^3\left(|X|+|C|\right)|C|+|X|\,|C|\,|D|\log\left(|D|\right)\right).$$

Proof. First, we apply Proposition 1 and compute $\mathrm{proj}(v)$ and $\mathrm{proj}(\overline{v})$, for every vertex v of T, in time $O\left({w_{\mathrm{proj}}}^2\log(w_{\mathrm{proj}})\left(|X|+|C|\right)|C|+|X|\,|C|\,|D|\log\left(|D|\right)\right)$.

Table. Next, our algorithm will construct, for each vertex v of T, a table Tab_v indexed by the shapes (Φ, Ψ) for v. For a shape (Φ, Ψ), the content of the table Tab_v at this index, which we denote by $\mathrm{Tab}_v(\Phi, \Psi)$, should be a pair $(\tau, \omega^{\Psi}(\tau))$, where τ is a highest-Ψ-weight assignment from

X_v to D of configuration Φ. We now explain how we can compute the table Tab_v, for every vertex v of T. This is done in a bottom up manner.

Leaves corresponding to constraints. Consider a leaf l of T corresponding to a constraint $c \in C$. From Remark 1 (i), there is only one assignment from X_l to D, the empty assignment ϵ, that can be constructed in time $O(1)$. Clearly, there is only one $\Phi \in \mathrm{proj}(l)$, and it satisfies (S1$^{\geq}$). Hence, we set $\mathrm{Tab}_l(\Phi, \Psi) := (\epsilon, 0)$ for every shape (Φ, Ψ) for l. Therefore, we can compute the table Tab_l in time $O(w_{\mathrm{proj}})$. Since there are $|C|$ leaves corresponding to constraints in C, in total they require $O(w_{\mathrm{proj}} |C|)$ time.

Leaves corresponding to variables. Consider a leaf l of T corresponding to a variable $x \in X$. From Remark 1 (ii), there are $|D|$ assignment from X_l to D, and they can be constructed in time $O(|D|)$. We now fix one such assignment τ, and observe that there is precisely one $\Phi \in \mathrm{proj}(l)$ such that τ has configuration Φ, and it can be constructed in time $O(C)$. Once we have considered all $|D|$ assignments, for every shape (Φ, Ψ) for l, which are at most ${w_{\mathrm{proj}}}^2$, we set $\mathrm{Tab}_l(\Phi, \Psi) := (\tau, 0)$, where τ is any assignment of configuration Φ. Therefore, we can compute the table Tab_l in time $O\left(|C|\,|D| + {w_{\mathrm{proj}}}^2\right)$. Since there are $|X|$ leaves corresponding to variables, in total they require $O\left(|X|\,|C|\,|D| + {w_{\mathrm{proj}}}^2\,|X|\right)$ time.

Inner vertices. Consider now an inner vertex v of T, with children v_1, v_2. We loop over all triples (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) of linked shapes for v, v_1, v_2, as described in the proof of Theorem 1. For each triple of linked shapes, let $\left(\tau_1, \omega^{\Psi_1}(\tau_1)\right) := \mathrm{Tab}_{v_1}(\Phi_1, \Psi_1)$ and $\left(\tau_2, \omega^{\Psi_2}(\tau_2)\right) := \mathrm{Tab}_{v_2}(\Phi_2, \Psi_2)$. We then construct the Ψ-weight of the "candidate assignment for (Φ, Ψ)" given by $\tau_1 \cup \tau_2$, and this can be done in time $O(|C|)$ due to Lemma 8.

For each shape (Φ, Ψ) for v, we then set the highest-Ψ-weight assignment $\tau_1 \cup \tau_2$, among all candidate assignments for (Φ, Ψ), as the content of $\mathrm{Tab}_v(\Phi, \Psi)$, together with its Ψ-weight. To improve runtime, here we do not explicitly construct $\tau_1 \cup \tau_2$; instead, we store pointers to (Φ_1, Ψ_1) and (Φ_2, Ψ_2) giving the highest-Ψ-weight, so that this is done in time $O(1)$ instead of $O(|X|)$. Therefore, for each inner vertex v of T, the table Tab_v can be computed (partially implicitly) in time $O({w_{\mathrm{proj}}}^3\,|C|)$. Since there are $|X| + |C| - 1$ inner vertices of T, in total they require $O({w_{\mathrm{proj}}}^3\,(|X| + |C|)\,|C|)$ time.

We now show that we set Tab_v correctly, for every inner vertex v of T. Since we already proved it for the leaves of T, we now assume inductively that Tab_{v_1} and Tab_{v_2} have been set correctly, where v_1, v_2 are the children of v. Let (Φ, Ψ) be a shape for v, and let τ' be an assignment from X_v to D of configuration Φ. Let τ'_1 and τ'_2 denote the restrictions of τ' to X_{v_1} and X_{v_2}, respectively, and set $\Phi_1 := \overline{C_{v_1}}/\tau'_1$ and $\Phi_2 := \overline{C_{v_2}}/\tau'_2$. Consider now our procedure, when it considers the triple of linked shapes (Φ, Ψ), (Φ_1, Ψ_1), (Φ_2, Ψ_2) originated from Ψ, Φ_1, Φ_2. Note that, according to Lemma 2 and (L1), the linked shape (Φ, Ψ) that we just obtained is indeed the shape for v we started from. Let τ_1 and τ_2 be the assignments in $\mathrm{Tab}_{v_1}(\Phi_1, \Psi_1)$ and $\mathrm{Tab}_{v_2}(\Phi_2, \Psi_2)$, respectively. So $\tau_1 \cup \tau_2$ is a candidate assignment for (Φ, Ψ). By induction, $\omega^{\Psi_1}(\tau'_1) \leq \omega^{\Psi_1}(\tau_1)$ and $\omega^{\Psi_2}(\tau'_2) \leq \omega^{\Psi_2}(\tau_2)$. Lemma 9 then implies $\omega^{\Psi}(\tau') \leq \omega^{\Psi}(\tau)$. Now let $\tau'' : X_v \to D$ be the assignment in $\mathrm{Tab}_v(\Phi, \Psi)$. τ'' is a candidate assignment for (Φ, Ψ), so it has configuration Φ, due to Lemma 2 and (L1). Furthermore, by construction, we have $\omega^{\Psi}(\tau) \leq \omega^{\Psi}(\tau'')$, therefore $\omega^{\Psi}(\tau') \leq \omega^{\Psi}(\tau'')$.

Root. Once the table Tab_v is computed, for every vertex v of T, we consider the root r of T, and the shape $(\epsilon, C/\epsilon)$ for r (see Remark 1 (iii) and (iv)). From Remark 3, the table Tab_r, indexed by $(\epsilon, C/\epsilon)$, implicitly contains a highest-weight assignment from X to D. Following our pointers, we can construct it explicitly in time $O(|X| + |C|)$. □

5 Some consequences

In this section, we obtain some corollaries of our main theorems, and we discuss how our results subsume previously known tractability results in integer linear optimization, binary polynomial optimization, and Boolean satisfiability.

5.1 Main consequences

In Section 5.1.1, we specialize some of our main results to integer separable (nonlinear) optimization; in Section 5.1.2, to integer linear optimization; in Section 5.1.3, to binary polynomial optimization; and in Section 5.1.4, to Boolean satisfiability. We focus in particular on the optimization problems, emphasizing the consequences of Theorem 1. All the results that we obtain are new, except for Boolean satisfiability, where we recover precisely the tractability of weighted MaxSAT and #SAT for CNF formulas with bounded PS-width in [STV14].

5.1.1 Integer separable optimization

The integer separable optimization problem is the special case of the optimization problem considered in this work (Section 4.1), where the domain consists of the integer points in a bounded interval, and only inequality constraints are allowed. Formally, a *separable inequality system* is a quadruple $\left(X, D_I, C_S^{\leq}, C_S^{\geq}\right)$, where X is a finite set of *variables*, D_I is a finite *domain* set of the form $D_I = \{-D_{\max}, -D_{\max}+1, \ldots, D_{\max}\}$ for some $D_{\max} \in \mathbb{Z}_{\geq 0}$, and where $C_S^{\leq}$, $C_S^{\geq}$ are sets of *separable inequality constraints* of the form

$$\sum_{x \in X} f_x^c(x) \leq \delta^c \qquad c \in C_S^{\leq},$$
$$\sum_{x \in X} f_x^c(x) \geq \delta^c \qquad c \in C_S^{\geq},$$

where $f_x^c : D_I \to \mathbb{Z}$ for every $x \in X$ and $c \in C_S^{\leq} \cup C_S^{\geq}$, and where $\delta^c \in \mathbb{Z}$ for every $c \in C_S^{\leq} \cup C_S^{\geq}$. Clearly, a constraint in $C_S^{\geq}$ can also be expressed as a constraint in $C_S^{\leq}$, and vice versa. However, we choose to keep both types of constraints, as moving one inequality from one set to the other may affect the resulting projection-width of the system.

The definition of projection-width of a separable inequality system follows from our original definition for separable systems in Section 1.1, since every inequality constraint $c \in C_S^{\leq}$ can be written as a set constraint in $C^{\in}$ with $\Gamma^c = \{0, 1, \ldots, \gamma^c\}$, where as usual

$$\gamma^c := \delta^c - \sum_{x \in X} \min\{f_x^c(d) \mid d \in D_I\}.$$

In the *integer separable optimization problem,* we are given a separable inequality system $\left(X, D_I, C_S^{\leq}, C_S^{\geq}\right)$ and $\nu_x : D_I \to \mathbb{R}$ for every $x \in X$. For every assignment $\tau : X \to D$, we define its *value*

$$\nu(\tau) := \sum_{x \in X} \nu_x(\tau(x)).$$

The goal is to find a highest-value assignment from X to D_I satisfying the constraints in $C_S^{\leq} \cup C_S^{\geq}$, or prove that no such assignment exists.

Since $|D_I| = 2D_{\max} + 1$, our Theorem 1 directly implies the following result:

Corollary 1 (Integer separable optimization)**.** *Consider a separable inequality system* $\left(X, D_I, C_S^{\leq}, C_S^{\geq}\right)$*, let* $C_S := C_S^{\leq} \cup C_S^{\geq}$*, and let* T *be a branch decomposition of* $X \cup C_S$ *of projection-width* w_{proj}*. Let* $\nu_x : D \to \mathbb{R}$ *for every* $x \in X$*. Then, the integer separable optimization problem can be solved in time*

$$O\left({w_{\text{proj}}}^3 \left(|X| + |C_S|\right)|C_S| + |X|\,|C_S|\,D_{\max}\log\left(D_{\max}\right)\right). \tag{5}$$

5.1.2 Integer linear optimization

The integer linear optimization problem is the special case of integer separable optimization considered in Section 5.1.1, where all functions are linear.

Formally, a *linear inequality system* is a separable inequality system $\left(X, D_I, C_L^{\leq}, C_L^{\geq}\right)$, where for every $c \in C_L^{\leq} \cup C_L^{\geq}$ and $x \in X$, the function $f_x^c : D_I \to \mathbb{Z}$ is of the form

$$f_x^c(x) = a_x^c x,$$

for some $a_x^c \in \mathbb{Z}$. In the *integer linear optimization problem,* we are given a linear inequality system $\left(X, D_I, C_L^{\leq}, C_L^{\geq}\right)$ and $\nu_x \in \mathbb{R}$ for every $x \in X$. For every assignment $\tau : X \to D_I$, we define its *value*

$$\nu(\tau) := \sum_{x \in X} \nu_x \cdot (\tau(x)).$$

The goal is to find a highest-value assignment from X to D_I satisfying the constraints in $C_L^{\leq} \cup C_L^{\geq}$, or prove that no such assignment exists. Corollary 1 directly implies the following result:

Corollary 2 (Integer linear optimization)**.** *Consider a linear inequality system* $\left(X, D_I, C_L^{\leq}, C_L^{\geq}\right)$*, let* $C_L := C_L^{\leq} \cup C_L^{\geq}$*, and let* T *be a branch decomposition of* $X \cup C_L$ *of projection-width* w_{proj}*. Let* $\nu_x \in \mathbb{R}$ *for every* $x \in X$*. Then, the integer linear optimization problem can be solved in time*

$$O\left({w_{\text{proj}}}^3 \left(|X| + |C_L|\right)|C_L| + |X|\,|C_L|\,D_{\max}\log\left(D_{\max}\right)\right). \tag{6}$$

5.1.3 Binary polynomial optimization

Important applications of our results arise in *binary polynomial optimization*, an area that has recently seen significant progress; see, for example, [DPK17, DPK18b, DPK18a, DPK21, DPK24b]. In a *binary polynomial optimization problem*, we are given a hypergraph $H = (V, E)$, together with $\nu_v \in \mathbb{R}$ for every $v \in V$, and $\nu_e \in \mathbb{R}$ for every $e \in E$. Let $X_V := \{x_v \mid v \in V\}$ denote the set of variables. The goal is to find an assignment $\tau : X_V \to \{0, 1\}$ maximizing

$$\sum_{v \in V} \nu_v \tau(x_v) + \sum_{e \in E} \nu_e \prod_{v \in e} \tau(x_v).$$

The objective function above is, in general, not separable. However, it is well-known how the binary polynomial optimization problem can be reformulated as an integer linear optimization problem. To do so, we apply Fortet's linearization [For59, For60], introducing auxiliary variables $Y_E = \{y_e \mid e \in E\}$. For every assignment $\tau : X_V \cup Y_E \to \{0, 1\}$, we define its *value* by the linear function

$$\nu(\tau) := \sum_{v \in V} \nu_v \tau(x_v) + \sum_{e \in E} \nu_e \tau(y_e).$$

Consistency between the auxiliary variables and the original ones is then enforced through linear inequalities. Among several possible formulations, we adopt the *standard linearization,* though exploring alternative formulations could be an interesting direction in light of the results of this paper.

As we discussed in Section 5.1.1, each linear inequality can be placed in either $C_L^{\leq}$ or $C_L^{\geq}$, and this choice may affect the resulting projection-width. Here, we place all inequalities in $C_L^{\geq}$. This yields the linear inequality system $S_{\text{BPO}} = (X_V \cup Y_E, \{0,1\}, \emptyset, C_{\text{BPO}})$, where the constraints in C_{BPO} are given by

$$\begin{aligned} &(1 - y_e) + x_v \geq 1 && \forall v \in e,\ \forall e \in E, \\ &\sum_{v \in e}(1 - x_v) + y_e \geq 1 && \forall e \in E. \end{aligned}$$

The binary polynomial optimization problem is thus equivalent to the integer linear optimization problem of finding a maximum-value assignment from $X_V \cup Y_E$ to $\{0,1\}$ satisfying the constraints in C_{BPO}. We can then directly apply Corollary 2 to obtain the following tractability results for binary polynomial optimization, where we denote by $\text{size}(H) := \sum_{e \in E} |e|$ the size of the hypergraph H.

Corollary 3 (Binary polynomial optimization). *Consider a hypergraph $H = (V, E)$ and let T be a branch decomposition of $X_V \cup Y_E \cup C_{\text{BPO}}$ of projection-width w_{proj}. Let $\nu_v \in \mathbb{R}$ for every $v \in V$, and $\nu_e \in \mathbb{R}$ for every $e \in E$. Then, the binary polynomial optimization problem can be solved in time*

$$O\left({w_{\text{proj}}}^3 \left(|V| + \text{size}(H)\right) \text{size}(H)\right).$$

Theorem 1 also allows us to solve classes of *constrained binary polynomial optimization,* which is the extension of binary polynomial optimization obtained by considering only assignments satisfying some given additional separable constraints. [1]

Formally, let $\left(X_V \cup Y_E, \{0,1\}, C_c^{\in}, C_c^{\geq}\right)$ be a separable system representing these constraints. We then define the *constrained binary polynomial optimization problem* as the problem of finding a maximum-value assignment from $X_V \cup Y_E$ to $\{0,1\}$ satisfying the constraints in $C_c^{\in} \cup C_c^{\geq} \cup C_{\text{BPO}}$. At this point, we can again directly apply Theorem 1 and obtain the following result.

Corollary 4 (Constrained binary polynomial optimization). *Consider a hypergraph $H = (V, E)$, a separable system $\left(X_V \cup Y_E, \{0,1\}, C_c^{\in}, C_c^{\geq}\right)$, and let T be a branch decomposition of $X_V \cup Y_E \cup C_c^{\in} \cup C_c^{\geq} \cup C_{\text{BPO}}$ of projection-width w_{proj}. Then, the constrained binary polynomial optimization problem can be solved in time*

$$O\left({w_{\text{proj}}}^3 \left(|V| + \text{size}(H)\right) \text{size}(H)\right).$$

We remark that literals can be naturally incorporated into our framework, in both the unconstrained and constrained settings. The only modification required is that, in the constraints in C_{BPO}, some variables x_v may be replaced by $1 - x_v$ and vice versa. Consequently, all results in this section remain valid in this more general setting. We have excluded literals purely for notational simplicity. For additional background on binary polynomial optimization with literals, also known as *pseudo-Boolean optimization,* we refer the reader to [BH02, CDPDG24, DPK24a, DPK25].

Note that most known tractability results for binary polynomial optimization rely on structural properties of the underlying hypergraph H. In contrast, Corollaries 3 and 4 are of a different

[1] If the additional separable constraints are linear, it suffices to use Corollary 2 instead of Theorem 1.

nature, as they are based instead on properties of the constraint system. In Section 5.2.3, we present an example illustrating how these these results can be leveraged to derive tractability conditions expressed in terms of the hypergraph, thereby bringing them closer in spirit to existing results in the literature.

5.1.4 Boolean satisfiability

In this section, we show that our main results subsume those of [STV14] on the tractability of weighted MaxSAT and #SAT for CNF formulas with bounded PS-width. This influential result accounts for nearly all known tractable cases of these SAT problems, including formulas with bounded primal treewidth, bounded incidence treewidth, bounded signed incidence clique-width, bounded incidence clique-width, bounded MIM-width, as well as γ-acyclic formulas and disjoint-branches formulas [STV14, Cap16].

We begin by defining PS-width, where "PS" stands for *precisely satisfiable*. Our notation closely parallels that used for projection-width, highlighting the structural similarity between the two concepts. A *CNF formula* is a pair (X, F), where X is a set of variables and F is a set of clauses. Recall that a *clause* is the disjunction of literals, that is, variables or negation of variables of X. Given $X' \subseteq X$, $F' \subseteq F$, and $\tau : X' \to \{0, 1\}$, we define F'/τ, as the set of clauses in F' that are satisfied by τ. Given $X' \subseteq X$ and $F' \subseteq F$, we denote by

$$\text{proj}(F', X') = \{F'/\tau \mid \tau : X' \to \{0, 1\}\}.$$

Given a branch decomposition T of $X \cup F$ and a vertex v of T, we denote by T_v the subtree of T rooted in v, by F_v the set clauses of F such that the corresponding vertex appears in the leaves of T_v and by X_v the set of variables of F that similarly appear in the leaves of T_v. We denote by

$$\begin{aligned} \text{proj}(v) &:= \text{proj}\left(F \setminus F_v, X_v\right), \\ \text{proj}(\overline{v}) &:= \text{proj}\left(F_v, X \setminus X_v\right). \end{aligned}$$

If we denote by $V(T)$ the set of vertices of T, the *PS-width* of the formula (X, F) and T is defined by

$$\max_{v \in V(T)} \max\left(|\text{proj}(v)|, |\text{proj}(\overline{v})|\right).$$

The *PS-width* of a formula (X, F) is then defined as the minimum among the PS-widths of the formula (X, F) and T, over all branch decompositions T of $X \cup F$.

Observation 2. *Let (X, F) be a CNF formula. Then, there exists a separable system $(X, D, \emptyset, C_F)$ with $|C_F| = |F|$, and $D = \{0, 1\}$ such that an assignments $\tau : X \to \{0, 1\}$ satisfies the constraints in C_F if and only if it satisfies the clauses in F. Moreover, a branch decomposition of $X \cup F$ of PS-width w_{ps} yields a branch decomposition of $X \cup C_F$ of projection-width w_{ps}. Furthermore, we have $g_x^c(x) \in \{0, x, 1 - x\}$ for every $x \in X$, $c \in C_F$, and $\gamma^c = 1$ for every $c \in C_F$.*

Proof. For every clause in F, we write a constraint in C_F over variables X with domain $D = \{0, 1\}$ defined by

$$\sum_{x \in X^+} x + \sum_{x \in X^-} (1 - x) \geq 1,$$

where X^+ denotes the set of variables that appear in the positive literals in the clause, and X^- denotes the set of variables that appear in the negative literals in the clause. Clearly, an assignment

$\tau : X \to \{0,1\}$ satisfies the clause if and only if it satisfies the obtained constraint. It then suffices to show that a branch decomposition of $X \cup F$ of PS-width w_{ps} yields a branch decomposition of $X \cup C_F$ of projection-width w_{ps}. To see this, let $X' \subseteq X$, $F' \subseteq F$, $\tau : X' \to \{0,1\}$, and let C' be the set of constraints in C_F originating from the clauses in F'. Observe that $(C'/\tau)^c \in \{0,1\}$ for every $c \in C_F$. Furthermore, a clause $f \in F'$ is in the set F'/τ if and only if $(C'/\tau)^c = 1$, where c is the constraint in C' corresponding to f. □

Thanks to Observation 2, in the special case where $C^{\in} = \emptyset$, $D = \{0,1\}$, $g_x^c(x) \in \{0, x, 1-x\}$ for every $x \in X$, $c \in C_F$, and $\gamma^c = 1$ for every $c \in C_F$, Theorem 2 recovers the tractability of #SAT on CNF formulas with bounded PS-width as established in [STV14]. Similarly, Theorem 4 subsumes the tractability of weighted MaxSAT for CNF formulas with bounded PS-width in the same work. The running time we obtain for these two problems is comparable to that in [STV14], and is given by

$$O\left(w_{\text{ps}}^3 \left(|X| + |F|\right) |F|\right),$$

where w_{ps} denotes the PS-width of the given branch decomposition.

5.2 Consequences for incidence treewidth

In this section, we show that our main results imply the tractability of optimization and counting problems over separable systems whose incidence graph has bounded treewidth. Consider a separable system $S = \left(X, D, C^{\in}, C^{\geq}\right)$, and let $C := C^{\in} \cup C^{\geq}$. The *incidence graph* of S, denoted $G_{\text{inc}}(S)$, is the bipartite graph with vertex bipartition $X \cup C$, where an edge connects $x \in X$ and $c \in C$ if and only if $g_x^c \not\equiv 0$.

In the next result, we show that, given a tree decomposition of $G_{\text{inc}}(S)$ with treewidth bounded by a constant, one can efficiently construct a branch decomposition of $X \cup C$ whose projection-width is polynomially bounded.

Lemma 10 (Treewidth and projection-width)**.** *Consider a separable system $S = \left(X, D, C^{\in}, C^{\geq}\right)$, let $C := C^{\in} \cup C^{\geq}$, let $\gamma := \max\{\gamma^c \mid c \in C\}$, and let T be a tree decomposition of $G_{\text{inc}}(S)$ of treewidth w_{inc}. Then, in time $O(|X| + |C|)$, we can construct a branch decomposition T' of $X \cup C$ of projection-width at most*

$$\max\{|D|, \gamma + 1\}^{w_{\text{inc}}+1}.$$

Proof. Without loss of generality, we can assume that T is binary. In fact, for any vertex with more than two children, we can replace it with a binary tree of new vertices, each associated with the same bag, i.e., subset of $X \cup C$. This transformation does not increase the size of any bag and therefore does not increase the treewidth. We can then construct a branch decomposition T' of $X \cup C$ as follows. First, we add a vertex r as the father of the root of T. Then, for every $v \in X \cup C$, let t be the vertex of T closest to the root of T such that v appears in the bag of t, and we hang a leaf with label v on the edge between t and its father. We then remove the leaves that have no label. The resulting tree T' is binary and for every $v \in X \cup C$, there exists precisely one leaf of T' with label v. Therefore, T' is a branch decomposition of $X \cup C$. We claim that the projection-width of S and T' is at most the one in the statement.

First, let l be a leaf of T' corresponding to a constraint c in C. Since $X_l = \emptyset$, we have $|\text{proj}(l)| = 1$. Moreover, $C_l = \{c\}$ implies $\left|\text{proj}(\bar{l})\right| \leq \gamma^c + 1$. Next, let l be a leaf of T' corresponding to a variable x in X. Since $X_l = \{x\}$, we have $|\text{proj}(l)| \leq |D|$. Moreover, $C_l = \emptyset$ implies $\left|\text{proj}(\bar{l})\right| = 1$.

Next, let v be an inner vertex of T'. If v is also a vertex of T, let $t := v$. Otherwise, let t be the vertex of T in T'_v closest to v. We observe that, by construction, the label of every leaf of T'_v appears only in bags of T_t. Moreover, the label of every leaf of $T' \setminus T'_v$ either only appears in bags of $T \setminus T_t$, or it appears in the bag of t.

Let $x \in X_v$ and $c \in \overline{C_v}$ such that $g_x^c \not\equiv 0$. By the previous observation, x appears only in bags of T_t and c only appears in bags of $T \setminus T_t$ or in the bag of t. Since $g_x^c \not\equiv 0$, x and c must appear in a common bag, so c must appear in the bag of t. Therefore, the constraints $c \in \overline{C_v}$ with $g_x^c \not\equiv 0$ for some $x \in X_v$ all appear in the bag corresponding to t, so they are at most $w_{\text{inc}} + 1$. Hence, $|\text{proj}(v)| \leq (\gamma + 1)^{w_{\text{inc}}+1}$.

Let $c \in C_v$ and $x \in \overline{X_v}$ such that $g_x^c \not\equiv 0$. By the previous observation, c appears only in bags of T_t and x only appears in bags of $T \setminus T_t$ or in the bag of t. Since $g_x^c \not\equiv 0$, x and c must appear in a common bag, so x must appear in the bag of t. Therefore, the variables $x \in \overline{X_v}$ with $g_x^c \not\equiv 0$ for some $c \in C_v$ all appear in the bag corresponding to t, so they are at most $w_{\text{inc}} + 1$. Hence, $|\text{proj}(v)| \leq |D|^{w_{\text{inc}}+1}$.

Thus the projection-width of T' is at most $\max\{|D|, \gamma + 1\}^{w_{\text{inc}}+1}$. □

Lemma 10, together with our main results Theorems 1 to 4, implies the tractability of problems defined over separable systems S whose incidence graph $G_{\text{inc}}(S)$ has bounded treewidth. In particular, from Theorem 1 and Lemma 10 we obtain the following result.

Corollary 5 (Optimization – incidence treewidth)**.** *Consider a separable system* $S = \left(X, D, C^{\in}, C^{\geq}\right)$*, let* $C := C^{\in} \cup C^{\geq}$*, let* $\gamma := \max\{\gamma^c \mid c \in C\}$*, and let* T *be a tree decomposition of* $G_{\text{inc}}(S)$ *of treewidth* w_{inc}*. Let* $\nu_x : D \to \mathbb{R}$ *for every* $x \in X$*. Then, the optimization problem can be solved in time* (2)*, where* w_{proj} *is replaced by* $\max\{|D|, \gamma + 1\}^{w_{\text{inc}}+1}$.

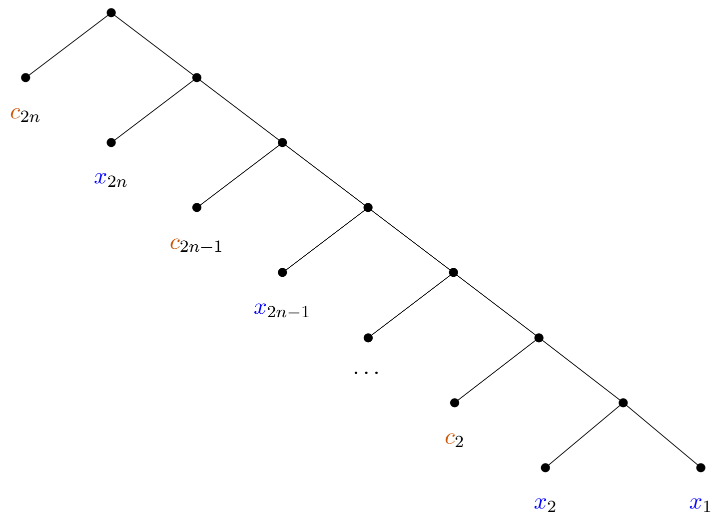
It is natural to ask whether one could design an algorithm with a running time similar to that in Corollary 5, but where γ is replaced by $\log(\gamma)$. However, this is unlikely, as it would yield a polynomial-time algorithm for the subset sum problem, or for the $0, 1$ knapsack problem, both of which are well-known to be weakly NP-complete. Indeed, such problems can be naturally expressed as optimization problems with $D = \{0, 1\}$ and $w_{\text{inc}} = 1$, since they involve only a single constraint.

It is worth emphasizing that the bounded incidence treewidth setting in Corollary 5 is substantially more restrictive than our main bounded projection-width setting in Theorem 1. Even within the more limited SAT framework discussed in Section 5.1.4, several important classes of formulas admit PS-width polynomially bounded, but incidence treewidth not bounded by a constant, including those with bounded signed incidence clique width, bounded incidence clique-width, bounded MIM-width, γ-acyclic formulas, and formulas with disjoint branches [STV14, Cap16]. In Observation 3 below, we present a family of linear inequality systems whose incidence treewidth is not bounded by a constant, while the projection-width remains polynomially bounded. For this family, Theorem 1 yields a polynomial-time algorithm, while Corollary 5 does not. To the best of our knowledge, also Corollary 5 is new. The main reason we wrote Corollary 5 is that it allows us to illustrate how our results subsume known tractability results in integer linear optimization (see Section 5.2.2) and binary polynomial optimization (see Section 5.2.3).

Observation 3. *There exists a family of linear inequality systems* S_L *with* $2n$ *variables such that the treewidth of* $G_{\text{inc}}(S_L)$ *is at least* n*, and the projection-width of* S_L *is at most* $2n$.

Proof. Consider the linear inequality system $S_L = \left(X, D_I, \emptyset, C_L^{\geq}\right)$, with variables $X = \{x_1, x_2, \ldots, x_{2n}\}$, domain $D_I = \{0, 1\}$, and linear inequality constraints $C_L^{\geq} = \{c_1, c_2, \ldots, c_{2n}\}$, where for $k \in$

Figure 1: Illustration of the branch decomposition T in the proof of Observation 3.



$\{1, 2, \ldots, 2n\}$, c_k is given by

$$\sum_{i=1}^{k} x_i \geq \gamma^{c_k},$$

where $\gamma^{c_k} \in \{0, 1, \ldots, 2n\}$. The subgraph of $G_{\text{inc}}(S_L)$ induced by variables $x_1, x_2, \ldots, x_n$ and constraints $c_{n+1}, c_{n+2}, \ldots, c_{2n}$ is complete bipartite, thus the treewidth of $G_{\text{inc}}(S_L)$ is at least n. Now let T be the linear branch decomposition of $X \cup C_L^{\geq}$ in Figure 1. It is simple to show that the projection-width of S_L and T is bounded by $2n$. □

We note that Corollary 5 immediately yields several classical tractability results. Given a hypergraph $H = (V, E)$, recall that the *incidence graph* of H, denoted $G_{\text{inc}}(H)$, is the bipartite graph with vertex bipartition $V \cup E$, where an edge connects $v \in V$ and $e \in E$ if and only if $v \in e$. Corollary 5 implies that, given a tree decomposition of the incidence graph $G_{\text{inc}}(H)$ of treewidth w_{inc}, the weighted set cover and weighted set packing problems on H can be solved in time $O(2^{3(w_{\text{inc}}+1)}(|E| + |V|)|V|)$, while the weighted hitting set and weighted independent set problems on H can be solved in time $O(2^{3(w_{\text{inc}}+1)}(|V| + |E|)|E|)$.

Finally, we note that Corollary 5 remains valid when the incidence treewidth is replaced by the primal treewidth. More precisely, the same result holds if, in its statement, the incidence graph of the separable system $G_{\text{inc}}(S)$ is replaced by the primal graph $G_{\text{pri}}(S)$. The *primal graph* of a separable system S is the graph $G_{\text{pri}}(S)$ with vertex set X, where two vertices $x, x' \in X$ are adjacent if and only if there exists a constraint $c \in C$ such that $g_x^c \not\equiv 0$ and $g_{x'}^c \not\equiv 0$. This follows from the well-known fact that any tree decomposition of $G_{\text{pri}}(S)$ of treewidth w_{pri} can be transformed into a tree decomposition of $G_{\text{inc}}(S)$ of treewidth at most $w_{\text{pri}} + 1$.

5.2.1 Integer separable optimization

From Corollary 1 and Lemma 10 we obtain the following result.

Corollary 6 (Integer separable optimization – incidence treewidth)**.** *Consider a separable inequality system* $S_S = \left(X, D_I, C_S^{\leq}, C_S^{\geq}\right)$, *let* $C_S := C_S^{\leq} \cup C_S^{\geq}$, *let* $\gamma := \max\{\gamma^c \mid c \in C_S\}$, *and let* T *be* *a tree decomposition of* $G_{\text{inc}}(S_S)$ *of treewidth* w_{inc}. *Let* $\nu_x : D \to \mathbb{R}$ *for every* $x \in X$. *Then, the integer separable optimization problem can be solved in time* (5), *where* w_{proj} *is replaced by* $\max\{|D_I|, \gamma+1\}^{w_{\text{inc}}+1}$.

5.2.2 Integer linear optimization

From Corollary 2 and Lemma 10 we obtain the result below.

Corollary 7 (Integer linear optimization – incidence treewidth)**.** *Consider a linear inequality system* $S_L = \left(X, D_I, C_L^{\leq}, C_L^{\geq}\right)$, *let* $C_L := C_L^{\leq} \cup C_L^{\geq}$, *let* $\gamma := \max\{\gamma^c \mid c \in C_L\}$, *and let* T *be a tree decomposition of* $G_{\text{inc}}(S_L)$ *of treewidth* w_{inc}. *Let* $\nu_x \in \mathbb{R}$ *for every* $x \in X$. *Then, the integer linear optimization problem can be solved in time* (6), *where* w_{proj} *is replaced by* $\max\{2D_{\max}+1, \gamma+1\}^{w_{\text{inc}}+1}$.

Bounding γ in Corollary 7, allows us to compare our result with the literature in integer linear optimization. We obtain the following result.

Corollary 8 (Integer linear optimization – incidence treewidth v2)**.** *Consider a linear inequality system* $S_L = \left(X, D_I, C_L^{\leq}, C_L^{\geq}\right)$, *let* $C_L := C_L^{\leq} \cup C_L^{\geq}$, *and let* T *be a tree decomposition of* $G_{\text{inc}}(S_L)$ *of treewidth* w_{inc}. *Let* $\nu_x \in \mathbb{R}$ *for every* $x \in X$. *Let* $A_{\max}$ *be the maximum absolute value of* a_x^c, *for* $x \in X$ *and* $c \in C_L$. *Then, the integer linear optimization problem can be solved in time* (6), *where* w_{proj} *is replaced by* $(2A_{\max}D_{\max}|X|+1)^{w_{\text{inc}}+1}$.

Proof. We apply Corollary 7. To bound $\gamma := \max\{\gamma^c \mid c \in C_L\}$, recall (see Section 1.1) that γ^c, for $c \in C_L$, is defined by:

$$\gamma^c := \delta^c - \sum_{x \in X} \min\left\{a_x^c d' \mid d' \in D_I\right\}.$$

As discussed in Section 1.1, we can assume $\gamma^c \geq 0$ for every $c \in C_L$. We can also assume $\delta^c \leq A_{\max}D_{\max}|X|$ for every $c \in C_L$. For $c \in C_L^{\leq}$, this is because otherwise all assignments from X to D_I satisfy the constraint c. For $c \in C_L^{\geq}$, this is because otherwise no assignment from X to D_I satisfies the constraint c. We then obtain $\gamma \leq 2A_{\max}D_{\max}|X|$. □

We observe that Corollary 8 recovers the tractability result for integer linear optimization established in [GOR17]. In fact, our running time is comparable to that of theorem 11 in [GOR17], which is

$$O((A_{\max}D_{\max}|X|)^{2w_{\text{inc}}+2}\, w_{\text{inc}}(|X|+|C_L|)).$$

5.2.3 Binary polynomial optimization

For a hypergraph H, it is shown in [CDPDG24] that a tree decomposition of $G_{\text{inc}}(H)$ of treewidth w_{inc} yields a tree decomposition of $G_{\text{inc}}(S_{\text{BPO}})$ of treewidth at most $2(w_{\text{inc}}+1)$. Combining this result with Corollary 3 and Lemma 10 gives the following.

Corollary 9 (Binary polynomial optimization – incidence treewidth)**.** *Consider a hypergraph* $H = (V, E)$ *and let* T *be a tree decomposition of* $G_{\text{inc}}(H)$ *of treewidth* w_{inc}. *Let* $\nu_v \in \mathbb{R}$ *for every* $v \in V$, *and* $\nu_e \in \mathbb{R}$ *for every* $e \in E$. *Then, the binary polynomial optimization problem can be solved in time*

$$O\left(2^{6w_{\text{inc}}+9}\left(|V| + \text{size}(H)\right)\text{size}(H)\right),$$

Corollary 9 recovers the tractability result in [CDPDG24] for binary polynomial optimization over hypergraphs with bounded incidence treewidth. Unlike the proof in [CDPDG24], ours does not rely on knowledge compilation.

For constrained binary polynomial optimization, we can employ Corollary 5 and Lemma 10 to obtain the following tractability result.

Corollary 10 (Constrained binary polynomial optimization – incidence treewidth)**.** *Consider a hypergraph $H = (V, E)$ and let T be a tree decomposition of $G_{\text{inc}}(H)$ of treewidth w_{inc}. Consider a separable system $S_c = \left(X_V \cup Y_E, \{0,1\}, C_c^{\in}, C_c^{\geq}\right)$ and let W be a vertex cover of $G_{\text{inc}}(S_c)$ of cardinality κ. Let $\nu_v \in \mathbb{R}$ for every $v \in V$, and $\nu_e \in \mathbb{R}$ for every $e \in E$. Then, the constrained binary polynomial optimization problem can be solved in time*

$$O\left(2^{6w_{\text{inc}}+9+3\kappa}\left(|V| + \text{size}(H)\right)\text{size}(H)\right).$$

Proof. From the proof of lemma 3 in [CDPDG24], a tree decomposition of $G_{\text{inc}}(H)$ of treewidth w_{inc} yields a tree decomposition of $G_{\text{inc}}(S_{\text{BPO}})$ of treewidth at most $2(w_{\text{inc}} + 1)$. Now denote by S' the separable system obtained by putting together the separable systems S_{BPO} and S_c, that is, $S' = \left(X_V \cup Y_E, \{0,1\}, C_c^{\in}, C_{\text{BPO}} \cup C_c^{\geq}\right)$. Note that $G_{\text{inc}}(S')$ is obtained by adding isolated vertices to $G_{\text{inc}}(S_{\text{BPO}})$ and $G_{\text{inc}}(S_c)$, and then taking their union. Clearly, adding isolated vertices does not increase the treewidth or the cardinality of a vertex cover. It then follows from lemma 3 in [ALQ+24], that $G_{\text{inc}}(S')$ has treewidth at most $2(w_{\text{inc}} + 1) + \kappa$. The result then follows from Corollary 5. □

Corollary 10 is just one illustration of how Corollary 5 can be applied to constrained binary polynomial optimization. Other results of the same type can be obtained by replacing lemma 3 in [ALQ+24] with analogous bounds on the incidence treewidth of the combined system. In particular, Corollary 10 is new and substantially extends the recent result of [CDPDG24], which establishes tractability of binary polynomial optimization with a single "extended cardinality constraint." More general tractability statements can be obtained by applying directly Theorem 1 instead of Corollary 5.

Funding. Alberto Del Pia is partially funded by AFOSR grant FA9550-23-1-0433 and ONR grant N00014-25-1-2490. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Air Force Office of Scientific Research or of the Office of Naval Research.

References

[ALQ+24] Bogdan Alecu, Vadim V. Lozin, Daniel A. Quiroz, Roman Rabinovich, Igor Razgon, and Viktor Zamaraev. The treewidth and pathwidth of graph unions. *SIAM Journal on Discrete Mathematics*, 38(1), 2024.

[BH02] Endre Boros and Peter L. Hammer. Pseudo-Boolean optimization. *Discrete applied mathematics*, 123(1):155–225, 2002.

[Cap16] Florent Capelli. *Structural restrictions of CNF-formulas: applications to model counting and knowledge compilation.* PhD thesis, Université Paris Diderot, Sorbonne Paris Cité, 2016.

[CCZ14] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming.* Springer, 2014.

[CDPDG24] Florent Capelli, Alberto Del Pia, and Silvia Di Gregorio. A knowledge compilation take on binary polynomial optimization. *arXiv:2311.00149*, 2024.

[DPK17] Alberto Del Pia and Aida Khajavirad. A polyhedral study of binary polynomial programs. *Mathematics of Operations Research*, 42(2):389–410, 2017.

[DPK18a] Alberto Del Pia and Aida Khajavirad. The multilinear polytope for acyclic hypergraphs. *SIAM Journal on Optimization*, 28(2):1049–1076, 2018.

[DPK18b] Alberto Del Pia and Aida Khajavirad. On decomposability of multilinear sets. *Mathematical Programming, Series A*, 170(2):387–415, 2018.

[DPK21] Alberto Del Pia and Aida Khajavirad. The running intersection relaxation of the multilinear polytope. *Mathematics of Operations Research*, 46(3):1008–1037, 2021.

[DPK24a] Alberto Del Pia and Aida Khajavirad. Beyond hypergraph acyclicity: limits of tractability for pseudo-boolean optimization. *arXiv:2410.23045*, 2024.

[DPK24b] Alberto Del Pia and Aida Khajavirad. A polynomial-size extended formulation for the multilinear polytope of beta-acyclic hypergraphs. *Mathematical Programming, Series A*, 207:269–301, 2024.

[DPK25] Alberto Del Pia and Aida Khajavirad. The pseudo-boolean polytope and polynomial-size extended formulations for binary polynomial optimization. *Mathematical Programming, Series A*, 2025.

[For59] R. Fortet. L'algèbre de boole et ses applications en recherche opérationnelle. *Cahiers du centre d'études de recherche opérationnelle*, 1:5–36, 1959.

[For60] R. Fortet. Applications de l'algèbre de boole en recherche opérationelle. *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, recherche opérationnelle. Recherche opérationnelle*, 4:17–26, 1960.

[GOR17] Robert Ganian, Sebastian Ordyniak, and M.S. Ramanujan. Going beyond primal treewidth for (M)ILP. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 815–821, 1917.

[MRC90] R. Kipp Martin, Ronald L. Rardin, and Brian A. Campbell. Polyhedral characterization of discrete dynamic programming. *Operations Research*, 38(1):127–138, 1990.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.

[STV14] Sigve Hortemo Sæther, Jan Arne Telle, and Martin Vatshelle. Solving MaxSAT and #SAT on structured CNF formulas. In *Theory and Applications of Satisfiability Testing*, pages 16–31, 2014.