On Local Search in Bilevel Mixed-Integer Linear Programming

Eneko C. Clemente and Oleg A. Prokopyev†

Department of Business Administration, University of Zurich, Zurich 8032, Switzerland

Abstract. Two-level hierarchical decision-making problems, where a leader's choice influences a follower's action, arise across key business and public-sector domains, from market design and pricing to defense. These problems are typically modeled as bilevel programs and are known to be notoriously hard to solve at scale. In single-level combinatorial optimization, especially for challenging instances, local search methods are often used to obtain good-quality solutions when problem size limits the use of specialized solvers. These methods also play a key role within state-of-the-art solvers to improve feasibility bounds. Their appeal lies in their conceptual simplicity and ease of implementation; however, applying them to bilevel problems presents two key challenges: (i) the potentially large number of iterations required to terminate, and (ii) in each iteration, evaluating the leader's objective function requires solving the follower's problem, which may be hard by itself. We address the first challenge by extending approximate local optimality to the bilevel setting. This solution concept guarantees that no neighboring solution improves the leader's objective function beyond some limit. To overcome the second challenge, we introduce the concept of weak local optimality, yet another generalization. Specifically, instead of computing the follower's rational response, we evaluate the leader's objective function using either a follower's approximate solution, or simply a feasible decision. By combining these two concepts, we demonstrate that a (weak) approximate local optimal solution can be efficiently computed through a local search-based approach. Computational experiments demonstrate that the proposed method significantly reduces runtime compared to standard local search while maintaining comparable solution quality.

Keywords. bilevel programming, local search, computational complexity, approximation algorithms

1 Introduction

Bilevel programs form a broad class of two-level hierarchical decision-making problems with two distinct decision-makers, a *leader* and a *follower*. Specifically, the leader (or the *upper-level* decision-maker), whose perspective is modeled and optimized, acts first and initiates the decision-making process. After the leader makes their decision, the follower (or the *lower-level* decision-maker) solves their own optimization problem, which, in turn, depends on the decision taken by the leader. The leader's objective function and the upper-level constraints may contain decision variables from

†Emails: {eneko.clemente, oleg.prokopyev}@business.uzh.ch

both the leader and the follower. Consequently, the leader has to anticipate the follower's rational response when making their own decision. Bilevel programming arises in various application domains, including resource allocation [17, 42], facility location [19, 52], price setting [48, 49], network and market design [11, 69] and defense [18, 34] related problems. For a comprehensive overview of bilevel programming, we refer the reader to the recent surveys [7] and [44].

Formally, we consider *bilevel mixed-integer linear programs* (bilevel MILPs) of the form:

$$[\mathbf{BP}]: \quad z^* := \min_{\mathbf{x},\mathbf{y}^*(\mathbf{x})} \ \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) \tag{1a}$$

$$\text{s.t. } \mathbf{x} \in \mathcal{X} := \big\{\mathbf{x} \in \{0,1\}^n : \ \mathbf{H}\mathbf{x} \le \mathbf{h}\big\}, \tag{1b}$$

$$\mathbf{y}^*(\mathbf{x}) \in \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \ \mathbf{c}^\top \mathbf{y}, \tag{1c}$$

where $\mathbf{a} \in \mathbb{Q}^n_+$, $\mathbf{d} \in \mathbb{Q}^m_+$, $\mathbf{c} \in \mathbb{Q}^m$, $\mathbf{H} \in \mathbb{Q}^{p \times n}$ and $\mathbf{h} \in \mathbb{Q}^p$. We refer to $\mathbf{x} \in \mathcal{X}$ as a *leader's feasible decision*, where $\mathcal{X}$ is the *leader's feasible set*, and $\mathbf{y}^*(\mathbf{x})$ as the *follower's optimal decision* (also known as the *follower's rational response*). For a given $\mathbf{x} \in \mathcal{X}$, the *follower's feasible set* is:

$$\mathcal{Y}(\mathbf{x}) := \big\{\mathbf{y} \in \{0,1\}^{m_1} \times \mathbb{R}^{m_2}_+ : \ \mathbf{F}\mathbf{y} + \mathbf{L}\mathbf{x} \le \mathbf{f}\big\}, \tag{2}$$

where $m := m_1 + m_2$, $\mathbf{F} \in \mathbb{Q}^{q \times m}$, $\mathbf{L} \in \mathbb{Q}^{q \times n}$ and $\mathbf{f} \in \mathbb{Q}^q$. Accordingly, the follower's problem can represent any mixed-integer linear program (MILP).

In this paper, we also examine two special classes of (2). Specifically, we consider follower's decisions that consist solely of either binary (i.e., $m_2 = 0$), or continuous (i.e., $m_1 = 0$) variables:

$$\mathcal{Y}_b(\mathbf{x}) := \big\{\mathbf{y} \in \{0,1\}^m : \ \mathbf{F}\mathbf{y} + \mathbf{L}\mathbf{x} \le \mathbf{f}\big\} \quad \text{and} \quad \mathcal{Y}_c(\mathbf{x}) := \big\{\mathbf{y} \in \mathbb{R}^m_+ : \ \mathbf{F}\mathbf{y} + \mathbf{L}\mathbf{x} \le \mathbf{f}\big\}.$$

If the follower's feasible set is defined by either $\mathcal{Y}_b$ or $\mathcal{Y}_c$, then the corresponding bilevel MILPs of the form (1) are referred to as [**B-BP**] and [**C-BP**], respectively.

Local and global optimality. Local search methods are typically used in two ways in single-level combinatorial optimization. First, they are integrated into MILP solvers [33, 38], which combine bounding techniques, such as refined cutting-plane methods to tighten lower bounds, with a set of heuristics, many of which rely on local search, to enhance feasibility bounds. Second, local search is widely applied as a standalone heuristic to find good-quality solutions to hard combinatorial optimization problems [1]. In this role, they are particularly valuable when global optimality is out of reach, as they still provide meaningful, albeit weak, optimality guarantees.

In comparison, the use of local search methods in bilevel MILPs remains relatively limited; see Section 2. Our goal is not to compete with global optimization methods, as specially constructed instances can exhibit local optimal solutions that are arbitrarily far from the global optimum. Rather, we aim to identify weaker optimality guarantees that can be *efficiently* achieved in bilevel MILPs.

Main contributions. Local search in bilevel MILPs faces two main challenges. First, it may require an exponential number of improving steps for the leader to converge. Second, evaluating the leader's objective function requires solving the follower's problem—a task that may itself be computationally hard. This work addresses these challenges, and our contributions are as follows:

(*i*) **We introduce three generalizations of local optimality in bilevel MILPs and explore their relationships:** To address the exponential worst-case behavior of local search, we extend the concept of *ε-local optimality*, which is initially proposed in [58] for single-level problems, to the bilevel setting. A leader's feasible decision that is ε-local optimal may still have better leader's feasible decisions in its neighborhood. Yet, these improving decisions can only reduce the leader's objective function value by a relatively small amount, which is "bounded" by ε.

Next, we introduce *weak local optimality*, where the leader's objective function is computed with an *inexact follower* rather than the follower's rational response. The inexact follower's response can be either an approximate solution with a performance guarantee δ or simply a feasible solution to the lower-level problem. This way, we can leverage efficient approximation schemes that are available for many classes of combinatorial optimization problems. Alternatively, we resort to off-the-shelf MILP solvers with a predefined optimality gap whenever such scheme is not available.

Lastly, we combine these concepts and define *weak ε-local optimality*. In particular, we demonstrate that if the follower's problem admits a δ-approximation algorithm, then any weak ε-local optimal solution is, in fact, $\mathcal{O}(\varepsilon + \delta)$-locally optimal.

(*ii*) **We propose a local search-based algorithm that finds a weak ε-local optimal solution in a polynomial number of improvement steps for the leader:** Our approach, referred to as $(\varepsilon, \mathcal{A})$-`LSA`, builds on two main ideas. First, the leader's objective function is scaled in a "strategic" manner, akin to the ε-local search [58]. Second, we evaluate the leader's objective function by using an inexact follower's response, preferably a δ-approximate solution, if one is available. We demonstrate that the proposed algorithm converges in a polynomial number of improving steps for the leader (where the leader's objective function might be evaluated with an inexact follower's response) to a weak ε-local optimal solution. Additionally, we identify a class of problems for which our approach is guaranteed to return a weak local optimal solution.

(*iii*) **We illustrate numerically the trade-offs revealed by our theoretical analysis:** In particular, our numerical results also support that the proposed approach reduces the runtime compared to the standard local search, while still returning solutions of comparable quality. We also show that neither scaling the leader's objective function nor solving approximately the follower's problem alone is sufficient; both must be employed in a joint manner.

Our technical assumptions. For any given leader's feasible decision, the follower's problem (1c) is a mixed-integer linear program that may have multiple optimal solutions. Consequently, the leader's problem [**BP**] may not be well-defined, as it depends on which optimal decision is chosen by the follower [44]. In this study, we focus on *optimistic* bilevel programs, where the follower selects the optimal decision to the lower-level problem that is most favorable to the leader. Accordingly, for any $\mathbf{x} \in \mathcal{X}$, the follower's rational response is given by:

$$\mathbf{y}^*(\mathbf{x}) \in \operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \left\{ \mathbf{d}^\top \mathbf{y} : \ \mathbf{c}^\top \mathbf{y} \geq \varphi(\mathbf{x}) \right\}, \tag{3}$$

where $\varphi(\mathbf{x})$ represents the *follower's value function* and is defined by $\varphi(\mathbf{x}) := \max \left\{ \mathbf{c}^\top \mathbf{y} \, : \, \mathbf{y} \in \mathcal{Y}(\mathbf{x}) \right\}$.

Any optimal decision $\mathbf{y}^*(\mathbf{x})$ picked by the follower, which satisfies (3), leads to the same leader's objective function value. Therefore, we assume, without loss of generality, that if the follower's optimal decision $\mathbf{y}^*(\mathbf{x})$ satisfies (3), then it is uniquely determined. Consequently, $\mathbf{y}^*(\cdot)$ can be defined as a function that returns the follower's rational response given a leader's feasible decision $\mathbf{x}$. The optimistic assumption is, perhaps, the most commonly used in the bilevel optimization literature [44]. There is also the *pessimistic* model, in which the follower chooses an optimal decision that is least favorable to the leader [75], as well as other generalizations [50].

To simplify notation and the analysis, we focus on the setting in which the leader's problem does not contain *coupling constraints*, i.e., the follower's rational response does not appear in (1b). Coupling constraints may be required in some practical application settings; thus, we discuss how our approach may accommodate these constraints in Supplemental Material S.M.1. Nevertheless, when both the leader's and follower's decision variables are continuous, such constraints do not increase modeling power, as they can be embedded into the leader's objective through suitable penalty terms [37]. However, identifying an appropriate penalization, and extending this result to bilevel MILPs remain open questions. Finally, the following assumptions are made throughout this paper:

- **A1:** $\mathcal{X} \neq \varnothing$, and $\mathcal{Y}(\mathbf{x}) \neq \varnothing$ for all $\mathbf{x} \in \mathcal{X}$.
- **A2:** There exists $U > 0$ such that $\|\mathbf{y}\|_\infty \leq U$ for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and for all $\mathbf{x} \in \mathcal{X}$.
- **A3:** $\mathbf{a} \in \mathbb{Q}^n_+$ and $\mathbf{d} \in \mathbb{Q}^m_+$.
- **A4:** $z^* > 0$ and $\mathbf{c}^\top \mathbf{y} \geq 0$ for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and for all $\mathbf{x} \in \mathcal{X}$.

Detailed justification of these technical assumptions is provided in Supplemental Material S.M.1.

The remainder of the paper. Section 2 reviews the relevant literature. Section 3 introduces generalizations of local optimality to the bilevel setting, and Section 4 examines how they relate to one another. Building on these solution concepts, Section 5 presents the weak approximate local search algorithm, or $(\varepsilon, \mathcal{A})$-LSA, and studies its theoretical properties, including its worst-case performance. Section 6 complements the theoretical analysis with numerical evidence. Section 7 concludes the paper. Table 1 summarizes the solution concepts and algorithms studied throughout. Appendix A gathers the main notation and concepts used in the manuscript, while the supplemental material provides additional discussions and numerical illustrations included for completeness.

	Follower's response		**Algorithm**
	Exact (Section 3.1)	**Inexact** (Section 3.2)	
Leader	local optimality	weak local optimality	(weak) local search
	ε-local optimality	weak ε-local optimality	$(\varepsilon, \mathcal{A})$-LSA (Section 5)

Table 1: Overview of the solution concepts and algorithms considered throughout the paper. If the follower's problem is solved exactly, then we examine local optimality and ε-local optimality. Conversely, if the follower's problem is not necessarily solved exactly, say, with an algorithm $\mathcal{A}$, then we explore weak local optimality and weak ε-local optimality. The relations between these solution concepts are further discussed in Section 4.

2 Literature review

Bilevel optimization. In contrast to single-level linear programs (LPs), solving bilevel LPs is known to be NP-hard [8]. In fact, even finding a local optimal solution is NP-hard [60]. Moreover, it is worth noting that even special classes of bilevel programs, such as linear min/max programs, remain strongly NP-hard [35]. This complexity also extends to the pessimistic model [75].

Given the inherent difficulty of solving bilevel programs, considerable efforts are directed at developing specialized algorithms for various classes of problems. Primary examples include interdiction games [15, 28, 41], matching interdiction [24, 66, 78] as well as decentralized versions of various network-related [17] and facility location [19] problems. Further insights into solution methods for various classes of bilevel MILPs can be found in the survey by [44].

For bilevel LPs, exact methods are reasonably well-established and typically rely on reformulating the problems as single-level MILPs by leveraging Karush–Kuhn–Tucker conditions [6, 23] or strong duality [43, 46, 76]. On the other hand, if the follower's decisions include integer restrictions, then the lower-level problem becomes non-convex. Consequently, reformulation-based approaches result in intractable models, even for modest instance sizes; see, e.g., [68]. In fact, introducing binary decision variables for the follower escalates the computational complexity to Σ_2^p-hardness [39]. Hence, under reasonable assumptions, these models cannot be reformulated as polynomial-sized single-level MILPs and hence, need to be solved with more sophisticated methods.

Exact methods for solving bilevel MILPs often exploit branch-and-bound techniques, as introduced in the seminal work by [56]. Other methods in the literature rely on decomposition [13, 61] or parametric programming [47, 53, 68] techniques. Although a few basic solvers for bilevel MILPs based on branch-and-cut ideas are available [27, 67], these solvers are effective only for relatively small-sized instances. Broadly speaking, it can be argued that exact methods for solving general types of bilevel MILPs are still in an early stage of their development.

Local search in the single-level setting. Advances in solving MILPs, as we mentioned earlier, have also benefited from the use of heuristics to obtain better feasibility bounds. The vast majority of these heuristics are based on local search ideas, and are heavily exploited for finding approximate or high-quality solutions to combinatorial optimization problems [5, 10, 40, 62]. On the other hand, local search can take an exponential number of improving steps to converge; see, for instance, [16] for the traveling salesman problem. This worst-case behavior is addressed in [58], where the solution concept of ε-local optimality is introduced. Importantly, it is shown that an ε-local optimal solution to any single-level combinatorial optimization problem can be found within a polynomial number of improving steps in the model's dimension and $1/\varepsilon$.

Local search in the bilevel setting. Recent advances in bilevel MILPs have largely been driven by the developments in cutting-plane techniques. Heuristic methods that are proposed in the literature often adapt established single-level strategies to the bilevel context [57, 73] and focus on specific problem classes [14, 25, 29]. Yet, these studies typically emphasize the computational performance of their approaches, while giving limited attention to the theoretical worst-case analysis.

From the leader's perspective, the results on local optimality available in the existing literature predominantly focus on bilevel programs with continuous variables at both levels [22, 65]. Alternative concepts, such as stationary solutions, have also been studied in the literature [45]. Additionally, local optimality can be approached from the follower's perspective [63], providing bounds on the leader's objective function by assuming the follower uses a local optimal solution to its 0–1 problem.

Naturally, local search ideas can be extended from the single-level setting to bilevel MILPs. However, a significant distinction arises in the bilevel setting due to the lower-level problem. Indeed, evaluating the leader's objective function requires solving the follower's problem, which is an MILP by itself and hence, NP-hard, in general [31]. Thus, searching for an improving solution in the neighborhood of a leader's feasible decision may require solving exactly multiple MILPs, which can be computationally prohibitive in practical settings.

3 Local optimality criteria

In this section, we extend the concepts of local optimality as originally introduced in single-level combinatorial optimization [1] to the bilevel setting. A local optimal solution is always defined with respect to some neighborhood, which is a subset of the leader's feasible set containing decisions that are sufficiently "close" to each other.

Neighborhood. Formally, given the leader's feasible set $\mathcal{X} \subseteq \{0,1\}^n$, we define the *neighborhood function with respect to* $\mathcal{X}$ as a mapping $N_{\mathcal{X}}$ from the set $\mathcal{X}$ to the set of all possible subsets $2^{\mathcal{X}}$ of $\mathcal{X}$, i.e., $N_{\mathcal{X}} : \mathcal{X} \longrightarrow 2^{\mathcal{X}}$. Specifically, for any $\mathbf{x}$ in $\mathcal{X}$, $N_{\mathcal{X}}(\mathbf{x}) \subseteq \mathcal{X}$ represents a set of neighbors of $\mathbf{x}$, which is referred to as the *neighborhood of* $\mathbf{x}$. Moreover, we assume that $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x})$. Throughout this study, the terms of a *neighborhood function* and a *neighborhood* are employed interchangeably.

There exists a rich variety of neighborhoods that have been explored in the combinatorial optimization literature [2, 3, 21]. Among those, the *k-flip neighborhood* function is, perhaps, the most commonly used one due to its simplicity [1]. Given a set $\mathcal{X} \subseteq \{0,1\}^n$, $\mathbf{x} \in \mathcal{X}$, and an integer $k \geq 1$, the k-flip neighborhood function at $\mathbf{x}$ with respect to $\mathcal{X}$ is defined as:

$$N_{\mathcal{X}}^{(k)}(\mathbf{x}) := \{\tilde{\mathbf{x}} \in \mathcal{X} : \ \|\tilde{\mathbf{x}} - \mathbf{x}\|_1 \leq k\}, \tag{4}$$

where $\|\cdot\|_1$ represents the 1-norm, also known as the *Hamming distance* for 0-1 vectors.

3.1 Exact follower: local and approximate local optimality

Local optimality. A local optimal solution to [**BP**] is simply a leader's feasible decision, which has no neighbor with a better leader's objective function value. Formally:

Definition 1. Given a neighborhood function $N_{\mathcal{X}}$, a leader's feasible decision $\mathbf{x}^0 \in \mathcal{X}$ is said to be *locally optimal* with respect to $N_{\mathcal{X}}$ if and only if:

$$\mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^0) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) \tag{5}$$

for all $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^0)$. If there exists $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^0)$ such that (5) does not hold, then $\mathbf{x}$ is referred to as a leader's *improving solution*, or simply a *better solution* in the neighborhood of $\mathbf{x}^0$. ■

In combinatorial optimization, a local optimal solution is typically found using the standard *local search algorithm* or, shortly, LSA; see [1]. This approach can be extended to bilevel MILPs to obtain a local optimal solution to [**BP**]. The algorithm begins with a feasible decision for the leader and explores its neighborhood to identify a better solution (*neighborhood search*). If an improving solution is found, then the algorithm updates to this new feasible decision (*improving step*), and the process repeats until no further improvements are possible. Hence, at each iteration, LSA requires solving the lower-level problem (1c) to evaluate the leader's objective function.

On the complexity of local search. Verifying whether a given leader's feasible decision is locally optimal is NP-hard, even when both the leader's and follower's decision variables are continuous; see [72]. Furthermore, it comes without surprise that local search requires an exponential number of improving steps to converge for the leader in the worst case. We demonstrate that any *quadratic binary program* (QBP) can be reformulated as a bilevel MILP. Consequently, any improving step (for local search) in QBP corresponds to an equivalent improving step in the corresponding bilevel problem. Then, by leveraging the existing results on the runtime complexity of local search in the context of QBP [59], we show that for bilevel MILPs, the standard local search requires an exponential number of improving steps to converge in the worst case.

Formally, consider the problem of finding an optimal solution to a QBP given by:

$$[\textbf{QBP}]: \quad z^*_{QP} := \min_{\mathbf{x}\in\{0,1\}^n} \ \mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{b}^\top \mathbf{x},$$

where $\mathbf{b} \in \mathbb{Q}^n$ and $\mathbf{Q} \in \mathbb{Q}^{n\times n}$ is a symmetric matrix, with its entries denoted by $\{q_{ij}\}_{i,j=1}^n$.

A QBP can be linearized by introducing a new variable $y_{ij} = x_i x_j$ for all $i, j \in [n]$. Then, standard linearization techniques for the resulting bilinear terms can be applied [32, 55]. Instead, we derive a bilevel problem as follows. To enforce that $y_{ij} = 0$ whenever either $x_i = 0$ or $x_j = 0$, we introduce a lower-level problem, which can be either a 0-1 or a linear program. That is, we obtain:

$$[\textbf{QP-BP}]: \quad z^*_{BP} := \min_{\mathbf{x},\mathbf{y}^*(\mathbf{x})} \ \mathbf{b}^\top \mathbf{x} + \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} y^*_{ij}(\mathbf{x}) \tag{6a}$$

$$\text{s.t. } \mathbf{x} \in \{0,1\}^n, \ \mathbf{y}^*(\mathbf{x}) \in \underset{\mathbf{y}\in\mathcal{Y}^{BP}(\mathbf{x})}{\operatorname{argmax}} \ \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij}, \tag{6b}$$

where, given a leader's feasible decision $\mathbf{x} \in \{0,1\}^n$, we define:

$$\mathcal{Y}^{BP}(\mathbf{x}) := \left\{\mathbf{y} \in [0,1]^{n\times n}: \ y_{ij} \le x_i \ , \ y_{ij} \le x_j \quad \forall\, i, j \in [n]\right\}, \tag{7}$$

as the follower's feasible set in (6b). Although the follower's decision variables are all continuous, the follower's optimal decisions are always binary. Therefore, without any loss of generality, we can also consider a follower's feasible set, where, instead of (7), the decision variables are all binary, i.e., $\mathbf{y} \in \{0,1\}^{n\times n}$, or any mix of binary and continuous decision variables.

To stay concise, we omit the definitions of the neighborhood and the improving solution for quadratic binary programs. Some of the concepts discussed before, including local optimality, can also be defined in the context of quadratic binary programming [59]. One can verify that the two considered problems, namely, [**QBP**] and [**QP-BP**], are essentially equivalent. Specifically, any feasible decision to the first problem can be derived from a feasible decision to the second problem, and vice versa. The proof is straightforward and is therefore omitted for brevity.

Proposition 1. *Given an integer $k \geq 1$, the standard local search with respect to $N_{\mathcal{X}}^{(k)}$ returns a local optimal solution to* (6) *in an exponential number of improving steps for the leader in the worst case.*

Using Theorem 2 by [59], there exists a class of [**QBP**], where local search requires an exponential number of improving steps to converge to a local optimal solution using the k-flip neighborhood function; recall (4). Then, this class can be reduced to a bilevel program. Additionally, we have that each improving solution to the former problem is an improving solution to the latter problem, and vice versa. Since Proposition 1 follows directly from this reduction, its proof is omitted for brevity.

Proposition 1 is negative, as it essentially implies that, in the worst case, local search algorithms offer no computational advantage over exhaustive global search. While we acknowledge that such extreme cases may be rare in practice, they highlight a key limitation: the decision-maker has no control over the runtime of these algorithms. As we later show in our computational study in Section 6, even for relatively modest instance sizes, standard local search may converge "slowly."

Approximate local optimality. The concept of ε-local optimality, which generalizes the traditional definition of local optimality, is introduced in [58] to address the exponential worst-case behavior of local search. We extend the definition of an *approximate local optimal solution*, or *ε-local optimal solution*, to the bilevel setting as follows:

Definition 2. Given $\varepsilon \geq 0$, and a neighborhood function $N_{\mathcal{X}}$, a leader's feasible decision $\mathbf{x}^{\varepsilon} \in \mathcal{X}$ is said to be *ε-locally optimal* with respect to $N_{\mathcal{X}}$ if and only if:

$$\mathbf{a}^{\top}\mathbf{x}^{\varepsilon} + \mathbf{d}^{\top}\mathbf{y}^{*}(\mathbf{x}^{\varepsilon}) \leq (1+\varepsilon)\Big(\mathbf{a}^{\top}\mathbf{x} + \mathbf{d}^{\top}\mathbf{y}^{*}(\mathbf{x})\Big)$$

for all $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon})$. ■

If $\varepsilon = 0$, then ε-local optimality coincides with local optimality. Conversely, if $\mathbf{x}^{\varepsilon}$ is ε-locally optimal for a given $\varepsilon > 0$, then there may still exist better leader's feasible decisions in its neighborhood. However, any such improvement might only reduce the leader's objective function value by ε in relative terms. For example, if the leader's objective function value at $\mathbf{x}^{\varepsilon}$ is 1, then no leader's feasible decision in its neighborhood has a leader's objective strictly less than $(1+\varepsilon)^{-1}$.

Furthermore, if both the leader's and the follower's variables are all binary, then verifying whether a given leader's decision is an ε-local optimal solution (for any $\varepsilon > 0$) is also NP-hard. While being rather technical, this observation is not that surprising. Therefore, this result and its proof are relegated to Supplemental Material S.M.2.1; see Proposition 4. Next, to address the difficulty arising with the need of solving the follower's problem, we introduce the notion of *weak local optimality*, where the leader's objective function is evaluated using an inexact follower's response.

3.2 Inexact follower: weak and weak approximate local optimality

Solving the follower's problem may be challenging, and hence, one may be interested in evaluating the leader's objective function using a follower's decision of "sufficiently good quality" rather than the follower's rational response. In fact, the idea of relaxing the optimality criteria for the follower is not new and has been exploited with some success [63, 77].

Approximate solutions. Formally, let $\delta \in [0, 1)$, $\mathbf{x} \in \mathcal{X}$ be a leader's feasible decision, and $\mathcal{A}$ be an algorithm that returns a feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the lower-level problem (1c) for any leader's feasible decision. Then, $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ is said to be a *δ-approximation* (or, simply, *δ-approximate solution*) of the follower's optimal solution $\mathbf{y}^*(\mathbf{x})$ if and only if:

$$(1 - \delta)\mathbf{c}^\top \mathbf{y}^*(\mathbf{x}) \leq \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \tag{8}$$

for any $\mathbf{x} \in \mathcal{X}$. Moreover, if $\mathcal{A}$ satisfies (8), then $\mathcal{A}$ is referred to as a *δ-approximation algorithm*. In particular, we do not make any restriction on $\mathcal{A}$ being a polynomial-time algorithm in (8).

Note that if the follower's decision variables are restricted to be all binary, then the combinatorial structure of the lower-level problem (1c) can occasionally be leveraged to find an approximate solution efficiently; see, e.g., [71]. Accordingly, if such procedure exists, then the performance guarantee is either a known constant or can somehow be controlled.

Conversely, a polynomial-time approximation algorithm for the lower-level problem does not necessarily exist. Hence, another option is to use an off-the-shelf solver, which can solve MILPs with a predefined optimality gap. Then, the resulting solution can serve as a δ-approximation to the follower's problem. However, such solvers typically rely on enumerative approaches, and are not guaranteed, in general, to find approximate solutions to MILPs in polynomial time.

Weak local optimality. The definition of local optimality can be generalized by assuming that the leader's objective function is computed using a follower's feasible decision obtained by a given algorithm $\mathcal{A}$ instead of the follower's rational (exact) response. When referring to algorithms to solve the follower's problem (1c), we assume, without loss of generality, that such algorithms always return a feasible decision for each $\mathbf{x} \in \mathcal{X}$. Examples include δ-approximation algorithms as outlined in (8), or specialized procedures. Formally, *weak local optimality* is defined as follows:

Definition 3. Let $N_{\mathcal{X}}$ be a neighborhood function and $\mathcal{A}$ be an algorithm that returns a feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the follower's problem (1c) for any $\mathbf{x} \in \mathcal{X}$. Then, a leader's feasible decision $\mathbf{x}^0 \in \mathcal{X}$ is said to be *weakly local optimal* with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$ if and only if:

$$\mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \tag{9}$$

for any $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^0)$. ■

If there exists $\mathbf{x}$ in the neighborhood of $\mathbf{x}^0$ that does not satisfy (9), then, following the previous discussion with the exact follower's response, $\mathbf{x}$ is referred to as an improving solution for the leader or, simply, a better solution in the neighborhood of $\mathbf{x}^0$. Moreover, for any $\mathbf{x} \in \mathcal{X}$, we refer to the follower's feasible decision $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ obtained by calling $\mathcal{A}$ as the *inexact follower's decision.* Similarly, the leader's objective function, where the follower's optimal decision $\mathbf{y}^*(\mathbf{x})$ is replaced by $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$, is referred to as the *leader's objective function value with an inexact follower.*

The difference between the leader's objective function values with an inexact follower at $\mathbf{x}^0 \in \mathcal{X}$ and $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^0)$, is denoted by $\Delta^{\mathcal{A}}(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d})$, and referred to as the *(absolute) gap.* Formally:

$$\Delta^{\mathcal{A}}\left(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d}\right) := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) - \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right). \tag{10}$$

If $\Delta^{\mathcal{A}}(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d})$ is strictly positive, then $\mathbf{x}$ is an improving solution in terms of the leader's objective function with an inexact follower. Conversely, if $\Delta^{\mathcal{A}}(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d})$ is non-positive for any $\mathbf{x}$ in the neighborhood of $\mathbf{x}^0$, then $\mathbf{x}^0$ is weakly local optimal.

On the complexity of weak local search. Weak local optimal solutions can be obtained via weak local search, an extension of local search in which the leader's objective function is evaluated using an inexact follower's decision rather than the follower's rational response. Its description is relegated to Supplemental Material S.M.2.2. As a direct consequence of Proposition 1, any weak local search algorithm in which the follower's problem is solved by a simple LP-rounding heuristic requires an exponential number of improving steps to converge in the worst case.

Indeed, while such a procedure does not in general guarantee optimality, it always returns an optimal solution for the follower's problem in [**QP-BP**]; recall our discussion in Section 3.1. Consequently, weak local search based on such an LP-rounding heuristic inherits the exponential worst-case behavior. We expect a similar phenomenon to hold for a broader class of procedures used to solve the follower's problem. Establishing such negative results for a broader class of procedures, however, appears technically challenging and remains an interesting direction for future research.

Weak approximate local optimality. We introduce the approximate counterpart of weak local optimality to address the exponential worst-case behavior outlined above. Formally, *weak approximate local optimality*, or *weak ε-local optimality* is defined as follows:

Definition 4. Let $\varepsilon \geq 0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be an algorithm that returns a feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the follower's problem (1c) for any $\mathbf{x} \in \mathcal{X}$. Then, a leader's feasible decision $\mathbf{x}^{\varepsilon,\mathcal{A}} \in \mathcal{X}$ is said to be *weakly ε-local optimal* with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$ if and only if:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right)$$

for any $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$. ■

A weak ε-local optimal solution is essentially weakly local optimal whenever $\varepsilon = 0$, or simply ε-locally optimal, whenever the lower-level problem is solved exactly. Interestingly, if a δ-approximation algorithm is available for the follower's problem, then there exists a relation between weak (approximate) and approximate local optimality, which we discuss in Section 4.

3.3 An illustrative example

We illustrate the generalized local optimality notions from the previous section on the maximum weighted clique interdiction problem [30]. Consider an undirected, weighted graph $G = (V, E)$ with vertex weights w_i for $i \in V$ as illustrated in Figure 1 below. The leader strategically interdicts (blocks) a subset of vertices; the follower then solves a maximum weighted clique problem on the residual graph. This interdiction problem can be formulated as the following bilevel program:

$$[\mathbf{CIP}]: \quad z^* := \min_{\mathbf{x}, \mathbf{y}^*(\mathbf{x})} \quad \sum_{i \in V} w_i y^*(\mathbf{x})_i \tag{11a}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{X} := \left\{ \mathbf{x} \in \{0,1\}^{|V|} \, : \, \sum_{i \in V} x_i \leq h \right\}, \tag{11b}$$

$$\mathbf{y}^*(\mathbf{x}) \in \underset{\mathbf{y} \in \{0,1\}^{|V|}}{\text{argmax}} \left\{ \sum_{i \in V} w_i y_i \, : \, y_i + y_j \leq 1 \; \forall \, (i,j) \notin E \, , \, \mathbf{y} \leq \mathbf{1} - \mathbf{x} \right\}, \tag{11c}$$

where $\mathbf{x}$ is the leader's decision, i.e., $i \in V$ is interdicted if and only if $x_i = 1$; $\mathbf{y}$ is the follower's decision, where $y_i = 1$ if and only if i belongs to a clique. Furthermore, if $(i,j) \notin E$, then i and j cannot be in the same clique; see (11c). Given a clique $\mathcal{C} \subseteq V$, its weight is given by $\omega(\mathcal{C}) := \sum_{i \in \mathcal{C}} w_i$.

Exact follower. We fix $h = 1$ in (11b) and consider the 2-flip neighborhood function. The maximum weighted clique in G without interdiction is given by $\mathcal{C} = \{1, 2, 3, 4\}$ with $\omega(\mathcal{C}) = 300$. Thus, the interdiction decision $\mathbf{x}^*$ of blocking vertex 1 is the leader's optimal solution, and hence, also locally optimal. The resulting maximum clique is $\mathcal{C}^* = \{2, 3, 4\}$ with $\omega(\mathcal{C}^*) = 220$.

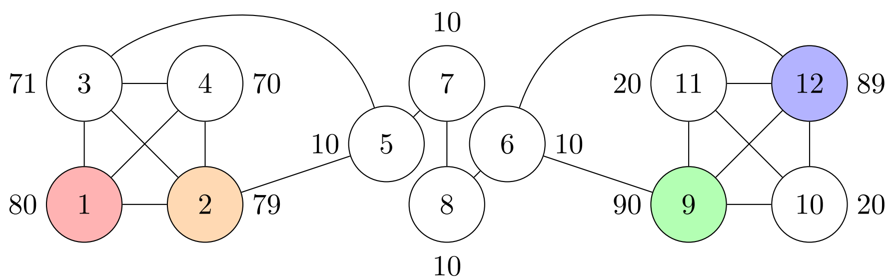


Figure 1: A weighted graph with maximum weighted clique $\mathcal{C} = \{1, 2, 3, 4\}$. Interdicting vertex 1 is locally optimal and leads to the clique $\mathcal{C}^* = \{2, 3, 4\}$. Interdicting vertex 2 is 1/220-locally optimal and leads to the clique $\mathcal{C}^\varepsilon = \{1, 3, 4\}$. Using a greedy heuristic, interdicting vertex 9 is weakly local optimal, while interdicting vertex 12 is weakly 1/129-local optimal, with corresponding cliques $\mathcal{C}^\mathcal{A} = \{10, 11, 12\}$ and $\mathcal{C}^{\varepsilon,\mathcal{A}} = \{9, 10, 11\}$, respectively.

On the other hand, if the leader's feasible decision $\mathbf{x}^\varepsilon$ consists of interdicting vertex 2, then the maximum clique becomes $\mathcal{C}^\varepsilon = \{1, 3, 4\}$ with $\omega(\mathcal{C}^\varepsilon) = 221$. Hence, $\mathbf{x}^\varepsilon$ is not a local optimal solution since $\mathbf{x}^*$ is an improving solution (unique) in its neighborhood. We have that $\frac{\omega(\mathcal{C}^\varepsilon)-\omega(\mathcal{C}^*)}{\omega(\mathcal{C}^*)} = \frac{221-220}{220} = \frac{1}{220}$, which, together with Definition 2, implies that $\mathbf{x}^\varepsilon$ is ε-local optimal for $\varepsilon = \frac{1}{220}$.

Inexact follower. Next, the follower's response is computed by using a greedy heuristic $\mathcal{A}$ as described in the Supplemental Material S.M.2.3. Starting with an empty clique, $\mathcal{A}$ iteratively selects and adds a vertex with the maximum degree to $\mathcal{C}$ such that it remains a clique. If multiple vertices have the same degree, then the vertex with the highest weight is chosen, with ties broken arbitrarily.

Denote by $\widetilde{\mathcal{C}} = \{9, 10, 11, 12\}$ the clique obtained by using $\mathcal{A}$ on G, with $\omega(\widetilde{\mathcal{C}}) = 219$. Then, let $\mathbf{x}^\mathcal{A}$ denote the leader's feasible decision that consists of interdicting vertex 9. Hence, $\mathbf{x}^\mathcal{A}$ is weakly local optimal (recall Definition 3), and $\mathcal{C}^\mathcal{A} = \{10, 11, 12\}$, where $\omega(\mathcal{C}^\mathcal{A}) = 129$. Similarly, let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ denote the leader's feasible decision that consists of interdicting vertex 12. Then, $\mathcal{C}^{\varepsilon,\mathcal{A}} = \{9, 10, 11\}$ with $\omega(\mathcal{C}^{\varepsilon,\mathcal{A}}) = 130$. According to Definition 4, $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly ε-local optimal, where $\varepsilon = \frac{1}{129}$.

4 Relationships between weak and approximate local optimality

Next, we explore links between weak (approximate) and approximate local optimality whenever the follower's problem is solved approximately. Understanding these relationships is crucial, as they allow us to quantify the error introduced from the leader's perspective whenever the lower-level problem is not solved exactly. In this section, we first address a special case, where there is a symmetry between the leader's and follower's objective functions. Afterwards, we establish a more general result that does not impose any assumptions on the parameters of the leader's objective function.

Symmetry between upper and lower levels. Vectors $\mathbf{c}$ and $\mathbf{d}$ in the leader's and follower's objective functions, respectively, are assumed to only differ by a scaling factor, i.e., $\mathbf{d} = \alpha \mathbf{c}$ for some $\alpha > 0$. Under this assumption, we assert that any weak (approximate) local optimal solution with respect to a δ-approximation algorithm is also approximate locally optimal. Importantly, this assertion does not rely on any assumption regarding vector $\mathbf{a}$ in the leader's objective function. Vectors $\mathbf{c}$ and $\mathbf{d}$ satisfying $\mathbf{d} = \alpha \mathbf{c}$ typically arise in "symmetric" interdiction problems [64]. If $\mathbf{a} = \mathbf{0}$ and $\alpha = 1$, then the leader and the follower essentially engage in a zero-sum game.

Proposition 2. *Let $\varepsilon \geq 0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be a δ-approximation algorithm for the lower-level problem* (1c)*. Also, assume that $\mathbf{d} = \alpha \mathbf{c}$ for some $\alpha > 0$. If $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly ε-local optimal with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is $\left(\frac{\delta+\varepsilon}{1-\delta}\right)$-locally optimal with respect to $N_{\mathcal{X}}$.*

Proof. Let $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$ be a leader's feasible decision in the neighborhood of $\mathbf{x}^{\varepsilon,\mathcal{A}}$. Then, by the definition of weak ε-local optimality (recall Definition 4 in Section 3.2), we have that:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq (1+\varepsilon) \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \right). \tag{12}$$

Note that $\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$ and $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ are both δ-approximate solutions to the follower's problem (1c), given $\mathbf{x}^{\varepsilon,\mathcal{A}}$ and $\mathbf{x}$, respectively. Hence, the following two inequalities are satisfied:

$$(1-\delta)\, \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \quad \text{and} \quad \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \leq \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}),$$

where the first inequality holds by (8) and the second one by the feasibility of $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$.

Next, recall our assumption that $\mathbf{d} = \alpha \mathbf{c}$, which implies:

$$\begin{aligned}
(1-\delta) \left(\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \right) &= (1-\delta) \left(\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \alpha \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \right) \\
&\leq (1-\delta)\, \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \alpha \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) && \text{by the definition of } \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \\
&\leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) && \text{since } 0 \leq \delta < 1 \text{ and } \mathbf{d} = \alpha \mathbf{c} \\
&\leq (1+\varepsilon) \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \right) && \text{by (12)} \\
&= (1+\varepsilon) \left(\mathbf{a}^\top \mathbf{x} + \alpha \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \right) \\
&\leq (1+\varepsilon) \left(\mathbf{a}^\top \mathbf{x} + \alpha \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}) \right) && \text{by the definition of } \mathbf{y}^*(\mathbf{x}) \\
&= (1+\varepsilon) \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) \right).
\end{aligned}$$

Therefore, by observing that $\frac{1+\varepsilon}{1-\delta} = 1 + \frac{\varepsilon+\delta}{1-\delta}$, it follows that the following inequality is satisfied:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \left(1 + \frac{\varepsilon+\delta}{1-\delta} \right) \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) \right),$$

which implies that $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is $\frac{\varepsilon+\delta}{1-\delta}$-locally optimal with respect to $N_{\mathcal{X}}$. ■

Note that Proposition 2 remains valid for $\varepsilon = 0$. A similar finding has been briefly discussed by [74] in the more restrictive context of bilevel knapsack (symmetric) interdiction problems. Thus, under the assumptions of Proposition 2, a natural relationship between weak and approximate local optimality is established whenever the follower's problem is solved approximately. Specifically:

Corollary 1. *Under the same assumptions as in Proposition 2, any weak local optimal solution with respect to* $N_{\mathcal{X}}$ *and* $\mathcal{A}$ *is* $\left(\frac{\delta}{1-\delta}\right)$*-locally optimal with respect to* $N_{\mathcal{X}}$.

No assumptions on the upper and lower levels. We apply classical proximity theory for the MILP value function [12, 54] and extend Proposition 2 to a more general setting, where we do not impose any assumption on the symmetry between $\mathbf{c}$ and $\mathbf{d}$. Given $\varepsilon \geq 0$, and $r \geq 0$, define:

$$\Pi\left(\varepsilon, r, \underline{z}\right) := \varepsilon + \frac{\left(2+\varepsilon\right) r d_{\max}}{\underline{z}}, \tag{14}$$

where $\underline{z} > 0$ is some lower bound for [**BP**] and $d_{max} := \max_{\ell \in [m]} \{d_\ell\}$ is the maximum element of vector $\mathbf{d}$. Similarly, define $d_{min} := \min_{\ell \in [m]} \{d_\ell \,:\, d_\ell > 0\}$ and $a_{min} := \min_{i \in [n]} \{a_i \,:\, a_i > 0\}$. If the follower's decision variables are all binary, then one can select the lower bound $\underline{z} = \min\{a_{\min}, d_{\min}\}$ since $z^* > 0$ by Assumption **A4**. More generally, one can construct such a lower bound by first relaxing the optimality criteria in [**BP**] and then solving its LP relaxation. If the resulting bound is not positive, however, further refinement may be required; we omit these details for space reasons.

Next, we demonstrate that, given $\underline{z} > 0$, a weak ε-local optimal solution with respect to an algorithm $\mathcal{A}$, which is assumed to always return a δ-approximation to the lower-level problem, is actually $\Pi\left(\varepsilon, \gamma_1\delta + \gamma_2, \underline{z}\right)$-locally optimal for some $\gamma_1, \gamma_2 \geq 0$. Formally:

Theorem 1. *Let* $\varepsilon \geq 0$*,* $\delta \in \mathbb{Q} \cap [0, 1)$*,* $N_{\mathcal{X}}$ *be a neighborhood function,* $\mathcal{A}$ *be a* δ*-approximation algorithm for the lower-level problem* (1c)*, and* $\underline{z} > 0$ *be a lower bound for the leader's optimal objective function value. There exist* $\gamma_1 \geq 0$ *and* $\gamma_2 \geq 0$ *such that, if* $\mathbf{x}^{\varepsilon,\mathcal{A}}$ *is weakly* ε*-local optimal with respect to* $N_{\mathcal{X}}$ *and* $\mathcal{A}$*, then* $\mathbf{x}^{\varepsilon,\mathcal{A}}$ *is* $\Pi\left(\varepsilon, \gamma_1\delta + \gamma_2, \underline{z}\right)$*-locally optimal with respect to* $N_{\mathcal{X}}$*. Also, if the follower's decision variables are either all continuous or all binary, then* $\gamma_2 = 0$.

If the follower's problem contains only binary variables and admits an approximation scheme with arbitrary performance guarantee δ, then Theorem 1 implies that the approximation guarantee Π is entirely controlled by ε and δ. In this case, Π satisfies $\lim_{(\varepsilon,\delta)\to(0,0)} \Pi\left(\varepsilon, \gamma_1\delta, \underline{z}\right) = 0$. The same reasoning applies when the lower-level problem is an LP solved approximately, e.g., via first-order methods [4]. Before proving Theorem 1, we establish a technical lemma and provide its proof.

Lemma 1. *Let $\mathcal{A}$ be an algorithm that returns a feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the follower's problem* (1c)*, which is within a neighborhood of the optimal solution $\mathbf{y}^*(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$, i.e.,*

$$\left\|\mathbf{y}^{\mathcal{A}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\right\|_1 \leq r, \tag{15}$$

where $r > 0$ does not depend on $\mathbf{x}$. If $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is a weak ε-local optimal solution with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is also $\Pi(\varepsilon, r, \underline{z})$-locally optimal with respect to $N_{\mathcal{X}}$.

Proof. Assume that $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is a weak ε-local optimal solution with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$. Then, consider the mapping $\mathbf{d} : \mathbb{R}^m \to \mathbb{R}$ defined as $\mathbf{y} \mapsto \mathbf{d}^\top \mathbf{y}$, which is $d_{\max}$-Lipschitz continuous. By the Lipschitz continuity property of $\mathbf{d}$, together with (15), we have that for any $\mathbf{x} \in \mathcal{X}$:

$$\left\|\mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 \leq d_{\max} \left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 \leq r d_{max},$$

which implies that the following inequalities are satisfied:

$$-r d_{\max} \leq \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \leq r d_{\max}. \tag{16}$$

Given that (16) holds for any leader's feasible decision $\mathbf{x}$, we conclude that:

$$\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) + r d_{\max}, \tag{17}$$

and, in particular, (16) holds for $\mathbf{x}^{\varepsilon,\mathcal{A}}$. That is:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) + r d_{\max}. \tag{18}$$

Next, fix an arbitrary leader's feasible decision in the neighborhood of $\mathbf{x}^{\varepsilon,\mathcal{A}}$, say, $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$. By the definition of weak ε-local optimality, the following inequality holds (recall Definition 4):

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right). \tag{19}$$

Using the previously derived inequalities and the definition of weak ε-local optimality, we obtain:

$$\begin{aligned}
\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) &\leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) + r d_{\max} && \text{by (18)} \\
&\leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right) + r d_{\max} && \text{by (19)} \\
&\leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) + r d_{max}\right) + r d_{\max} && \text{by (17)} \\
&\leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x})\right) + (2+\varepsilon) r d_{\max}.
\end{aligned}$$

Since $\underline{z} > 0$ is a strictly positive lower bound to [**BP**], for any $\mathbf{x} \in \mathcal{X}$, we have that:

$$1 \leq \frac{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x})}{\underline{z}}.$$

Hence, the following inequality is satisfied:

$$(2+\varepsilon) r d_{\max} \leq (2+\varepsilon) r d_{\max} \frac{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x})}{\underline{z}},$$

which implies that the following inequality is satisfied:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \left(1 + \varepsilon + \frac{(2+\varepsilon)\, r d_{\max}}{\underline{z}}\right) \Big(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x})\Big).$$

Therefore, $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is $\Pi\left(\varepsilon, r, \underline{z}\right)$-locally optimal with respect to $N_{\mathcal{X}}$. ■

We now present the proof of Theorem 1. The argument relies on proximity theory to establish a relation between the approximation quality of $\mathcal{A}$, denoted by δ, and the parameter r in Lemma 1. In essence, if the approximation guarantee δ is sufficiently small, then the follower's feasible decision obtained with $\mathcal{A}$ lies within a neighborhood of the follower's rational response, with the neighborhood size shrinking as δ decreases. Formally:

Proof of Theorem 1. For simplicity of our discussion, we assume that the follower's rational response is uniquely defined. At the end of the proof, we discuss how this assumption can be relaxed in an appropriate manner to make sure that the general result holds. To proceed, we show that, if $\mathcal{A}$ returns a follower's feasible decision that is within a certain neighborhood of the follower's rational response, then a weak ε-local optimal solution with respect to $\mathcal{A}$ is approximate locally optimal.

Next, we discuss several classical proximity theory results, which we exploit to show Theorem 1. Assume that a δ-approximation algorithm $\mathcal{A}$ is available to solve the follower's problem (1c) for any given leader's decision $\mathbf{x} \in \mathcal{X}$, where $\delta \in \mathbb{Q} \cap [0, 1)$. Also, fix a leader's feasible decision $\mathbf{x} \in \mathcal{X}$.

Furthermore, we introduce $\mathbf{F}_1$ and $\mathbf{F}_2$, the constraint matrices associated with the binary and continuous follower's decision variables, respectively, in the follower's feasible set (2). Similarly, we introduce $\mathbf{c}_1$ and $\mathbf{c}_2$, the cost vectors in the follower's objective function associated with the binary and continuous follower's decision variables, respectively. That is, given $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$, we have:

$$\mathbf{F}\mathbf{y} = \mathbf{F}_1 \mathbf{y}_1 + \mathbf{F}_2 \mathbf{y}_2 \quad \text{and} \quad \mathbf{c}^\top \mathbf{y} = \mathbf{c}_1^\top \mathbf{y}_1 + \mathbf{c}_2^\top \mathbf{y}_2,$$

where $\mathbf{y}_1 \in \{0,1\}^{m_1}$ and $\mathbf{y}_2 \in \mathbb{R}_+^{m_2}$. We define the parameterized vector $\mathbf{b}\left(\delta\right) \in \mathbb{R}^{q+1}$ as follows:

$$\mathbf{b}(\delta) := \Big((\mathbf{f} - \mathbf{L}\mathbf{x})^\top, -(1-\delta)\varphi(\mathbf{x})\Big)^\top,$$

where $\varphi(\mathbf{x})$ is the follower's optimal value function at $\mathbf{x}$. Next, we define matrices $\tilde{\mathbf{F}}_{\mathbf{1}}$ and $\tilde{\mathbf{F}}_{\mathbf{2}}$:

$$\tilde{\mathbf{F}}_{\mathbf{1}} := \left(\mathbf{F}_1^\top, -\mathbf{c}_1\right)^\top \quad , \quad \tilde{\mathbf{F}}_{\mathbf{2}} := \left(\mathbf{F}_2^\top, -\mathbf{c}_2\right)^\top.$$

Moreover, the follower's optimal decision, obtained by solving the follower's problem (1c) in [**BP**], can actually be found by solving the following mixed-integer optimization problem:

$$\begin{aligned} \min_{\mathbf{y}_1,\mathbf{y}_2,\mathbf{s}} \quad & 0 \\ \text{s.t.} \quad & \tilde{\mathbf{F}}_1\mathbf{y}_1 + \tilde{\mathbf{F}}_2\mathbf{y}_2 + \mathbf{s} = \mathbf{b}\left(0\right), \\ & \mathbf{y}_1 \in \{0,1\}^{m_1}, \ \mathbf{y}_2 \in \mathbb{R}_+^{m_2}, \ \mathbf{s} \in \mathbb{R}_+^{q+1}, \end{aligned} \tag{21}$$

where variables $\mathbf{s}$ (referred to as "slack" variables), are introduced to formulate the follower's feasible set with equality constraints rather than inequalities.

Also, $\mathbf{y}^*(\mathbf{x})$ is a follower's feasible decision with binary and continuous components that are assumed to be given by $\mathbf{y}_1^*(\mathbf{x})$ and $\mathbf{y}_2^*(\mathbf{x})$, respectively. In fact, $\mathbf{y}^*(\mathbf{x})$ is the optimal solution of (1c) if and only if there exists $\mathbf{s}^*(\mathbf{x}) \in \mathbb{R}_+^{q+1}$ such that $(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))$ is the optimal solution of (21).

Similarly, we consider a problem, where the follower's optimality condition is relaxed. That is:

$$\begin{aligned} \min_{\mathbf{y}_1,\mathbf{y}_2,\mathbf{s}} \quad & 0 \\ \text{s.t.} \quad & \tilde{\mathbf{F}}_1\mathbf{y}_1 + \tilde{\mathbf{F}}_2\mathbf{y}_2 + \mathbf{s} = \mathbf{b}\left(\delta\right) \\ & \mathbf{y}_1 \in \{0,1\}^{m_1}, \ \mathbf{y}_2 \in \mathbb{R}_+^{m_2}, \ \mathbf{s} \in \mathbb{R}_+^{q+1}, \end{aligned} \tag{22}$$

which is feasible since a feasible solution can be derived from the follower's feasible decision $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ obtained by calling $\mathcal{A}$. Indeed, one can select the binary and continuous components $\mathbf{y}_1^{\mathcal{A}}(\mathbf{x})$ and $\mathbf{y}_2^{\mathcal{A}}(\mathbf{x})$, respectively, and then compute the slack variable $\mathbf{s}^{\mathcal{A}}(\mathbf{x})$.

We are interested in the relationship between changes in the right-hand side (r.h.s.) and changes in the optimal solution of (22). We rely on results by [54], which show that, if (22) does not contain any binary variables (i.e., $m_1 = 0$), then the mapping from the set of r.h.s. vectors to the set of feasible solutions to (22) is Lipschitz continuous. Formally:

Theorem 2 (Theorem 2.4 by [54])**.** *Assume that $\tilde{\mathbf{F}}_1 = \mathbf{0}$ and that $\left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top$ is an optimal solution of (22). Then, there exists $(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))^\top$, an optimal solution of (21), and a positive constant $\mu_1\left(\tilde{\mathbf{F}}_2\right) > 0$ that depends only on $\tilde{\mathbf{F}}_2$ such that:*

$$\left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - (\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))^\top \right\|_\infty \le \mu_1\left(\tilde{\mathbf{F}}_2\right) \|\mathbf{b}(0) - \mathbf{b}(\delta)\|_1 .$$

A direct consequence of Theorem 2 is that whenever $\tilde{\mathbf{F}}_1 = 0$, we have:

$$\begin{aligned} \left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 &\le \left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - (\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))^\top \right\|_1 \\ &\le (m+q+1) \left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - (\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))^\top \right\|_\infty \\ &\le \mu_1\left(\tilde{\mathbf{F}}_2\right) \cdot (m+q+1)\, \delta\varphi(\mathbf{x}), \end{aligned}$$

where the last inequality follows from the definition of $\mathbf{b}(\delta)$, i.e., $\|\mathbf{b}(0) - \mathbf{b}(\delta)\|_1 = \delta\varphi(\mathbf{x})$.

Interestingly, a similar result has been presented earlier by [12], where the decision variables are mixed-integer or pure integer. To leverage the result by [12], the parameters in (22) have to be integers. This is not a restrictive assumption in our case.

Indeed, recall that all the parameters in the bilevel program are assumed to be rational. Therefore, without loss of generality, we assume that the entries of $\mathbf{F}$, $\mathbf{L}$, and $\mathbf{f}$ in the follower's feasible set (1c) are all integers. This assumption is justified by the fact that we can multiply every component of $\mathbf{F}$, $\mathbf{L}$, and $\mathbf{f}$ by some large integer in order to obtain integer entries. As a result, for a given $\mathbf{x} \in \mathcal{X}$, and since $\delta \in \mathbb{Q}$, we can assume without loss of generality that $\mathbf{b}(\delta)$, $\tilde{\mathbf{F}}_1$ and $\tilde{\mathbf{F}}_2$ have only rational entries. Hence, we assume they have integral entries. Consequently, the slack vector $\mathbf{s}$ as described earlier can also be assumed to have integer components whenever $m_2 = 0$.

Theorem 3 (Theorem 2.1 from [12])**.** *There exist two constants $\mu_2 > 0$ and $\mu_3 \geq 0$ that do not depend on the r.h.s of (22) such that if $\left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top$ is the optimal solution of (22), then there exists an optimal solution $\left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top$ of (21) such that:*

$$\left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top \right\|_1 \leq \mu_2 \left\| \mathbf{b}(0) - \mathbf{b}(\delta) \right\|_1 + \mu_3.$$

In addition, if $\tilde{\mathbf{F}}_{\mathbf{2}} = \mathbf{0}$, that is, if the follower's decision variables are all binary, then $\mu_3 = 0$.

The constants μ_2 and μ_3 in Theorem 3 depend only on the follower's parameters, that is, $\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2$ and $\mathbf{c}$. Moreover, the optimal solution of the follower's problem (1c) is always non-negative and bounded; recall Assumptions **A2** and **A4**. Thus, we have that $\varphi(\mathbf{x}) \leq \|\mathbf{c}\|_\infty mU$. Next, we define:

$$\gamma_1 = \max\left\{\mu_1\left(\tilde{\mathbf{F}}_{\mathbf{2}}\right) \cdot m(m+q+1)\|\mathbf{c}\|_\infty U, \mu_2 \cdot m\|\mathbf{c}\|_\infty U\right\} \quad \text{and} \quad \gamma_2 = \mu_3.$$

As a direct consequence of Theorems 2 and 3, given the follower's feasible decision $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the follower's problem (1c), and the follower's optimal decision $\mathbf{y}^*(\mathbf{x})$, we have that:

$$\left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 \leq \left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top \right\|_1 \leq \gamma_1\delta + \gamma_2, \tag{23}$$

where $\gamma_2 = 0$ whenever the follower's decision variables are either all binary or all continuous. Another observation is that γ_1 and γ_2 do not depend on the leader's feasible decision $\mathbf{x}$.

From (23), we conclude that r in Lemma 1 can be set to $r = \gamma_1\delta + \gamma_2$. Consequently, Theorem 1 follows from Lemma 1. That is, any weak ε-local optimal solution with respect to a δ-approximation algorithm is, indeed, an approximate local optimal solution to [**BP**].

Finally, recall our assumption on the uniqueness of the follower's rational response. We exploit this assumption in our proof above as Theorems 2 and 3 only provide the existence of a follower's rational response to problem (21). Indeed, without the uniqueness assumption, the follower's decisions obtained in Theorems 2 and 3 may not be the optimistic ones. To relax this uniqueness assumption and to provide the proof for the more general case, the objective functions of problems (21) and (22) can be modified by replacing 0 with $\mathbf{d}_1^\top \mathbf{y}_1 + \mathbf{d}_2^\top \mathbf{y}_2$ (where $\mathbf{d}_1$ and $\mathbf{d}_2$ are defined in a similar manner to $\mathbf{c}_1$ and $\mathbf{c}_2$). However, Theorem 3 requires that if the right-hand sides equal zero in problems (21) and (22), then the corresponding optimal objective function values must also be equal to zero [12]. Hence, to ensure that this additional requirement holds, we need to introduce another variable and an extra constraint into problems (21) and (22), similar to the approach, which is used to justify Assumption **A4** in Supplemental Material S.M.1. ■

Empirical analysis. The approximation guarantee $\Pi(\varepsilon, \gamma_1\delta + \gamma_2, \underline{z})$ in Theorem 1 satisfies $\lim_{(\varepsilon,\delta)\to(0,0)} \Pi(\varepsilon, \gamma_1\delta, \underline{z}) = 0$ whenever $\gamma_2 = 0$. In what follows, we assess this convergence empirically. Accordingly, for any leader's feasible decision $\mathbf{x} \in \mathcal{X}$, we define the *empirical maximum gap* as:

$$\max_{\tilde{\mathbf{x}} \in N_{\mathcal{X}}(\mathbf{x})} \left(\frac{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) - \mathbf{a}^\top \tilde{\mathbf{x}} - \mathbf{d}^\top \mathbf{y}^*(\tilde{\mathbf{x}})}{\mathbf{a}^\top \tilde{\mathbf{x}} + \mathbf{d}^\top \mathbf{y}^*(\tilde{\mathbf{x}})} \right)^+ .$$

We evaluate this quantity on an asymmetric interdiction problem related to the maximum weighted clique interdiction problem described in Section 3.3. Figure 2 reports the empirical maximum gap attained by $\Pi(\varepsilon, \gamma_1\delta, \underline{z})$-local optimal solutions across various values of δ and ϵ. Instance construction and computational details are relegated to the Supplemental Material S.M.3.

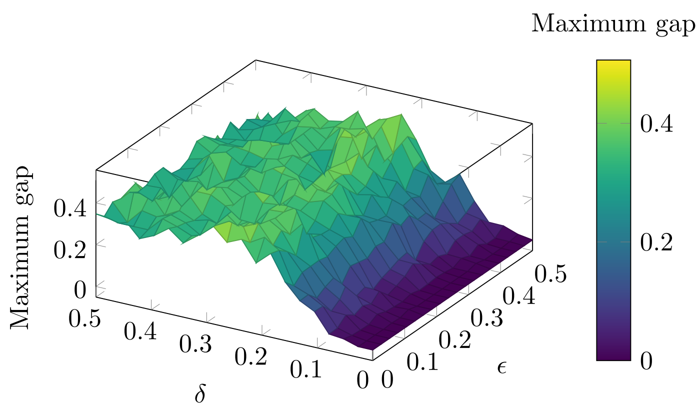


Figure 2: Surface plot of the empirical maximum gap of a leader's decision $x^{\varepsilon,\mathcal{A}}$, which is weakly ε-local optimal with respect to the 2-flip neighborhood function and a δ-approximation algorithm. The plot shows the gap as a function of δ and ε for an instance of the asymmetric interdiction clique problem. Details on the construction of the instances and the figure are provided in the Supplemental Material S.M.3.

Consistent with Theorem 1, we observe that as both δ and ε approach zero, the empirical gap converges to zero. Notably, for this instance, and again as anticipated by Theorem 1, the parameter δ exerts a more significant influence than ε on the magnitude of the empirical gap. Conversely, for larger values of δ, the empirical gap may become relatively large, as there may still exist leader's feasible decisions within the neighborhood of $\mathbf{x}$ that reduce the leader's objective by 40%.

5 Finding a weak approximate local optimal solution

In Section 5.1, we introduce the weak approximate local search, also referred to as $(\varepsilon, \mathcal{A})$-`LSA`. Then, in Section 5.2, we explore the worst-case performance guarantees of $(\varepsilon, \mathcal{A})$-`LSA`, both its runtime complexity and the quality of the obtained solutions, under the assumption that the follower's decision variables are all binary. Finally, in Section 5.3, we extend these results to capture bilevel problems for which the follower's decision variables are mixed-integer.

5.1 Weak approximate local search

A key step in $(\varepsilon, \mathcal{A})$-`LSA` consists of selecting, possibly strategically, an improving solution in the neighborhood of a given leader's feasible decision. This phase is known as a *neighborhood search*. Naturally, we can extend this procedure to improving solutions in terms of the leader's objective function evaluated with an inexact follower; recall our discussion in Section 3.2.

Neighborhood search. The neighborhood of a leader's feasible decision might be very large, making it challenging to effectively find an improving solution. Various strategies exist for the neighborhood search in the single-level optimization context, which typically depend on the considered neighborhood function, or the structure of the underlying problem [3]. Also, multiple improving solutions may exist within the neighborhood, necessitating some tie-breaking rules.

Algorithm 1 - Improved - oracle for the neighborhood search

function Improved($\mathbf{x}^k, N_{\mathcal{X}}, \mathbf{a}, \mathbf{d}, \mathcal{A}, \gamma$)
  improved $\leftarrow$ FALSE, $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$
  if $\exists\, \mathbf{x}^{k+1} \in N_{\mathcal{X}}(\mathbf{x}^k)$ such that $\Delta^{\mathcal{A}}\left(\mathbf{x}^k, \mathbf{x}^{k+1}, \mathbf{a}, \mathbf{d}\right) > \gamma$ **then**
    improved $\leftarrow$ TRUE
  return *improved*, $\mathbf{x}^{k+1}$

Importantly, in our discussion, we do not impose any restrictions on the specific approach used for the neighborhood search. Hence, we introduce an oracle Improved, which is described in Algorithm 1. Specifically, given a leader's feasible decision $\mathbf{x}^k \in \mathcal{X}$ and $\gamma \geq 0$, Improved answers the following question: is there an improving solution in the neighborhood of $\mathbf{x}^k$ that decreases the leader's objective function value with an inexact follower (i.e., obtained by calling $\mathcal{A}$) by at least γ?

Weak approximate local search. In the context of single-level combinatorial optimization, *scaling* the entries of the vector in the objective function is a commonly used technique, which plays an important role in various fully polynomial-time approximation schemes (FPTAS); see, e.g., [51]. Loosely speaking, scaling is employed to reduce the number of distinct values of the objective

Algorithm 2 - $(\varepsilon, \mathcal{A})$-LSA - weak approximate local search

1: **function** $(\varepsilon, \mathcal{A})$-LSA($\mathbf{x}^0$, $N_{\mathcal{X}}$, $\mathbf{a}$, $\mathbf{d}$, $\mathcal{A}$, ε)
2: $\quad i \leftarrow 0$, $\mathbf{x}^i \leftarrow \mathbf{x}^0$, *scaling* $\leftarrow$ TRUE
3: $\quad$ **while** *scaling* **do**
4: $\quad\quad$ Obtain $\mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)$ by calling $\mathcal{A}$ // Inexact follower
5: $\quad\quad$ $K \leftarrow \mathbf{a}^\top \mathbf{x}^i + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)$, $q_a \leftarrow \frac{K\varepsilon}{4n(1+\varepsilon)}$, $q_d \leftarrow \frac{K\varepsilon}{4(m+1)U(1+\varepsilon)}$
6: $\quad\quad$ $a'_j \leftarrow q_a \left\lceil \frac{a_j}{q_a} \right\rceil$ for $j \in [n]$, $d'_\ell \leftarrow q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil$ for $\ell \in [m]$ // Scaling
7: $\quad\quad$ $\gamma \leftarrow U \left(m + q_d^{-1} \sum_{\ell=1}^m d_\ell\right)^{-1} \sum_{\ell=1}^m d'_\ell$ if $m_2 > 0$ else $\gamma \leftarrow 0$ // Improvement gap
8: $\quad\quad$ $k \leftarrow 0$, $\mathbf{x}^{i,k} \leftarrow \mathbf{x}^i$
9: $\quad\quad$ **while** *scaling* and $\left(\mathbf{a}^\top \mathbf{x}^{i,k} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k}) \geq \frac{K}{2}\right)$ **do**
10: $\quad\quad\quad$ $(\mathit{improved}, \mathbf{x}^{i,k+1}) \leftarrow \textsc{Improved}(\mathbf{x}^{i,k}, N_{\mathcal{X}}, \mathbf{a}', \mathbf{d}', \mathcal{A}, \gamma)$ // Neighborhood search
11: $\quad\quad\quad$ **if** *improved* **then**
12: $\quad\quad\quad\quad$ Obtain $\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k+1})$ by calling $\mathcal{A}$, $\mathbf{x}^{i,k} \leftarrow \mathbf{x}^{i,k+1}$, $k \leftarrow k+1$ // Inexact follower
13: $\quad\quad\quad$ **else**
14: $\quad\quad\quad\quad$ $\mathbf{x}^{\varepsilon,\mathcal{A}} \leftarrow \mathbf{x}^{i,k}$, *scaling* $\leftarrow$ FALSE
15: $\quad\quad$ $\mathbf{x}^{i+1} \leftarrow \mathbf{x}^{i,k}$, $i \leftarrow i+1$
16: $\quad$ **Return** $\mathbf{x}^{\varepsilon,\mathcal{A}}$

function's vector by grouping components that are "sufficiently close" to each other. Using scaling as a preprocessing step can accelerate the convergence of optimization algorithms, but might also diminish the quality of the obtained solutions. To mitigate this issue, a carefully chosen scaling factor is essential to balance the trade-off between efficiency and performance.

The idea of scaling the objective function in local search for single-level combinatorial optimization is explored by [58], where the concept of ε-local search is introduced. Our algorithm is motivated by their approach, but differs in two key ways. *First*, we need to account for the lower-level problem, which requires estimating the follower's response and, therefore, necessitates somewhat different scaling rules as well as more involved proofs. *Second*, the follower's decision variables may be continuous, adding another difficulty compared to [58], which only considers the pure 0-1 case. We address this additional issue by introducing a minimum improving gap $\gamma > 0$, ensuring that any improving solution reduces the leader's objective function by at least γ.

Algorithm 2 begins with an initial leader's feasible decision $\mathbf{x}^0$ and computes the corresponding inexact follower's response (by calling $\mathcal{A}$), together with the leader's objective function value. Subsequently, the cost vectors $\mathbf{a}$ and $\mathbf{d}$ from the leader's objective are scaled; see line 6. The scaling factors depend (adaptively) on the leader's objective function value computed before. If $K = 0$ in line 5, then we use the convention that both scaled vectors are set to $\mathbf{0}$ and $\gamma = 0$.

Then, a weak local search is performed as a subroutine using these adjusted vectors within the while loop at line 9. This subroutine may terminate before finding a weak local optimal solution with respect to $\mathbf{a}'$ and $\mathbf{d}'$. Indeed, if no such solution is found within a "reasonable" number of iterations, as determined by the predefined stopping criteria in the loop at line 9, then $\mathbf{a}$ and $\mathbf{d}$ need to undergo further scaling. Specifically, the loop stops whenever the leader's objective function with inexact follower is reduced by more than half, and the entire procedure repeats. The algorithm continues until it successfully identifies a weak local optimum with respect to $\mathbf{a}'$ and $\mathbf{d}'$.

5.2 Runtime and performance guarantees of $(\varepsilon, \mathcal{A})$-LSA

We assume that both Improved and $\mathcal{A}$ are given, with their runtime complexity denoted by $\mathcal{C}_I$ and $\mathcal{C}_{\mathcal{A}}$, respectively. We examine the theoretical performance guarantees of $(\varepsilon, \mathcal{A})$-LSA, together with the properties of the obtained solution. In particular, our results capture both the case in which the follower's problem is solved approximately, and the other, where it is solved exactly.

Throughout this section, the follower's decision variables are assumed to be all binary. That is, the follower's feasible set is given by $\mathcal{Y}_b(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{X}$. Thus, the upper bound for the follower's decisions from Assumption **A2** is naturally given by $U = 1$; also, $\gamma = 0$ by default in Algorithm 2. In what follows, we present the following results:

- We develop a lower bound for the minimum gap between a leader's feasible decision and an improving solution in its neighborhood; see Lemma 2. We use this result to establish the runtime complexity of $(\varepsilon, \mathcal{A})$-LSA; see Theorem 4.
- By Theorem 4, $(\varepsilon, \mathcal{A})$-LSA terminates for any $\varepsilon > 0$, with a leader's feasible decision $\mathbf{x}^{\varepsilon,\mathcal{A}}$. Then, we demonstrate that the obtained solution is weakly ε-local optimal; see Theorem 5.
- Finally, we present a class of problems (namely, $\mathbf{a} = \mathbf{0}$ and $\mathbf{d} = \alpha\mathbf{1}$ for some $\alpha > 0$) for which $(\varepsilon, \mathcal{A})$-LSA is guaranteed to return a weak local optimal solution; see Theorem 6.

Lemma 2. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Assume that q_a, q_d, K, $\mathbf{a}'$ and $\mathbf{d}'$ are given as in Algorithm 2 at iteration $i \in \mathbb{Z}_{\geq 0}$ and fix $\Delta := \frac{\varepsilon K}{4(1+\varepsilon)(m+1)n}$. Then, for any $\mathbf{x}^{i,k} \in \mathcal{X}$ and any improving solution in its neighborhood $\mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})$, we have that:*

$$\min\left\{\Delta^{\mathcal{A}}(\mathbf{x}^{i,k}, \mathbf{x}^{i,k+1}, \mathbf{a}', \mathbf{d}') \ : \ \mathbf{x}^{i,k} \in \mathcal{X},\ \mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})\right\} \geq \Delta,$$

where $\Delta^{\mathcal{A}}$ is given by (10).

Proof. If $K = 0$, then the desired claim holds trivially because $\mathbf{a}' = \mathbf{0}$ and $\mathbf{d}' = \mathbf{0}$. Hence, we assume $K > 0$ for the remainder of the proof. Consider some $\mathbf{x}^{i,k} \in \mathcal{X}$ and suppose that Improved returns $\left(\text{TRUE}, \mathbf{x}^{i,k+1}\right)$. Then, $\mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})$ is a neighbor of $\mathbf{x}^{i,k}$ that improves the leader's objective function evaluated with the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$.

Recall line 5 of Algorithm 2 (see Section 5.1) for the definition of q_a and q_d. We have that:

$$\begin{aligned}\Delta^{\mathcal{A}}\left(\mathbf{x}^{i,k}, \mathbf{x}^{i,k+1}, \mathbf{a}', \mathbf{d}'\right) &= \mathbf{a}'^{\top}\mathbf{x}^{i,k} + \mathbf{d}'^{\top}\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k}) - \mathbf{a}'^{\top}\mathbf{x}^{i,k+1} - \mathbf{d}'^{\top}\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k+1}) \\ &= \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil \left(x_j^{i,k} - x_j^{i,k+1}\right) + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil \left(y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1})\right) \\ &= q_a \sum_{j=1}^{n} \left\lceil \frac{a_j}{q_a} \right\rceil \Delta_j^{\mathbf{x},i,k} + q_d \sum_{\ell=1}^{m} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k} > 0, \qquad (24a)\end{aligned}$$

where the strict positive inequality in (24a) comes from the fact that $\mathbf{x}^{i,k+1}$ is an improving solution; also, we define $\Delta^{\mathbf{x},i,k}$ and $\Delta^{\mathbf{y},i,k}$ as follows:

$$\Delta_j^{\mathbf{x},i,k} := x_j^{i,k} - x_j^{i,k+1} \text{ and } \Delta_\ell^{\mathbf{y},i,k} := y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}).$$

It follows from (24a) that the gap $\Delta^{\mathcal{A}}\left(\mathbf{x}^{i,k}, \mathbf{x}^{i,k+1}, \mathbf{a}', \mathbf{d}'\right)$ is a linear combination of q_a and q_d with integer coefficients. That is:

$$\Delta^{\mathcal{A}}\left(\mathbf{x}^{i,k}, \mathbf{x}^{i,k+1}, \mathbf{a}', \mathbf{d}'\right) = k_a q_a + k_d q_d,$$

where $k_a = \sum_{j=1}^{n} \left\lceil \frac{a_j}{q_a} \right\rceil \Delta_j^{\mathbf{x},i,k}$, and, similarly, $k_d = \sum_{\ell=1}^{m} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k}$.

Next, we present a bound for the maximum absolute value that can be taken by the integer coefficients $k_a \in \mathbb{Z}$ and $k_d \in \mathbb{Z}$ in this linear combination. We start with the definition of k_a:

$$\begin{aligned}|k_a| = \left| \sum_{j=1}^{n} \left\lceil \frac{a_j}{q_a} \right\rceil \Delta_j^{\mathbf{x},i,k} \right| &\le \sum_{j=1}^{n} \left(\frac{a_j}{q_a} + 1 \right) \qquad (25a) \\ &\le n \left(\frac{a_{max}}{q_a} + 1 \right) = n \left(\frac{4a_{max} n(1+\varepsilon)}{K\varepsilon} + 1 \right) =: b_a, \qquad (25b)\end{aligned}$$

where the equality in (25b) follows from the definition of q_a; see line 5 in Algorithm 2. In the same spirit, we can derive a similar bound for k_d, namely:

$$\begin{aligned}|k_d| = \left| \sum_{\ell=1}^{m} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k} \right| &\le \sum_{\ell=1}^{m} \left(\frac{d_\ell}{q_d} + 1 \right) \qquad (26a) \\ &\le m \left(\frac{d_{max}}{q_d} + 1 \right) = m \left(\frac{4d_{max}(m+1)(1+\varepsilon)}{K\varepsilon} + 1 \right) =: b_d, \qquad (26b)\end{aligned}$$

where the equality in (26b) follows from the definition of q_d and the fact that $U = 1$; recall that the follower's decision variables are assumed to be all binary.

Next, consider the following pure integer linear program:

$$\min_{k_a \in \mathbb{Z},\, k_d \in \mathbb{Z}} \left\{ k_a q_a + k_d q_d \,:\, k_a q_a + k_d q_d > 0\,,\, |k_a| \le b_a\,,\, |k_d| \le b_d \right\}, \qquad (27)$$

which is clearly feasible by simply using (24a). Since the feasible set of (27) is finite, then (27) has at least one optimal solution.

Let (k_a^*, k_d^*) be an optimal solution of problem (27). Consequently, a lower bound for the improvement obtained with $\mathbf{x}^{i,k+1}$ for the leader's objective function value with an inexact follower, and with respect to the vectors $\mathbf{a}'$ and $\mathbf{d}'$, is given by $k_a^* q_a + k_d^* q_d$. That is:

$$k_a^* q_a + k_d^* q_d = k_a^* \frac{\varepsilon K}{4(1+\varepsilon)n} + k_d^* \frac{\varepsilon K}{4(1+\varepsilon)(m+1)} = \frac{\varepsilon K}{4(1+\varepsilon)} \left(\frac{k_a^*}{n} + \frac{k_d^*}{(m+1)} \right),$$

where we again apply the definitions of q_a and q_d.

Observe that $(0, 0)$ is not a feasible solution for (27). Also note that k_a^* and k_d^* cannot be simultaneously strictly negative. Hence, we need to consider only two cases, as outlined below.

First, if $k_a^* \geq 0$ and $k_d^* \geq 0$, then we have that either $k_a^* \geq 1$ or $k_d^* \geq 1$. Therefore:

$$k_a^* q_a + k_d^* q_d \geq \frac{\varepsilon K}{4(1+\varepsilon)} \cdot \frac{1}{\max\{n, m+1\}} \geq \Delta. \tag{28}$$

Assume now that $k_a^* k_d^* < 0$. Without loss of generality, we can assume that $k_a^* < 0$, as the arguments presented below can also be applied whenever $k_d^* < 0$.

By the definitions of k_a^* and k_d^*, we have $k_a^* q_a + k_d^* q_d > 0$, which implies $k_d^* q_d > -k_a^* q_a = |k_a^*| \, q_a$. Dividing both sides of the latter inequality by q_d, which is strictly positive, and given that $\frac{q_a}{q_d} = \frac{m+1}{n}$ from their definitions, we get $k_d^* > \frac{m+1}{n} |k_a^*|$. Recall that (k_a^*, k_d^*) forms an optimal solution of (27). Then, k_d^* is the smallest integer that satisfies $k_d^* > \frac{m+1}{n} |k_a^*|$, i.e:

$$k_d^* = \left\lceil \frac{m+1}{n} |k_a^*| \right\rceil > 0,$$

and, by using the definitions of both q_a and q_d again, we obtain the following relation:

$$k_a^* q_a + k_d^* q_d = \frac{\varepsilon K}{4(1+\varepsilon)(m+1)} \left(\frac{m+1}{n} k_a^* + \left\lceil \frac{m+1}{n} |k_a^*| \right\rceil \right). \tag{29}$$

If $(m+1)\,|k_a^*|$ is a multiple of n, then (k_a^*, k_d^*) satisfies $k_a^* q_a + k_d^* q_d = 0$ by (29), i.e., (k_a^*, k_d^*) is not a feasible solution of (27). Consequently, we assume that $(m+1)\,|k_a^*|$ is not a multiple of n. Therefore, there exists $\alpha \in \mathbb{Z}_{\geq 0}$ and $\beta \in \{1, \cdots, n-1\}$ such that $(m+1)\,|k_a^*| = \alpha n + \beta$, and,

$$\begin{aligned} k_a^* q_a + k_d^* q_d &= \frac{\varepsilon K}{4(1+\varepsilon)(m+1)} \left(-(\alpha + \frac{\beta}{n}) + \left\lceil \alpha + \frac{\beta}{n} \right\rceil \right) \\ &= \frac{\varepsilon K}{4(1+\varepsilon)(m+1)} \left(-\frac{\beta}{n} + \left\lceil \frac{\beta}{n} \right\rceil \right) \geq \frac{\varepsilon K}{4(1+\varepsilon)(m+1)n} =: \Delta, \end{aligned}$$

where the second equality is obtained by using the additive property of the ceiling function, i.e., $\lceil u + k \rceil = \lceil u \rceil + k$ for any $u \in \mathbb{R}$ and $k \in \mathbb{Z}_{\geq 0}$. The last inequality is derived from the fact that the minimum absolute difference between a rational number that is not an integer and the closest integer above it is the inverse of the denominator of the rational number. ■

We exploit Lemma 2 to derive the runtime complexity of $(\varepsilon, \mathcal{A})$-`LSA`. We demonstrate that, in the worst case, $(\varepsilon, \mathcal{A})$-`LSA` requires a polynomial number of calls to Improved and $\mathcal{A}$. Formally:

Theorem 4. *Let $\varepsilon > 0$, $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision, and $\mathbf{x}^0 \in \mathcal{X}$ be an initial leader's feasible decision with the corresponding leader's objective function value $0 < K_0 := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq n a_{\max} + m d_{\max}$. Then, $(\varepsilon, \mathcal{A})$-`LSA` terminates and its runtime complexity is in the order of $\mathcal{O}\big(\frac{1}{\varepsilon} nm \log\big(K_0\big)\left(\mathcal{C}_I + \mathcal{C}_{\mathcal{A}}\right)\big)$.*

Proof. Fix $i \geq 0$. Then, assume that K is the leader's objective function value with an inexact follower that is computed at the iteration i in line 5 of Algorithm 2. We are interested in the worst-case runtime of the algorithm. If $K = 0$ at some iteration i, then the algorithm stops and returns the current leader's feasible decision. Hence, we assume $K > 0$ for the remainder of the proof.

From Lemma 2, if *improved* is TRUE in line 11 of Algorithm 2 when calling Improved, then the improving solution that is obtained by Improved always reduces the leader's objective function value with respect to $\mathbf{a}'$ and $\mathbf{d}'$ by at least:

$$\Delta := \frac{\varepsilon K}{4(1+\varepsilon)(m+1)n}.$$

Hence, the number of calls to Improved between two consecutive iterations i and $i+1$ is:

$$\frac{1}{\Delta}\left(\mathbf{a}'^\top \mathbf{x}^i + \mathbf{d}'^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)\right) = \frac{1}{\Delta}\left(\sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^i + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^i)\right) \tag{31a}$$

$$\leq \frac{1}{\Delta}\left(\sum_{j=1}^{n} q_a \left(\frac{a_j}{q_a} + 1\right) x_j^i + \sum_{\ell=1}^{m} q_d \left(\frac{d_\ell}{q_d} + 1\right) y_\ell^{\mathcal{A}}(\mathbf{x}^i)\right) \tag{31b}$$

$$\leq \frac{1}{\Delta}\left(\sum_{j=1}^{n} \left(a_j + q_a\right) x_j^i + \sum_{\ell=1}^{m} \left(d_\ell + q_d\right) y_\ell^{\mathcal{A}}(\mathbf{x}^i)\right) \tag{31c}$$

$$\leq \frac{1}{\Delta}\left(K + n q_a + (m+1) q_d\right), \tag{31d}$$

where (31a) comes from the definitions of the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. Then, (31b) comes from the property of the ceiling function. Also, the leader's and the follower's variables are all binary. Thus, (31d) holds. We use $m+1$ instead of m in the third term for convenience in the derivations below.

Additionally, by using the definitions of q_a and q_d as computed in the line 5 of Algorithm 2:

$$\frac{n q_a}{\Delta} = \frac{(m+1)\, q_d}{\Delta} = \frac{K\varepsilon}{4\Delta\,(1+\varepsilon)} = n\,(m+1)\,.$$

On the other hand, observe that:

$$\frac{K}{\Delta} = \frac{4n(m+1)(1+\varepsilon)}{\varepsilon} = \mathcal{O}\left(\frac{nm}{\varepsilon}\right),$$

and therefore, using (31d), we have that:

$$\frac{\mathbf{a'}^\top \mathbf{x}^i + \mathbf{d'}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)}{\Delta} = \mathcal{O}\left(\frac{nm}{\varepsilon}\right).$$

By the stopping criteria in the inner loop in line 9 of Algorithm 2, the leader's objective function value with an inexact follower is divided by two between two iterations i and $i+1$. Thus, the number of iterations is bounded by $\mathcal{O}(\log K_0)$, where $K_0 := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) > 0$; recall Assumption **A4**. Therefore, the maximum number of calls to Improved can be estimated as:

$$\mathcal{O}\left(\frac{1}{\varepsilon} nm \log K_0\right).$$

In the worst case, each call to Improved, which has a runtime complexity $\mathcal{C}_I$, is followed by a call to $\mathcal{A}$, which has a runtime complexity $\mathcal{C}_{\mathcal{A}}$. Therefore, each call to Improved contributes to the runtime complexity by the order of $\mathcal{O}(\mathcal{C}_I + \mathcal{C}_{\mathcal{A}})$, concluding the proof. ∎

Next, we show that the resulting leader's decision is weakly ε-local optimal.

Theorem 5. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Then,* $(\varepsilon, \mathcal{A})$-`LSA` *is guaranteed to return a weak ε-local optimal solution of* [**B-BP**] *with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$.*

Proof. By Theorem 4, $(\varepsilon, \mathcal{A})$-`LSA` terminates. Let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ be the solution that is returned by the algorithm, and let $\mathbf{x} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$ be an arbitrary leader's feasible decision in its neighborhood. Let K, q_a and q_d denote the values obtained at the last iteration i_f before the algorithm ends. If $K = 0$ at iteration i_f, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is already weakly ε-local optimal for [**B-BP**] with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$. Hence, we assume $K > 0$ for the remainder of the proof, which implies $q_a, q_d > 0$. Then:

$$\begin{aligned}\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) &= \sum_{j=1}^{n} a_j x_j^{\varepsilon,\mathcal{A}} + \sum_{\ell=1}^{m} d_\ell y_\ell^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \\ &\leq \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^{\varepsilon,\mathcal{A}} + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}),\end{aligned}$$

where the transition from the second to the third inequality follows from the fact that $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is a weak local optimal solution with respect to the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$.

Notably, the algorithm stops exclusively after finding a weak local optimal solution with respect to these scaled vectors; see line 14 of Algorithm 2. Then, it follows that:

$$\begin{aligned}\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) &\leq \sum_{j=1}^{n} q_a \left(\frac{a_j}{q_a} + 1\right) x_j + \sum_{\ell=1}^{m} q_d \left(\frac{d_\ell}{q_d} + 1\right) y_\ell^{\mathcal{A}}(\mathbf{x}) \\ &= \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + nq_a + mq_d \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + nq_a + (m+1)q_d,\end{aligned}$$

where we simply exploit the properties of the ceiling function.

Furthermore, the subsequent inequality results directly from the stopping criterion that is used within the inner loop in line 9 of Algorithm 2:

$$\frac{K}{2} \leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + nq_a + (m+1)q_d,$$

which we exploit to obtain:

$$\frac{\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) - \mathbf{a}^\top \mathbf{x} - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})} \leq \frac{nq_a + (m+1)q_d}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})} \leq \frac{nq_a + (m+1)q_d}{\frac{K}{2} - nq_a - (m+1)q_d} = \varepsilon,$$

which then implies that $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly ε-local optimal; recall Definition 4. ■

The theoretical maximum gap ε obtained by our approach is asymptotically sharp whenever the follower's problem is solved exactly. Specifically, in the Supplemental Material S.M.4.1, we construct an instance of [**B-BP**], where the maximum gap between the solution obtained by $(\varepsilon, \mathcal{A})$-`LSA` and an improving solution in its neighborhood is in the order of $\mathcal{O}(\varepsilon)$.

Approximate local optimality. If algorithm $\mathcal{A}$ provides an exact solution to the follower's problem, then Theorem 5 implies that $(\varepsilon, \mathcal{A})$-`LSA` returns an ε-local optimal solution. Furthermore, if the lower-level problem (1c) possesses a special structure (e.g., unimodularity of $\mathbf{F}$ in $\mathcal{Y}_b$) and Improved is a polynomial-time algorithm (i.e., the neighborhood can be searched efficiently), then an ε-local optimal solution to [**B-BP**] can be found in polynomial time. Conversely, if $\mathcal{A}$ is a δ-approximation algorithm, then Theorem 1 can be leveraged to strengthen Theorem 5. That is:

Corollary 2. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, $\mathcal{A}$ be a polynomial-time algorithm that returns a δ-approximate solution to the lower-level problem* (1c) *for any leader's feasible decision, and $\underline{z} > 0$ be a strictly positive lower bound for the leader's objective function. If* Improved *is a polynomial-time algorithm, then $(\varepsilon, \mathcal{A})$-`LSA` is a polynomial-time algorithm that finds a $\Pi(\varepsilon, \gamma_1\delta, \underline{z})$-local optimal solution with respect to $N_{\mathcal{X}}$, for some $\gamma_1 \geq 0$ and where Π is given by* (14)*.*

Next, we demonstrate the existence of a relatively broad class of problems for which our approach returns a weak local optimal solution. We assume that $\mathbf{a} = \mathbf{0}$ and $\mathbf{d} = \alpha\mathbf{1}$ for some $\alpha > 0$. This setting typically arises in some interdiction problems; see, e.g., [30]. Formally:

Theorem 6. *Let $N_{\mathcal{X}}$ be a neighborhood function, and let $\mathcal{A}$ be an algorithm that returns a feasible decision to the follower's problem* (1c) *for any leader's feasible decision. Let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ be the solution obtained by $(\varepsilon, \mathcal{A})$-`LSA`. Furthermore, assume that $\mathbf{a} = \mathbf{0}$, $\mathbf{d} = \alpha\mathbf{1}$ for some $\alpha > 0$. Then, $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly local optimal with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$.*

Proof. If $K = 0$ at some iteration i of $(\varepsilon, \mathcal{A})$-LSA, then the algorithm terminates and the desired claim holds. Hence, we assume $K > 0$ for the remainder of the proof and only consider iterations i for which $K > 0$, so that $q_a, q_d > 0$. We show that any improvement in the leader's objective function in terms of the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$ is also an improvement in terms of the original leader's objective function and vice versa. Fix $i \geq 0$. Let $\mathbf{x}^{i,k}$ denote a leader's feasible decision obtained within the inner loop at line 9 of Algorithm 2. Assume there exists an improving solution $\mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})$ in terms of the leader's objective function, with an inexact follower and with scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. In other words, the response obtained by IMPROVED is TRUE.

Then, the following holds:

$$
\begin{aligned}
0 &< \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^{i,k} + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^{i,k+1} - \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}) \\
&= \sum_{\ell=1}^{m} q_d \left\lceil \frac{\alpha}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - \sum_{\ell=1}^{m} q_d \left\lceil \frac{\alpha}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}),
\end{aligned}
$$

where the first strict inequality comes from the definition of an improving solution, and the second equality follows from the assumption that $\mathbf{a} = \mathbf{0}$ and $\mathbf{d} = \alpha\mathbf{1}$. Consequently, we have:

$$
0 < \sum_{\ell=1}^{m} y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - \sum_{\ell=1}^{m} y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}), \tag{34}
$$

which holds since $q_d \left\lceil \frac{\alpha}{q_d} \right\rceil$ does not depend on ℓ. Multiplying both sides of (34) by α, we see that $\mathbf{x}^{i,k+1}$ is an improving solution in terms of the original leader's objective function value.

Additionally, the converse statement holds. Indeed, starting from an improving solution in terms of the original leader's objective function, one can show that it is an improving solution with respect to the leader's objective function with scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. The argument is independent of q_a and q_d, so the relation holds for every iteration i, completing the proof. ∎

Theorem 6 also suggests that if, first, the neighborhood can be searched efficiently and, second, $\mathcal{A}$ is both polynomial-time and exact, then $(\varepsilon, \mathcal{A})$-LSA is guaranteed to efficiently return a local optimal solution. Finally, results similar to those developed in this section (including Theorems 4 and 5) remain valid when we relax the integrality constraints for some of the follower's decision variables. Although the corresponding proofs require somewhat different approaches, the obtained results closely mirror those presented in this section. Hence, we only discuss briefly the mixed-integer case in Section 5.3 below, and refer to Supplemental Material S.M.4.2 for the complete discussion. Some relevant extensions of $(\varepsilon, \mathcal{A})$-LSA are detailed in Supplemental Material S.M.4.3.

5.3 Extension to mixed-integer and pure continuous follower

In this section, we highlight the worst-case runtime of $(\varepsilon, \mathcal{A})$-LSA when the lower level is an MILP. Accordingly, we assume that $m_2 > 0$, indicating that at least one of the follower's decision variables is continuous. Consequently, $(\varepsilon, \mathcal{A})$-LSA behaves differently, particularly in line 7 of Algorithm 2, where γ is introduced as a minimum acceptable gap for any improving solution.

If the follower's problem involves continuous variables, then the convergence of $(\varepsilon, \mathcal{A})$-LSA within a polynomial number of improving steps may not be guaranteed through scaling alone. Indeed, there may exist neighbors that only marginally improve the leader's objective function by an infinitesimally small amount. To address this issue, we introduce γ, an adaptively chosen threshold that represents the minimum gap required for a neighbor to be considered as an improving solution.

Proposition 3. *Let $\varepsilon > 0$, $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision, and $\mathbf{x}^0$ be an initial leader's feasible decision with associated leader's objective function value $0 < \tilde{K}_0 := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq n a_{\max} + mU d_{\max}$. Then, $(\varepsilon, \mathcal{A})$-LSA terminates and its runtime complexity is in the order of $\mathcal{O}\big(\frac{1}{\varepsilon} m \log\big(\tilde{K}_0\big)\left(\mathcal{C}_I + \mathcal{C}_{\mathcal{A}}\right)\big)$.*

The complete proof of Proposition 3 is deferred to Supplemental Material S.M.4.2. Even when γ is nonzero, $(\varepsilon, \mathcal{A})$-LSA still returns a weak ε-local optimal solution, and so, in a polynomial number of improving steps. Yet, the factor n disappears from the runtime complexity in Proposition 3 (compared to Theorem 4). Note, however, that n appears implicitly through $\tilde{K}_0$ and $\mathcal{C}_{\mathcal{A}}$ in the runtime complexity. The intuition behind this somewhat surprising observation is that γ ensures that no improving solutions leading to negligible improvements in the leader's objective function, particularly through the follower's decision, are selected during the neighborhood search.

6 Computational study

In this section, we support our theoretical developments by exploring the empirical performance of $(\varepsilon, \mathcal{A})$-LSA introduced in Section 5. We emphasize that, in the worst case, local optimal solutions and their generalizations introduced in our study can be arbitrarily far from the leader's optimal decision. Accordingly, our experiments are not intended to demonstrate the superiority of our approach over exact methods, but rather to explore the trade-offs between efficiency and solution quality relative to standard local search. We are interested in the trade-offs that arise from the choice of ε, and from solving the follower's problem approximately. Our experimental setup along with the metrics that evaluate the quality of the obtained solutions are discussed in Section 6.1.

We select problem instances with increasing lower-level computational complexity, focusing on well-established classes for bilevel MILPs [70]. Specifically, we consider two distinct bilevel MILP classes, where the lower-level problems are known to be computationally difficult. First, in Section 6.2, we consider the knapsack interdiction problem. That is, the follower's problem is a linear mixed 0-1 knapsack problem, which is NP-hard only in a weak sense as it admits a FPTAS in the pure 0-1 case; see, e.g., [9, 31]. Then, in Section 6.3, we examine the maximum weighted clique interdiction problem, where the follower's problem is known to be strongly NP-hard [31]; recall our earlier example in Section 3.3. Loosely speaking, despite the fact that the two considered follower's problems are both NP-hard, they represent, in a sense, two opposite ends of the complexity spectrum.

6.1 Preliminaries and performance measures

All procedures start with $\mathbf{x}^0 = \mathbf{0}$. The experiments restrict attention to the k-flip neighborhood defined in (4), with $k \in \{2, 3\}$, and $k = 2$ unless stated otherwise. Furthermore, Improved iterates over the neighborhood and selects the first improving solution. More precisely, the procedure first considers all neighbors obtained by flipping exactly one component, then all neighbors obtained by flipping exactly two components, and so on up to k flips. Within each fixed flip cardinality, the subsets of flipped indices are generated in lexicographic order. We assume that $(\varepsilon, \mathcal{A})$-`LSA` performs a weak local search when $\varepsilon = 0$; see Supplemental Material S.M.2.2 for the pseudo-code.

Efficiency. We evaluate all considered algorithms using three distinct measures. That is: (i) *Runtime* (Time): it measures the time to termination in seconds; (ii) *Number of improving steps* (ImpSteps): it corresponds to the number of times that Improved returns true; (iii) *Number of calls to* $\mathcal{A}$ ($\text{Call}_{\mathcal{A}}$): it is defined as the number of times the follower's problem is solved using $\mathcal{A}$.

Performance. We use the following three distinct measures. Specifically:

(i) *Improving ratio* (ImpRatio): this metric measures the relative improvement in the leader's objective function (evaluated using the exact follower's response) by comparing solutions obtained by $(\varepsilon, \mathcal{A})$-`LSA` to those obtained by LSA. Formally, given $\mathbf{x}^{\varepsilon,\mathcal{A}}$ and $\mathbf{x}^L$, the leader's feasible decisions returned by $(\varepsilon, \mathcal{A})$-`LSA` and LSA, respectively, ImpRatio is defined as:

$$\text{ImpRatio} = \frac{\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) - \left(\mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^0)\right)}{\mathbf{a}^\top \mathbf{x}^L + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^L) - (\mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^0))},$$

where $\mathbf{x}^0$ is the initial leader's feasible solution used as the starting point for both algorithms.

(ii) *Percentage of better solutions* (BetterSol): it represents the percentage of improving solutions (in terms of the exact follower's response) in the neighborhood of the solution $\mathbf{x}^{\varepsilon,\mathcal{A}}$ obtained by $(\varepsilon, \mathcal{A})$-`LSA`. Indeed, recall from Definition 2 that there may exist improving solutions in the neighborhood of $\mathbf{x}^{\varepsilon,\mathcal{A}}$. By "percentage," we refer to the ratio of improving solutions to the total number of leader's feasible decisions in the neighborhood. Specifically, we count all feasible decisions and determine how many of them are improving ones. If BetterSol= 0%, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is locally optimal.

(*iii*) *Maximum gap* (MaxGap): it measures the largest empirical difference in terms of the leader's objective function values between the solution obtained by one of our algorithms and any improving solution in its neighborhood, considering the exact follower's response. Formally, given $\mathbf{x}^{\varepsilon,\mathcal{A}}$, the leader's feasible decision obtained by calling $(\varepsilon,\mathcal{A})$-`LSA`, MaxGap is computed as follows:

$$\textsc{MaxGap} := \max_{\tilde{\mathbf{x}} \in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})} \left(\frac{\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) - \mathbf{a}^\top \tilde{\mathbf{x}} - \mathbf{d}^\top \mathbf{y}^*(\tilde{\mathbf{x}})}{\mathbf{a}^\top \tilde{\mathbf{x}} + \mathbf{d}^\top \mathbf{y}^*(\tilde{\mathbf{x}})} \right)^+,$$

where $(a)^+ = \max\{a, 0\}$ for any $a \in \mathbb{R}$. If $\mathcal{A}$ always returns the exact follower's response, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is ε-local optimal, and therefore, MaxGap $\leq \varepsilon$. A similar theoretical bound exists whenever the lower-level problem is solved approximately; recall our discussion in Section 4.

Hardware and software. Our algorithms are implemented in Python 3.10. The experiments are conducted in parallel on a cluster equipped with 32 Intel(R) Xeon(R) Gold 6126 CPUs (1 core each), operating at 2.60GHz, running Ubuntu 22.04.3. The MILP solver is Gurobi 11.0.0 and the optimization model is rebuilt at each call. This implementation choice may account for a large share of the computational cost, which suggests that implementation improvements are possible.

6.2 Knapsack interdiction problem (KIP)

Given a set of items, the leader's goal is to strategically interdict some of the items that can be picked by the follower. The leader has an interdiction budget; in addition, the leader is penalized with some cost for each interdiction action. In return, the follower solves a linear mixed 0-1 knapsack problem with the remaining non-interdicted items. Formally:

$$\begin{aligned}
[\textbf{KIP}]: \quad & \min_{\mathbf{x},\mathbf{y}^*(\mathbf{x})} \ \mathbf{a}^\top \mathbf{x} + \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}) \\
& \text{s.t. } \mathbf{x} \in \mathcal{X} := \left\{ \mathbf{x} \in \{0,1\}^n \ : \ \mathbf{1}^\top \mathbf{x} \leq 0.3n \right\}, \\
& \qquad \mathbf{y}^*(\mathbf{x}) \in \operatorname{argmax} \left\{ \mathbf{c}^\top \mathbf{y} : \mathbf{y} \in \{0,1\}^{n_1} \times [0,1]^{n_2} \ , \ \mathbf{F}\mathbf{y} \leq \mathbf{f}, \ \mathbf{y} \leq \mathbf{1} - \mathbf{x} \right\},
\end{aligned}$$

where $n := n_1 + n_2$. In our experiments, the cost vectors are generated using a uniform distribution $\mathcal{U}$. That is, $\mathbf{a}' \sim \mathcal{U}(\{1000, \ldots, 1100\}^n)$, $\mathbf{a} = 0.01 \times \mathbf{a}'$, and $\mathbf{c} \sim \mathcal{U}(\{1000, \ldots, 1100\}^n)$. We use such ranges since they create harder instances in which a leader's feasible decision may have many improving solutions that lead only to small improvements in the leader's objective function value, thereby better illustrating the worst-case behavior studied in our theoretical analysis.

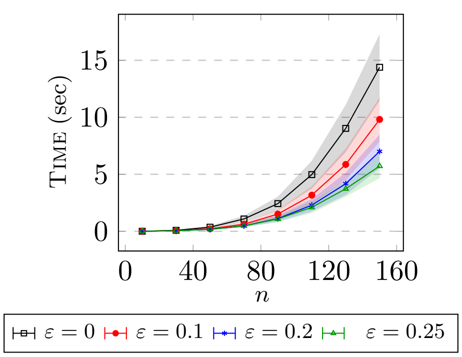
The constraints are generated in a similar manner with $\mathbf{F} \sim \mathcal{U}(\{1000, \ldots, 1100\}^{q \times n})$ and $\mathbf{f} =$

0.4**F1**. Additionally, the leader can interdict up to 30% of the follower's items. In the experiments below, given some parameters n_1, n_2 and q, we always generate 200 instances of [**KIP**], and report the *average* (Avg) and the *mean absolute deviation* (MAD) over these instances.

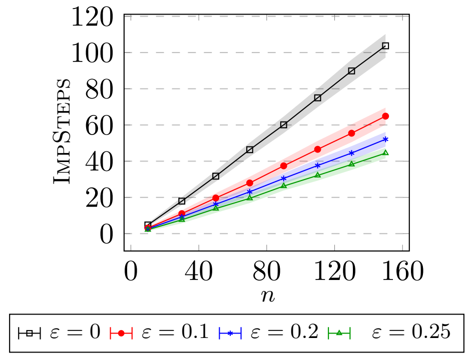
The discussion below is divided into three parts based on whether the follower's decision variables are all continuous ($n_1 = 0$), all binary ($n_2 = 0$), or mixed-integer ($n_1 n_2 > 0$). The three corresponding classes of the follower's problems are distinguished by their complexity: the first one is polynomial-time solvable as it is simply a linear program, while the latter two are NP-hard.

6.2.1 Pure continuous lower level (i.e., $\mathcal{Y} = \mathcal{Y}_c$)

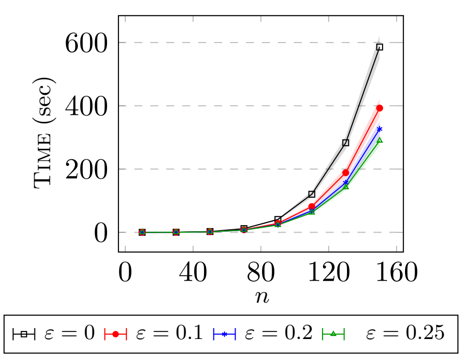
Next, the follower's decision variables are assumed to be all continuous, i.e., $n_1 = 0$, and $n = n_2$, where $n \in \{10, 30, \ldots, 150\}$. Moreover, the lower-level problem contains a single constraint, i.e., $q = 1$. Hence, it is a standard continuous knapsack problem, which is solvable by the greedy algorithm [20]. We let $\varepsilon \in \{0, 0.1, 0.2, 0.25\}$, where $\varepsilon = 0$ corresponds to LSA.



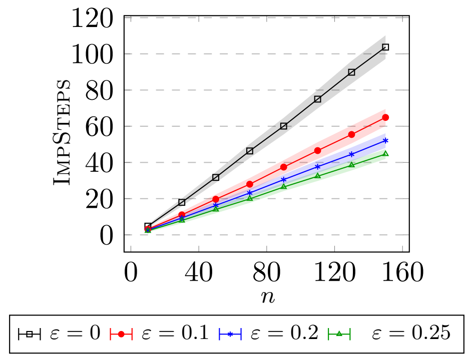
(a) $k = 2$: runtime



(b) $k = 2$: number of improving steps



(c) $k = 3$: runtime



(d) $k = 3$: number of improving steps

Figure 3: **Continuous follower - exact follower ($\delta = 0$) - 2- and 3-flip neighborhoods ($k \in \{2, 3\}$):** comparison of the runtime and the number of improving steps for the knapsack interdiction problem; see Section 6.2.1. Recall that $\varepsilon = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-`LSA`. Each line shows the average (Avg), with the shaded region indicating Avg $\pm$ MAD. Figures 3a and 3b as well as Figure 3c and 3d correspond to the 2- and 3-flip neighborhood, respectively.

The primary interest with the purely continuous case is to isolate the impact of the scaling technique when the follower's problem can be solved efficiently. Figure 3 includes TIME and IMPSTEPS, and offers only a snapshot of our broader analysis, which also covers $\textsc{Call}_{\mathcal{A}}$, BetterSol, MaxGap, and ImpRatio for both the 2-and 3-flip neighborhoods. The complete set of figures is available in the Supplemental Material S.M.5.1. We then make the following observations:

• First, **scaling** the vectors in the leader's objective does **improve the runtime**; see Figure 3a. Specifically, the improvement is more pronounced for larger values of ε, which is intuitive. Note that the runtime does not explode for large values of n, which is consistent with Proposition 3. Indeed, the lower-level problem is a linear program and the leader's neighborhood can be efficiently searched. Therefore, $(\varepsilon, \mathcal{A})$-LSA is guaranteed to be a polynomial-time algorithm.

• The number of improving steps grows roughly linearly in n, even for local search; see Figure 3b. That said, as ε increases, the slope becomes less steep, which is consistent with the theoretical worst-case performance of order $\mathcal{O}\left(\frac{n}{\varepsilon}\right)$ for $(\varepsilon, \mathcal{A})$-LSA; see Lemma 3 in Supplemental Material S.M.4.2.

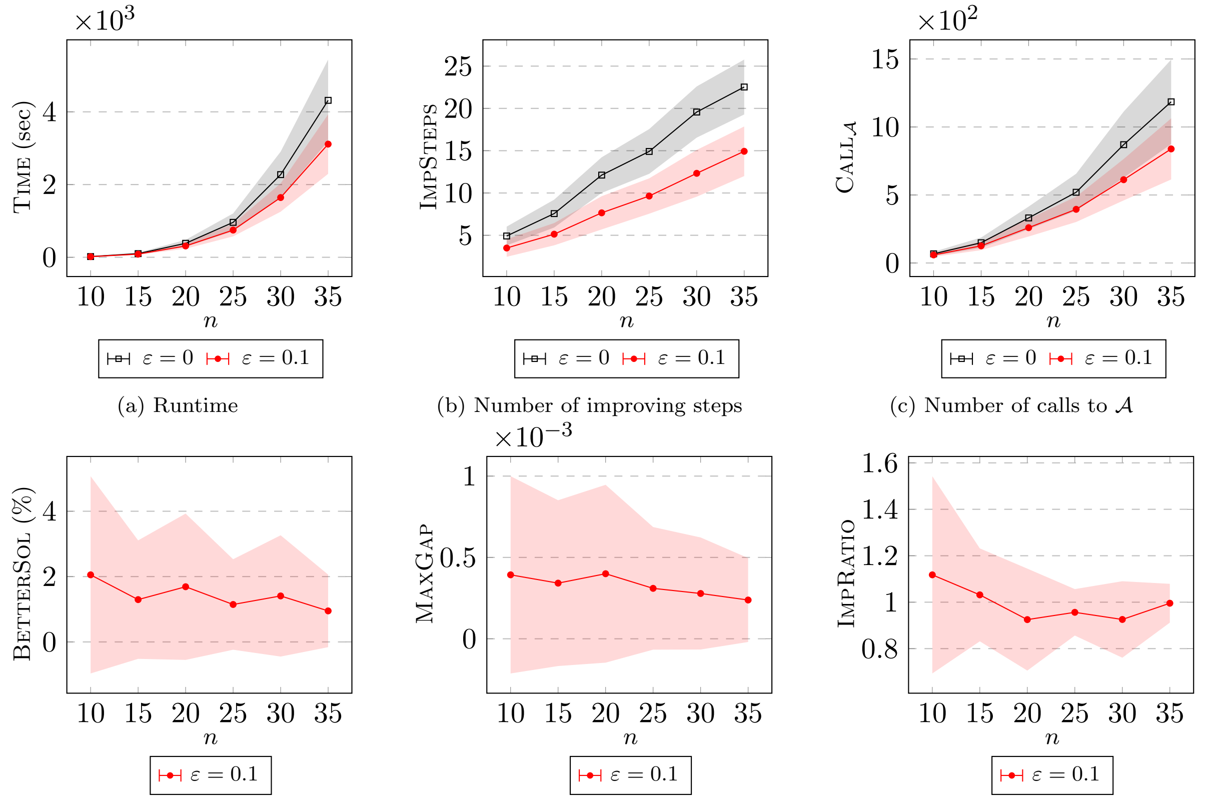
• The above observations remain valid for the 3-flip neighborhood. However, the impact of scaling on the runtime is even more pronounced as the 3-flip neighborhood has a larger cardinality (order of $\mathcal{O}(n^3)$) than the 2-flip neighborhood (order of $\mathcal{O}(n^2)$); compare Figures 3a and 3c. On the other hand, as observed in Figures 3b and 3d, the number of improving steps for the leader remains stable even for a larger neighborhood, which is consistent with Proposition 3.

6.2.2 Pure binary lower level (i.e., $\mathcal{Y} = \mathcal{Y}_b$)

We assume that the follower's decision variables are all binary, i.e., $n_2 = 0$, and $n = n_1$, such that $n \in \{10, 15, \dots, 35\}$. The lower-level problem, a 0-1 knapsack problem with one constraint (i.e., $q = 1$), is solved either exactly using a pseudo-polynomial time algorithm, or approximately with a classical FPTAS, where its performance guarantee is controlled by δ; see, e.g., [9].

We consider four pairs of parameters $(\delta, \varepsilon) \in \{(0,0), (0.1,0), (0,0.1), (0.1,0.1)\}$. Recall that $(\delta, \varepsilon) = (0,0)$ corresponds to standard local search. Pair $(\delta, \varepsilon) = (0.1, 0.1)$ refers to $(\varepsilon, \mathcal{A})$-LSA, where both the scaling technique is used and the follower's problem is solved approximately. Pair $(\delta, \varepsilon) = (0, 0.1)$ represents $(\varepsilon, \mathcal{A})$-LSA, where only scaling is employed, while the follower's problem is solved exactly. Finally, $(\delta, \varepsilon) = (0.1, 0)$ corresponds to the weak local search, where there is no scaling, but the follower's problem is solved approximately. We point out the following observations:

• If the follower's problem is solved exactly, then the isolated effect of scaling can be found in Figure 4. The results on both the runtime (Figure 4a), and the number of improving steps (Figure 4b) are consistent with the ones from Section 6.2.1. Naturally, these observations extend to the number of calls to $\mathcal{A}$, which is also reduced after scaling; see Figure 4c.



(a) Runtime

(b) Number of improving steps

(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

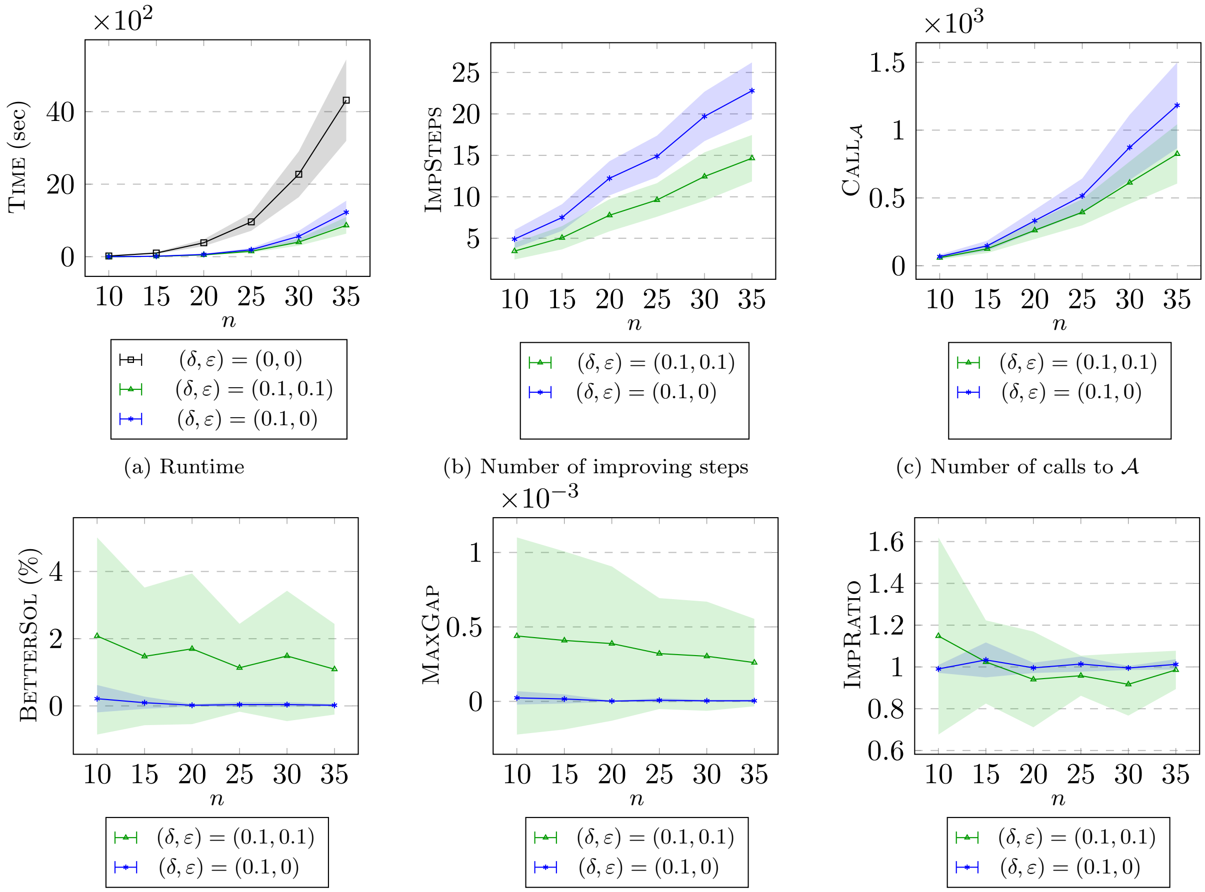
(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-LSA compared to the improvement of LSA

Figure 4: **Binary follower - exact follower ($\delta = 0$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.2. Recall that $\varepsilon = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-LSA. Each line shows the average (Avg), with the shaded region indicating Avg ± MAD.

- Recall that the leader's decision obtained by $(\varepsilon, \mathcal{A})$-LSA is not guaranteed to be locally optimal. We observe this phenomenon in Figure 4d, as there are still on average, around 2% of improving solutions in the neighborhood of the obtained solution. Recall that the maximum theoretical gap of the solutions returned by $(\varepsilon, \mathcal{A})$-LSA is given by $\frac{\varepsilon+\delta}{1-\delta} = \varepsilon$ for $\delta = 0$; see Proposition 2. The **empirical maximum gap**, see Figure 4e, is **much lower** than the theoretical one given by $\varepsilon = 0.1$. Furthermore, the improvement ratio is 1 ± 0.1; see Figure 4f. That is, the solutions obtained by $(\varepsilon, \mathcal{A})$-LSA are of comparable quality to the ones obtained by LSA.

- If we simply **solve the lower-level problem approximately**, but with no scaling, i.e., $(\delta, \varepsilon) = (0.1, 0)$, then we see a **significant reduction in the runtime** compared to LSA; see Figure 5a. Moreover, the runtime is better for $(\varepsilon, \mathcal{A})$-LSA when $\delta = 0.1$ and $\varepsilon = 0.1$ compared to when $\delta = 0.1$ and $\varepsilon = 0$. In fact, whenever the approximation algorithm for the lower-level problem is efficient, we systematically observe this behavior, which is consistent with Theorem 4.

- When the follower's problem is solved exactly, scaling reduces both the number of improving



(a) Runtime

(b) Number of improving steps

(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-`LSA` compared to the improvement of LSA

Figure 5: **Binary follower - inexact follower** ($\delta = 0.1$) **- 2-flip neighborhood** ($k = 2$)**:** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.2. Recall that $\varepsilon = \delta = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-`LSA`. Each line shows the average (Avg), with the shaded region indicating Avg ± MAD.

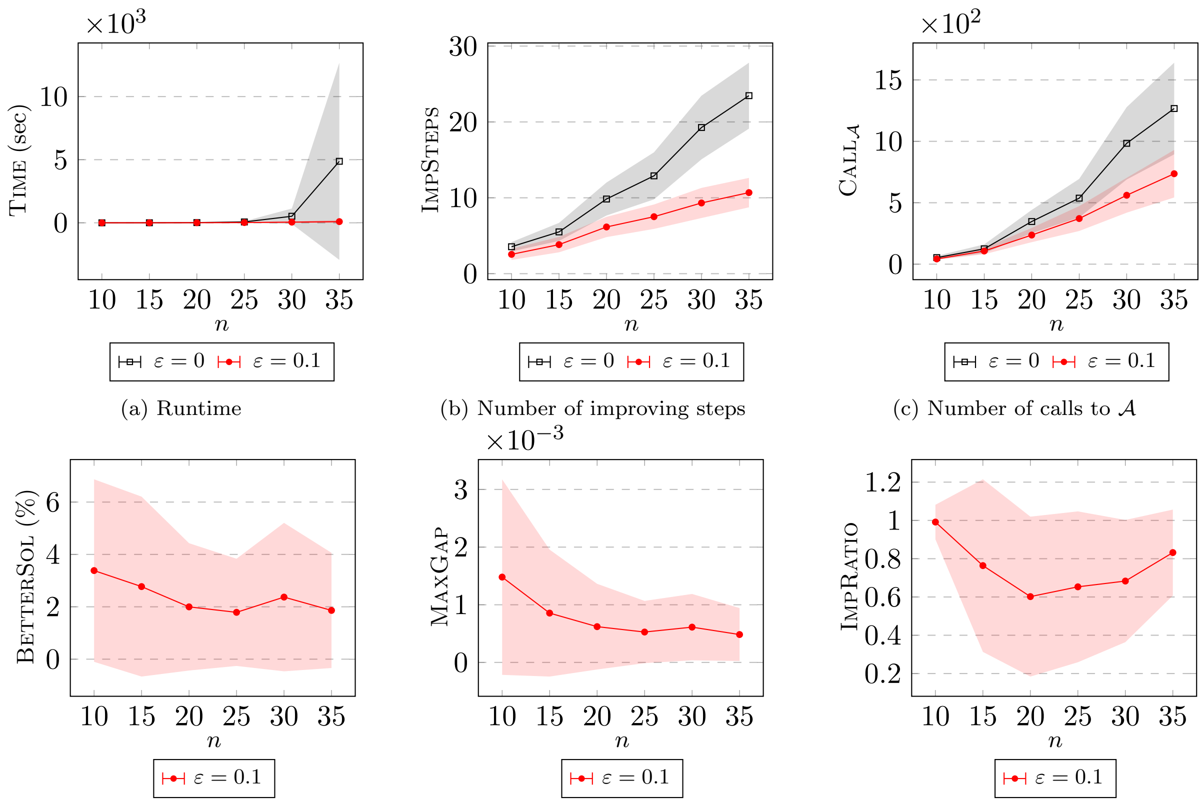
steps and the number of calls to $\mathcal{A}$; recall Figures 4b and 4c. Similarly, it is also the case even when the follower's problem is not solved exactly, but approximately; compare Figures 5b and 5c.

• The significant improvement in the runtime achieved by solving the lower-level problem approximately is accompanied by only a subtle deterioration in the solution quality, as illustrated in Figures 5d, 5e and 5f. In fact, here, scaling has a larger negative effect on the solution quality.

The concluding remark from the experiments in Sections 6.2.1 and 6.2.2 is that both scaling the leader's objective function and solving the lower level approximately improve the runtime. Nonetheless, solving the follower's problem approximately has a more pronounced impact on the runtime for this class of problems; recall $\mathcal{Y} = \mathcal{Y}_b$. Importantly, both techniques still produce similar quality solutions compared to the standard local search algorithm.

6.2.3 Mixed-integer lower level

Next, we assume that the follower's decision variables can be both binary and continuous. That is, we set $n_1 = 0.8n$, $n_2 = 0.2n$, where $n \in \{10, 15, \ldots, 35\}$, and $q = 10$. The mixed-integer knapsack problem is solved using Gurobi, with the MILP optimality gap set to $\delta \in [0, 1)$. We explore pairs of parameters $\varepsilon \in \{0, 0.1\}$ and $\delta \in \{0, 0.005\}$.



(a) Runtime

(b) Number of improving steps

(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

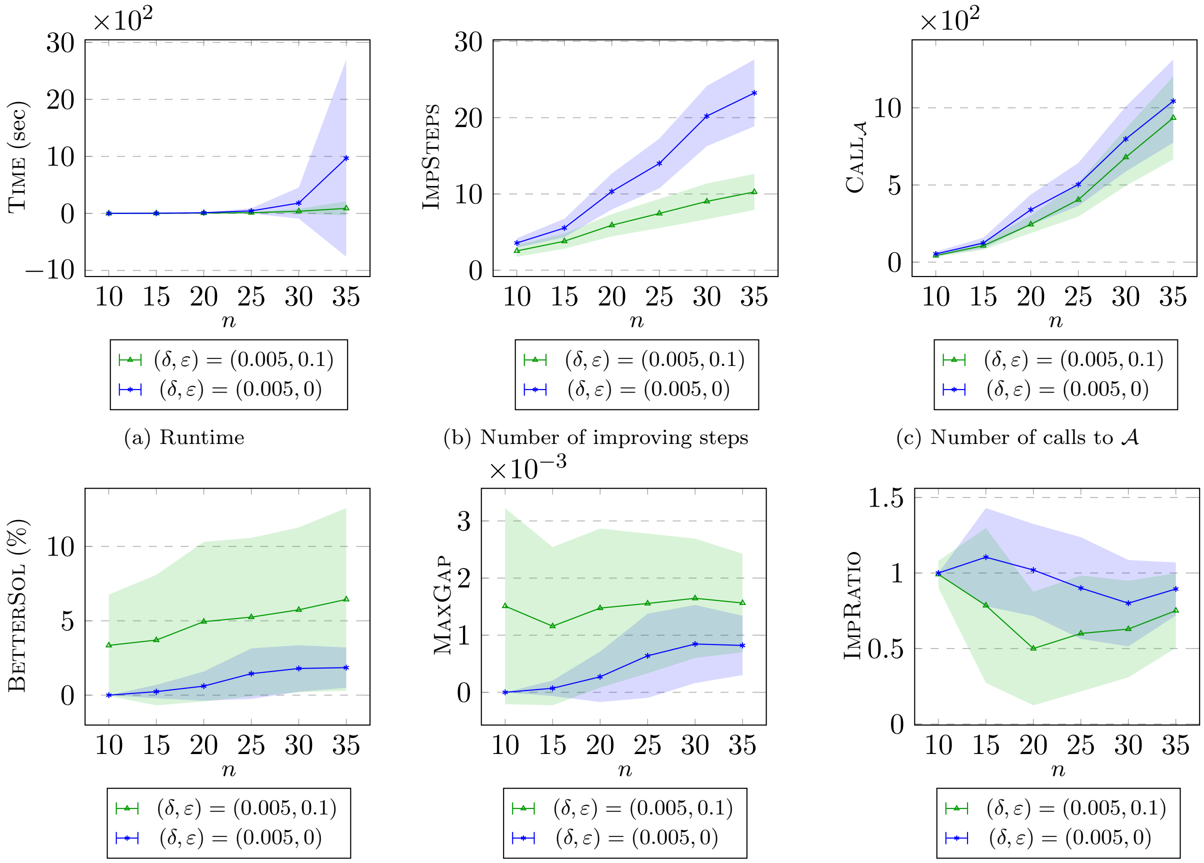
(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-LSA compared to the improvement of LSA

Figure 6: **Mixed-integer follower - exact follower ($\delta = 0$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.3. Recall that $\varepsilon = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-LSA. Each line shows the average (Avg) of the metric, with the shaded region indicating Avg ± MAD.

From our experiments, we observe that:

• As shown in Figure 6a, the **runtime of LSA explodes**, averaging over an hour for a single instance. This behavior can be explained by the computational difficulty of solving the follower's problem. In contrast, **scaling drastically reduces the runtime**. The observed improvement is consistent with the one in the number of improving steps (Figure 6b) and in the number of calls to $\mathcal{A}$ (Figure 6c), and aligns with the previous results in Sections 6.2.1 and 6.2.2.

• The percentage of improving solutions is relatively stable around 2-3%; see Figure 6d. The



(a) Runtime

(b) Number of improving steps

(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-`LSA` compared to the improvement of LSA

Figure 7: **Mixed-integer follower - inexact follower ($\delta = 0.005$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.3. Recall that $\varepsilon = \delta = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-`LSA`. Each line shows the average (Avg), with the shaded region indicating Avg $\pm$ MAD.

maximum gap, in Figure 6e, is much smaller than the theoretical one given by $\frac{\varepsilon+\delta}{1-\delta} = \varepsilon$ for $\delta = 0$; recall Proposition 2. The improvement ratio, between 0.6 and 1, is lower compared to the ones from Section 6.2.1 and Section 6.2.2; see Figure 6f. This quality deterioration is intuitive since, as often with approximate algorithms, there is a trade-off between the solution quality and runtime.

• Recall from Section 6.2.2 that the reduction in the runtime is mostly driven by solving the follower's problem approximately. In contrast, in Figure 7a with $(\delta, \varepsilon) = (0.005, 0)$, the **runtime also explodes when scaling is not applied, but the lower level is solved approximately**. On the other hand, scaling has an effect on the number of improving steps (Figure 7b) and the number of calls to $\mathcal{A}$ (Figure 7c), which is consistent with the observations from the previous sections.

• The percentage of improving solutions (Figure 7d) and the maximum gap (Figure 7e) are also consistent with our previous results. However, for the improving ratio, we observe that it can be down to 0.5; see Figure 7f. This lower ratio is not surprising given the significant reduction in the runtime. Our interpretation is that it serves as a reminder that there is "no free lunch." That is, the decrease in runtime inevitably comes at the cost of a decrease in the solution quality performance.

• It is worth mentioning that the runtime results in Figure 6a and Figure 7a exhibit more noisy behavior compared to those in Section 6.2.1 or Section 6.2.2. This somehow higher noise level is likely due to solving small MILP instances with Gurobi, which finds an approximate solution in less than a second, resulting in larger differences than when solving larger instances.

We conclude from Figures 6 and 7 that either applying, in isolation, scaling or solving the lower level approximately reduces the runtime. However, the latter does not have as much impact as it does in Section 6.2.2. Instead, **scaling is the primary driver of the runtime improvement**.

6.3 Maximum weighted clique interdiction problem

The last set of experiments is on the maximum weighted clique interdiction problem; recall (11) in Section 3.3. We use randomly generated graphs G with n vertices and edge density d. Specifically, we generate 50 instances of Erdős-Rényi graphs [26], where $n \in \{40, 50, 60\}$ and $d \in \{0.5, 0.7, 0.9\}$. The weight of $i \in V$ is $w_i = 10\tilde{w}_i + 1000$, where $\tilde{w}_i \sim \mathcal{U}(\{1, \ldots, deg(i)\})$ and $\deg(i)$ denotes the degree of vertex $i \in V$. The leader's interdiction budget is set to $h = 0.1n$.

		TIME (sec)		IMPSTEPS		CALL$_{\mathcal{A}}$	
n	d	Avg	MAD	Avg	MAD	Avg	MAD
40	0.5	31.5	7.6	7.6	1.7	368	90
40	0.7	26.4	7.0	9.8	2.1	421	103
40	0.9	7.7	2.2	12.0	2.3	470	117
50	0.5	100.2	28.4	9.6	1.9	666	181
50	0.7	84.8	21.4	12.6	2.4	740	191
50	0.9	23.3	6.6	14.9	2.7	794	232
60	0.5	195.6	51.3	11.1	2.3	927	249
60	0.7	178.8	43.7	13.5	2.5	970	243
60	0.9	76.4	23.1	18.6	3.3	1,260	348

Table 2: **Standard LSA - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and performance metric for LSA, where $(\delta, \varepsilon) = (0, 0)$, applied to the maximum weighted clique interdiction problem; recall (11) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

Table 2 contains the results for standard local search, i.e., $\varepsilon = \delta = 0$. Similarly, the results for $(\varepsilon, \mathcal{A})$-`LSA` can be found in Table 3 (below), where both scaling ($\varepsilon = 0.1$) and solving the follower's problem approximately ($\delta = 0.1$) are applied. The follower's maximum weighted clique problem is solved using Gurobi, with the optimality gap set to δ. The isolated effects of either scaling or solving the follower's problem approximately are relegated to the Supplemental Material S.M.5.2.

Comparing Tables 2 and 3, we observe a decrease in the runtime across all instances when

		Time (sec)		ImpSteps		Call$_{\mathcal{A}}$		MaxGap		ImpRatio	
n	d	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD
40	0.5	23.6	7.5	7.1	1.8	341	103	$8.1 \cdot 10^{-3}$	$9.6 \cdot 10^{-3}$	0.89	$1.8 \cdot 10^{-1}$
40	0.7	16.5	7.2	9.2	2.5	394	139	$2.2 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$	0.83	$1.6 \cdot 10^{-1}$
40	0.9	5.9	1.2	10.8	2.2	404	83	$2.2 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	0.89	$8.3 \cdot 10^{-2}$
50	0.5	90.4	31.2	9.2	2.3	627	207	$8.4 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	0.94	$1.0 \cdot 10^{-1}$
50	0.7	65.4	23.6	11.8	2.6	659	207	$7.6 \cdot 10^{-3}$	$8.8 \cdot 10^{-3}$	0.91	$1.3 \cdot 10^{-1}$
50	0.9	15.0	4.0	14.4	2.9	765	203	$2.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	0.88	$9.2 \cdot 10^{-2}$
60	0.5	152.8	40.5	9.7	2.1	742	205	$9.3 \cdot 10^{-3}$	$8.7 \cdot 10^{-3}$	0.89	$1.7 \cdot 10^{-1}$
60	0.7	168.9	47.9	13.3	2.2	948	264	$1.5 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$	0.91	$9.2 \cdot 10^{-2}$
60	0.9	32.5	10.3	18.7	4.1	1,080	314	$2.6 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	0.84	$7.7 \cdot 10^{-2}$

Table 3: **$(\varepsilon, \mathcal{A})$-LSA - inexact follower ($\delta = 0.1$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and performance metric for $(\varepsilon, \mathcal{A})$-LSA, where $(\delta, \varepsilon) = (0.1, 0.1)$, applied to the maximum weighted clique interdiction problem; recall (11) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

using $(\varepsilon, \mathcal{A})$-LSA, instead of LSA. However, this decrease is not as pronounced as in Section 6.2. We believe that this observation can be attributed to the increased computational difficulty of the lower-level problem, which is strongly NP-hard and not approximable. Furthermore, both the number of improving steps and the number of calls to $\mathcal{A}$ are reduced, although the decrease is also less significant compared to Section 6.2. The maximum empirical gap remains stable and much smaller than the theoretical one in Proposition 2. Finally, the improvement ratio is relatively stable around 0.9 ± 0.1. Thus, the solutions from $(\varepsilon, \mathcal{A})$-LSA are of comparable quality to those from LSA.

6.4 Summary insights

We believe that the disparity in efficiency gains achieved by $(\varepsilon, \mathcal{A})$-LSA when applied to the knapsack interdiction and the maximum weighted clique interdiction problems can be attributed to the different computational complexity of their respective lower-level problems. Although the knapsack problem is NP-hard [31], it is only weakly so and admits an FPTAS. In contrast, the maximum clique problem is strongly NP-hard, and hence, it does not admit an FPTAS. In fact, unless P=NP, for any $\varepsilon > 0$, no polynomial-time algorithm can provide an $\mathcal{O}(n^{\frac{1}{2}-\varepsilon})$-approximate solution to the maximum clique problem [36]. Consequently, these two problem classes occupy very different positions within the NP-hardness spectrum. One could argue that most problems of interest in the bilevel optimization literature typically fall between these two extremes, suggesting that the empirical efficiency of $(\varepsilon, \mathcal{A})$-LSA is likely to vary similarly, falling somewhere between the results obtained for the two considered extremes.

The theoretical and empirical efficiency of $(\varepsilon, \mathcal{A})$-LSA relies on two key ideas: adaptively scaling the leader's objective function and evaluating it via an inexact follower's response. Our approach integrates these two ideas within a local search-based algorithm. We observe that applying either idea in isolation is generally insufficient; both are needed to ensure efficiency. Moreover, our computational study demonstrates a trade-off between the runtime gains achieved by using $(\varepsilon, \mathcal{A})$-`LSA` instead of the standard local search algorithm and the quality of the obtained solutions. By accepting a reasonably small decrease in the solution quality compared to LSA, we can achieve (depending heavily on the difficulty of the lower-level problem) a significant reduction in the runtime and in the number of improving steps required for the leader to converge.

To conclude this section, we acknowledge that our computational experiments are primarily restricted to interdiction problems. For completeness, we have therefore also conducted experiments on non-interdiction instances borrowed from the literature [70]. The results of these experiments are provided in the Supplemental Material S.M.5.3.

7 Conclusion and further research directions

In this study, we address two primary challenges encountered by local search in the context of bilevel MILPs. First, we mitigate the worst-case exponential behavior typically associated with the standard local search method by employing advanced scaling techniques and extending the concept of ε-local optimality to the bilevel setting. Second, we tackle the difficulty of computing the follower's optimal decision during the neighborhood search by estimating the follower's response with either an approximate or merely a feasible decision to the lower-level problem.

Our theoretical contributions are supported by numerical experiments. The results demonstrate that both techniques, namely, scaling the leader's objective function and solving approximately the lower-level problem, improve the runtime while preserving a solution quality comparable to that of standard local search. Notably, applying these techniques in isolation is generally insufficient; instead, they must be applied in a unified manner. Moreover, we observe that the runtime improvements are significantly influenced by the complexity of the lower-level problem.

The development of solution methods for mixed-integer linear programs has historically benefited from both cutting-plane techniques and heuristics, including local search methods. While cutting-plane approaches have been successfully adapted to bilevel MILPs, local search has received little attention. Local optimal solutions, though not necessarily globally optimal, remain valuable. Alongside the traditional focus on improving lower bounds in exact methods, exploring approaches that enhance feasibility bounds, similar to the one proposed in this study, may offer a promising and underexplored direction for advancing the field. Integrating these ideas within exact methods could significantly advance solution techniques for bilevel MILPs. Finally, our numerical experiments suggest that the computational complexity of the lower-level problem impacts the performance of our approach. Further empirical and theoretical analysis, focused on expected rather than worst-case behavior, is needed to better understand this phenomenon.

Declarations

Funding and/or Conflicts of Interests/Competing Interests. The authors have no funding to report and no competing interests to declare.

Acknowledgements

We thank the Associate Editor and the Reviewers for their helpful comments and suggestions.

References

[1] Aarts, E. and Lenstra, J. K. *Local Search in Combinatorial Optimization.* Princeton: Princeton University Press, 2003.

[2] Ahuja, R. K., Ergun, Ö., Orlin, J. B., and Punnen, A. P. "A survey of very large-scale neighborhood search techniques". In: *Discrete Applied Mathematics* 123.1-3 (2002), pp. 75–102.

[3] Altner, D. S., Ahuja, R. K., Ergun, Ö., and Orlin, J. B. "Very large-scale neighborhood search". In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques.* Ed. by Burke Edmund K.and Kendall, G. Boston, MA: Springer US, 2013, pp. 339–367.

[4] Applegate, D., Hinder, O., Lu, H., and Lubin, M. "Faster first-order primal-dual methods for linear programming using restarts and sharpness". In: *Mathematical Programming* 201.1 (2023), pp. 133–184.

[5] Arkin, E. M. and Hassin, R. "On local search for weighted k-set packing". In: *Mathematics of Operations Research* 23.3 (1998), pp. 640–648.

[6] Audet, C., Hansen, P., Jaumard, B., and Savard, G. "Links between linear bilevel and mixed 0–1 programming problems". In: *Journal of Optimization Theory and Applications* 93 (1997), pp. 273–300.

[7] Beck, Y., Ljubić, I., and Schmidt, M. "A survey on bilevel optimization under uncertainty". In: *European Journal of Operational Research* 311.2 (2023), pp. 401–426.

[8] Ben-Ayed, O. and Blair, C. E. "Computational difficulties of bilevel linear programming". In: *Operations Research* 38.3 (1990), pp. 556–560.

[9] Bernhard, K. and Vygen, J. *Combinatorial optimization: Theory and algorithms*. Springer, 2008.

[10] Bertsimas, D., Iancu, D. A., and Katz, D. "A new local search algorithm for binary optimization". In: *INFORMS Journal on Computing* 25.2 (2013), pp. 208–221.

[11] Bichler, M. and Waldherr, S. "Core pricing in combinatorial exchanges with financially constrained buyers: Computational hardness and algorithmic solutions". In: *Operations Research* 70.1 (2022), pp. 241–264.

[12] Blair, C. E. and Jeroslow, R. G. "The value function of a mixed integer program: I". In: *Discrete Mathematics* 19.2 (1977), pp. 121–138.

[13] Bolusani, S. and Ralphs, T. K. "A framework for generalized Benders' decomposition and its application to multilevel optimization". In: *Mathematical Programming* 196.1 (2022), pp. 389–426.

[14] Brotcorne, L., Labbé, M., Marcotte, P., and Savard, G. "A bilevel model for toll optimization on a multicommodity transportation network". In: *Transportation Science* 35.4 (2001), pp. 345–358.

[15] Caprara, A., Carvalho, M., Lodi, A., and Woeginger, G. J. "Bilevel knapsack with interdiction constraints". In: *INFORMS Journal on Computing* 28.2 (2016), pp. 319–333.

[16] Chandra, B., Karloff, H., and Tovey, C. "New results on the old k-opt algorithm for the traveling salesman problem". In: *SIAM Journal on Computing* 28.6 (1999), pp. 1998–2029.

[17] Correa, J., Harks, T., Kreuzen, V. J., and Matuschke, J. "Fare evasion in transit networks". In: *Operations Research* 65.1 (2017), pp. 165–183.

[18] Dahan, M., Sela, L., and Amin, S. "Network inspection for detecting strategic attacks". In: *Operations Research* 70.2 (2022), pp. 1008–1024.

[19] Dan, T. and Marcotte, P. "Competitive facility location with selfish users and queues". In: *Operations Research* 67.2 (2019), pp. 479–497.

[20] Dantzig, G. B. "Discrete-variable extremum problems". In: *Operations Research* 5.2 (1957), pp. 266–288.

[21] Deineko, V. G. and Woeginger, G. J. “A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem”. In: *Mathematical Programming* 87 (2000), pp. 519–542.

[22] Dempe, S. “A simple algorithm for the-linear bilevel programming problem”. In: *Optimization* 18.3 (1987), pp. 373–385.

[23] Dempe, S. and Zemkoho, A. B. “The bilevel programming problem: reformulations, constraint qualifications and optimality conditions”. In: *Mathematical Programming* 138 (2013), pp. 447–473.

[24] Dinitz, M. and Gupta, A. “Packing interdiction and partial covering problems”. In: *Integer Programming and Combinatorial Optimization.* Ed. by Goemans, M. and Correa, J. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 157–168.

[25] Dussault, J.-P., Marcotte, P., Roch, S., and Savard, G. “A smoothing heuristic for a bilevel pricing problem”. In: *European Journal of Operational Research* 174.3 (2006), pp. 1396–1413.

[26] Erdos, P., Rényi, A., et al. “On the evolution of random graphs”. In: *Publ. math. inst. hung. acad. sci* 5.1 (1960), pp. 17–60.

[27] Fischetti, M., Ljubić, I., Monaci, M., and Sinnl, M. “A new general-purpose algorithm for mixed-integer bilevel linear programs”. In: *Operations Research* 65.6 (2017), pp. 1615–1637.

[28] Fischetti, M., Ljubić, I., Monaci, M., and Sinnl, M. “Interdiction games and monotonicity, with application to knapsack problems”. In: *INFORMS Journal on Computing* 31.2 (2019), pp. 390–410.

[29] Fischetti, M., Monaci, M., and Sinnl, M. “A dynamic reformulation heuristic for generalized interdiction problems”. In: *European Journal of Operational Research* 267.1 (2018), pp. 40–51.

[30] Furini, F., Ljubić, I., Martin, S., and San Segundo, P. “The maximum clique interdiction problem”. In: *European Journal of Operational Research* 277.1 (2019), pp. 112–127.

[31] Garey, M. R. and Johnson, D. S. *Computers and intractability.* Vol. 174. USA: Freeman San Francisco, 1979.

[32] Glover, F. “Improved linear integer programming formulations of nonlinear integer problems”. In: *Management Science* 22.4 (1975), pp. 455–460.

[33] Gurobi. *Gurobi Optimization, LLC.* `https://www.gurobi.com`. Accessed: 2024-07-26. 2024.

[34] Gutin, E., Kuhn, D., and Wiesemann, W. “Interdiction games on Markovian PERT networks”. In: *Management Science* 61.5 (2015), pp. 999–1017.

[35] Hansen, P., Jaumard, B., and Savard, G. “New branch-and-bound rules for linear bilevel programming”. In: *SIAM Journal on Scientific and Statistical Computing* 13.5 (1992), pp. 1194–1217.

[36] Håstad, J. “Clique is hard to approximate within $n^{1-\varepsilon}$”. In: *Acta Mathematica* 182.1 (1999), pp. 105–142.

[37] Henke, D., Lefebvre, H., Schmidt, M., and Thürauf, J. “On coupling constraints in linear bilevel optimization”. In: *Optimization Letters* 19.3 (2025), pp. 689–697.

[38] IBM. *CPLEX Optimizer*. `https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer`. Accessed: 2024-07-26. 2024.

[39] Jeroslow, R. G. “The polynomial hierarchy and a simple model for competitive analysis”. In: *Mathematical Programming* 32.2 (1985), pp. 146–164.

[40] Kanellakis, P.-C. and Papadimitriou, C. H. “Local search for the asymmetric traveling salesman problem”. In: *Operations Research* 28.5 (1980), pp. 1086–1099.

[41] Ketkov, S. S. and Prokopyev, O. A. “On a class of interdiction problems with partition matroids: complexity and polynomial-time algorithms”. In: *INFORMS Journal on Computing* (2025). To appear.

[42] Khorramfar, R., Özaltın, O. Y., Kempf, K. G., and Uzsoy, R. “Managing Product Transitions: A Bilevel Programming Approach”. In: *INFORMS Journal on Computing* 34.5 (2022), pp. 2828–2844.

[43] Kleinert, T., Grimm, V., and Schmidt, M. “Outer approximation for global optimization of mixed-integer quadratic bilevel problems”. In: *Mathematical Programming* 188.2 (2021), pp. 461–521.

[44] Kleinert, T., Labbé, M., Ljubić, I., and Schmidt, M. “A survey on mixed-integer programming techniques in bilevel optimization”. In: *EURO Journal on Computational Optimization* 9 (2021), p. 100007.

[45] Kleinert, T. and Schmidt, M. “Computing feasible points of bilevel problems with a penalty alternating direction method”. In: *INFORMS Journal on Computing* 33.1 (2021), pp. 198–215.

[46] Kleinert, T. and Schmidt, M. “Why there is no need to use a big-M in linear bilevel optimization: A computational study of two ready-to-use approaches”. In: *Computational Management Science* 20.1 (2023), p. 3.

[47] Köppe, M., Queyranne, M., and Ryan, C. T. “Parametric integer programming algorithm for bilevel mixed integer programs”. In: *Journal of Optimization Theory and Applications* 146.1 (2010), pp. 137–150.

[48] Kuiteing, A. K., Marcotte, P., and Savard, G. “Network pricing of congestion-free networks: The elastic and linear demand case”. In: *Transportation Science* 51.3 (2017), pp. 791–806.

[49] Labbé, M., Marcotte, P., and Savard, G. “A bilevel model of taxation and its application to optimal highway pricing”. In: *Management Science* 44.12-part-1 (1998), pp. 1608–1622.

[50] Lagos, T. and Prokopyev, O. A. “On complexity of finding strong-weak solutions in bilevel linear programming”. In: *Operations Research Letters* 51.6 (2023), pp. 612–617.

[51] Lawler, E. L. “Fast approximation algorithms for knapsack problems”. In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. IEEE. 1977, pp. 206–213.

[52] Lin, Y. H., Tian, Q., and Zhao, Y. “Unified framework for choice-based facility location problem”. In: *INFORMS Journal on Computing* (2024). To appear.

[53] Lozano, L. and Smith, J. C. “A value-function-based exact approach for the bilevel mixed-integer programming problem”. In: *Operations Research* 65.3 (2017), pp. 768–786.

[54] Mangasarian, O. L. and Shiau, T.-H. “Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems”. In: *SIAM Journal on Control and Optimization* 25.3 (1987), pp. 583–595.

[55] McCormick, G. P. “Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems”. In: *Mathematical Programming* 10.1 (1976), pp. 147–175.

[56] Moore, J. T. and Bard, J. F. “The mixed integer linear bilevel programming problem”. In: *Operations Research* 38.5 (1990), pp. 911–921.

[57] Nishizaki, I. and Sakawa, M. “Computational methods through genetic algorithms for obtaining Stackelberg solutions to two-level integer programming problems”. In: *Cybernetics and Systems: An International Journal* 36.6 (2005), pp. 565–579.

[58] Orlin, J. B., Punnen, A. P., and Schulz, A. S. "Approximate local search in combinatorial optimization". In: *SIAM Journal on Computing* 33.5 (2004), pp. 1201–1214.

[59] Papp, D. "On the complexity of local search in unconstrained quadratic binary optimization". In: *SIAM Journal on Optimization* 26.2 (2016), pp. 1257–1261.

[60] Prokopyev, O. A. and Ralphs, T. K. "On the complexity of finding locally optimal solutions in bilevel linear optimization". In: *Operations Research* (2026). To appear.

[61] Saharidis, G. K. and Ierapetritou, M. G. "Resolution method for mixed integer bi-level linear problems based on decomposition technique". In: *Journal of Global Optimization* 44 (2009), pp. 29–51.

[62] Schuurman, P. and Vredeveld, T. "Performance guarantees of local search for multiprocessor scheduling". In: *INFORMS Journal on Computing* 19.1 (2007), pp. 52–63.

[63] Shi, X., Prokopyev, O. A., and Ralphs, T. K. "Mixed integer bilevel optimization with a k-optimal follower: a hierarchy of bounds". In: *Mathematical Programming Computation* 15.1 (2023), pp. 1–51.

[64] Smith, J. C. and Song, Y. "A survey of network interdiction models and algorithms". In: *European Journal of Operational Research* 283.3 (2020), pp. 797–811.

[65] Still, G. "Linear bilevel problems: Genericity results and an efficient method for computing local minima". In: *Mathematical Methods of Operations Research* 55 (2002), pp. 383–400.

[66] Stozhkov, V., Boginski, V., Prokopyev, O. A., and Pasiliao, E. L. "A simple greedy heuristic for linear assignment interdiction". In: *Annals of Operations Research* 249.1 (2017), pp. 39–53.

[67] Tahernejad, S., Ralphs, T. K., and DeNegre, S. T. "A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation". In: *Mathematical Programming Computation* 12.4 (2020), pp. 529–568.

[68] Tavaslıoğlu, O., Prokopyev, O. A., and Schaefer, A. J. "Solving stochastic and bilevel mixed-integer programs via a generalized value function". In: *Operations Research* 67.6 (2019), pp. 1659–1677.

[69] Tawfik, C. and Limbourg, S. "A bilevel model for network design and pricing based on a level-of-service assessment". In: *Transportation Science* 53.6 (2019), pp. 1609–1626.

[70] Thürauf, J., Kleinert, T., Ljubić, I., Ralphs, T., and Schmidt, M. "BOBILib: Bilevel Optimization (Benchmark) Instance Library". In: *Mathematical Programming Computation* (2026), pp. 1–58.

[71] Vazirani, V. V. *Approximation algorithms.* Springer Berlin, Heidelberg, 2001.

[72] Vicente, L., Savard, G., and Júdice, J. "Descent approaches for quadratic bilevel programming". In: *Journal of Optimization Theory and Applications* 81.2 (1994), pp. 379–399.

[73] Wen, U. and Huang, A. "A simple tabu search method to solve the mixed-integer linear bilevel programming problem". In: *European Journal of Operational Research* 88.3 (1996), pp. 563–571.

[74] Weninger, N. and Fukasawa, R. "A fast combinatorial algorithm for the bilevel knapsack problem with interdiction constraints". In: *Mathematical Programming* 210.1 (2025), pp. 847–879.

[75] Wiesemann, W., Tsoukalas, A., Kleniati, P.-M., and Rustem, B. "Pessimistic bilevel optimization". In: *SIAM Journal on Optimization* 23.1 (2013), pp. 353–380.

[76] Zare, M. H., Borrero, J. S., Zeng, B., and Prokopyev, O. A. "A note on linearized reformulations for a class of bilevel linear integer problems". In: *Annals of Operations Research* 272 (2019), pp. 99–117.

[77] Zare, M. H., Prokopyev, O. A., and Sauré, D. "On bilevel optimization with inexact follower". In: *Decision Analysis* 17.1 (2020), pp. 74–95.

[78] Zenklusen, R. "Matching interdiction". In: *Discrete Applied Mathematics* 158.15 (2010), pp. 1676–1690.

Appendix

A Local optimality concepts and notation

Tables 4 and 5 summarize the main optimality concepts from the follower's and leader's perspectives, respectively, together with the corresponding references and a brief explanation of each concept. Table 6 then collects the main notation used throughout the paper.

Follower's perspective		
Concept	**Reference**	**Explanation**
Follower's rational response	(3)	Optimistic optimal solution to the follower's problem
Inexact follower's decision	-	Feasible decision to the follower's problem returned by an algorithm $\mathcal{A}$; the procedure $\mathcal{A}$ may be an approximation algorithm, but it may also simply be a heuristic that only returns a feasible decision to the follower's problem
δ-approximate follower's solution	(8)	Inexact follower's decision returned by a δ-approximation algorithm $\mathcal{A}$

Table 4: Summary of the main optimality concepts from the follower's perspective; see Section 3.

We use the concept of weak ε-local optimality because it captures the two relaxations needed in the bilevel setting. The *approximate* part, through ε, relaxes the leader's objective. The *weak* part reflects a relaxation on the follower's side, since the follower's problem may not be solved exactly. In some settings, one has access to a δ-approximation algorithm for the follower. However, in other cases, one may have access only to a heuristic $\mathcal{A}$ or to an off-the-shelf solver that returns a feasible solution to the follower's problem, without any approximation guarantee. The *weak* notion is therefore broad enough to cover both cases.

Leader's perspective		
Concept	**Reference**	**Explanation**
Local optimality	Definition 1	Leader's feasible decision for which no neighbor gives a strictly better leader's objective function value when the follower's problem is solved exactly
ε-local optimality	Definition 2	Approximate version of local optimality in which the follower's problem is solved exactly; improving neighbors may still exist for the leader, but none of them improves the leader's objective function value by more than a multiplicative factor $1+\varepsilon$
Weak local optimality	Definition 3	Local optimality where the leader's objective function is computed with an inexact follower's decision instead of the rational response
Weak ε-local optimality	Definition 4	Combination of the previous two relaxations: the leader only requires approximate local optimality up to a multiplicative factor $1+\varepsilon$, where the leader's objective function is computed with an inexact follower's decision instead of the rational response

Table 5: Summary of the main local optimality concepts from the leader's perspective; see Section 3.

Notation	Reference	Section	Meaning
$\mathcal{X}$	(1b)	Section 1	Leader's feasible set
$\mathbf{x} \in \mathcal{X}$	(1b)	Section 1	Leader's feasible decision
$\mathcal{Y}(\mathbf{x})$	(2)	Section 1	Follower's feasible set given a leader's feasible decision $\mathbf{x} \in \mathcal{X}$
$\mathbf{y}^*(\mathbf{x})$	(3)	Section 1	Follower's rational response given a leader's feasible decision $\mathbf{x} \in \mathcal{X}$
$\varphi(\mathbf{x})$	-	Section 1	Follower's value function given a leader's feasible decision $\mathbf{x} \in \mathcal{X}$
$N_{\mathcal{X}}$	-	Section 3	Neighborhood function (from the leader's perspective)
$N_{\mathcal{X}}^{(k)}$	(4)	Section 3	k-flip neighborhood function (from the leader's perspective), a special case of $N_{\mathcal{X}}$
LSA	-	Section 3.1	Standard local search algorithm
$\mathbf{x}^{\varepsilon}$	Definition 2	Section 3.1	ε-local optimal leader's feasible decision
$\mathcal{A}$	-	Section 3.2	Algorithm that returns a follower's feasible decision for each leader's feasible decision $\mathbf{x} \in \mathcal{X}$
$\mathbf{y}^{\mathcal{A}}(\mathbf{x})$	-	Section 3.2	Inexact follower's decision returned by an algorithm $\mathcal{A}$ given a leader's feasible decision $\mathbf{x} \in \mathcal{X}$
$\mathbf{x}^{\varepsilon,\mathcal{A}}$	Definition 4	Section 3.2	Weak ε-local optimal leader's feasible decision
$\Delta^{\mathcal{A}}\left(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d}\right)$	(10)	Section 3.2	Difference in the leader's objective function values for the leader's feasible decisions $\mathbf{x}^0$ and $\mathbf{x}$, respectively, computed using an inexact follower's decision returned by an algorithm $\mathcal{A}$
Improved	Algorithm 1	Section 5.1	Oracle for the neighborhood search
$(\varepsilon, \mathcal{A})$-LSA	Algorithm 2	Section 5.1	Weak approximate local search algorithm

Table 6: Summary of the main notation used throughout the paper.

Online Supplemental Material

(On Local Search in Bilevel Mixed-Integer Linear Programming)

Eneko C. Clemente and Oleg A. Prokopyev†

Department of Business Administration, University of Zurich, Zurich 8032, Switzerland

S.M.1 Justification of our technical assumptions in Section 1

Recall from Section 1 that four technical assumptions are made throughout this paper. Assumptions **A1** and **A2** together guarantee the existence of an optimal solution for both the follower's problem given any leader's feasible decision, and the bilevel problem itself [11]. Assumption **A1** is standard in bilevel optimization; see, e.g., [7], while Assumption **A2** implies that the follower's feasible set (2) is bounded, which is also a relatively common assumption in the related studies. For more details on the unbounded setting, we refer to the survey by [6].

To ensure that any problem instance satisfies Assumption **A3**, a straightforward transformation can be applied. Specifically, in the context of [**B-BP**], if vectors $\mathbf{a}$ or $\mathbf{d}$ contain negative components, then the corresponding variables can be adjusted, or "flipped." For example, if $d_\ell < 0$ for some $\ell \in [m]$, then variables y_ℓ and $y^*_\ell(\mathbf{x})$ are replaced by $1 - y_\ell$ and $1 - y^*_\ell(\mathbf{x})$, respectively, in both the follower's and leader's objective functions. This adjustment renders $\mathbf{d}$ non-negative while the leader's optimal objective function value of the modified problem remains the same, up to a constant. Next, for continuous follower's decisions, Assumption **A2** is used to introduce a new variable $\tilde{y}_\ell$ defined as $\tilde{y}_\ell = U - y_\ell$, and then the same arguments apply.

The leader's objective function in [**BP**] is assumed to be strictly positive in Assumption **A4**. The non-negativity of vectors $\mathbf{a}$ and $\mathbf{d}$ in Assumption **A3** ensures that the leader's optimal objective function value of [**BP**] is non-negative, i.e., $z^* \geq 0$. When $a_i > 0$ for all $i \in [n]$ and $d_\ell > 0$ for all $\ell \in [m]$, the condition $z^* > 0$ can be verified directly. Indeed, $z^* = 0$ holds if and only if $\mathbf{0} \in \mathcal{X}$ and the corresponding follower's rational response satisfies $y^*_\ell(\mathbf{0}) = 0$ for all $\ell \in [m]$. Verifying $z^* > 0$ in full generality requires additional refinements, which are beyond the scope of the present discussion.

Finally, the second part of Assumption **A4** can be ensured by another transformation. Indeed, given that the follower's feasible set is bounded, Assumption **A2** implies the existence of a lower bound, say, denoted by M, for the follower's objective function value in (1c). Specifically, an appropriate value for M can be derived such that it is independent of any leader's feasible decision. This lower bound M can then be used to reformulate the lower-level problem (1c) into an equivalent

†Emails: {eneko.clemente, oleg.prokopyev}@business.uzh.ch

problem with the updated cost vector and decision variables, given by $\widetilde{\mathbf{c}} := (\mathbf{c}, -M)^\top$ and $\tilde{\mathbf{y}} := (\mathbf{y}, y_{m+1})^\top$, respectively, along with the additional constraint $y_{m+1} = 1$.

Comments on our assumptions. Assumptions **A3** and **A4** are essential for establishing the approximation results in Section 5, but they also introduce certain limitations. Assumption **A3** requires the cost vector $\mathbf{d}$ to be non-negative, which is achieved by "flipping" variables when negative components are present. Although this transformation preserves the optimal objective function value up to a constant, it can alter the structure of the follower's feasible set and may disrupt properties such as total unimodularity or other structure critical for some approximation algorithms.

That said, many bilevel optimization problems with such exploitable lower-level structure, e.g., interdiction problems, naturally use cost vectors with non-negative entries; see, for example, [12]. The non-negativity of $\mathbf{a}$ and $\mathbf{d}$ is mainly required for the scaling step in Algorithm 2. This requirement aligns with classical assumptions imposed by many scaling-based approximation algorithms in single-level combinatorial optimization [10].

Assumption **A4** can be enforced by "flipping" decision variables or introducing an artificial variable in the lower-level problem. In such cases, when the lower-level problem is solved approximately, the bound becomes $\mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \geq (1-\delta)\mathbf{c}^\top \mathbf{y}^*(\mathbf{x}) + \delta M$, which is aligned with the standard definition in the literature [9]. This construction may weaken the approximation guarantees in Section 4, including Theorem 1, by a term dependent on M, with the additional error scaling with δ. Moreover, "flipping" variables can affect the gap in the definition of ε-local optimality (refer to our discussion in Section 3) by a term dependent on $\|\mathbf{a}\|_1$, $\|\mathbf{d}\|_1$, U, and that scales with ε. These considerations highlight the need for systematic theoretical and empirical analysis of how such transformations impact solution quality, which we leave for future research.

Coupling constraints. Coupling constraints arise when the follower's optimal decision is part of the leader's feasible set, defined as $\mathcal{X} = \{\mathbf{x} \in \{0,1\}^n \ : \ \mathbf{H}_1\mathbf{x} + \mathbf{H}_2\mathbf{y}^*(\mathbf{x}) \leq \mathbf{h}\}$. Consequently, the feasibility of a leader's decision can be influenced by the follower's rational response. To address this issue, our approach can be extended by modifying the definition of a feasible solution for the leader. Indeed, a step that verifies whether $(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ is feasible can simply be added to IMPROVED.

Extending the inexact follower approach to bilevel programs with coupling constraints presents additional challenges. Specifically, while $\left(\mathbf{x}, \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right)$ might satisfy the coupling constraints, $(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ might not. Yet, if the lower-level problem is solved using a δ-approximation algorithm $\mathcal{A}$, then a sufficient condition can be derived under which we ensure that if both $\mathcal{X}(\delta) \subseteq \mathcal{X}$, and $\left(\mathbf{x}, \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right) \in \mathcal{X}(\delta)$, then $(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \in \mathcal{X}$. This modification is straightforward. Hence, we omit it for brevity.

S.M.2 Omitted discussions for Section 3

We divide the discussion into three parts. First, in Supplemental Material S.M.2.1, we demonstrate that for any $\varepsilon \geq 0$, verifying whether a given leader's feasible decision is ε-locally optimal is an NP-hard problem in general. Second, in Supplemental Material S.M.2.2, we describe the weak local search algorithm and illustrate it through the example from Section 3.3. Finally, Supplemental Material S.M.2.3 presents the greedy heuristic $\mathcal{A}$ that is used in the example from Section 3.3.

S.M.2.1 Verifying (approximate) local optimality in bilevel MILP is NP-hard

We show that when the follower's decisions are all binary, verifying whether a given leader's feasible decision is an ε-local optimal solution is an NP-hard problem. This result is established by reducing the *Subset Sum Problem* (**SSP**), which is known to be NP-complete [4], to verifying whether a given leader's feasible decision is ε-locally optimal for any fixed $\varepsilon \geq 0$. The **SSP** problem consists of answering the following question: given a set of non-negative integers $w_1, \ldots, w_n$ and a positive target W, is there a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} w_i = W$?

Proposition 4. *Fix $\varepsilon \geq 0$. There exists an instance of* [**B-BP**] *and a leader's feasible decision $\mathbf{x}^\varepsilon \in \mathcal{X}$ such that the answer to* **SSP** *is "Yes" if and only if $\mathbf{x}^\varepsilon$ is ε-locally optimal with respect to $N_{\mathcal{X}}^{(1)}$.*

Proof. Fix $\varepsilon \geq 0$. The proof consists of reducing **SSP** to checking ε-local optimality of some given leader's feasible decision $\mathbf{x}$ to [**B-BP**]. Consider the following bilevel MILP:

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{n} w_i y_i + x_1 + \frac{W + \frac{1}{2}}{1+\varepsilon} x_2 && \text{(S.M.1a)}\\
\text{s.t.} \quad & \mathbf{x} \in \{0,1\}^2, && \text{(S.M.1b)}\\
& \mathbf{y} \in \arg\max_{\tilde{y}} \quad -\sum_{i=1}^{n} w_i \tilde{y}_i && \text{(S.M.1c)}\\
& \qquad W - x_1/2 - W x_2 \leq \sum_{i=1}^{n} w_i \tilde{y}_i \leq \sum_{i=1}^{n} w_i, && \text{(S.M.1d)}\\
& \qquad \tilde{\mathbf{y}} \in \{0,1\}^n. && \text{(S.M.1e)}
\end{aligned}
$$

Solving **SSP** can be reduced to verifying that $(x_1, x_2) = (0, 0)$ is ε-locally optimal for (S.M.1). We first introduce $f(x_1, x_2, \text{"}answer\text{"})$, a function that returns the leader's objective function value of (S.M.1) for a given pair (x_1, x_2), where "*answer*"$\in$ {"Yes", "No"} corresponds to the answer to **SSP**. All possible values that can be taken by f are given in Table S.M.1.

If *"answer"* is "Yes," then $(x_1, x_2) = (0, 0)$ is ε-locally optimal. Indeed, from Table S.M.1, note that $f(0, 0, \text{"Yes"}) = W \leq (1+\varepsilon)(W+1) = (1+\varepsilon) f(1, 0, \text{"Yes"})$. Also, $f(0, 0, \text{"Yes"}) \leq (1+\varepsilon) f(0, 1, \text{"Yes"}) \leq (1+\varepsilon) f(1, 1, \text{"Yes"})$. Thus, $(x_1, x_2) = (0, 0)$ is ε-locally optimal.

"*answer*"	(x_1, x_2)			
	$(0,0)$	$(1,0)$	$(0,1)$	$(1,1)$
"Yes"	W	$W+1$	$\frac{2W+1}{2(1+\varepsilon)}$	$\frac{2W+1}{2(1+\varepsilon)}+1$
"No"	$\geq W+1$	$\geq W+2$	$\frac{2W+1}{2(1+\varepsilon)}$	$\frac{2W+1}{2(1+\varepsilon)}+1$

Table S.M.1: Summary of the leader's objective function values of (S.M.1), denoted by f, for different pairs (x_1, x_2) of leader's feasible decisions and the answer to **SSP**.

Conversely, if $(x_1, x_2) = (0, 0)$ is ε-locally optimal, then *"answer"* must be equal to "Yes." Indeed, if *"answer"* is equal to "No," then $(x_1, x_2) = (0, 1)$ satisfies $f(0, 0, \text{"No"}) \geq W + 1 > W + \frac{1}{2} = (1 + \varepsilon)f(0, 1, \text{"No"})$. Consequently, $(x_1, x_2) = (0, 0)$ is not ε-locally optimal. Therefore, verifying that $(x_1, x_2) = (0, 0)$ is ε-locally optimal is equivalent to solving **SSP**. ■

S.M.2.2 On weak local search

To begin, we introduce the concept of weak local search, which generalizes the standard local search algorithm. The main distinction between the two lies in their approach to computing the leader's objective function when searching for an improving solution. In standard local search, the leader's objective function is evaluated using the follower's rational response. In contrast, weak local search relies on evaluating the leader's objective function using an inexact follower's response, e.g., an approximate solution to the lower-level problem (1c), or any follower's feasible decision.

Finding (weak) local optimal solutions. The weak local search algorithm is essentially a standard local search, which uses an inexact follower's response (obtained with an algorithm $\mathcal{A}$) to compute the leader's objective function. It iteratively performs a neighborhood search by calling Improved (see Algorithm 1 in Section 5.1 and the corresponding discussion) until no new improving solution (in terms of the leader's objective function with the inexact follower's response) is found.

Algorithm 3 - $\mathcal{A}$-`LSA` - weak local search

```
function A-LSA(x^0, N_X, a, d, A)
    x ← x^0
    (improved, x^+) ←Improved(x, N_X, a, d, A, 0)
    while improved do
        x ← x^+
        (improved, x^+) ←Improved(x, N_X, a, d, A, 0)
    return x
```

The weak local search algorithm, denoted as $\mathcal{A}$-`LSA` and described in Algorithm 3, takes as inputs an initial leader's feasible decision $\mathbf{x}^0$, a neighborhood function $N_{\mathcal{X}}$, vectors $\mathbf{a}$ and $\mathbf{d}$, and an algorithm $\mathcal{A}$ that returns a feasible solution to the lower-level problem (1c). Note that, in contrast to $(\varepsilon, \mathcal{A})$-`LSA` (see Algorithm 2 in Section 5.1), no minimum gap is required to accept a neighbor as an improving solution within Improved. That is, $\gamma = 0$; recall Algorithm 1.

Clearly, $\mathcal{A}$-LSA returns a weak local optimal solution within a finite number of calls to Improved, but does not necessarily return a local optimal solution. Unsurprisingly, if the lower-level problem is solved exactly, then $\mathcal{A}$-LSA is essentially the standard local search. An illustrative example of $\mathcal{A}$-LSA applied to the problem from Section 3.3 is provided below.

Example 1 We consider the example from Section 3.3. The initial leader's feasible decision $\mathbf{x}^0$ is $\mathbf{0}$. Improved selects the first improving solution while enumerating the neighborhood. It first considers all one-flip neighbors, then all two-flip neighbors; within each flip cardinality, subsets of flipped indices are generated in lexicographic order.

Exact follower. $\mathcal{A}$-LSA starts by searching for an improving solution in the neighborhood of $\mathbf{x}^0$. When calling Improved on $\mathbf{x}^0$, we obtain $(\text{TRUE}, \mathbf{x}^1)$, where $\mathbf{x}^1 \in N_{\mathcal{X}}(\mathbf{x}^0)$ consists of interdicting vertex 1, that is $x_1^1 = 1$ and $x_i^1 = 0$ otherwise. The obtained clique $\mathcal{C}^*$ has value $\omega(\mathcal{C}^*) = 220 < 300$. Next, $\mathbf{x}^0$ is replaced by $\mathbf{x}^1$. Then, Improved is called on $\mathbf{x}^1$ to obtain $(\text{FALSE}, \mathbf{x}^1)$. Therefore, the algorithm ends and returns $\mathbf{x}^* = \mathbf{x}^1$.

Inexact follower. If the maximum clique problem is solved using $\mathcal{A}$ from the Supplemental Material S.M.2.3, then $\mathcal{A}$-LSA returns the decision $\mathbf{x}^{\mathcal{A}}$ that consists of interdicting vertex 9. ■

S.M.2.3 Greedy heuristic for the maximum weighted clique problem

For completeness, we present the algorithm $\mathcal{A}$ used in the illustrative example from Section 3.3. The procedure, described in Algorithm 4, is a simple greedy heuristic [13].

Algorithm 4 - $\mathcal{A}$ - Sequential greedy heuristic for maximum weighted clique

Require: Graph $G = (V, E)$
function $\mathcal{A}(V, E)$
    $\mathcal{C} \leftarrow \emptyset$
    while $V \neq \emptyset$ **do**
        Choose $v \in V$ with maximum degree, break ties by weight
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{v\}$
        $V \leftarrow \{u \in V \setminus \{v\} \mid u \text{ is adjacent to all vertices in } \mathcal{C}\}$
    return $\mathcal{C}$

Algorithm 4 takes a weighted graph $G = (V, E)$ as input and returns a weighted clique $\mathcal{C}$. It begins with an empty clique and iteratively selects a vertex with the maximum degree. Ties are broken by choosing the vertex with the highest weight (then ties are broken arbitrarily). Next, the selected vertex is added to the clique $\mathcal{C}$, and the graph G is then updated by retaining only vertices connected to all current members of $\mathcal{C}$. This process continues until no vertices remain in G.

S.M.3 Omitted discussions for Section 4

To empirically evaluate the empirical maximum gap from Figure 2 on the interdiction maximum clique problem (see Section 3.3 for further details), we construct our instances as follows:

Graph generation. We generate 50 Erdős–Rényi random graphs $G = (V, E)$, each with $n = 40$ nodes and edge density $p = 0.5$; see [2]. For the follower's problem, each node v receives a weight drawn uniformly at random from $\{1000 + 1, \ldots, 1000 + 10 \cdot \deg(v)\}$, where $\deg(v)$ denotes the degree of node v. The leader's objective function vectors are generated uniformly at random, i.e., $\mathbf{a} \sim \mathcal{U}([1, 10]^{40})$, $\mathbf{d} \sim \mathcal{U}([10, 100]^{40})$, and the interdiction budget is set to $h = 4$.

Experimental details. For each instance, we construct a parameter grid over δ and ε, ranging from 0 to 0.5 in increments of 0.0263 (i.e., 20 evenly spaced values for each parameter). For each pair, to obtain a leader's feasible decision with the desired property (as in Theorem 1), we run the $(\varepsilon, \mathcal{A})$-`LSA` algorithm using the 2-flip neighborhood function, starting from the initial leader's solution $\mathbf{x} = \mathbf{0}$. Moreover, Improved selects the first improving solution while enumerating the neighborhood. It first considers all one-flip neighbors, then all two-flip neighbors; within each flip cardinality, subsets of flipped indices are generated in lexicographic order. The follower's problem is solved using the MILP solver by [5] with a pre-specified optimality gap equal to δ.

Performance evaluation. For each (δ, ε) pair, the performance metric (specifically, the maximum empirical gap) is computed for all 50 independently generated instances. The reported results represent the *average* empirical maximum gap across all instances.

S.M.4 Omitted discussions for Section 5

Supplemental Material S.M.4.1 discusses the sharpness of $(\varepsilon, \mathcal{A})$-LSA. We extend the discussion from Section 5.2 to the case of a follower with mixed-integer decision variables in Supplemental Material S.M.4.2. Supplemental Material S.M.4.3 discusses extensions of our approach.

S.M.4.1 Sharpness of $(\varepsilon, \mathcal{A})$-LSA

We construct an instance of the knapsack interdiction problem for which the bound $\varepsilon > 0$ in Theorem 5 is asymptotically attained when the follower's problem is solved exactly. We refer to [1] for further background on the problem. The knapsack interdiction problem can be formulated as:

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mathbf{c}^\top \mathbf{y}, \tag{S.M.2}$$

where the leader's feasible set is defined as $\mathcal{X} := \left\{\mathbf{x} \in \{0,1\}^n \ : \ \mathbf{1}^\top \mathbf{x} \leq h\right\}$, and $h \in \mathbb{Z}_+$. Moreover, $\mathbf{x} \in \mathcal{X}$ indicates the decision made by the leader, i.e., $x_i = 1$ if and only if item i is interdicted. Given $\mathbf{x} \in \mathcal{X}$, we define $\mathcal{Y}(\mathbf{x}) := \left\{\mathbf{y} \in \{0,1\}^n \ : \ \mathbf{1}^\top \mathbf{y} \leq f \ , \ \mathbf{y} \leq \mathbf{1} - \mathbf{x}\right\}$. Also, $\mathbf{y}^*(\mathbf{x})$ is the follower's rational response given $\mathbf{x}$, i.e., $y_i(\mathbf{x})^* = 1$ if and only if the follower selects item i.

In the following, we assume that $h = 1$ and $f = 1$. Next, we define $\varepsilon > 0$ by:

$$\varepsilon := \frac{4(n+1)}{\xi + \psi - \frac{1}{3}} \ \text{ and } \ \theta := \frac{\varepsilon}{12(n+1)(1+\varepsilon)},$$

where ξ is a strictly positive integer and $\frac{2}{3} < \psi < 1$.

We define $\mathbf{c} \in \mathbb{R}^n_+$ by $c_i = \theta$, for any $i \in \{1, \ldots, n-2\}$, $c_{n-1} = 1$ and $c_n = 1 - \theta$. Also, let $\mathbf{e}_i \in \{0,1\}^n$ be the vector, where the only non-zero element is equal to 1 in the i-th component.

We consider the 2-flip neighborhood $N^{(2)}_{\mathcal{X}}$. We call $(\varepsilon, \mathcal{A})$-LSA with initial feasible solution $\mathbf{x}^0 = \mathbf{e}_n$. It follows that $\mathbf{y}^*(\mathbf{x}^0) = \mathbf{e}_{n-1}$, and $K = \mathbf{c}^\top \mathbf{y}^*(\mathbf{x}^0) = 1$. Hence, $q_d = \frac{K\varepsilon}{4(n+1)(1+\varepsilon)}$, and:

$$\frac{c_i}{q_d} = \begin{cases} \theta\left(4(n+1) + \xi + \psi - 1/3\right) & \text{if } i \in \{1, \ \ldots \ , n-2\}, \\ 4(n+1) + \xi + \psi - 1/3 & \text{if } i = n-1, \\ 4(n+1) + \xi + \psi - 1/3 - \theta\left(4(n+1) + \xi + \psi - 1/3\right) & \text{if } i = n. \end{cases}$$

Next, observe that $\theta\left(4(n+1) + \xi + \psi - 1/3\right) = 1/3$. Since ξ is an integer, if we apply the ceiling function to $\frac{\mathbf{c}}{q_d}$, then we obtain the following scaled vector $\mathbf{c}' = q_d \left\lceil \frac{\mathbf{c}}{q_d} \right\rceil$:

$$\left\lceil \frac{c_i}{q_d} \right\rceil = \begin{cases} 1 & \text{if } i \in \{1, \ \ldots \ , n-2\}, \\ 4n + \xi + 5 & \text{if } i = n-1, \\ 4n + \xi + 5 & \text{if } i = n. \end{cases}$$

We enter the inner loop at line 9 of Algorithm 2 with starting feasible solution $\mathbf{x}^0$ and call Improved with the scaled vector $\mathbf{c}'$. The 2-flip neighborhood of $\mathbf{x}^0$ is given by:

$$N_{\mathcal{X}}^{(2)}(\mathbf{x}^0) := \Big\{\mathbf{0}\Big\} \bigcup \Big\{\mathbf{e}_i : i \in \{1, \ldots, n\}\Big\}.$$

Then, Improved returns false since there are no improving solutions in the neighborhood of $\mathbf{x}^0$ (in terms of the scaled vectors). Hence, the algorithm terminates and returns $\mathbf{x}^0 = \mathbf{e}_n$.

In fact, the only solution in the neighborhood $\mathbf{x}^0$ that has a better leader's objective function value is $\mathbf{x}^1 = \mathbf{e}_{n-1}$. Indeed, observe that $\mathbf{y}^*(\mathbf{x}^1) = \mathbf{e}_n$. Consequently, the following holds:

$$\frac{\mathbf{c}^\top \mathbf{y}^*(\mathbf{x}^0) - \mathbf{c}^\top \mathbf{y}^*\left(\mathbf{x}^1\right)}{\mathbf{c}^\top \mathbf{y}^*\left(\mathbf{x}^1\right)} = \frac{1-(1-\theta)}{1-\theta} = \varepsilon \frac{1}{12(n+1) + (12(n+1)-1)\,\varepsilon} = \mathcal{O}(\varepsilon),$$

which implies that the worst-case bound ε obtained by $(\varepsilon, \mathcal{A})$-`LSA` in Theorem 5 is in the order of ε (asymptotically), assuming $0 < \varepsilon < 1$. Thus, $(\varepsilon, \mathcal{A})$-`LSA` can be considered "sharp" in this sense.

S.M.4.2 Proofs and further discussions for Section 5.3

We assume that at least one of the follower's decision variables is continuous, i.e., $m_2 > 0$. Also, without loss of generality, we assume that $\|\mathbf{d}\|_\infty > 0$. Indeed, if $\mathbf{d} = \mathbf{0}$, then the leader's problem [**BP**] is reduced to a single-level combinatorial optimization problem.

We mirror the discussion in Section 5.2 while taking into account the differences arising with the relaxation of the integrality restriction at the lower level. Accordingly, we present the following:

- We provide an upper bound for the maximum number of calls to Improved between two iterations i and $i+1$ in $(\varepsilon, \mathcal{A})$-`LSA`; see Lemma 3. We leverage this observation to derive the runtime complexity of the weak approximate local search; see Proposition 3.
- As $(\varepsilon, \mathcal{A})$-`LSA` is guaranteed to terminate, we show that the leader's feasible decision $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is a weak ε-local optimal solution; see Proposition 5.

Lemma 3. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Assume that q_a, q_d, K, γ, $\mathbf{a}'$ and $\mathbf{d}'$ are given as in Algorithm 2 at iteration $i \in \mathbb{Z}_{\geq 0}$. Then, the maximum number of calls to* Improved *between two subsequent iterations i and $i+1$, denoted by $\tau(m, \varepsilon)$, is of order of $\tau(m, \varepsilon) = \mathcal{O}\left(\frac{m}{\varepsilon}\right)$, i.e., the maximum number of calls to* Improved *is polynomial in m and $1/\varepsilon$.*

Proof. Recall that at the iteration i, the leader's objective function value with an inexact follower and scaled vectors is given by $\mathbf{a}'^\top \mathbf{x}^i + \mathbf{d}'^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)$, see line 6 of Algorithm 2. We are interested in the maximum number of calls to Improved. If $K = 0$ at iteration i, then the algorithm stops after one call to Improved and returns the current leader's feasible decision. Hence, we assume $K > 0$ for the remainder of the proof. Moreover, without loss of generality, we assume that $d_\ell > 0$ for all $\ell \in [m]$. If some components of $\mathbf{d}$ are zero, the same argument applies after restricting the sums to the positive components of $\mathbf{d}$; we omit this notational refinement to keep the proof clear.

At each call of Improved, if *improved* is true, then the improving solution returned by the sub-procedure reduces the leader's objective function value with an inexact follower and with scaled vectors by at least γ. Hence, the maximum number of calls τ to Improved between iterations i and $i+1$ is given by the following identity:

$$\tau := \frac{\mathbf{a'}^\top \mathbf{x}^i + \mathbf{d'}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)}{\gamma},$$

where $\mathbf{a'}$ and $\mathbf{d'}$ are the scaled vectors obtained at line 6 of Algorithm 2. An upper bound for τ can be derived as follows:

$$\begin{aligned}
\tau &= \frac{\mathbf{a'}^\top \mathbf{x}^i + \mathbf{d'}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)}{\gamma} && \text{(S.M.4a)}\\
&= \frac{\sum_{j=1}^n q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^i + \sum_{\ell=1}^m q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^i)}{\gamma} && \text{(S.M.4b)}\\
&\le \frac{\mathbf{a}^\top \mathbf{x}^i + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i) + n q_a + q_d\,(m+1)\,U}{\gamma} && \text{(S.M.4c)}\\
&= \left(K + \frac{K\varepsilon}{4\,(1+\varepsilon)} + \frac{K\varepsilon}{4\,(1+\varepsilon)}\right) \frac{m q_d + \sum_{\ell=1}^m d_\ell}{U q_d^2 \sum_{\ell=1}^m \left\lceil \frac{d_\ell}{q_d} \right\rceil}, && \text{(S.M.4d)}
\end{aligned}$$

where (S.M.4a) comes from the definition of the scaled vectors $\mathbf{a'}$ and $\mathbf{d'}$. Additionally, (S.M.4c) is obtained by the property of the ceiling function. Specifically,

$$\sum_{\ell=1}^m q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^i) \le \sum_{\ell=1}^m q_d \left(\frac{d_\ell}{q_d} + 1 \right) y_\ell^{\mathcal{A}}(\mathbf{x}^i) \le \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i) + \sum_{\ell=1}^m q_d y_\ell^{\mathcal{A}}(\mathbf{x}^i).$$

Moreover, both the leader's and the follower's decision variables are bounded by 1 and U, respectively; recall Assumption **A2**. Therefore,

$$\sum_{\ell=1}^m q_d y_\ell^{\mathcal{A}}(\mathbf{x}^i) \le q_d m U \le q_d\,(m+1)\,U,$$

and the other terms in (S.M.4c) can be derived in a similar manner. Finally, (S.M.4d) is obtained by the definitions of K, q_a, q_d, and γ; see lines 5-7 of Algorithm 2.

We once again use the property of the ceiling function to obtain:

$$\sum_{\ell=1}^m \left\lceil \frac{d_\ell}{q_d} \right\rceil \ge \max\left\{ \sum_{\ell=1}^m \frac{d_\ell}{q_d}, \sum_{\ell=1}^m 1 \right\},$$

which is then exploited, along with (S.M.4), to derive the following sequence of inequalities:

$$\begin{aligned}\tau &\leq \frac{K}{Uq_d}\left(\frac{2+3\varepsilon}{2\left(1+\varepsilon\right)}\right)\frac{mq_d+\sum_{\ell=1}^{m}d_\ell}{q_d\sum_{\ell=1}^{m}\left\lceil\frac{d_\ell}{q_d}\right\rceil}\\ &\leq \frac{K}{Uq_d}\left(\frac{2+3\varepsilon}{2\left(1+\varepsilon\right)}\right)\left(1+1\right)\\ &\leq \frac{K}{Uq_d}\left(\frac{2+3\varepsilon}{1+\varepsilon}\right)\leq\frac{4U\left(m+1\right)\left(1+\varepsilon\right)K}{KU\varepsilon}\left(\frac{2+3\varepsilon}{1+\varepsilon}\right)\leq\frac{4\left(m+1\right)\left(2+3\varepsilon\right)}{\varepsilon},\end{aligned}$$

where the fourth inequality is obtained by using the definition of q_d; see line 5 of Algorithm 2. Hence, the maximum number of calls depends only on m and ε. Finally, we have that $\tau\left(m,\varepsilon\right)=\mathcal{O}\left(\frac{m}{\varepsilon}\right)$, which concludes the proof. ∎

Proof of Proposition 3. By Lemma 3, the maximum number of calls to Improved (and hence, calls to $\mathcal{A}$) between iterations i and $i+1$ is given by $\tau\left(m,\varepsilon\right)$. Moreover, $\tilde{K}_0$ is an upper-bound of the optimal leader's objective value with an inexact follower.

Additionally, between each iteration i and $i+1$, the leader's objective function value is divided by at least 2 by the stopping criteria of the loop of line 9 of Algorithm 2. Therefore, the maximum number of iterations is in the order of $\mathcal{O}\left(\log\tilde{K}_0\right)$, and the total number of calls of Improved and $\mathcal{A}$ is in the order of $\mathcal{O}\left(\tau\left(m,\varepsilon\right)\log\tilde{K}_0\right)$. Consequently, $(\varepsilon,\mathcal{A})$-LSA terminates within a finite number of improving steps. In fact, the number of the improving steps is polynomial. ∎

The procedure $(\varepsilon,\mathcal{A})$-LSA is guaranteed to terminate, even when the follower's problem contains continuous decision variables. We next show that the solution it returns is in fact weakly ε-local optimal. The following result is analogous to Theorem 5 in Section 5.2.

Proposition 5. *Let $\varepsilon>0$, $N_{\mathcal{X}}$ be a neighborhood function, and $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Then,* $(\varepsilon,\mathcal{A})$-LSA *is guaranteed to return a weak ε-local optimal solution of* [**BP**] *with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$.*

Proof. Let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ be the leader's feasible decision obtained by calling $(\varepsilon,\mathcal{A})$-LSA, and let $\mathbf{x}\in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$. Let q_a,q_d,K and γ denote the values obtained before $(\varepsilon,\mathcal{A})$-LSA terminates in the last iteration i_f; recall lines 5-7 from Algorithm 2. If $K=0$ at iteration i_f, then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is already weakly ε-local optimal for [**BP**] with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$ and the proposition follows immediately. Hence, we assume $K>0$ for the remainder of the proof, which implies $q_a,q_d>0$. Then:

$$\mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}}+\mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}})=\sum_{j=1}^{n}a_jx_j^{\varepsilon,\mathcal{A}}+\sum_{\ell=1}^{m}d_\ell y_\ell^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$$

$$\leq \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^{\varepsilon,\mathcal{A}} + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$$
$$\leq \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}) + \gamma,$$

where the first inequality comes from the property of the ceiling function, and the second one comes from the local optimality property of $\mathbf{x}^{\varepsilon,A}$ with respect to the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. Indeed, recall that $(\varepsilon, \mathcal{A})$-LSA stops whenever *improved*, which is returned by IMPROVED, is FALSE. Consequently, the gap between $\mathbf{x}^{\varepsilon,A}$ and $\mathbf{x}$ with respect to the scaled vectors satisfies $\Delta^{\mathcal{A}}(\mathbf{x}^{\varepsilon,A}, \mathbf{x}, \mathbf{a}', \mathbf{d}') \leq \gamma$.

Hence, using the definition of γ and the property of the ceiling function, we obtain:

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil \left(y_\ell^{\mathcal{A}}(\mathbf{x}) + U \left(m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d} \right)^{-1} \right) \tag{S.M.7a}$$
$$\leq \sum_{j=1}^{n} q_a \left(\frac{a_j}{q_a} + 1 \right) x_j + \sum_{\ell=1}^{m} q_d \left(\frac{d_\ell}{q_d} + 1 \right) \left(y_\ell^{\mathcal{A}}(\mathbf{x}) + U \left(m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d} \right)^{-1} \right) \tag{S.M.7b}$$
$$\leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + n q_a + m U q_d + U \frac{\sum_{\ell=1}^{m} d_\ell + m q_d}{m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d}} \tag{S.M.7c}$$
$$\leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + n q_a + (m+1) U q_d, \tag{S.M.7d}$$

where (S.M.7a) and (S.M.7b) follow by the definition of γ and by the properties of the ceiling function, respectively; recall line 7 of Algorithm 2.

In addition, according to the stopping criteria of the loop in line 9 of Algorithm 2, together with the previous inequality, we have that:

$$\frac{K}{2} \leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + n q_a + (m+1) U q_d.$$

Therefore, the following inequalities are satisfied:

$$\frac{\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) - \mathbf{a}^\top \mathbf{x} - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})} \leq \frac{n q_a + (m+1) U q_d}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})}$$
$$\leq \frac{n q_a + (m+1) U q_d}{\frac{K}{2} - n q_a - (m+1) U q_d} = \varepsilon,$$

where the last equality follows by the definition of q_a and q_d. Finally, we can conclude that $x^{\varepsilon,\mathcal{A}}$ is weakly ε-local optimal with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$. ■

If the follower's decision variables are all continuous, then the follower's problem (1c) is a linear program and can be solved in polynomial time. Therefore:

Corollary 3. *If* IMPROVED *and $\mathcal{A}$ run in polynomial time, and $\mathcal{A}$ solves the follower's LP exactly, then* $(\varepsilon, \mathcal{A})$-LSA *returns an ε-local optimal solution to* [**C-BP**] *in polynomial time.*

Furthermore, if the neighborhood always remains of polynomial size, similarly to the k-flip neighborhood, then it can be efficiently explored by exhaustively enumerating all of its elements. In that case, the assumption of the previous corollary holds.

On the other hand, if the follower's variables can also be binary and $\mathcal{A}$ is a δ-approximation algorithm, then we can show that the solution obtained by $(\varepsilon, \mathcal{A})$-`LSA` is an approximate local optimal solution. Indeed, $(\varepsilon, \mathcal{A})$-`LSA` is guaranteed to return a weak ε-local optimal solution by Proposition 5. Then, recall our discussion from Section 4 on the relation between weak (approximate) and approximate local optimality whenever a δ-approximation algorithm is used to solve the follower's problem.

Corollary 4. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, $\mathcal{A}$ be a polynomial-time algorithm that returns a δ-approximate solution to the lower-level problem* (1c) *for any leader's feasible decision, and $\underline{z} > 0$ be a strictly positive lower bound for the leader's objective function value. If* Improved *is a polynomial-time algorithm, then $(\varepsilon, \mathcal{A})$-`LSA` is a polynomial-time algorithm that finds a $\Pi(\varepsilon, \gamma_1\delta + \gamma_2, \underline{z})$-local optimal solution with respect to $N_{\mathcal{X}}$, for some $\gamma_1, \gamma_2 \geq 0$, and Π as given by* (14).

S.M.4.3 Extensions of our approach

In this section, we discuss several meaningful extensions of our approach, initially omitted to maintain clarity. These extensions are particularly valuable from a practical standpoint, as they address considerations that are important to effectively apply our approach. Specifically:

Initial feasible solution. An initial leader's feasible decision, denoted as $\mathbf{x}^0$, is required as an input for $(\varepsilon, \mathcal{A})$-`LSA`. This initial decision can be obtained by solving the *single-level relaxation* of [**BP**], which relaxes the leader's optimization problem by dropping the follower's optimality condition [6]. This relaxation is formulated as: $z^{SLR} := \min_{\mathbf{x},\mathbf{y}} \left\{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \;:\; \mathbf{x} \in \mathcal{X},\; \mathbf{y} \in \mathcal{Y}(\mathbf{x})\right\}$. A feasible solution to the single-level relaxation problem can serve as a starting point for $(\varepsilon, \mathcal{A})$-`LSA`. Such feasible decisions can be obtained through classical heuristics [3].

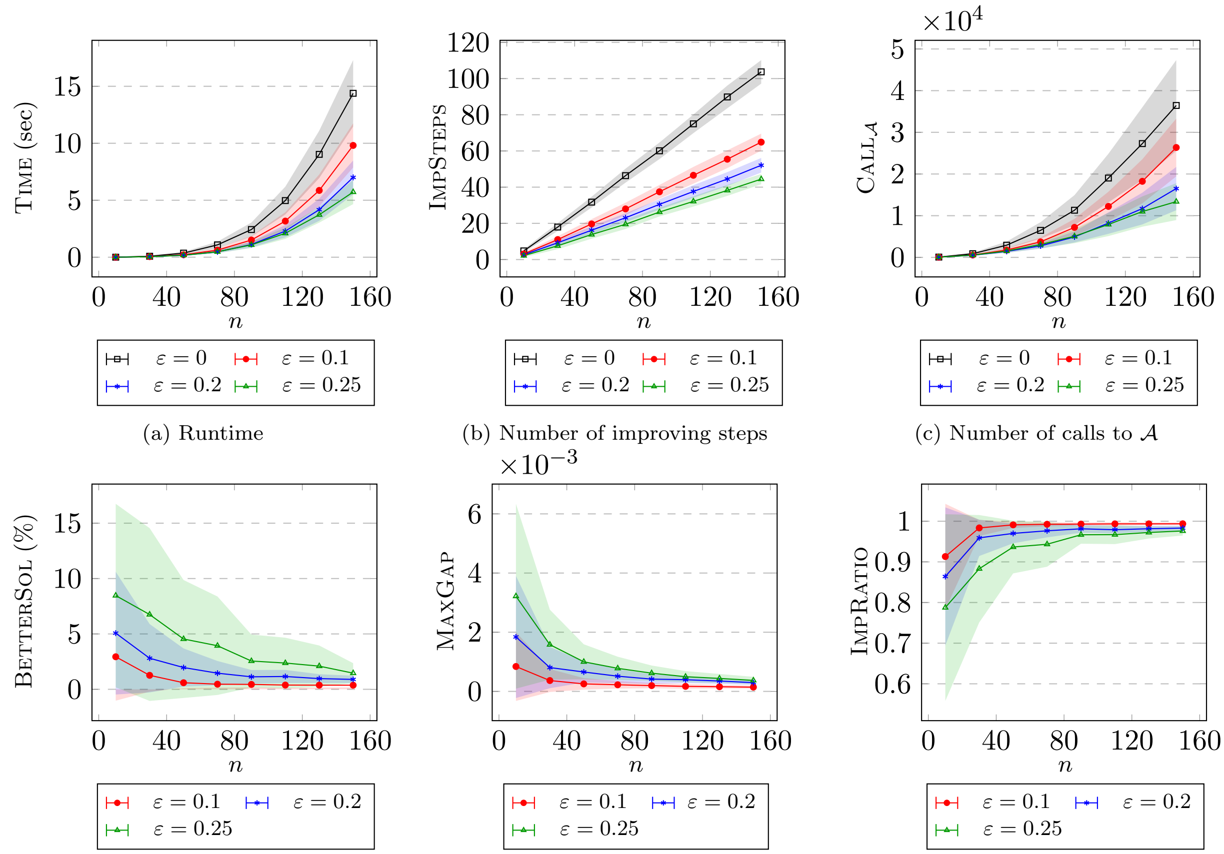
General integrality at the upper level. Next, we consider another extension, where the leader's decision variables are generalized to integer values, i.e., $\mathbf{x} \in \{0, \ldots, u\}^n$, for some integer $u \in \mathbb{Z}_{>0}$. If u is known in advance, then the leader's decisions can be replaced by binary decisions using a standard binary expansion technique. Alternatively, one could modify the definitions introduced in Section 3 and generalize them to accommodate the leader's general integer decisions. In this case, a minor adjustment is required in Algorithm 2. Specifically, in line 6, the scaling factor q_a applied to vector $\mathbf{a}$ is replaced by $q_a = \frac{K\varepsilon}{4nu(1+\varepsilon)}$. The results from Section 5.2 then follow.

S.M.5 Omitted discussions for Section 6

We present the remaining figures and tables from the computational experiments in Section 6 that were omitted for conciseness. Supplemental Material S.M.5.1 contains the additional figures for the knapsack interdiction problem, Supplemental Material S.M.5.2 provides the additional tables for the maximum weighted clique interdiction problem, and Supplemental Material S.M.5.3 presents additional experiments on non-interdiction instances from the literature.

S.M.5.1 Knapsack interdiction problem: continuous lower level (i.e., $\mathcal{Y} = \mathcal{Y}_c$)

Below are the complete figures from Section 6.2.1 whenever the follower's decision variables are all continuous. Specifically, Figure S.M.1 shows the results for the 2-flip neighborhood function, while Figure S.M.2 presents those for the 3-flip neighborhood function.



(a) Runtime

(b) Number of improving steps

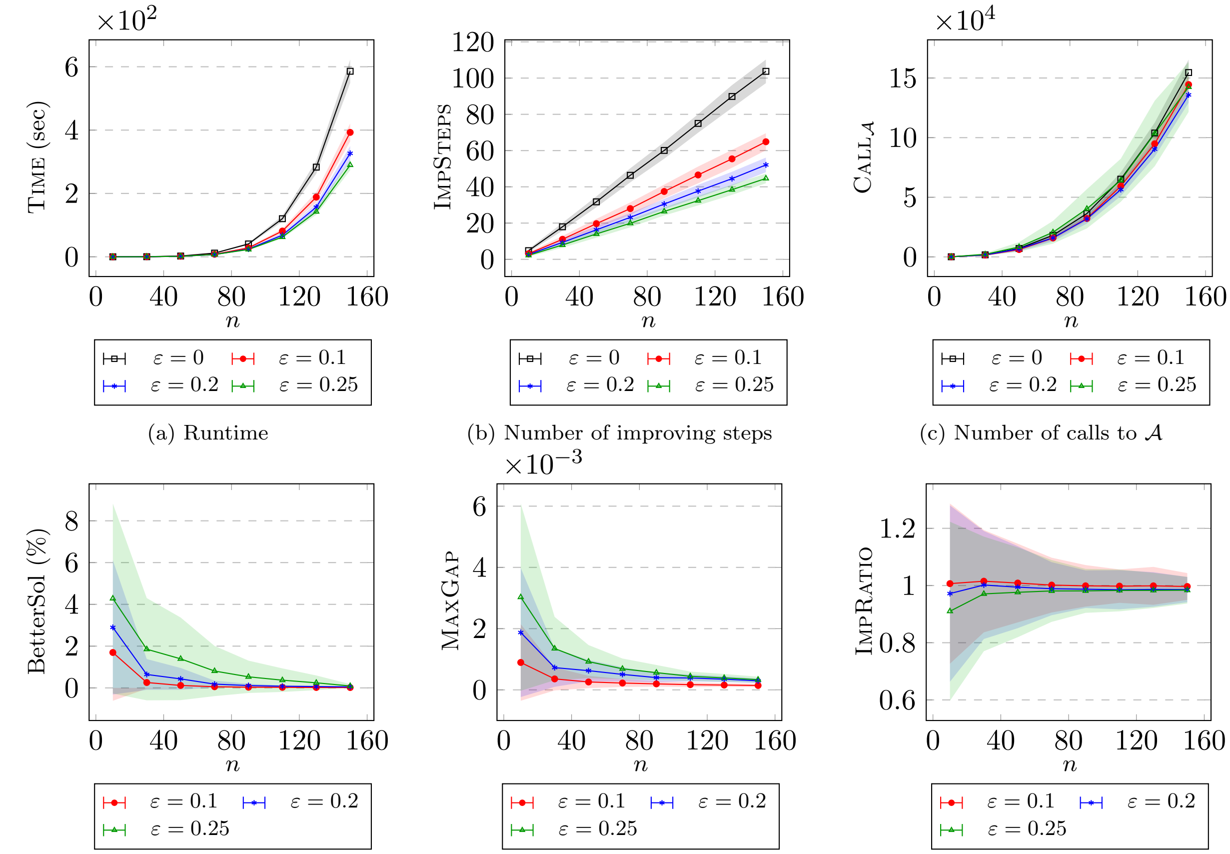
(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-LSA compared to the improvement of LSA

Figure S.M.1: **Continuous follower - exact follower ($\delta = 0$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.1. Recall that $\varepsilon = \delta = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-LSA. Each line shows the average (Avg), with the shaded region indicating Avg ± MAD.



(a) Runtime

(b) Number of improving steps

(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solutions compared to the size of the neighborhood

(e) Maximum gap between the solution obtained and improving solution in its neighborhood

(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-`LSA` compared to the improvement of LSA

Figure S.M.2: **Continuous follower - exact follower ($\delta = 0$) - 3-flip neighborhood ($k = 3$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.1. Recall that $\varepsilon = \delta = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-`LSA`. Each line shows the average (Avg), with the shaded region indicating Avg ± MAD.

S.M.5.2 Maximum weighted clique interdiction problem

Below are the missing tables from Section 6.3. Specifically, Table S.M.2 shows the results for the weak local search, where the lower-level problem is solved approximately, while Table S.M.3 presents the results for $(\varepsilon, \mathcal{A})$-LSA where the follower's problem is solved exactly.

		Time (sec)		ImpSteps		Call$_{\mathcal{A}}$		MaxGap		ImpRatio	
n	d	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD
40	0.5	23.6	7.7	7.2	1.8	340	104	$8.3 \cdot 10^{-3}$	$9.5 \cdot 10^{-3}$	0.88	$1.9 \cdot 10^{-1}$
40	0.7	17.2	7.6	9.5	2.5	415	145	$1.9 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	0.86	$1.5 \cdot 10^{-1}$
40	0.9	5.8	1.2	11.0	2.3	399	87	$2 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	0.89	$8.4 \cdot 10^{-2}$
50	0.5	94.0	32.1	9.3	2.3	637	214	$7.9 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	0.96	$8.8 \cdot 10^{-2}$
50	0.7	66.9	23.6	12.3	2.6	678	213	$8.1 \cdot 10^{-3}$	$9.9 \cdot 10^{-3}$	0.92	$1.3 \cdot 10^{-1}$
50	0.9	13.9	3.8	14.1	3.0	719	204	$2.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	0.88	$9.5 \cdot 10^{-2}$
60	0.5	158.7	40.7	10.1	2.2	766	205	$9.4 \cdot 10^{-3}$	$9 \cdot 10^{-3}$	0.90	$1.5 \cdot 10^{-1}$
60	0.7	168.5	53.7	13.3	2.2	944	297	$1.2 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	0.91	$8.9 \cdot 10^{-2}$
60	0.9	32.2	8.8	18.9	3.6	1,083	277	$2.6 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	0.84	$7.3 \cdot 10^{-2}$

Table S.M.2: $(\varepsilon, \mathcal{A})$-LSA **- inexact follower ($\delta = 0.1$) - 2-flip neighborhood function ($k = 2$):** comparison of the efficiency and performance metric for $(\varepsilon, \mathcal{A})$-LSA, where $(\delta, \varepsilon) = (0.1, 0)$, applied to the maximum weighted clique interdiction problem; recall (11) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

		Time (sec)		ImpSteps		Call$_{\mathcal{A}}$		MaxGap		ImpRatio	
n	d	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD	Avg	MAD
40	0.5	30.9	7.9	7.4	1.6	366	89	$6.1 \cdot 10^{-6}$	$1.2 \cdot 10^{-5}$	1.00	$1.5 \cdot 10^{-2}$
40	0.7	25.6	7.2	9.2	2.0	404	96	$4.7 \cdot 10^{-5}$	$8.4 \cdot 10^{-5}$	0.99	$1.2 \cdot 10^{-2}$
40	0.9	7.7	1.9	12.2	2.4	473	110	$3.7 \cdot 10^{-5}$	$7 \cdot 10^{-5}$	1.01	$1.3 \cdot 10^{-2}$
50	0.5	98.7	30.5	9.4	1.9	648	188	$5.6 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	0.99	$1.5 \cdot 10^{-2}$
50	0.7	82.1	22.5	12.0	2.1	714	197	$3 \cdot 10^{-5}$	$5.7 \cdot 10^{-5}$	0.98	$5.1 \cdot 10^{-2}$
50	0.9	22.3	5.5	14.6	2.8	767	200	$2 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	1.00	$8.4 \cdot 10^{-3}$
60	0.5	181.2	45.6	10.7	2.3	864	225	$6.5 \cdot 10^{-5}$	$1.1 \cdot 10^{-4}$	0.99	$2.9 \cdot 10^{-2}$
60	0.7	167.3	44.7	12.7	2.5	903	234	$7.9 \cdot 10^{-5}$	$1.3 \cdot 10^{-4}$	0.98	$3.8 \cdot 10^{-2}$
60	0.9	73.3	23.7	18.0	3.3	1,203	352	$2.7 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	1.00	$9 \cdot 10^{-3}$

Table S.M.3: $(\varepsilon, \mathcal{A})$-LSA **- exact follower ($\delta = 0$) - 2-flip neighborhood function ($k = 2$):** comparison of the efficiency and performance metric for $(\varepsilon, \mathcal{A})$-LSA, where $(\delta, \varepsilon) = (0, 0.1)$, applied to the maximum weighted clique interdiction problem; recall (11) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

S.M.5.3 Additional computational experiments on non-interdiction instances

We extend our computational experiments on some non-interdiction instances from the library compiled by [8] that are indicated by *general-bilevel*. We select instances, which are classified as *hard*, with $n, m < 300$, binary variables at both levels and no coupling constraints, resulting in 17 instances. For each, we solve the single-level relaxation to obtain a leader's feasible decision and discard any instance where this solution is already locally optimal, leaving 12 instances (their characteristics are summarized in Table S.M.4). If an instance does not satisfy our assumptions (in particular Assumption **A4**), then we apply the transformation described in Supplemental Material S.M.1.

Instance	n	m	q	p
cov1075-0-100	60	60	0	637
glass-sc-0-100	107	107	0	6119
iis-100-0-cov-0-100	50	50	0	3831
iis-bupa-cov-0-100	173	172	0	4803
lseu-0.500000	45	44	0	28
lseu-0.900000	9	80	0	28
mad-0-100	110	110	0	51
mas74-0-100	76	75	0	13
p0201-0.500000	101	100	0	133
p0282-0.500000	141	141	0	241
p0282-0.900000	29	253	0	241
p0548-0.500000	274	274	0	176

Table S.M.4: Summary of the instance sizes used in our additional computational experiments. Columns n and m denote the numbers of the leader's and follower's decision variables, respectively. Columns q and p correspond to the numbers of constraints at the upper- and lower-level problem, respectively. All instances are binary, do not have any coupling constraints, and are classified as *hard* by [8].

We then compare $(\varepsilon, \mathcal{A})$-`LSA` to the standard local search algorithm. First, we solve the single-level relaxation to obtain a leader's feasible decision, which serves as the initial solution, i.e., $\mathbf{x}_0$, for both approaches. Our computations are restricted to the 2-flip neighborhood function. As a side observation, none of the instances in Table S.M.4 contain constraints at the upper level, making them particularly challenging for local search methods due to the resulting large neighborhood size.

For $(\varepsilon, \mathcal{A})$-`LSA`, we set $\varepsilon = 0.25$ and solve the lower-level problem approximately (using $\mathcal{A}$, i.e., the same solver and computational setup as in Section 6, except that we allow Gurobi to use 32 threads) with a pre-specified gap $\delta = 0.25$ (recall Algorithm 2). The standard local search corresponds to the parameters $\varepsilon = \delta = 0$. For each method, we report the runtime, number of improving steps, number of calls to $\mathcal{A}$ and the improvement ratio as initially defined in Section 6.

We observe from Table S.M.5 below that the results are relatively consistent with those in Section 6. The main takeaway is that, for most instances, using $(\varepsilon, \mathcal{A})$-`LSA` reduces runtime while preserving solution quality. However, there are two notable exceptions.

Instance	Time (sec)			ImpSteps		Call$_{\mathcal{A}}$		ImpRatio
	0	0.25	relaxation	0	0.25	0	0.25	0.25
cov1075-0-100	40.06	39.40	1.38	1	1	1,289	1,289	1.00
glass-sc-0-100	342.01	352.19	285.82	3	3	6,073	6,073	1.00
iis-100-0-cov-0-100	59.30	56.18	83.24	7	7	1,294	1,294	1.00
iis-bupa-cov-0-100	844.14	810.32	223.00	5	5	15,617	15,617	1.00
lseu-0.500000	10.10	7.87	0.26	7	5	821	715	0.66
lseu-0.900000	1.50	0.31	0.25	1	0	51	45	0.00
mad-0-100	98.53	54.14	0.42	5	3	4,953	6,414	0.67
mas74-0-100	20.61	20.32	0.02	17	17	3,507	3,507	1.00
p0201-0.500000	98.60	92.81	0.15	1	2	11	14	1.00
p0282-0.500000	192.34	536.07	0.05	1	7	2,209	5,173	5.90
p0282-0.900000	4.55	4.60	0.08	10	10	523	523	1.00
p0548-0.500000	12,913.21	927.48	0.12	16	2	53,099	5,799	0.79

Table S.M.5: **LSA and $(\varepsilon, \mathcal{A})$-LSA - 2-flip neighborhood function ($k = 2$):** comparison of efficiency and performance metrics for both standard local search, indicated by column 0, and $(\varepsilon, \mathcal{A})$-LSA with $(\delta, \varepsilon) = (0.25, 0.25)$, indicated by column 0.25, on the instances described in Table S.M.4. Runtime for solving the single-level relaxation is also reported.

The first exception is *glass-sc-0-100*, where the number of improving steps and the number of calls to $\mathcal{A}$ for both the standard local search and $(\varepsilon, \mathcal{A})$-LSA are identical. We attribute this outcome to the fact that solving the follower's problem is relatively easy in this instance; hence, the solver finds the optimal solution almost immediately and is not slowed down by a nonzero optimality gap. The second exception is the instance *p0282-0.500000*, where the runtime for $(\varepsilon, \mathcal{A})$-LSA is higher because the algorithm follows a different improvement path, as indicated by the increased number of ImpSteps and Call$_{\mathcal{A}}$. Notably, this increase in runtime is accompanied by a high improvement ratio (see the ImpRatio column).

References

[1] Caprara, A., Carvalho, M., Lodi, A., and Woeginger, G. J. “Bilevel knapsack with interdiction constraints”. In: *INFORMS Journal on Computing* 28.2 (2016), pp. 319–333.

[2] Erdos, P., Rényi, A., et al. “On the evolution of random graphs”. In: *Publ. math. inst. hung. acad. sci* 5.1 (1960), pp. 17–60.

[3] Fischetti, M., Glover, F., and Lodi, A. “The feasibility pump”. In: *Mathematical Programming* 104 (2005), pp. 91–104.

[4] Garey, M. R. and Johnson, D. S. *Computers and intractability.* Vol. 174. USA: Freeman San Francisco, 1979.

[5] Gurobi. *Gurobi Optimization, LLC.* `https://www.gurobi.com`. Accessed: 2024-07-26. 2024.

[6] Kleinert, T., Labbé, M., Ljubić, I., and Schmidt, M. “A survey on mixed-integer programming techniques in bilevel optimization”. In: *EURO Journal on Computational Optimization* 9 (2021), p. 100007.

[7] Tavaslıoğlu, O., Prokopyev, O. A., and Schaefer, A. J. “Solving stochastic and bilevel mixed-integer programs via a generalized value function”. In: *Operations Research* 67.6 (2019), pp. 1659–1677.

[8] Thürauf, J., Kleinert, T., Ljubić, I., Ralphs, T., and Schmidt, M. “BOBILib: Bilevel Optimization (Benchmark) Instance Library”. In: *Mathematical Programming Computation* (2026), pp. 1–58.

[9] Vavasis, S. A. “Polynomial time weak approximation algorithms for quadratic programming”. In: *Complexity in Numerical Optimization.* World Scientific, 1993, pp. 490–500.

[10] Vazirani, V. V. *Approximation algorithms.* Springer Berlin, Heidelberg, 2001.

[11] Vicente, L., Savard, G., and Judice, J. “Discrete linear bilevel programming problem”. In: *Journal of Optimization Theory and Applications* 89 (1996), pp. 597–614.

[12] Weninger, N. and Fukasawa, R. “A fast combinatorial algorithm for the bilevel knapsack problem with interdiction constraints”. In: *Mathematical Programming* 210.1 (2025), pp. 847–879.

[13] Wu, Q. and Hao, J.-K. “A review on algorithms for maximum clique problems”. In: *European Journal of Operational Research* 242.3 (2015), pp. 693–709.