The Complexity of Stackelberg Pricing Games

Christoph Grüne
Department of Computer Science, RWTH Aachen University, Germany

Dorothee Henke
University of Passau, Germany

Eva Rotenberg
IT University of Copenhagen, Denmark

Lasse Wulf
IT University of Copenhagen, Denmark

Abstract

We consider Stackelberg pricing games, which are also known as bilevel pricing problems, or combinatorial price-setting problems. This family of problems consists of games between two players: the leader and the follower. There is a market that is partitioned into two parts: the part of the leader and the part of the leader's competitors. The leader controls one part of the market and can freely set the prices for products. By contrast, the prices of the competitors' products are fixed and known in advance. The follower, then, needs to solve a combinatorial optimization problem in order to satisfy their own demands, while comparing the leader's offers to the offers of the competitors. Therefore, the leader has to hit the intricate balance of making an attractive offer to the follower, while at the same time ensuring that their own profit is maximized.

Pferschy, Nicosia, Pacifici, and Schauer considered the Stackelberg pricing game where the follower solves a knapsack problem. They raised the question whether this problem is complete for the second level of the polynomial hierarchy, i.e., Σ_2^p-complete. The same conjecture was also made by Böhnlein, Schaudt, and Schauer. In this paper, we positively settle this conjecture. Moreover, we show that this result holds actually in a much broader context: The Stackelberg pricing game is Σ_2^p-complete for over 50 underlying problems whose decision versions are NP-complete, including most classics such as TSP, vertex cover, clique, subset sum, etc. This result falls in line of recent meta-theorems about higher complexity in the polynomial hierarchy by Grüne and Wulf.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Theory of computation → Computational pricing and auctions; Theory of computation → Representations of games and their complexity; Theory of computation → Complexity classes; Theory of computation → Problems, reductions and completeness

Keywords and phrases Bilevel pricing problems, Stackelberg games, bilevel optimization, polynomial hierarchy

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.141

Related Version *Full Version*: `https://arxiv.org/abs/2511.05700`

Funding Supported by the Carlsberg Foundation CF21-0302 "Graph Algorithms with Geometric Applications" and the VILLUM Foundation grant (VIL37507) "Efficient Recomputations for Changeful Problems".
Christoph Grüne: Funded by the German Research Foundation (DFG) – GRK 2236/2.

1 Introduction

Stackelberg pricing games are economic games between two players, usually called the leader and the follower. In these games, we assume that the leader owns one part of the relevant market and may set prices of products to their liking. The rest of the market belongs to competitors of the leader and has fixed prices, which we assume to be known in advance.

After the leader has set the prices, the follower solves a combinatorial optimization problem, freely combining products of the leader and the competitors. The leader therefore has to balance both players' interests carefully. On the one hand, if the leader sets the prices too low, the leader will not make a large profit. On the other hand, if they set the prices too high, the follower will simply switch to the competitors, and the leader will not receive any profit. Stackelberg pricing games were introduced by Labbé, Marcotte and Savard, in order to model optimal highway road tolls [15]. In this context, the follower solves a shortest-path problem. Early works have studied NP-hardness, approximation algorithms, hardness of approximation, or multiple followers for the highway toll pricing problem [3, 4, 6, 14, 21]. Later works have extended this to many other settings where the follower solves different combinatorial optimization problems. This includes the case of minimum spanning tree (Cardinal et al. [7], Labbé et al. [17]), knapsack (Briest et al. [2], Pferschy et al. [20], Bui et al. [5]), set cover (Briest et al. [2], Bui et al. [5]), independent set (Böhnlein et al. [1], Bui et al. [5]), or the bipartite vertex cover problem (Briest at al. [3]). Pferschy et al. consider a variant of the knapsack pricing problem [19]. A general overview of network pricing problems can be found in the articles of van Hoesel [23] and of Labbé and Violin [16]. While the follower's problem in the original toll pricing problem of Labbé et al. is the shortest path problem, which can be solved efficiently, we remark that this is a serious restriction of our modeling capacities. Indeed, it might be argued that, in the real world, the vast majority of optimization problems that one encounters are NP-hard. As such, in this paper we restrict our attention to such pricing games where the follower's problem already is NP-hard, vastly enriching our modeling capabilities. Several of the above mentioned papers have already studied this setting.

Example. As one possible example of a Stackelberg pricing game with an NP-hard follower's problem, let us mention the *Stackelberg knapsack game with profit selection* as defined by Pferschy, Nicosia, Pacifici, and Schauer [20]. We are given a set $\mathcal{U}$ of items and a partition $\mathcal{U} = \mathcal{U}_L \cup \mathcal{U}_F$ into leader's and follower's items (with $\mathcal{U}_L \cap \mathcal{U}_F = \emptyset$). Just like in the standard knapsack problem, each item $e \in \mathcal{U}$ has a weight $w(e)$. Together with some given weight threshold $W \geq 0$, this determines the set of *feasible knapsack solutions* by $\mathcal{F} = \{X \subseteq \mathcal{U}_L \cup \mathcal{U}_F : \sum_{e \in X} w(e) \leq W\}$. Each item has a profit, specified separately for the leader by $p_L : \mathcal{U}_L \to \mathbb{Z}_{\geq 0}$ and for the follower by $p_F : \mathcal{U}_F \to \mathbb{Z}_{\geq 0}$. The idea is now that, for all items $e \in \mathcal{U}_L$, the leader receives their profit $p_L(e)$ only if the follower actually buys item e. In order to achieve this, the leader is allowed to "tempt" the follower, by offering so-called *incentives* $i(e)$ for each $e \in \mathcal{U}_L$ to the follower. The leader then wants to solve

$$\begin{aligned} &\max_{i\,:\,\mathcal{U}_L \to \mathbb{R}} \ p_L(X^\star) - i(X^\star) \\ &\qquad \text{s.t. } X^\star \in \arg\max_{X \in \mathcal{F}} \ i(X) + p_F(X). \end{aligned} \tag{1}$$

(For the sake of readability, we write $p_L(X^\star)$, $i(X^\star)$, and $p_F(X)$ instead of $p_L(X^\star \cap \mathcal{U}_L)$, $i(X^\star \cap \mathcal{U}_L)$, and $p_F(X \cap \mathcal{U}_F)$.) The leader therefore has to consider the tradeoff between offering the follower good incentives $i(e)$ and still receiving a large payout $p_L(X^\star) - i(X^\star)$. Even though we described this problem from the viewpoint of giving incentives, we will later see in Section 3 that an easy reformulation can equivalently frame the problem in terms of pricing. Therefore, as has also been observed by past researchers [1, 5], problem (1) fits into the framework of combinatorial pricing games.

The polynomial hierarchy. Many problems in bilevel optimization are expected to be even harder than NP-hard – they are naturally complete for the second level of the *polynomial hierarchy*, the so-called complexity class Σ_2^p. If a problem is Σ_2^p-complete, then, under complexity-theoretic assumptions, it cannot be expressed as a mixed-integer linear formulation with only polynomially many variables and constraints (unless the polynomial hierarchy collapses). Therefore, the study and understanding of Σ_2^p-complete problems is important. We refer to Woeginger [24] for an excellent introduction to Σ_2^p-completeness in bilevel optimization. Past researchers have recognized the potential relevance of the complexity class Σ_2^p in the context of bilevel pricing problems. Indeed, Pferschy, Nicosia, Pacifici, and Schauer [20] considered the knapsack pricing problem (1). They were able to show NP-hardness as well as several other interesting approximability and inapproximability results for different variants of the follower's behavior. They could show that the variant of problem (1) where the leader can only choose between two different incentives per item is Σ_2^p-hard. However, in the general case, they could not resolve the complexity status and stated it as an open question whether the problem is Σ_2^p-complete. Later, Böhnlein, Schaudt, and Schauer [1] considered the combinatorial pricing problem for multiple underlying problems. They obtained a polynomial-time algorithm for interval scheduling pricing and an inapproximability result for independent set pricing. Furthermore, they showed that there exists some artificial, hand-crafted problem Π such that Π is in NP and the pricing problem for Π is Σ_2^p-complete. Naturally, they remarked that such a result would be much more interesting for a more natural problem, and conjectured that both the knapsack pricing problem, and the set packing pricing problem (where set packing is a generalization of independent set) are Σ_2^p-complete.

Our contribution. In this paper, we positively settle the open question and conjecture from [1, 20] by proving that the knapsack pricing problem (1) and the independent set pricing problem are Σ_2^p-complete. Moreover, we are able to extend our techniques beyond a proof of the initial conjecture in order to prove a much more general result: We show that the combinatorial pricing problem is in fact Σ_2^p-complete for many underlying optimization problems whose decision versions are NP-complete. This includes *every of the following problems*:

> knapsack, independent set, clique, vertex cover, set cover, hitting set, feedback vertex set, dominating set, feedback arc set, uncapacitated facility location, p-center, p-median, TSP, Steiner tree, 3-dimensional matching, two-disjoint directed path, satisfiability, subset sum, Hamiltonian cycle,

as well as many more problems (but at least 50) on an ever growing list of so-called *SSP-NP-complete* problems. This is a subclass of NP-complete problems introduced by Grüne and Wulf [9, 12]. The class contains most classically known NP-complete problems, and is defined as all NP-complete problems that satisfy a certain extra condition, more precisely explained in Section 5. This extra condition is on the one hand general enough so that it seems to be satisfied by most "natural" NP-complete problems, on the other hand it is restrictive enough to enable broad meta-theorems about Σ_2^p-completeness. In fact, our main theorem states that, for every SSP-NP-complete problem Π, its pricing variant is Σ_2^p-complete. Our result complements recent other meta-theorems about completeness in the polynomial hierarchy of multi-level optimization problems, regarding min-cost interdiction [9], min-max regret [9], unit cost interdiction [11], and recoverable optimization [10].

We remark that, for each individual problem from the list, it is not surprising that the pricing variant is Σ_2^p-complete. However, we argue that the main insight that our paper offers is that all these pricing problems are Σ_2^p-complete "for the same reason". We further remark that our results are not a trivial consequence of [9]. Instead, we first need to show Σ_2^p-completeness of the pricing variant of (an optimization version of) Sat, which requires a carefully chosen instance and already partially answers the open question of [1]. Additionally, due to the complicated interaction between the leader and the follower, where the leader could choose any price function from a potentially unbounded union of polyhedra, both the Σ_2^p-containment proof (Theorem 2) as well as the hardness proof (Section 7) are significantly more involved than in [9].

Outline. We quickly sketch the remainder of the paper, and our general proof strategy to obtain our meta-theorem. In Section 3, we define the pricing problem, and in Section 4, we show that it is contained in the class Σ_2^p. In Section 5, we summarize the main ideas of the SSP framework of Grüne and Wulf. In Section 6, we first consider the pricing problem for only $\Pi =$ Satisfiability (Sat) and show its Σ_2^p-completeness. Finally, in Section 7, we show how the Σ_2^p-completeness of the Sat pricing problem extends to *all* other problems in the class SSP-NPc. This results in our main theorems, Theorems 7, 10, and 11.

2 Preliminaries

A *language* is a set $L \subseteq \{0,1\}^*$. A language L is contained in Σ_2^p if and only if there exists some polynomial-time computable function V (also called *verifier*) such that, for all $w \in \{0,1\}^*$ and suitable $m_1, m_2 = |w|^{O(1)}$,

$$w \in L \;\Leftrightarrow\; \exists y_1 \in \{0,1\}^{m_1} \;\forall y_2 \in \{0,1\}^{m_2} : V(w, y_1, y_2) = 1.$$

An introduction to the polynomial hierarchy and the class Σ_2^p can be found in the book by Papadimitriou [18] or in the article by Jeroslow [13]. An introduction specifically in the context of bilevel optimization can be found in the article of Woeginger [24].

A *many-one reduction* or *Karp reduction* from a language L to a language L' is a map $f : \{0,1\}^* \to \{0,1\}^*$ such that $w \in L$ if and only if $f(w) \in L'$ for all $w \in \{0,1\}^*$. A language L is Σ_2^p-hard if every $L' \in \Sigma_2^p$ can be reduced to L with a polynomial-time many-one reduction. If L is both Σ_2^p-hard and contained in Σ_2^p, it is Σ_2^p-complete.

A *Boolean variable* x is a variable that takes one of the values 0 or 1. Let $X = \{x_1, \ldots, x_n\}$ be a set of variables. The corresponding *literal set* is given by $L = \{x_1, \ldots, x_n\} \cup \{\overline{x}_1, \ldots, \overline{x}_n\}$. A *clause* is a disjunction of literals. A Boolean formula is in *conjunctive normal form (CNF)* if it is a conjunction of clauses. It is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals. In this paper, we use the notation where a clause is represented by a subset of the literals, and a CNF formula φ is represented by a set of clauses. For example, the set $\{\{x_1, \overline{x}_2\}, \{x_2, x_3\}\}$ corresponds to the formula $(x_1 \vee \overline{x}_2) \wedge (x_2 \vee x_3)$. We write $\varphi(X)$ to indicate that formula φ depends only on X. Sometimes we are interested in cases where the variables are partitioned, we indicate this case by writing $\varphi(X, Y, \ldots)$. An *assignment* of variables is a map $\alpha : X \to \{0,1\}$. The evaluation of the formula φ under assignment α is denoted by $\varphi(\alpha) \in \{0,1\}$. Given a formula $\varphi(X, Y)$ with partitioned variables and a partial assignment α on X, we also denote the remaining formula by $\varphi(\alpha, Y)$.

For some cost function $c : U \to \mathbb{R}$ and some subset $U' \subseteq U$, we define the cost of the subset U' as $c(U') := \sum_{u \in U'} c(u)$. For a map $f : A \to B$ and some subset $A' \subseteq A$, we define the image of the subset A' as $f(A') = \{f(a) : a \in A'\}$. The preimage of $B' \subseteq B$ is $f^{-1}(B') = \{a \in A : f(a) \in B'\}$. We write $f^{-1}(b)$ instead of $f^{-1}(\{b\})$ for $b \in B$.

3 Problem Definition

We formally define the pricing problem. Since we later wish to argue about the properties of all pricing problems simultaneously, our definition necessarily has to be somewhat technical. Every pricing problem is based on an *underlying* optimization problem. We assume in this paper that the underlying problem is a combinatorial optimization problem with a linear objective function, and that it has the following structural properties.

▶ **Definition 1** (Linear Optimization Problem, from [9]). *A linear optimization problem (or short LOP problem)* Π *is a tuple* $(\mathcal{I}, \mathcal{U}, \mathcal{F}, w, t)$ *such that*

- $\mathcal{I} \subseteq \{0,1\}^*$ *is a language. We call* $\mathcal{I}$ *the set of instances of* Π.
- *To each instance* $I \in \mathcal{I}$*, there is some*
  - *set* $\mathcal{U}(I)$ *called the universe associated to the instance* I.
  - *set* $\mathcal{F}(I) \subseteq 2^{\mathcal{U}(I)}$ *called the feasible solution set associated to instance* I.
  - *function* $w^{(I)} : \mathcal{U}(I) \to \mathbb{Z}$ *mapping each universe element to its weight.*
  - *threshold* $t^{(I)} \in \mathbb{Z}$.

For $I \in \mathcal{I}$, we define the solution set $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : w^{(I)}(S) \leq t^{(I)}\}$ as the set of feasible solutions below the weight threshold. The instance I is a yes-instance if and only if $\mathcal{S}(I) \neq \emptyset$. We assume that the universe size $|\mathcal{U}(I)|$ as well as the bit-complexity of $w^{(I)}$ and $t^{(I)}$ are polynomial in the input size $|I|$ and that it can be checked in polynomial time in $|I|$ whether some proposed set $F \subseteq \mathcal{U}(I)$ is feasible. The LOP problem is called a *minimization LOP problem* if $w^{(I)}, t^{(I)} \geq 0$ for all instances $I \in \mathcal{I}$. It is called a *maximization LOP problem* if $w^{(I)}, t^{(I)} \leq 0$ for all instances $I \in \mathcal{I}$. The following is an example of a minimization LOP problem (in this example, the function $w^{(I)}$ is always the unit weight function).

Vertex Cover
Instances: Graph $G = (V, E)$, number $k \in \mathbb{N}$.
Universe: Vertex set $V =: \mathcal{U}$.
Feasible solution set: The set of all vertex covers of G.
Solution set: The set of all vertex covers of G of size at most k.

How to turn any problem into a pricing problem. As explained, we want to consider the Vertex Cover problem, and define its pricing variant Pricing-Vertex Cover. Our goal is to make this definition in such a way that it naturally and seamlessly generalizes to all other problems from the class LOP, and such that the resulting pricing problem can again be considered a natural problem. However, there are some technical details that cannot be avoided when making such a definition. This is because trying to find a definition that fits all problems at once naturally arises some complications. If some team of researchers studies a single pricing problem, they can adapt to the complications tailored to this specific problem, but here we do not possess this luxury. In the full version of this paper, we informally explain these complications and which choices we make in our modeling in order to overcome them.

Formal definition. We are now ready to give the definition of the pricing problem. Our problem definition is slightly different for minimization and maximization problems. First, let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{F}, w, t)$ be a maximization LOP problem as defined earlier. We now formally define the problem Pricing-Π. The input is a tuple $(I, \mathcal{U}_L, \mathcal{U}_F, p_L, p_F)$ such that $I \in \mathcal{I}$, $\mathcal{U}(I) = \mathcal{U}_L \cup \mathcal{U}_F$, $\mathcal{U}_L \cap \mathcal{U}_F = \emptyset$, $p_L : \mathcal{U}_L \to \mathbb{Z}_{\geq 0}$, and $p_F : \mathcal{U}_F \to \mathbb{Z}_{\geq 0}$. The problem is to compute the value of

$$\begin{aligned} \max_{i : \mathcal{U}_L \to \mathbb{R}} \; & p_L(X^\star) - i(X^\star) \\ \text{s.t. } & X^\star \in \arg\max_{X \in \mathcal{F}(I)} \; i(X) + p_F(X). \end{aligned} \tag{2}$$

Since we are concerned with computational complexity, we want to treat the problem as a decision problem. We therefore assume that we are given an additional threshold $T_{\mathrm{pr}} \in \mathbb{Q}_{\geq 0}$, and the question is whether the value of (2) is at least T_{pr}. The problem Pricing-Π is then the set of all such decision problems for all possible tuples $(I, \mathcal{U}_L, \mathcal{U}_F, p_L, p_F)$ with properties as specified above, and all possible thresholds $T_{\mathrm{pr}} \in \mathbb{Q}_{\geq 0}$.

As it is often done in bilevel optimization, we follow the optimistic assumption: If the follower can choose between different solutions X that have the same maximal follower's objective value $i(X) + p_F(X)$, we assume optimistically that they will select one with the best objective value $p_L(X) - i(X)$ for the leader.

In the remainder of the paper, it will be more convenient to instead consider an equivalent reformulation of (2). Let us define for all items $e \in \mathcal{U}_L$ the term $d(e) := p_L(e) - i(e)$, and let $p(e) := p_L(e)$ for $e \in \mathcal{U}_L$ and $p(e) := p_F(e)$ for $e \in \mathcal{U}_F$. Then, since the function $p_L : \mathcal{U}_L \to \mathbb{Z}_{\geq 0}$ is a fixed function that is part of the input, we have that optimizing over $i : \mathcal{U}_L \to \mathbb{R}$ is equivalent to optimizing over $d : \mathcal{U}_L \to \mathbb{R}$. Therefore, (2) is equivalent to

$$\begin{aligned} \max_{d\,:\,\mathcal{U}_L \to \mathbb{R}} \;& d(X^\star) \\ \text{s.t. } & X^\star \in \operatorname*{arg\,max}_{X \in \mathcal{F}(I)} \; p(X) - d(X). \end{aligned} \tag{P$_{\max}$}$$

Observe that (P$_{\max}$) can be understood as a problem where the follower has a gross profit $p(e) \geq 0$ for every item $e \in \mathcal{U}_L \cup \mathcal{U}_F$, and is aiming to maximize their profit. The leader can place taxes on the items in $\mathcal{U}_L$, which increases the leader's profit, while decreasing the follower's profit. Note that, if $X \cap \mathcal{U}_L \neq \emptyset$ for all $X \in \mathcal{F}(I)$, then the leader can raise the tax to infinity since the follower has to include such leader's items into a feasible solution. Therefore, (P$_{\max}$) is unbounded. In the remainder of the paper, we assume that this is not the case, that is, we make sure that there is always at least one feasible solution $F \subseteq \mathcal{U}_F$.

Now, let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{F}, w, t)$ be a minimization LOP problem as described earlier. Each instance of Π can be described as $\min_{X \in \mathcal{F}(I)} w(X) = \max_{X \in \mathcal{F}(I)} -w(X)$ for some weight function $w \geq 0$. It is therefore natural to define the problem Pricing-Π in this case as

$$\begin{aligned} \max_{d\,:\,\mathcal{U}_L \to \mathbb{R}} \;& d(X^\star) \\ \text{s.t. } & X^\star \in \operatorname*{arg\,min}_{X \in \mathcal{F}(I)} \; c(X) + d(X). \end{aligned} \tag{P$_{\min}$}$$

for a function $c : \mathcal{U}(I) \to \mathbb{Z}_{\geq 0}$. Problem (P$_{\min}$) can be interpreted as a pricing problem, where the follower tries to pay minimal cost $c(X) + d(X)$ and the leader can increase the prices for their part of the universe, trying to make attractive offers to the follower while still aiming for a high profit.

Difference between the domains $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$. In both problems (P$_{\max}$) and (P$_{\min}$), the leader is unconstrained, i.e., they can select any $d : \mathcal{U}_L \to \mathbb{R}$. Other natural choices, which have also been considered by past authors [5, 20], are to only allow $d(e) \geq 0$ or $d(e) \leq p(e)$ (which is equivalent to $i(e) \geq 0$ in (2)) for all $e \in \mathcal{U}_L$. For our exposition, it is most convenient to consider $d : \mathcal{U}_L \to \mathbb{R}$. However, we later show that this does not make a difference for our main result, i.e., all the other natural choices lead to Σ_2^p-complete problems as well.

Example. If Π = knapsack, then Π is a maximization problem, and our problem Pricing-Π is given by (P$_{\max}$), where $p : \mathcal{U}(I) \to \mathbb{Z}_{\geq 0}$ are the profits and $\mathcal{F}(I)$ is the family of sets of items of total weight below the weight threshold. By our explanations, this is equivalent to the

Stackelberg knapsack problem (1) from [20]. If $\Pi =$ vertex cover, then Π is a minimization problem, and Pricing-Π is given by ($\mathrm{P}_{\min}$), where $c : \mathcal{U}(I) \to \mathbb{Z}_{\geq 0}$ are the vertex weights and $\mathcal{F}(I)$ is the family of all vertex covers of the input graph.

4 Containment in Σ_2^p

Given the above definition of Pricing-Π for any LOP problem Π, we can show the containment in Σ_2^p under the following theorem.

▶ **Theorem 2.** *For all LOP problems Π, we have Pricing-$\Pi \in \Sigma_2^p$.*

Proof. In order to prove the containment in Σ_2^p, we would like to find a witness $\exists y_1 \forall y_2$ of polynomial size that can be verified by a polynomial-time computable function V with input y_1, y_2 and the instance of the problem. Consider w.l.o.g. only problem ($\mathrm{P}_{\max}$). The instance of the corresponding decision problem Pricing-$\Pi = (I, \mathcal{U}_L, \mathcal{U}_F, p, T_{\mathrm{pr}})$ consists of an instance I as given in the underlying problem Π, leader elements $\mathcal{U}_L \subseteq \mathcal{U}(I)$, follower elements $\mathcal{U}_F \subseteq \mathcal{U}(I)$, a profit function $p : \mathcal{U} \to \mathbb{Z}_{\geq 0}$, and a threshold $T_{\mathrm{pr}} \in \mathbb{Q}_{\geq 0}$. Let $n := |\mathcal{U}(I)|$ and $n_L := |\mathcal{U}_L|$. A possible witness-verifier pair to this problem can be defined as follows:

$$\begin{aligned} &\exists d : \mathcal{U}_L \to \mathbb{R}, \exists X^\star \in \mathcal{F}(I) : \forall X \in \mathcal{F}(I) : \\ &\qquad [p(X^\star) - d(X^\star) \geq p(X) - d(X)] \wedge [d(X^\star) \geq T_{\mathrm{pr}}] \,. \end{aligned} \tag{3}$$

Since $d : \mathcal{U}_L \to \mathbb{R}$ involves real numbers, we have to worry about their encoding length. Thus, it is not entirely obvious why the witness is of polynomial size and can be checked efficiently. With the following polyhedral arguments, however, we can show that d can be encoded in polynomial size. For this, let $x_i \in \{0,1\}^n$ be the characteristic vector of the i-th feasible solution $X_i \in \mathcal{F}(I)$. Since $\Pi \in$ SSP-NP, each x_i is a string of length $n = |\mathcal{U}(I)|$. However, there may be exponentially many such x_i's. Then, (3) is true if and only if the following nonlinear mixed-integer program contains a feasible point.

$$\begin{aligned} \min\ & 0 \\ \text{s.t. } & d \in \mathbb{R}^{n_L} \\ & x^\star \in \mathcal{F}(I) \\ & p^t x^\star - (d,0)^t x^\star \geq p^t x_i - (d,0)^t x_i \quad \forall x_i \in \mathcal{F}(I) \\ & (d,0)^t x^\star \geq T_{\mathrm{pr}} \end{aligned} \tag{4}$$

Note that, in this formulation, we let the leader choose the characteristic vector $x^\star$ to the feasible solution $X^\star \in \mathcal{F}(I)$ that maximizes the follower's objective. This is possible due to the optimistic assumption. Furthermore, the constraint "$x^\star \in \mathcal{F}(I)$" can in fact be expressed as a MIP constraint because we have assumed Π to be a problem in SSP-NP. More precisely, the feasibility of Π can be checked in polynomial time, and this Turing machine can be translated into polynomially many MIP constraints (by the Cook-Levin theorem and a corresponding reduction from Sat to MIP).

If we have a yes-instance to the problem Pricing-Π, there is a feasible solution $(d, x^\star)$ to (4). Then, $x^\star$ encodes an optimal follower's solution $X^\star \in \mathcal{F}(I)$ corresponding to the leader's solution d. Since $\exists X^\star \in \mathcal{F}(I)$ is on the same quantifier level as $\exists d : \mathcal{U}_L \to \mathbb{R}$, we can fix the characteristic vector $x^\star$. Then, let $P_{x^\star}$ denote the resulting polyhedron associated to the linear program where we fix $x^\star$ and vary $d \in \mathbb{R}^{n_L}$ in (4). We remark that this linear program may have exponentially many constraints and is not necessarily bounded. However, $P_{x^\star}$ is nonempty since $d \in P_{x^\star}$. Let $L_{\max}$ be the largest size of a number in the linear inequality system of (4) (where the bit-size of a rational number r/q is given by $\log(|r|) + \log(|q|) + 1$). Furthermore, $P_{x^\star} \subseteq \mathbb{R}^{n_L}$ and the size of each constraint is bounded by $2n_L L_{\max} + 2nL_{\max} \leq 4nL_{\max} =: m$ (note that $x^\star, x_i \in \{0,1\}^n$ for all $x_i \in \mathcal{F}(I)$).

We now claim that, for all thresholds T_{pr}, the following holds: If there is a leader's solution d with profit at least T_{pr}, then there is also a leader's solution $\widehat{d}$ with profit at least T_{pr} such that $\widehat{d}$ has a bit-size of at most $16n^3 L_{\max}$. For the proof, we use a theorem of Schrijver.

▶ **Fact 3** (Theorem 10.2 in [22]). *For any rational N-dimensional polyhedron, its vertex complexity is bounded by $4N^2$ times its facet complexity.*

An upper bound to the facet complexity of $P_{x^\star}$ is given by $m = 4nL_{\max}$. This induces an upper bound to the vertex complexity of $P_{x^\star}$ of $4n_L^2 m \leq 16n^3 L_{\max}$. Then, $P_{x^\star}$ can be described by vectors $a_1, \ldots, a_k, b_1, \ldots, b_t$, each of bit-size at most $16n^3 L_{\max}$, with $P_{x^\star} = \text{conv}\{a_1, \ldots, a_k\} + \text{cone}\{b_1, \ldots, b_t\}$. Since $P_{x^\star} \neq \emptyset$ and any point in $P_{x^\star}$ is a feasible point in which the leader receives a profit of at least T_{pr}, we can set $\widehat{d} = a_1$. Thus, there is a suitable $\widehat{d}$ of bit-size at most $16n^3 L_{\max}$, which shows the claim.

Note that the encoding of $\widehat{d}$ contains the encodings of the vector components as rational numbers. It follows that there are an $x^\star \in \{0,1\}^n$, encoding an $X^\star \in \mathcal{F}(I)$, and a $\widehat{d} : \mathcal{U}_L \to \mathbb{R}$ of polynomial size such that for all $x \in \{0,1\}^n$, encoding $X \in \mathcal{F}(I)$, $\left[p(X^\star) - \widehat{d}(X^\star) \geq p(X) - \widehat{d}(X)\right] \wedge \left[\widehat{d}(X^\star) \geq T_{\text{pr}}\right]$ holds. ◀

▶ **Lemma 4.** *If* $(\mathrm{P}_{\max})$ *is bounded, then there is an optimal $d^\star : \mathcal{U}_L \to \mathbb{R}$ of polynomial bit-complexity in the size of the input $(I, \mathcal{U}_L, \mathcal{U}_F, p_L, p_F)$.*

Proof. This is a corollary of the proof of Theorem 2. Similarly to (4), consider the polyhedron $P_{x^\star} = \{d \in \mathbb{R}^{n_L} : p^t x^\star - (d,0)^t x^\star \geq p^t x_i - (d,0)^t x_i \ \forall x_i \in \mathcal{F}(I)\}$ for an appropriate fixed $x^\star \in \mathcal{F}(I)$ (an optimal follower's solution corresponding to some optimal leader's solution $\tilde{d}$), and let $L_{\max}$ again be the largest size of a number in the inequality system. Analogously to the proof of Theorem 2, $P_{x^\star}$ is nonempty because $\tilde{d} \in P_{x^\star}$, and we can write it as $P_{x^\star} = \text{conv}\{a_1, \ldots, a_k\} + \text{cone}\{b_1, \ldots, b_t\}$, where the vectors $a_1, \ldots, a_k, b_1, \ldots, b_t$ each have a bit-complexity of at most $16n^3 L_{\max}$. Optimal solutions of $\max\{(d,0)^t x^\star : d \in P_{x^\star}\}$ are now optimal solutions of $(\mathrm{P}_{\max})$. Note that this maximization problem over $P_{x^\star}$ is bounded because $(\mathrm{P}_{\max})$ is bounded. Thus, by the fundamental theorem of linear programming, we may choose an optimal $d^\star \in P_{x^\star}$ that is one of the vectors $a_1, \ldots, a_k$. This gives an optimal solution of $(\mathrm{P}_{\max})$ that has a bit-complexity of at most $16n^3 L_{\max}$. ◀

5 SSP Framework

We repeat the most important definitions from [9] necessary to understand our meta-theorem. In Section 3, we introduced the concept of LOP problems. It turns out that oftentimes the mathematical discussion is a lot clearer when one omits the concepts $\mathcal{F}(I)$, $w^{(I)}$, and $t^{(I)}$ since, for the abstract proofs of the theorems, only $\mathcal{I}$, $\mathcal{U}$, and $\mathcal{S}$ are important. This leads to the following abstraction from the concept of an LOP problem:

▶ **Definition 5** (Subset Search Problem (SSP), from [9]). *A subset search problem (or short SSP problem) Π is a tuple $(\mathcal{I}, \mathcal{U}, \mathcal{S})$ such that*

- *$\mathcal{I} \subseteq \{0,1\}^*$ is a language. We call $\mathcal{I}$ the set of instances of Π.*
- *To each instance $I \in \mathcal{I}$, there is some set $\mathcal{U}(I)$, which we call the universe associated to the instance I.*
- *To each instance $I \in \mathcal{I}$, there is some (potentially empty) set $\mathcal{S}(I) \subseteq 2^{\mathcal{U}(I)}$, which we call the solution set associated to the instance I.*

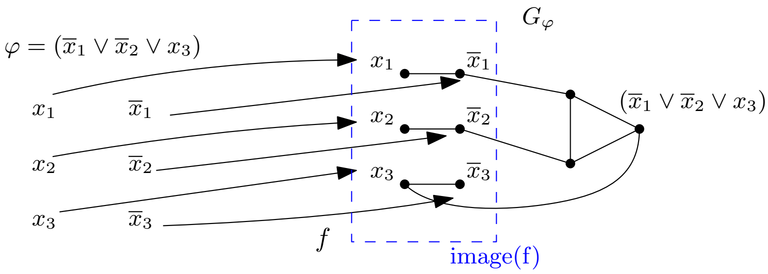

Figure 1 The injective mapping of an exemplary SSP reduction between SATISFIBILITY and VERTEX COVER.

An instance of an SSP problem is a yes-instance if $\mathcal{S}(I) \neq \emptyset$. Every LOP problem becomes an SSP problem with the definition $\mathcal{S}(I) := \{S \in \mathcal{F}(I) : w^{(I)}(S) \leq t^{(I)}\}$. We call this the *SSP problem derived from an LOP problem.* Some problems are more naturally modeled as an SSP problem to begin with, rather than as an LOP problem. For example, the satisfiability problem becomes an SSP problem with the following definition.

SATISFIABILITY (SAT)

Instances: Literal set $L = \{\ell_1, \dots, \ell_n\} \cup \{\overline{\ell}_1, \dots, \overline{\ell}_n\}$, clause set $C = \{C_1, \dots, C_m\}$ such that $C_j \subseteq L$ for all $j \in \{1, \dots, m\}$.

Universe: $L =: \mathcal{U}$.

Solution set: The set of all subsets $L' \subseteq \mathcal{U}$ of the literals such that $|L' \cap \{\ell_i, \overline{\ell}_i\}| = 1$ for all $i \in \{1, \dots, n\}$ and $|L' \cap C_j| \geq 1$ for all clauses $C_j \in C$.

Grüne and Wulf [9] introduce a new type of reduction, called *SSP reduction*. This type of reduction is best explained with an example. Consider Figure 1. Garey and Johnson [8] show a reduction from SAT to vertex cover. Given a Boolean formula φ with n variables and m clauses, they construct a graph G_φ such that G_φ has a vertex cover of size $n + 2m$ if and only if φ is satisfiable. Garey and Johnson associate to each literal $x_1 \dots, x_n, \overline{x}_1, \dots \overline{x}_n$ a vertex in the graph, which could be called the *representative* $f(x_i)$ of the literal. These representatives have very strong structural properties. Let $V' = \text{image}(f)$ be the set of all representatives. Then V' "encodes" the SAT problem: Precisely, this means that the following structural properties are true:

- The function f is injective.
- Given a vertex cover of size $n + 2m$, if we restrict it to V', we directly obtain a satisfying assignment.
- On the other hand, given a satisfying assignment, we can easily translate it to V' and complete this partial vertex cover to a vertex cover of size $n + 2m$.

Whenever one has all these structural properties together, we call this an SSP reduction. Completely formal, the definition is as follows.

▶ **Definition 6** (SSP Reduction, from [9]). *Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ and $\Pi' = (\mathcal{I}', \mathcal{U}', \mathcal{S}')$ be two SSP problems. We say that there is an SSP reduction from Π to Π', and write $\Pi \leq_{SSP} \Pi'$, if*

- *There exists a function $g : \mathcal{I} \to \mathcal{I}'$, computable in polynomial time in the input size $|I|$, such that I is a yes-instance if and only if $g(I)$ is a yes-instance (i.e., $\mathcal{S}(I) \neq \emptyset$ if and only if $\mathcal{S}'(g(I)) \neq \emptyset$).*
- *There exist functions $(f_I)_{I \in \mathcal{I}}$, computable in polynomial time in $|I|$, such that, for all instances $I \in \mathcal{I}$, we have that $f_I : \mathcal{U}(I) \to \mathcal{U}'(g(I))$ is an injective function mapping from the universe of the instance I to the universe of the instance $g(I)$ such that*

$$\{f_I(S) : S \in \mathcal{S}(I)\} = \{S' \cap f_I(\mathcal{U}(I)) : S' \in \mathcal{S}'(g(I))\}.$$

The main insight of [9] is that a surprising amount of problems admit such SSP reductions. The class of SSP-NP-complete problems is denoted by SSP-NPc and consists of all SSP problems Π that are polynomial-time verifiable and such that $\textsc{Sat} \leq_{\text{SSP}} \Pi$.

Finally, we introduce a new concept, which is not included in [9]. An LOP problem Π is called *LOP-complete* if the SSP problem derived from it is SSP-NP-complete, and there exists an SSP reduction that shows this SSP-NP-completeness and has the additional property that $\{F \in \mathcal{F}(I) : w^{(I)}(F) \leq t^{(I)} - 1\} = \emptyset$, where I, $w^{(I)}$, and $t^{(I)}$ are the instance, the weights, and the threshold used in the reduction. Intuitively, this means that there exists an NP-completeness proof of Π, where the threshold $t^{(I)}$ used for this NP-completeness reduction is tight in the sense that there can never exist any solutions below weight $t^{(I)}$, and there exists a solution of weight equal to $t^{(I)}$ if and only if the original instance is a yes-instance. As an example, in the same reduction as above, Garey and Johnson prove that G_φ has no vertex covers of size less than $n + 2m =: t^{(I)}$. This proves that the vertex cover problem is LOP-complete.

Note that the usual meaning of "completeness" of Π means that every member of the class can be reduced to Π using a Karp reduction. We mean completeness here in a slightly different context. It can be shown that, if Π is LOP-complete, then every member Π' of LOP can be reduced to Π with a reduction that has the properties described above.

In contrast to previous papers, we require this additional tightness guarantee for our calculations to be correct. However, note that a lot of natural occurring NP-hardness proofs indeed already have the tightness guarantee, so this is not too much of a restriction.

Having introduced LOP-completeness, we can now state our main result that for every LOP-complete minimization or maximization problem, the corresponding pricing problem is Σ_2^p-complete (Theorems 10 and 11). The main idea is that we can make use of the strong structural properties of SSP reductions, which are guaranteed to exist for all LOP-complete problems.

6 Feasibility Pricing

Apart from only the pricing problems ($\text{P}_{\max}$) and ($\text{P}_{\min}$) corresponding to linear optimization problems, we wish to additionally consider problems such as $\Pi = \textsc{Sat}$, which are neither maximization nor minimization problems, as they are concerned only with feasibility. As explained in the introduction, we wish to prove our meta-theorem starting from problems of this kind. Let $\Pi = (\mathcal{I}, \mathcal{U}, \mathcal{S})$ be an SSP problem as described in the previous section. Given a tuple $(I, \mathcal{U}_L, \mathcal{U}_F, p)$ with $p : \mathcal{U} \to \mathbb{Z}_{\geq 0}$, we define the problem $\textsc{Feas-Pricing-}\Pi$ as the decision problem corresponding to

$$\begin{aligned} \max_{d\colon \mathcal{U}_L \to \mathbb{R}} \ & d(X^\star) \\ \text{s.t. } \ & X^\star \in \arg\max_{X \in \mathcal{S}(I)} \ p(X) - d(X). \end{aligned} \tag{P_{feas}}$$

By the exact same proof as in Theorem 2, we can prove that, for all SSP-NP problems Π, we have $\textsc{Feas-Pricing-}\Pi \in \Sigma_2^p$. Our main result for problems of this kind can then be stated as follows and is a corrollary of Theorem 10.

▶ **Theorem 7.** *For any problem Π that is SSP-NP-complete (see Section 5), the problem $\textsc{Feas-Pricing-}\Pi$ as in (P_{feas}) is Σ_2^p-complete.*

In this section, we consider $\textsc{Feas-Pricing-Sat}$, and start by proving that the main result is true for the case of $\textsc{Sat}$. We are given a Boolean formula I in *conjunctive* normal form over the literals $\mathcal{U}(I) := \{\ell_1, \dots, \ell_N\} \cup \{\bar{\ell}_1, \dots, \bar{\ell}_N\}$, where the set $\mathcal{S}(I) \subseteq 2^{\mathcal{U}(I)}$ of satisfying subsets is defined as in Section 5. Moreover, we are given a partition $\mathcal{U}(I) = \mathcal{U}_L \cup \mathcal{U}_F$ (with $\mathcal{U}_L \cap \mathcal{U}_F = \emptyset$), a function $p : \mathcal{U}(I) \to \mathbb{Z}_{\geq 0}$, and a threshold $T_{\text{pr}} \in \mathbb{Q}_{\geq 0}$. The problem then is to decide whether the optimal value of (P_{feas}) is at least T_{pr}.

We give a sketch of the proof idea and remark that some of our arguments are based on the elegant proof of Böhnlein et al. [1]. We reduce from the canonical Σ_2^p-complete problem ∃∀-DNF-Sat. Here, we are given a Boolean formula $\varphi(A, B)$ in *disjunctive* normal form, and we are asked to evaluate whether there exists an assignment of the A-variables such that, for all assignments of the B-variables, $\varphi(A, B)$ is true. We may assume that $|A| = |B| =: n$. We now want to transform the DNF formula φ into a new CNF formula I, and find $\mathcal{U}_L \cup \mathcal{U}_F = \mathcal{U}(I)$, p, and T_{pr} such that the decision version of (P_{feas}) becomes difficult. In particular, the idea is that the choice of $d : \mathcal{U}_L \to \mathbb{R}$ forces the follower to select elements that correspond to the assignment of A, and residual elements from the follower's solution should correspond to the assignment of B. The first idea is that our formula I begins with a clause $(h_1 \vee h_2)$, where $h_1 \in \mathcal{U}_F$. If the leader sets the prices too high, the follower simply chooses h_1 and obtains a fixed profit. In the other case, the follower chooses h_2 and the following considerations apply: The instance I contains variables v_i^t, v_i^f, v_i^o for $i \in \{1, \ldots, n\}$, where $v_i^t, v_i^f \in \mathcal{U}_L$ and $v_i^o \in \mathcal{U}_F$. The idea is that the leader will set prices for v_i^t and v_i^f, and the follower will generally prefer the element that maximizes their profit, i.e., the element with the smaller price. This implicitly determines an assignment of A, by having $a_i = 1$ if $d(v_i^t) < d(v_i^f)$ and $a_i = 0$ otherwise. However, if the prices are too high, the follower simply switches to v_i^o. Finally, there is a pair of variables z_1, z_2 such that the solution contains exactly one of them. The leader would like to collect prices on z_1, but the value of z_1 is linked to the satisfiability of $\exists A \forall B \varphi(A, B)$. If this formula is false, the follower can evade the leader's prices. In summary, we are able to show that the leader can collect high prices if and only if $\exists A \forall B \varphi(A, B)$.

▶ **Theorem 8.** *Feas-Pricing-Sat is Σ_2^p-complete.*

Proof. The containment in Σ_2^p can be shown analogously to Theorem 2. Given an instance $\exists A \forall B \varphi(A, B)$ of ∃∀-DNF-Sat with $|A| = |B| = n$, we define the following instance $(I, \mathcal{U}_L, \mathcal{U}_F, p)$ of Feas-Pricing-Sat. Let $M := 2n$, let $\texttt{once}(y_1, y_2, y_3)$ be the CNF formula that is true if and only if exactly one of y_1, y_2, and y_3 is true, and let $\oplus$ denote the XOR relation.

Variables: $V := \{h_1, h_2, z_1, z_2\} \cup \bigcup_{i=1}^n \{v_i^t, v_i^f, v_i^o, a_i, b_i\}$

Literals: $\mathcal{U}(I) = V \cup \overline{V}$ with $\mathcal{U}_L = \bigcup_{i=1}^n \{v_i^t, v_i^f\} \cup \{z_1\}, \quad \mathcal{U}_F = \mathcal{U}(I) \setminus \mathcal{U}_L$

Profits ($\mathcal{U}_L$): $p(v_i^t) = p(v_i^f) = M \ \forall i \in \{1, \ldots, n\}, \quad p(z_1) = M$

Profits ($\mathcal{U}_F$): $p(v_i^o) = 1 \ \forall i \in \{1, \ldots, n\}, \quad p(h_1) = n, \quad p(z_2) = 1, \quad p(e) = 0$ else

Formula:

$$
\begin{aligned}
&(h_1 \vee h_2) \wedge \Big(h_1 \to \big(\textstyle\bigwedge_{x \in V \setminus \{h_1\}} \overline{x}\big)\Big) \\
&\wedge (h_2 \to ((z_1 \oplus z_2) \wedge (z_2 \to \neg\varphi(A, B)))) \\
&\wedge \Big(h_2 \to \textstyle\bigwedge_{i=1}^n \Big(\texttt{once}(v_i^t, v_i^f, v_i^o) \wedge (v_i^t \to a_i) \wedge (v_i^f \to \overline{a}_i) \wedge (v_i^o \to z_1)\Big)\Big)
\end{aligned}
$$

This completes the description of the instance. It is easily seen that I is in CNF, using that φ is in DNF and De Morgan's law. Let $k^\star := (n + 1)M - n$. We claim that the optimal value of (P_{feas}) is at least $k^\star$ if and only if $\exists A \forall B \varphi(A, B)$ is true.

"If" direction. Assume that there exists an assignment $\alpha : A \to \{0,1\}$ that makes $\exists A \forall B \varphi(A,B)$ true. We define a pricing function $d : \mathcal{U}_L \to \mathbb{R}$ by having $d(z_1) = M$ and, for all $i \in \{1,\ldots,n\}$,

$$\begin{aligned} d(v_i^t) &= M-1, & d(v_i^f) &= M & &\text{if } \alpha(a_i)=1,\\ d(v_i^t) &= M, & d(v_i^f) &= M-1 & &\text{if } \alpha(a_i)=0. \end{aligned}$$

We claim that this choice of d enables the leader to obtain $k^\star$ profit. First, we argue that the follower's optimal value is n.

If the follower chooses h_1, their profit is exactly n, since h_1 implies that, for all other variables, the negative literal is selected, and only positive literals induce profit for the follower. Note that this is independent of the choice of d.

Otherwise, the follower chooses h_2. Then, they obtain at most a profit of 1 from every clause $\texttt{once}(v_i^t, v_i^f, v_i^o)$. Furthermore, they cannot obtain profit from z_1, since $d(z_1) = p(z_1) = M$. At last, we have to consider the element z_2. For this, suppose that the follower collects a profit of 1 from z_2. Then, due to $v_i^o \to z_1$ and $z_1 \oplus z_2$, the follower's solution X does not contain any v_i^o, but contains either v_i^t or v_i^f for all $i \in \{1,\ldots,n\}$. If X contains exactly the element from $\{v_i^t, v_i^f\}$ that has cheaper prices d, for all $i \in \{1,\ldots,n\}$, then, due to $v_i^t \to a_i$ and $v_i^f \to \overline{a}_i$, the variables A have to be set according to the assignment α. Then the clause $z_2 \to \neg\varphi(A,B)$ and the fact that α makes $\exists A \forall B \varphi(A,B)$ true imply $z_2 \notin X$. If X contains at least one element v_i^t or v_i^f with leader's price M, then the total profit obtained from the clauses $\texttt{once}(v_i^t, v_i^f, v_i^o)$ is at most $n-1$. In all cases, the total follower's profit is at most n. In total, the follower's optimal value is indeed n. Finally, observe that the follower's solution

$$X = \{\overline{h}_1, h_2, z_1, \overline{z}_2\} \cup \bigcup_{i=1}^{n} \{\overline{v}_i^o\} \cup \bigcup_{\alpha(a_i)=1} \{v_i^t, \overline{v}_i^f, a_i\} \cup \bigcup_{\alpha(a_i)=0} \{\overline{v}_i^t, v_i^f, \overline{a}_i\}$$

achieves the follower's optimal value n and has $d(X) = k^\star$. Due to the optimistic assumption, the leader can make at least $k^\star$ profit.

"Only if" direction. Assume that there exists $d : \mathcal{U}_L \to \mathbb{R}$ such that d achieves profit at least $k^\star$ for the leader. We need to show that $\exists A \forall B \varphi(A,B)$ is satisfiable. Let X_d be an optimistic follower's response to d, i.e., $d(X_d) \geq k^\star$ and $p(X_d) - d(X_d) = \mathrm{OPT}_F$, where OPT_F is the follower's optimal value.

First, we show that $\mathrm{OPT}_F = n$. The follower can always choose h_1, hence, $\mathrm{OPT}_F \geq n$. By construction of the instance, for all $X \in \mathcal{S}(I)$, we have $p(X) \leq (n+1)M$. Therefore, the follower's profit is $\mathrm{OPT}_F = p(X_d) - d(X_d) \leq (n+1)M - k^\star = n$. It follows that $\mathrm{OPT}_F = n$.

Second, if the follower chooses h_1, then the leader's profit is 0, since h_1 implies that, for all other variables, the negative literal is selected, and only positive literals induce profit for the leader. Hence, $h_1 \notin X_d$ and $h_2 \in X_d$ because the leader's profit is at least $k^\star > 0$.

Third, observe that $d(z_1) \geq M$. Indeed, if $d(z_1) < M$, then the follower can pick $\{v_1^o, \ldots, v_n^o\} \cup \{z_1\}$ as a partial solution and obtain strictly more than n profit, contradicting $\mathrm{OPT}_F = n$.

Fourth, we claim that both $z_1 \in X_d$ and $|X_d \cap \{v_i^t, v_i^f\}| = 1$ for all $i \in \{1,\ldots,n\}$. Indeed, assume otherwise, then by the structure of the instance, X_d contains at most n elements e with $p(e) = M$. Therefore, $p(X_d) \leq nM + 1$. But, since $d(X_d) \geq k^\star$, we have $p(X_d) - d(X_d) \leq nM + 1 - k^\star = (n+1) - M \leq 0$, which is a contradiction to $\mathrm{OPT}_F \geq n$.

We claim that these four properties implicitly define a satisfying assignment to the formula $\exists A \forall B \varphi(A,B)$ by setting $\alpha(a_i) = 1$ if $v_i^t \in X_d$ (and $a_i \in X_d$) and $\alpha(a_i) = 0$ if $v_i^f \in X_d$ (and $\overline{a}_i \in X_d$). For this, assume that α does not satisfy the formula. Then the follower can choose an assignment of $b_1, \dots, b_n$ that makes $\varphi(\alpha, B)$ false. Then $X' := X_d \setminus \{z_1, \overline{z}_2\} \cup \{\overline{z}_1, z_2\}$ is also a valid solution to I. But since $d(z_1) \geq M$ and $p(z_2) = 1$, the follower's profit of X' is strictly greater than the profit of X_d, a contradiction to the fact that X_d is an optimal follower's solution. We conclude that $\exists A \forall B \varphi(A, B)$ is true. ◀

▶ **Corollary 9.** *Feas-Pricing-Sat remains Σ_2^p-complete, even if $d : \mathcal{U}_L \to \mathbb{R}$ in* ($\mathrm{P}_{\mathrm{feas}}$) *is restricted to $0 \leq d$ or $d \leq p$ or $0 \leq d \leq p$.*

Proof. We check that both directions of the proof of Theorem 8 remain true. For the "if" direction, observe that the function d defined there already has the property $0 \leq d \leq p$. For the "only if" direction, the assumption that a function d with any of the three restrictions exists is stronger than only assuming $d : \mathcal{U}_L \to \mathbb{R}$. Clearly, starting with a stronger assumption, the reasoning remains correct. ◀

7 Outline of the Proof of the Main Theorem

Due to space constraints, we can only give an outline of the proof of the main theorem. The complete formal proof is presented in the full version of this paper.

▶ **Theorem 10.** *For any problem Π that is an LOP-complete maximization problem (see Section 5), the problem Pricing-Π as in* ($\mathrm{P}_{\max}$) *is Σ_2^p-hard.*

▶ **Theorem 11.** *For any problem Π that is an LOP-complete minimization problem (see Section 5), the problem Pricing-Π as in* ($\mathrm{P}_{\min}$) *is Σ_2^p-hard.*

In order to prove our main theorems, we give a reduction from Feas-Pricing-Sat to Pricing-Π for the problem Π. We make use of the fact that Π is LOP-complete. This implies that the strong structural properties explained in Section 5 hold.

More technically, this implies that, given the instance I_{SAT} of Sat, one can compute in polynomial time a tuple $(I_\Pi, w^{(I_\Pi)}, t^{(I_\Pi)})$, where I_Π is an instance of Π with universe $\mathcal{U}(I_\Pi)$, set of feasible solutions $\mathcal{F}(I_\Pi)$, weight function $w^{(I_\Pi)} : \mathcal{U}(I_\Pi) \to \mathbb{Z}_{\geq 0}$, and threshold $t^{(I_\Pi)} \in \mathbb{Z}_{\geq 0}$. Note that a maximization LOP problem can be expressed by non-negative weights $w^{(I_\Pi)}$ and threshold $t^{(I_\Pi)}$ by defining $\mathcal{S}(I_\Pi) = \{F \in \mathcal{F}(I_\Pi) : w^{(I_\Pi)}(F) \geq t^{(I_\Pi)}\}$. In order to simplify the notation, we may write $\mathcal{U}_{\text{SAT}}$, $\mathcal{U}_\Pi$, $\mathcal{F}$, w, and t from now on for these objects. For the proof of the main theorem, we show how, using these objects, one can define an instance of the pricing problem for Π, such that we can reduce the pricing variant of Sat (which we know is Σ_2^p-hard by the previous section) to the pricing variant of Π. Therefore the pricing variant of Π is Σ_2^p-hard. More concretely, we assume that we have as input an instance of the Sat pricing problem, as follows:

$$\begin{aligned} &\max_{d\,:\,\mathcal{U}_L \to \mathbb{R}} \ d(X^\star) \\ &\quad \text{s.t. } X^\star \in \operatorname*{arg\,max}_{X \in \mathcal{S}(I_{\text{SAT}})} \ p(X) - d(X). \end{aligned} \tag{5}$$

We now wish to take this instance, defined by the function $p : \mathcal{U}(I_{\text{SAT}}) \to \mathbb{Z}_{\geq 0}$ and transform it into a new instance of the pricing problem for Π and a function $p' : \mathcal{U}(I_\Pi) \to \mathbb{Z}_{\geq 0}$, such that the two instances are equivalent.

$$\begin{aligned} &\max_{d'\,:\,\mathcal{U}'_L \to \mathbb{R}} \ d'(Y^\star) \\ &\quad \text{s.t. } Y^\star \in \operatorname*{arg\,max}_{Y \in \mathcal{F}(I_\Pi)} \ p'(Y) - d'(Y) \end{aligned} \tag{6}$$

In particular, one main idea is to define the new profit function p' for the Π pricing problem as follows, given the profits p of the SAT pricing problem:

$$p'(e) := \begin{cases} Mw(e) + p(f^{-1}(e)) & \text{if } e \in f(\mathcal{U}_{\text{SAT}}) \\ Mw(e) & \text{if } e \in \mathcal{U}_\Pi \setminus f(\mathcal{U}_{\text{SAT}}). \end{cases}$$

Here $M \gg 0$ is some large integer. The main idea why we choose the function p' in this way is that we want to show that this lifts the NP-completeness proof of Π to a Σ_2^p-completeness proof of PRICING-Π. In particular, M gives larger importance to all those elements in $\mathcal{U}_\Pi$ that are contained in the image of the SSP reduction f. There is one major technical difficulty: The NP-completeness proof of Π is inherently concerned with the set $\mathcal{S}(I)$, and not with the set $\mathcal{F}(I)$. On the other hand, PRICING-Π (in the variant ($\text{P}_{\max}$)) is inherently concerned with the set $\mathcal{F}(I)$ and not with the set $\mathcal{S}(I)$. Bringing these two concepts together is the main technical feat of the proof. We show that choosing M large enough can get rid of this issue and unify the two concepts. Finally, there is one last complication, which can be explained as a certain kind of "bootstrapping" issue. Note that M is part of the definition of p'. In the pricing problem, no matter how large we choose M, the leader can always choose a function d' with $d' < 0$ and $|d'| \gg M$, i.e., set highly negative taxes (which is unreasonable regarding the leader's objective, but could lead to optimal follower's solutions outside of $\mathcal{S}(I)$). This shows that a very simple argument along the lines of "simply choose M large enough" is *not* enough for our argument. Instead, we additionally show how to overcome this final bootstrapping issue.

In summary, we prove the following two technical lemmas:

▶ **Lemma 12.** *If γ is the optimal value of* (5)*, then the optimal value of* (6) *is at least γ.*

▶ **Lemma 13.** *If γ' is the optimal value of* (6)*, then the optimal value of* (5) *is at least γ'.*

Hence, their optimal values are equal, and this is a valid reduction between the two pricing problems, showing Σ_2^p-hardness.

8 Conclusion

In this work, we showed that the knapsack pricing problem is Σ_2^p-complete, answering an open question of Pferschy, Nicosia, Pacifi, and Schauer [20] and a conjecture of Böhnlein, Schaudt, and Schauer [1]. Furthermore, we extended this result by showing that, when considering an arbitrary NP-complete problem, only very few assumptions are needed to ensure that the corresponding pricing problem is Σ_2^p-complete. These natural assumptions are met by most classical NP-complete problems. Our result holds for both natural maximization and minimization variants of the pricing problem, and for any choice of constraints $0 \leq d$, $d \leq c$, or $0 \leq d \leq c$ for the leader.

As a future research direction, it would be interesting to consider the complexity of approximating the Stackelberg pricing problem. Moreover, other models involving several decision makers could be considered in the future. Finally, the approach of this paper is limited to underlying problems that are NP-hard. It would be very interesting to extend our techniques to underlying problems in P, showing NP-completeness of the pricing problem in a unified way. However, we do not know if such a result is possible.

References

1 Toni Böhnlein, Oliver Schaudt, and Joachim Schauer. Stackelberg packing games. *Theoretical Computer Science*, 943:16–35, 2023. doi:10.1016/j.tcs.2022.12.006.

2 Patrick Briest, Luciano Guala, Martin Hoefer, and Carmine Ventre. On Stackelberg pricing with computationally bounded customers. *Networks*, 60(1):31–44, 2012. doi:10.1002/net.20457.

3 Patrick Briest, Martin Hoefer, and Piotr Krysta. Stackelberg network pricing games. *Algorithmica*, 62(3):733–753, 2012. doi:10.1007/s00453-010-9480-3.

4 Quang Minh Bui, Margarida Carvalho, and José Neto. Asymmetry in the complexity of the multi-commodity network pricing problem. *Mathematical Programming*, 208(1):425–461, 2024. doi:10.1007/s10107-023-02043-2.

5 Quang Minh Bui, Margarida Carvalho, and José Neto. Solving combinatorial pricing problems using embedded dynamic programming models. *INFORMS Journal on Computing*, 2025. doi:10.1287/ijoc.2024.0686.

6 Quang Minh Bui, Bernard Gendron, and Margarida Carvalho. A catalog of formulations for the network pricing problem. *INFORMS Journal on Computing*, 34(5):2658–2674, 2022. doi:10.1287/ijoc.2022.1198.

7 Jean Cardinal, Erik D Demaine, Samuel Fiorini, Gwenaël Joret, Stefan Langerman, Ilan Newman, and Oren Weimann. The Stackelberg minimum spanning tree game. *Algorithmica*, 59(2):129–144, 2011. doi:10.1007/s00453-009-9299-y.

8 Michael R. Garey and David S. Johnson. *Computers and Intractability.* A Series of Books in the Mathematical Sciences. W. H. Freeman and Company, 1979.

9 Christoph Grüne and Lasse Wulf. Completeness in the polynomial hierarchy for many natural problems in bilevel and robust optimization. In Nicole Megow and Amitabh Basu, editors, *Integer Programming and Combinatorial Optimization - 26th International Conference, IPCO 2025, Baltimore, MD, USA, June 11-13, 2025, Proceedings*, volume 15620 of *Lecture Notes in Computer Science*, pages 256–269. Springer, 2025. doi:10.1007/978-3-031-93112-3_19.

10 Christoph Grüne and Lasse Wulf. On the complexity of recoverable robust optimization in the polynomial hierarchy. In Pawel Gawrychowski, Filip Mazowiecki, and Michal Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, Warsaw, Poland, August 25-29, 2025*, volume 345 of *LIPIcs*, pages 52:1–52:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.MFCS.2025.52.

11 Christoph Grüne and Lasse Wulf. The complexity of blocking all solutions. *Theor. Comput. Sci.*, 1069:115820, 2026. doi:10.1016/J.TCS.2026.115820.

12 Christoph Grüne and Femke Pfaue. A compendium of reductions: reductions.network. *arXiv preprint*, 2025. doi:10.48550/arXiv.2511.04308.

13 Robert G. Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Math. Program.*, 32(2):146–164, 1985. doi:10.1007/BF01586088.

14 Gwenaë Joret. Stackelberg network pricing is hard to approximate. *Networks*, 57(2):117–120, 2011. doi:10.1002/net.20391.

15 Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12-part-1):1608–1622, 1998. doi:10.1287/mnsc.44.12.1608.

16 Martine Labbé and Alessia Violin. Bilevel programming and price setting problems. *Ann. Oper. Res.*, 240(1):141–169, 2016. doi:10.1007/S10479-015-2016-0.

17 Martine Labbé, Miguel A. Pozo, and Justo Puerto. Computational comparisons of different formulations for the Stackelberg minimum spanning tree game. *International Transactions in Operational Research*, 28(1):48–69, 2021. doi:10.1111/itor.12680.

18 Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

19 Ulrich Pferschy, Gaia Nicosia, and Andrea Pacifici. A Stackelberg knapsack game with weight control. *Theoretical Computer Science*, 799:149–159, 2019. doi:10.1016/j.tcs.2019.10.007.

20 Ulrich Pferschy, Gaia Nicosia, Andrea Pacifici, and Joachim Schauer. On the Stackelberg knapsack game. *European Journal of Operational Research*, 291(1):18–31, 2021. doi:10.1016/j.ejor.2020.09.007.

21 Sébastien Roch, Gilles Savard, and Patrice Marcotte. An approximation algorithm for Stackelberg network pricing. *Networks*, 46(1):57–67, 2005. doi:10.1002/NET.20074.

22 Alexander Schrijver. *Theory of linear and integer programming.* John Wiley & Sons, 1998.

23 Stan van Hoesel. An overview of Stackelberg pricing in networks. *European Journal of Operational Research*, 189(3):1393–1402, 2008. doi:10.1016/j.ejor.2006.08.064.

24 Gerhard J. Woeginger. The trouble with the second quantifier. *4OR*, 19(2):157–181, 2021. doi:10.1007/S10288-021-00477-Y.