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. Additionally, these methods also play a key role within state-of-the-art solvers to improve feasibility bounds. Their appeal lies in fast implementation and scalability; 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 mixed-integer optimization, local search, computational complexity

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 offers significant modeling capabilities, particularly in applications that involve settings with decentralized decision-making processes. Accordingly, bilevel programs arise in various application domains, including resource allocation (Correa et al. 2017; Khorramfar et al. 2022), facility location (Dan and Marcotte 2019; Lin et al. 2024), price setting (Labbé et al. 1998; Kuiteing et al. 2017), network and market design (Tawfik and Limbourg 2019; Bichler and Waldherr 2022) as well as law enforcement (Arslan et al. 2018) and defense (Gutin et al. 2015; Dahan et al. 2022) related problems. We refer to recent surveys by Kleinert et al. (2021b) and Beck et al. (2023) for a comprehensive overview of bilevel programming.

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

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

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

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

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:

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

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:

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

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 (Gurobi 2024; IBM 2024), 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 problems (Aarts and Lenstra 2003). In this role, they are particularly valuable when global optimality is out of reach, as they still provide meaningful 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. In this study, we identify two key challenges that limit the practical use of local search for bilevel MILPs. First, the algorithm may require an exponential number of improving steps before converging. Second, at each step, evaluating the leader's objective function involves solving the follower's problem, which is often computationally demanding in real-world settings. This work directly addresses these challenges, and our main 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 by Orlin et al. (2004) 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 by Orlin et al. (2004). 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 might be evaluated with an inexact follower's response) to a weak ε-local optimal solution. Additionally, we identify a sufficient condition and specific problem classes, where our approach 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 (Kleinert et al. 2021b). 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:

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

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 $\mathbf{x}$.

While the optimistic assumption is, perhaps, the most commonly used in the bilevel optimization literature (Kleinert et al. 2021b), there also exists the *pessimistic* model in which the follower chooses an optimal decision that is the least favorable from the leader's perspective (Wiesemann et al. 2013). Furthermore, the optimistic and pessimistic responses can be considered simultaneously, resulting in what is known as the strong-weak model (Lagos and Prokopyev 2023).

To simplify notation and 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 Appendix A. Moreover, recent work has shown that, when both the leader's and follower's decision variables are continuous, coupling constraints do not increase modeling power since they can be incorporated as penalty terms in the leader's objective function (Henke et al. 2025). However, identifying appropriate penalizations, and extending this result to bilevel MILPs, remains an open question.

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 $\|\mathbf{y}\|_1 \leq U$ for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and for all $\mathbf{x} \in \mathcal{X}$.
- **A3:** $\mathbf{a} \in \mathbb{R}_+^n$ and $\mathbf{d} \in \mathbb{R}_+^m$.
- **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 Appendix A.

The remainder of the paper. Section 2 provides an overview of the relevant literature. In Section 3, we introduce generalizations of local optimality to the bilevel setting, which are then explored in details in Section 4. Section 5 presents the weak approximate local search algorithm, or $(\varepsilon, \mathcal{A})$-`LSA`, and discusses its theoretical properties, including its worst-case performance. Section 6 offers numerical evidence supporting our theoretical findings. The paper concludes with Section 7. All proofs, additional discussions, and extensions of our approach, are provided in the appendix. Finally, Table 1 summarizes the solution concepts and algorithms discussed in this study.

<table>
<tr><th rowspan="2"></th><th colspan="2">Follower's response</th><th rowspan="2">Algorithm</th></tr>
<tr><th>Exact (Section 3.1)</th><th>Inexact (Section 3.2)</th></tr>
<tr><td rowspan="2">Leader</td><td>local optimality</td><td>weak local optimality</td><td>(weak) local search</td></tr>
<tr><td>ε-local optimality</td><td>weak ε-local optimality</td><td>$(\varepsilon, \mathcal{A})$-LSA (Section 5)</td></tr>
</table>

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. The significant modeling capabilities of bilevel programs come with increased computational complexity. Indeed, if both the leader's and the follower's decision variables are all continuous, then bilevel programs are known to be NP-hard (Ben-Ayed and Blair 1990). In fact, even finding a local optimal solution is NP-hard (Prokopyev and Ralphs 2024). Moreover, it is worth noting that even special classes of bilevel programs, such as linear min/max programs, remain strongly NP-hard (Hansen et al. 1992). This complexity extends to pessimistic (Wiesemann et al. 2013) and strong-weak (Lagos and Prokopyev 2023) models.

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 (Caprara et al. 2016; Fischetti et al. 2019), matching interdiction (Zenklusen 2010; Dinitz and Gupta 2013), and critical node detection (Mahdavi Pajouh et al. 2014; Furini et al. 2020) as well as decentralized versions of various network-related (Correa et al. 2017) and facility location (Dan and Marcotte 2019) problems. Further insights into solution methods for various classes of bilevel MILPs can be found in the survey by Kleinert et al. (2021b).

For bilevel MILPs with continuous follower's decision variables, exact methods are reasonably well-established and typically rely on reformulating the problems as single-level MILPs by leveraging Karush–Kuhn–Tucker conditions (Audet et al. 1997; Dempe and Zemkoho 2013) or strong duality (Zare et al. 2019; Kleinert et al. 2021a; Kleinert and Schmidt 2023). On the other hand, if the follower's decision variables involve integrality 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., Tavaslıoğlu et al. (2019). In fact, introducing binary decision variables for the follower escalates the computational complexity to Σ_2^p-hardness (Jeroslow 1985). 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 (Moore and Bard 1990). Other methods in the literature rely on decomposition (Saharidis and Ierapetritou 2009; Bolusani and Ralphs 2022) or parametric programming (Domínguez and Pistikopoulos 2010; Köppe et al. 2010; Lozano and Smith 2017; Tavaslıoğlu et al. 2019) techniques. Although a few basic solvers for bilevel MILPs based on branch-and-cut ideas are available (Fischetti et al. 2017; Tahernejad et al. 2020), 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; see, to name a few, the studies by Kanellakis and Papadimitriou (1980), Arkin and Hassin (1998), Schuurman and Vredeveld (2007), and Bertsimas et al. (2013). On the other hand, local search can take an exponential number of improving steps to converge; see, for instance, a classical example by Chandra et al. (1999) for the traveling salesman problem. This worst-case behavior is addressed by Orlin et al. (2004), 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 (Wen and Huang 1996; Nishizaki and Sakawa 2005) and focus on specific problem classes (Brotcorne et al. 2001; Dussault et al. 2006; Fischetti et al. 2018). 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 (Dempe 1987; Still 2002). Alternative concepts, such as stationary solutions, have also been studied in the literature (Kleinert and Schmidt 2021). Additionally, local optimality can be approached from the follower's perspective (Shi et al. 2023), 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 (Garey and Johnson 1979). 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 (Aarts and Lenstra 2003) 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 (Deineko and Woeginger 2000; Ahuja et al. 2002; Altner et al. 2013). Among those, the *k-flip neighborhood* function is, perhaps, the most commonly used one due to its simplicity (Aarts and Lenstra 2003). 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:

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

where $\|\cdot\|_1$ represents the 1-norm, also known 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 (Aarts and Lenstra 2003). 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. In bilevel programming, 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 Vicente et al. (1994). 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 explicitly construct a class of bilevel MILPs, where the lower-level problem is a linear program (LP) with binary optimal decisions, that exhibit this exponential behavior for LSA. Since this phenomenon is common for many combinatorial optimization problems, the detailed construction and the related discussion are relegated to Appendix B.1.

This result 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, even for relatively modest instance sizes, standard local search may converge "slowly"; see Section 6.

Approximate local optimality. The concept of ε-local optimality, which generalizes the traditional definition of local optimality, is introduced by Orlin et al. (2004) 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:

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

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 feasible 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, as in the above, this result and its proof are relegated to Appendix B.2 for conciseness. 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 (Zare et al. 2020; Shi et al. 2023).

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 unique 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:

$$(1 - \delta)\mathbf{c}^\top \mathbf{y}^*(\mathbf{x}) \leq \mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \tag{6}$$

for any $\mathbf{x} \in \mathcal{X}$. Moreover, if $\mathcal{A}$ satisfies (6), 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 (6).

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., Vazirani (2001). 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 unique feasible decision for each $\mathbf{x} \in \mathcal{X}$. Examples include δ-approximation algorithms as outlined in (6), 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 unique feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the lower-level problem (1c). 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:

$$\mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \tag{7}$$

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 (7), 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:

$$\Delta^{\mathcal{A}}\left(\mathbf{x}^0, \mathbf{x}, \mathbf{a}, \mathbf{d}\right) := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) - \left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right). \tag{8}$$

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 by using weak local search, an extension of local search, where the leader's objective function is computed with an inexact follower's response rather than the follower's optimal decision. Its description is relegated to Appendix B.1, along with an analysis of its computational complexity.

Specifically, we construct a class of bilevel MILPs for which weak local search converges in an exponential number of improving steps. The considered weak local search is based on a naive heuristic that takes a linear 0-1 program as an input, solves its LP relaxation, and returns a feasible decision by rounding. Note that, in general, this procedure is not even guaranteed to return either an optimal, or even a follower's feasible decision.

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 unique feasible solution $\mathbf{y}^{\mathcal{A}}(\mathbf{x})$ to the lower-level problem (1c). 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 Example: maximum weighted clique interdiction problem

Interdiction problems are among the most widely studied models in bilevel MILPs. They often represent a zero-sum game between an attacker disrupting a system and a defender (Smith and Song 2020), but can also capture non-zero-sum interactions, such as a decision-maker seeking to hedge against the worst-case outcome of its environment (e.g., natural disasters). Common applications include cyber defense, infrastructure security, and disrupting illicit supply chains. Also, these problems constitute a large portion of the available computational benchmarks (Thürauf et al. 2024).

In particular, we consider the problem of reducing the influence of tightly connected groups (for instance, to limit the spread of fake news in social medias) by strategically blocking key actors in a network to weaken the most influential clusters. To proceed, we introduce the maximum weighted clique interdiction problem, or shortly `CIP`, which involves an undirected weighted graph $G = (V, E)$, where V and E represent the sets of vertices and edges, respectively (Furini et al. 2019). The leader's goal is to strategically interdict (i.e., block) certain vertices in G. Following the leader's interdictions, the follower's task is to solve the maximum weighted clique problem in the interdicted graph. Let w_i denote the weight of vertex $i \in V$. Then, `CIP` is defined as follows:

$$[\textbf{CIP}] : \quad z^* := \min_{\mathbf{x}, \mathbf{y}^*(\mathbf{x})} \quad \sum_{i \in V} w_i y^*(\mathbf{x})_i \tag{9a}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{X} := \Big\{ \mathbf{x} \in \{0,1\}^{|V|} \ : \ \sum_{i \in V} x_i \leq h \Big\}, \tag{9b}$$

$$\mathbf{y}^*(x) \in \underset{\mathbf{y} \in \{0,1\}^{|V|}}{\operatorname{argmax}} \Big\{ \sum_{i \in V} w_i y_i \ : \ y_i + y_j \leq 1 \ \forall \, (i,j) \notin E \, , \, \mathbf{y} \leq \mathbf{1} - \mathbf{x} \Big\}, \tag{9c}$$

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 (9c). Given a clique $\mathcal{C} \subseteq V$, its weight is given by $\omega(\mathcal{C}) := \sum_{i \in \mathcal{C}} w_i$.

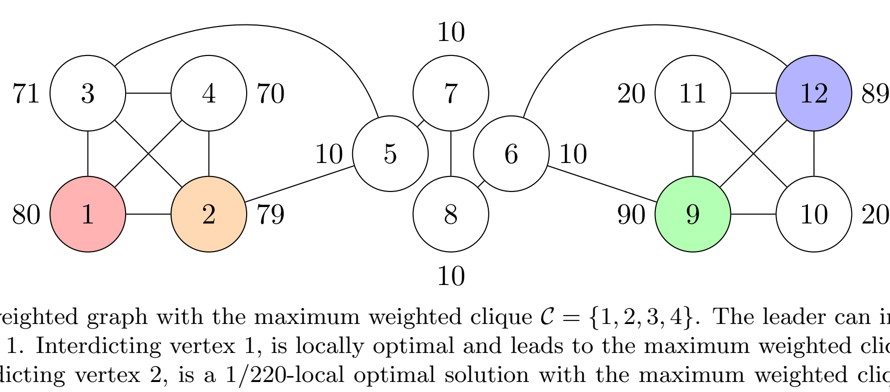


Figure 1: A weighted graph with the maximum weighted clique $\mathcal{C} = \{1,2,3,4\}$. The leader can interdict up to one vertex, i.e., $h = 1$. Interdicting vertex 1, is locally optimal and leads to the maximum weighted clique $\mathcal{C}^* = \{2,3,4\}$. Similarly, interdicting vertex 2, is a 1/220-local optimal solution with the maximum weighted clique $\mathcal{C}^\varepsilon = \{1,3,4\}$. Using a greedy search procedure, interdicting vertex 9, is weakly local optimal, while interdicting vertex 12, is weakly 1/129-local optimal. The estimated cliques are given by $\mathcal{C}^{\mathcal{A}} = \{10,11,12\}$ and $\mathcal{C}^{\varepsilon,\mathcal{A}} = \{6,9,11\}$, respectively.

Exact follower. We fix $h = 1$ in (9b) 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 $\omega(\mathcal{C}^*) = 220$.

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, we assume that the follower's response is obtained using a greedy heuristic $\mathcal{A}$ (see a pseudo-code in Appendix B.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 (Smith et al. 2013). If $\mathbf{a} = \mathbf{0}$ and $\alpha = 1$, then the leader and the follower essentially engage in a zero-sum game.

Proposition 1. *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 $N_{\mathcal{X}}$.*

Note that Proposition 1 remains valid for $\varepsilon = 0$. Thus, under the assumptions of Proposition 1, a natural relationship between weak and approximate local optimality is established. Specifically:

Corollary 1. *Under the same assumptions as in Proposition 1, 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 (Blair and Jeroslow 1977; Mangasarian and Shiau 1987) and extend Proposition 1 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:

$$\Pi\left(\varepsilon, r, \underline{z}\right) := \varepsilon + \frac{\left(2+\varepsilon\right) r d_{\max}}{\underline{z}}, \tag{10}$$

where $\underline{z} > 0$ is some lower bound for [**BP**] and $d_{max} := \max_i \{d_i\}$ is the maximum element of vector $\mathbf{d}$. Similarly, let $d_{min} := \min_i \{d_i\}$. 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**. Alternatively, if the leader's optimal decision is known to be non-zero, i.e., $\mathbf{x} \neq \mathbf{0}$, then $\underline{z} = a_{\min}$ can be chosen accordingly. A more sophisticated lower bound can be obtained by, first, relaxing the optimality criteria in [**BP**], and then solving the LP relaxation of the resulting single-level problem.

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. If $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly ε-local optimal with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$, then there exist $\gamma_1 \geq 0$ and $\gamma_2 \geq 0$ such that $\mathbf{x}^{\varepsilon,\mathcal{A}}$ 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 a controllable performance guarantee δ, then Theorem 1 implies that the approximation guarantee Π can be entirely controlled using ε and δ. In particular, in this case, Π is defined such that $\lim_{(\varepsilon,\delta)\to(0,0)} \Pi\left(\varepsilon, \gamma_1\delta + \gamma_2, \underline{z}\right) = 0$. Finally, the proofs for this section can be found in Appendix C.

Empirical analysis. At first glance, the bound established in Theorem 1 may seem rather loose, particularly in worst-case scenarios. Accordingly, we examine the convergence behavior and empirically evaluate the maximum gap for the maximum clique interdiction problem discussed in Section 3.3. For a leader's feasible decision $\mathbf{x} \in \mathcal{X}$, the *empirical maximum gap* is defined as follows:

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

Specifically, our analysis focuses on the asymmetric setting, which yields the weakest theoretical guarantees. We report the empirical maximum gap attained by $\Pi\left(\varepsilon, \gamma_1 \delta, \underline{z}\right)$-local optimal solutions across various values of δ and ϵ in Figure 2. Further details on the construction of the instances and computational setup are provided in the Appendix C.3.

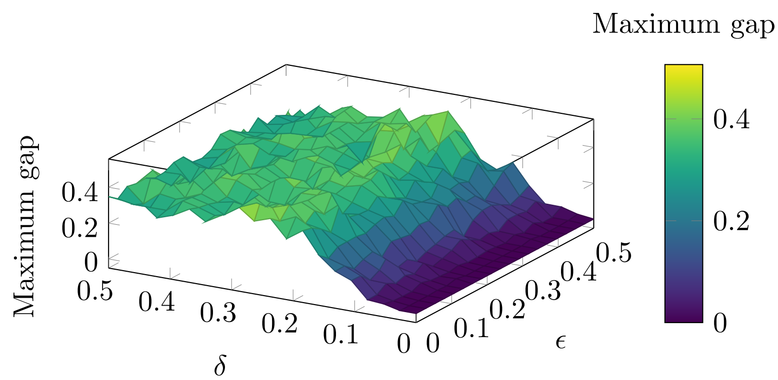


Figure 2: Surface plot of the empirical maximum gap of a leader's decision $x^{\varepsilon,\mathcal{A}}$, which is weakly ε-local optimal with respect to the 2-flip neighborhood function and a δ-approximation algorithm. The plot shows the gap as a function of δ and ε for an instance of the asymmetric interdiction clique problem. Details on the construction of the instances and the figure are provided in Appendix C.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 can become arbitrarily large, as there may still exist leader's feasible decisions within the neighborhood of $\mathbf{x}$ that reduce the leader's objective by 40%.

Remark 1. We note that a similar finding have been briefly discussed by Weninger and Fukasawa (2025) in the more restrictive context of bilevel knapsack (symmetric) interdiction problems. They highlight that solving the follower's problem approximately may offer a promising direction for addressing bilevel programs, an idea also supported by our study. ■

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 for its running-time performance 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. Most of the discussion for the latter case is relegated to Appendix D to streamline our discussion.

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 (Altner et al. 2013). Also, multiple improving solutions may exist within the neighborhood, necessitating some tie-breaking rules.

Algorithm 1 - IMPROVE - oracle for the neighborhood search

function IMPROVE($\mathbf{x}^k$, $N_{\mathcal{X}}$, $\mathbf{a}$, $\mathbf{d}$, $\mathcal{A}$, γ)
    if $\exists\, \mathbf{x}^{k+1} \in N_{\mathcal{X}}(\mathbf{x}^k)$ such that $\Delta^{\mathcal{A}}\left(\mathbf{x}^k, \mathbf{x}^{k+1}\right) > \gamma$ **then**
        "answer" $\leftarrow$ "Yes"
    else
        "answer" $\leftarrow$ "No"
        $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k$
    Return *"answer"*, $\mathbf{x}^{k+1}$

Importantly, in our discussion, we do not impose any restrictions on the specific approach used for the neighborhood search. Hence, we introduce an oracle IMPROVE, which is described in Algorithm 1. Specifically, given a leader's feasible decision $\mathbf{x}^k \in \mathcal{X}$ and $\gamma \geq 0$, IMPROVE 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?

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

function $(\varepsilon, \mathcal{A})$-$\texttt{LSA}(\mathbf{x}^0, N_{\mathcal{X}}, \mathbf{a}, \mathbf{d}, \mathcal{A}, \varepsilon)$
  $i \leftarrow 0$, $\mathbf{x}^i \leftarrow \mathbf{x}^0$, *scaling* $\leftarrow$ True
  while *scaling* **do**
    Obtain $\mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)$ by calling $\mathcal{A}$ // Inexact follower
    $K \leftarrow \mathbf{a}^\top \mathbf{x}^i + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)$
    $q_a \leftarrow \frac{K\varepsilon}{4n(1+\varepsilon)}$, $q_d \leftarrow \frac{K\varepsilon}{4(m+1)U(1+\varepsilon)}$
    $a'_j \leftarrow q_a \left\lceil \frac{a_j}{q_a} \right\rceil$ for $j \in [n]$, $d'_\ell \leftarrow q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil$ for $\ell \in [m]$ // Scaling
    $\gamma \leftarrow U\left(m + q_d^{-1} \sum_{\ell=1}^m d_\ell\right)^{-1} \sum_{\ell=1}^m d'_\ell$ if $m_2 > 0$ else $\gamma \leftarrow 0$ // Improvement gap
    $k \leftarrow 0$, $\mathbf{x}^{i,k} \leftarrow \mathbf{x}^i$
    while *scaling* and $\left(\mathbf{a}^\top \mathbf{x}^{i,k} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k}) > \frac{K}{2}\right)$ **do**
      (*"answer"*, $\mathbf{x}^{k+1}) \leftarrow \textsc{Improve}(\mathbf{x}^k, N_{\mathcal{X}}, \mathbf{a}', \mathbf{d}', \mathcal{A}, \gamma)$ // Neighborhood search
      if *"answer"* is "Yes" **then**
        $\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k+1})$ by calling $\mathcal{A}$ // Inexact follower
        $\mathbf{x}^{i,k} \leftarrow \mathbf{x}^{i,k+1}$, $k \leftarrow k+1$
      else
        $\mathbf{x}^{\varepsilon,\mathcal{A}} \leftarrow \mathbf{x}^{i,k}$, *scaling* $\leftarrow$ False
    $\mathbf{x}^i \leftarrow \mathbf{x}^{i,k}$, $i \leftarrow i+1$
  Return $\mathbf{x}^{\varepsilon,\mathcal{A}}$

Weak approximate local search. In the context of single-level linear 0-1 programming, *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., Lawler (1977). Loosely speaking, scaling is employed to reduce the number of distinct values of the objective 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 Orlin et al. (2004), 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 Orlin et al. (2004), 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 7. The scaling factors depend (adaptively) on the leader's objective function value computed before.

Then, a weak local search is performed as a subroutine using these adjusted vectors within the while loop at line 10. 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 10, 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 $\mathbf{d}'$.

A detailed example of $(\varepsilon, \mathcal{A})$-`LSA`, applied to the interdiction problem discussed in Section 3.3, is provided in Appendix D.1. The example illustrate both exact and inexact follower's responses.

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

We assume that both Improve 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. The proofs from this section can be found in Appendix D.2. Next, 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 1. We use this result to establish the runtime complexity of $(\varepsilon, \mathcal{A})$-`LSA`; see Theorem 2.
- By Theorem 2, $(\varepsilon, \mathcal{A})$-`LSA` terminates for any $\varepsilon \geq 0$, 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 3.

- Finally, we propose two further results under additional assumptions. That is, we introduce a sufficient condition for $\mathbf{x}^{\varepsilon,\mathcal{A}}$ to be a weak local optimal solution (i.e., $\varepsilon = 0$); see Proposition 2. Then, 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 4.

Lemma 1. *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 $i \in \mathbb{Z}_{\geq 0}$. Then, there exists $\Delta > 0$ such that 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 (8) *and Δ is a constant parameter defined as $\Delta := \frac{\varepsilon K}{4(1+\varepsilon)(m+1)n}$.*

We exploit Lemma 1 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 Improve and $\mathcal{A}$. Formally:

Theorem 2. *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 $K_0 := \mathbf{a}^\top\mathbf{x}^0 + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq na_{\max} + md_{\max}$. Then, $(\varepsilon, \mathcal{A})$-`LSA` terminates and its runtime complexity is in the order of $\mathcal{O}\big(\frac{1}{\varepsilon}nm\log\big(K_0\big)\left(\mathcal{C}_I + \mathcal{C}_{\mathcal{A}}\right)\big)$.*

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

Theorem 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. 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 $\mathcal{A}$.*

The theoretical maximum gap ε obtained by our approach is asymptotically sharp whenever the follower's problem is solved exactly. Specifically, in Appendix D.3, 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 3 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 Improve 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 δ-approximation algorithm, then Theorem 1 can be leveraged to strengthen Theorem 3. 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* Improve *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* (10).

Weak local optimality. Interestingly, a sufficient condition can be identified for which $(\varepsilon, \mathcal{A})$-`LSA` is ensured to return a weak local optimal solution to [**B-BP**]. This condition serves as a posteriori optimality certificate, as it depends on the values of q_a and q_d obtained during the last iteration i_f, before the algorithm stops. Deriving this result requires a somewhat deeper understanding of the scaling process within $(\varepsilon, \mathcal{A})$-`LSA`. For simplicity, we focus our discussion on vector $\mathbf{a}$ in the leader's objective function, though similar considerations apply to vector $\mathbf{d}$.

Consider $\varepsilon > 0$, and let $q_a > 0$ denote the scaling factor obtained before $(\varepsilon, \mathcal{A})$-`LSA` terminates. We define the step-wise function $f : \mathbb{R} \to \mathbb{R}$ as $f(u) := q_a \left\lceil \frac{u}{q_a} \right\rceil$, for $u > 0$. Clearly, if u is a multiple of q_a, then the scaling has no effect on u, i.e., $u = k_a q_a$ for $k_a \in \mathbb{Z}_{\geq 0}$ implies $f(u) = u$. If, in the last iteration i_f in $(\varepsilon, \mathcal{A})$-`LSA`, each component of $\mathbf{a}$ and $\mathbf{d}$ is a multiple of q_a and q_d, respectively, then $(\varepsilon, \mathcal{A})$-`LSA` finds a local optimal solution to [**B-BP**]. We generalize this observation for instances, where the components of $\mathbf{a}$ and $\mathbf{d}$ are sufficiently close to a multiple of q_a and q_d, respectively.

For each $j \in [n]$, there exist $p_j \in \mathbb{Z}_{\geq 0}$ and $\alpha_j \in [0, q_a)$ such that $a_j = p_j q_a - \alpha_j$. Similarly, for each $\ell \in [m]$, there exist $s_\ell \in \mathbb{Z}_{\geq 0}$ and $\beta_\ell \in [0, p_d)$ such that $d_\ell = s_\ell q_d - \beta_\ell$. Then, we define:

$$\Delta_x^* := \max_{\mathbf{x} \in \mathcal{X},\, \tilde{\mathbf{x}} \in N_{\mathcal{X}}(\mathbf{x})} \|\mathbf{x} - \tilde{\mathbf{x}}\|_1 \quad \text{and} \quad \Delta_y^{\mathcal{A}} := \max_{\mathbf{x} \in \mathcal{X},\, \tilde{\mathbf{x}} \in N_{\mathcal{X}}(\mathbf{x})} \left\|\mathbf{y}^{\mathcal{A}}(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\tilde{\mathbf{x}})\right\|_1,$$

which represent, respectively, the maximum distance between a leader's feasible decision and another one in its neighborhood, and the maximum difference between the follower's responses corresponding to two leader's decisions, where one of them is in the neighborhood of the other. Observe that the former is bounded by n, while the latter is bounded by m. Then:

Proposition 2. *Let $\varepsilon > 0$, $N_{\mathcal{X}}$ be a neighborhood function, $\mathcal{A}$ be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Assume that K, q_a and q_d are the parameters obtained at the last iteration i_f in Algorithm 2 before it terminates. Let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ be the solution obtained by $(\varepsilon, \mathcal{A})$-`LSA`. If the following condition holds:*

$$\Delta_x^* \sum_{j=1}^{n} \alpha_j + \Delta_y^{\mathcal{A}} \sum_{\ell=1}^{m} \beta_\ell \leq \frac{K\varepsilon}{4\,(m+1)\,n\,(1+\varepsilon)}, \tag{11}$$

then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is weakly local optimal with respect to $N_{\mathcal{X}}$ and $\mathcal{A}$.

Thus, the condition in (11) offers a sufficient criterion for verifying (weak) local optimality a posteriori. Furthermore, if the follower's problem is solved exactly, then Proposition 2 ensures that the solutions obtained by $(\varepsilon, \mathcal{A})$-`LSA` are locally optimal whenever they satisfy (11).

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., Furini et al. (2019). Formally:

Theorem 4. *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 $\mathcal{A}$.*

In particular, Theorem 4 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 2 and 3) 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 outline only some brief details for the mixed-integer case in Section 5.3 bellow, and refer to Appendix D.4 for the complete discussion. Some relevant extensions of $(\varepsilon, \mathcal{A})$-`LSA` are detailed in Appendix D.5.

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 8 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 improvement 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* $\tilde{K}_0 := \mathbf{a}^\top \mathbf{x}^0 + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^0) \leq n a_{\max} + mU d_{\max}$*. Then,* $(\varepsilon, \mathcal{A})$-`LSA` *terminates and its runtime complexity is in the order of* $\mathcal{O}\big(\frac{1}{\varepsilon} m \log\big(\tilde{K}_0\big)\,(\mathcal{C}_I + \mathcal{C}_{\mathcal{A}})\big)$.

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 2). 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 naive local search. In particular, 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 (Thürauf et al. 2024). 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; see, e.g., Garey and Johnson (1979) and Bernhard and Vygen (2008). 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 (Garey and Johnson 1979); 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 the leader's feasible decision $\mathbf{x}^0 = \mathbf{0}$. In the experiments, only the k-flip neighborhood as defined in (4), is considered, with $k \in \{2, 3\}$, where by default $k = 2$. Finally, Improve iterates over the neighborhood and selects the first improving solution found. Also, we assume that $(\varepsilon, \mathcal{A})$-`LSA` performs a weak local search whenever $\varepsilon = 0$ (we refer to Appendix B.1 for the corresponding pseudo-code).

Efficiency. We evaluate all considered algorithms using three distinct measures. That is: (i) *Running time* (Time): it measures the runtime in seconds; (ii) *Number of improving steps* (ImpSteps): it corresponds to the number of times that Improve returns "Yes;" (iii) *Number of calls to* $\mathcal{A}$ (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 (evaluted 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:

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

where $\mathbf{x}^0$ is the initial feasible solution used as the starting point for the 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 (with setting "Threads = 1").

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:

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

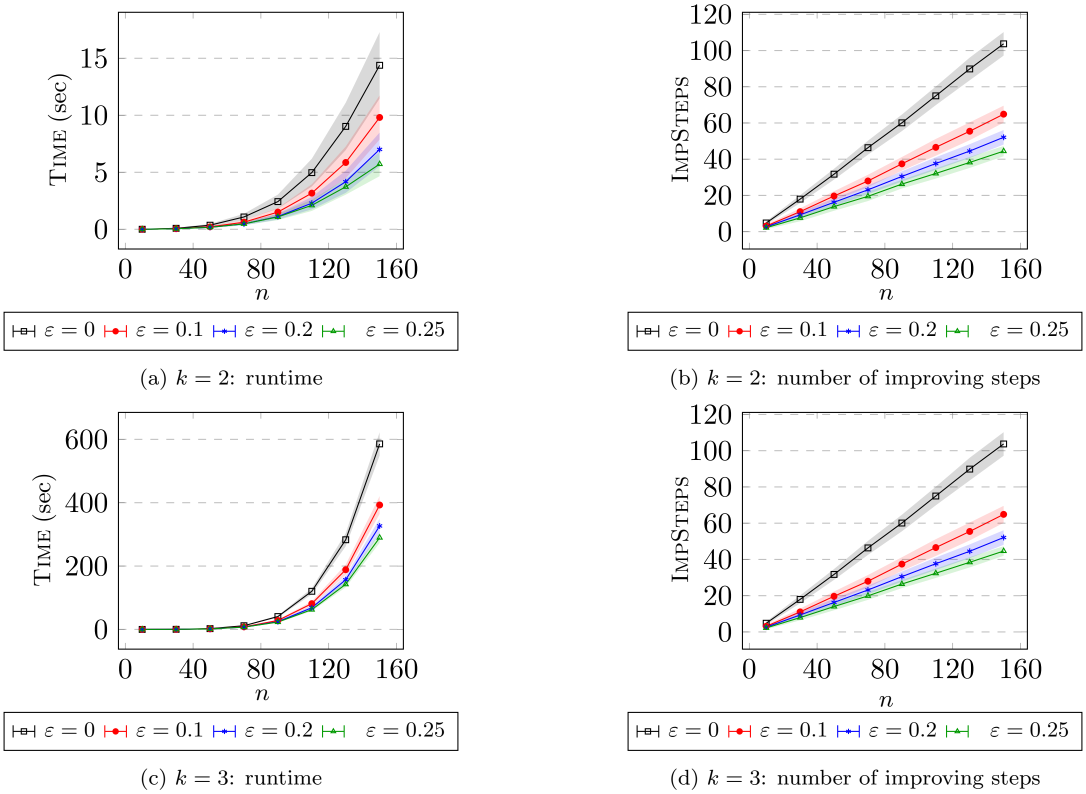
where $n := n_1 + n_2$. In our experiments, the cost vectors are generated using a uniform distribution $\mathcal{U}$. That is, $\mathbf{a}' \sim \mathcal{U}(\{1000, \ldots, 1100\}^n)$, $\mathbf{a} = 0.01 \times \mathbf{a}'$, and $\mathbf{c} \sim \mathcal{U}(\{1000, \ldots, 1100\}^n)$. The constraints are generated in a similar manner with $\mathbf{F} \sim \mathcal{U}(\{1000, \ldots, 1100\}^{q \times n})$ and $\mathbf{f} = 0.4\mathbf{F}\mathbf{1}$. Additionally, the leader can interdict up to 30% of the follower's items. In the experiments bellow, 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 value* (MAD) over these instances.

The discussion bellow 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 (Dantzig 1957). We let $\varepsilon \in \{0, 0.1, 0.2, 0.25\}$, where $\varepsilon = 0$ corresponds to LSA.

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 $\text{Call}_{\mathcal{A}}$, BetterSol, MaxGap,



(a) $k = 2$: runtime

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

(c) $k = 3$: runtime

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

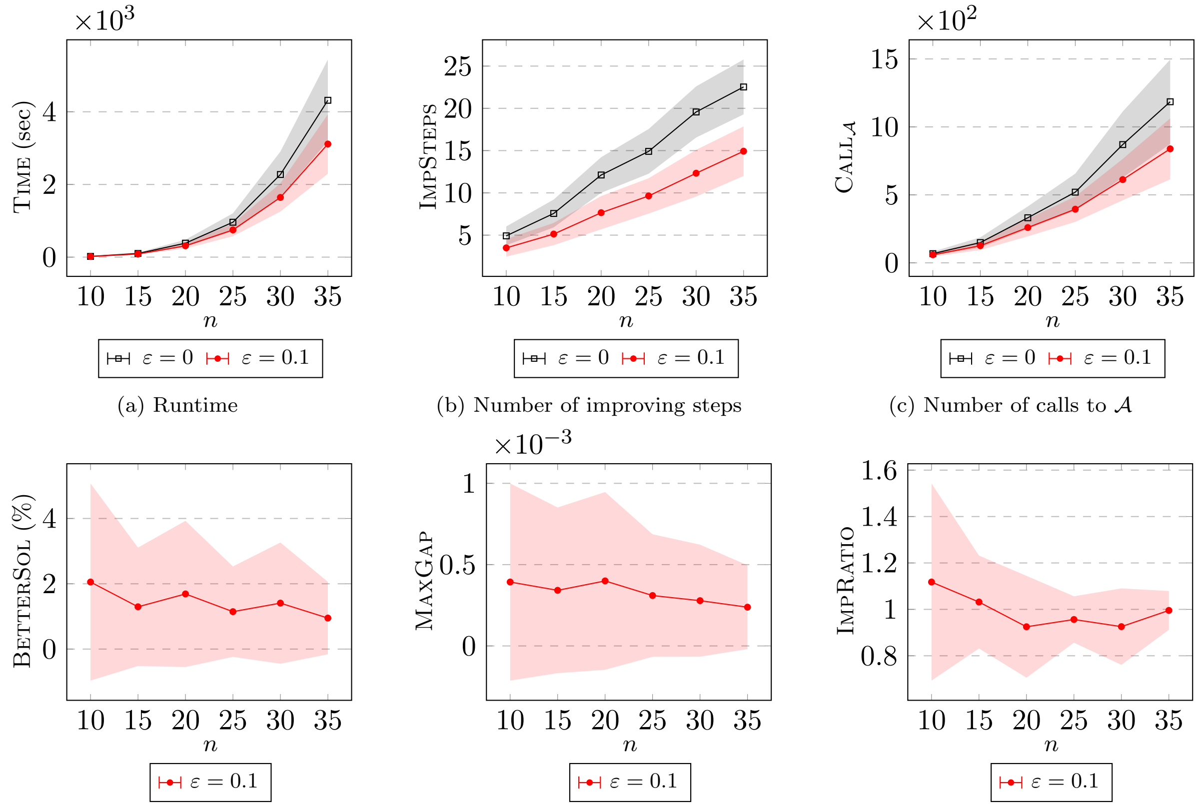
Figure 3: **Continuous follower - exact follower ($\delta = 0$) - 2- and 3-flip neighborhoods ($k \in \{2, 3\}$):** comparison of the runtime and the number of improving steps for the knapsack interdiction problem; see Section 6.2.1. Recall that $\varepsilon = 0$ corresponds to LSA, while $\varepsilon > 0$ 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.

and ImpRatio for both the 2-and 3-flip neighborhoods. The complete set of figures is available in Appendix E.1. We then make the following observations:

- First, **scaling** the vectors in the leader's objective does **improve the running time**; see Figure 3a. Specifically, the improvement is more pronounced for larger values of ε, which is intuitive. Note that the running time 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 more or less linearly in n, even for local search; see Figure 3b. Nevertheless, as ε increases, the slope becomes less steep, which aligns with the theoretical worst-case performance of $(\varepsilon, \mathcal{A})$-`LSA`, which is of the order $\mathcal{O}\left(\frac{n}{\varepsilon}\right)$; see Lemma 4 in Appendix D.4.

- 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.



(a) Runtime

(b) Number of improving steps

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

(d) Percentage of improving solution compare 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 compare to the improvement of LSA

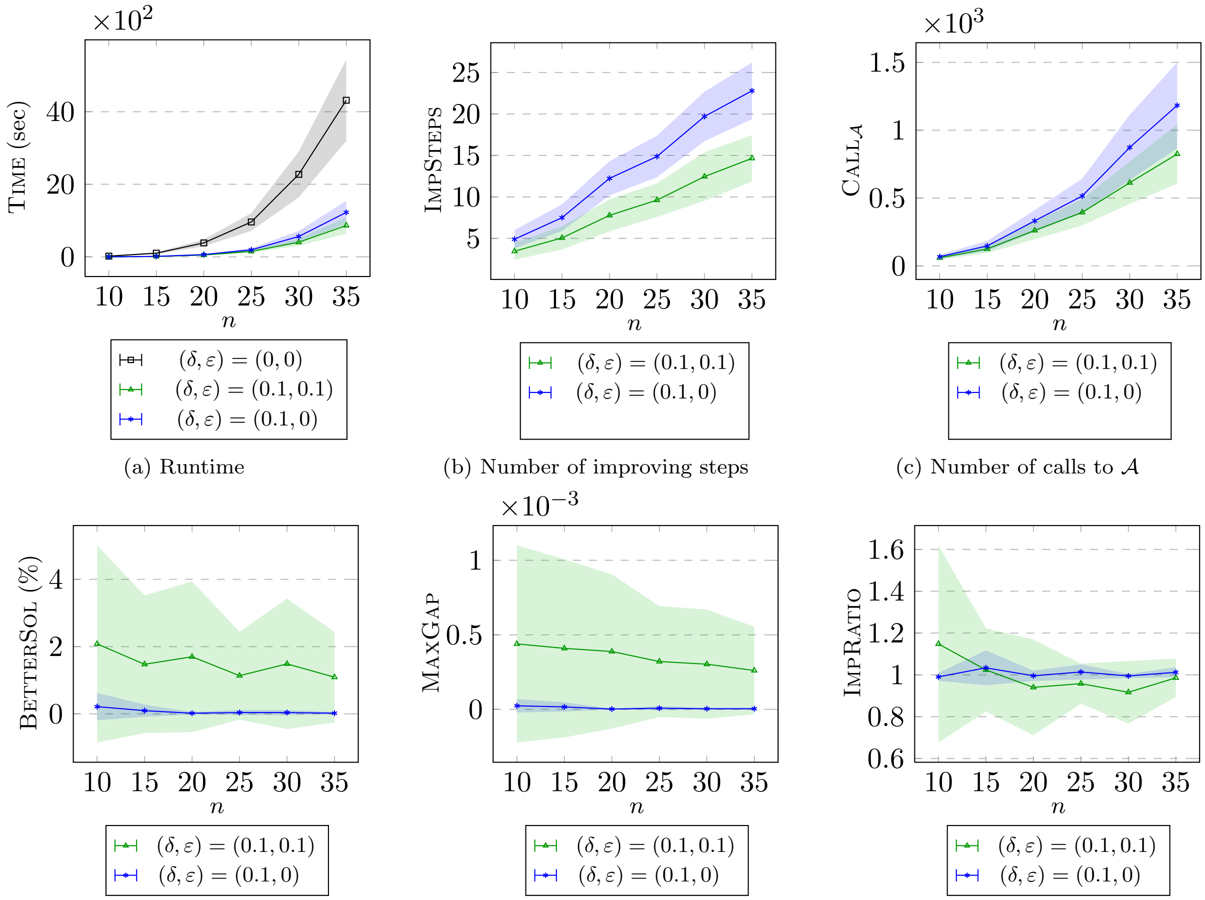
Figure 4: **Binary follower - exact follower** ($\delta = 0$) **- 2-flip neighborhood** ($k = 2$)**:** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.2. Recall that $\varepsilon = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon,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

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 $n \in \{10, 15, \ldots, 35\}$. 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 (Bernhard and Vygen 2008).

We consider four pairs of parameter $(\delta, \varepsilon) \in \{(0, 0), (0.1, 0), (0, 0.1), (0.1, 0.1)\}$. 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



(a) Runtime

(b) Number of improving steps

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

(d) Percentage of improving solution compare 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` compare 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.

in Figure 4. The results on both the running time (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.

• Recall that the leader's decision obtained by $(\varepsilon, \mathcal{A})$-`LSA` is not guaranteed to be locally optimal. We observe this phenomena 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 1. 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 behaviour, which is consistent with Theorem 2.

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

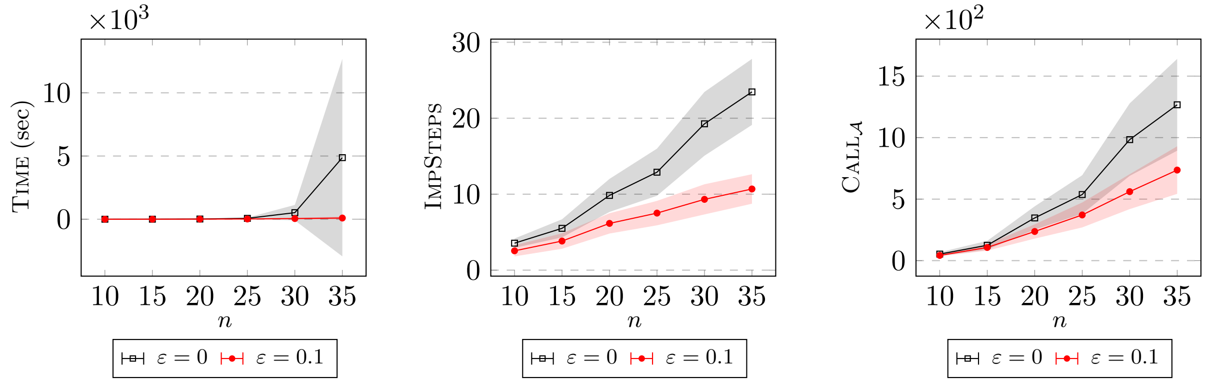
6.2.3 Mixed-integer lower level

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 $\delta \in \{0, 0.005\}$. 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 behaviour 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 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 1. 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.

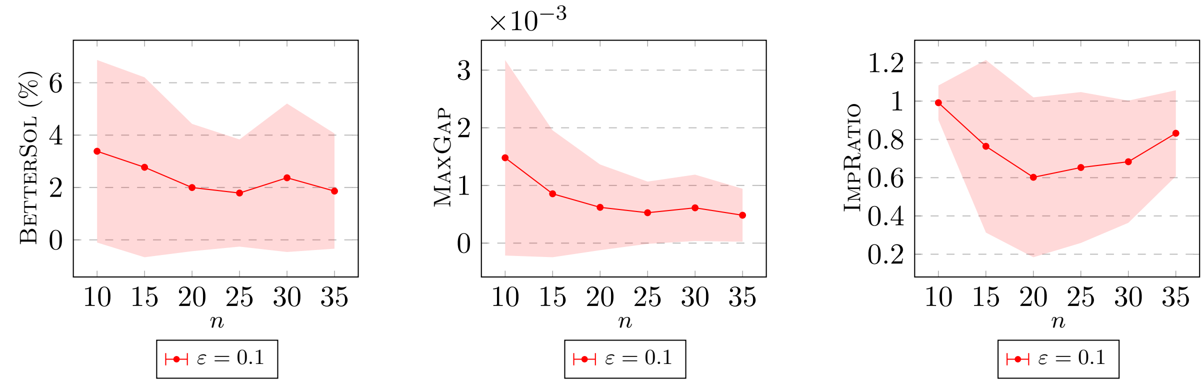
• Recall from Section 6.2.2 that the reduction in the runtime is mostly driven by solving the



(a) Runtime

(b) Number of improving steps

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



(d) Percentage of improving solution compare to the size of the neighborhood

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

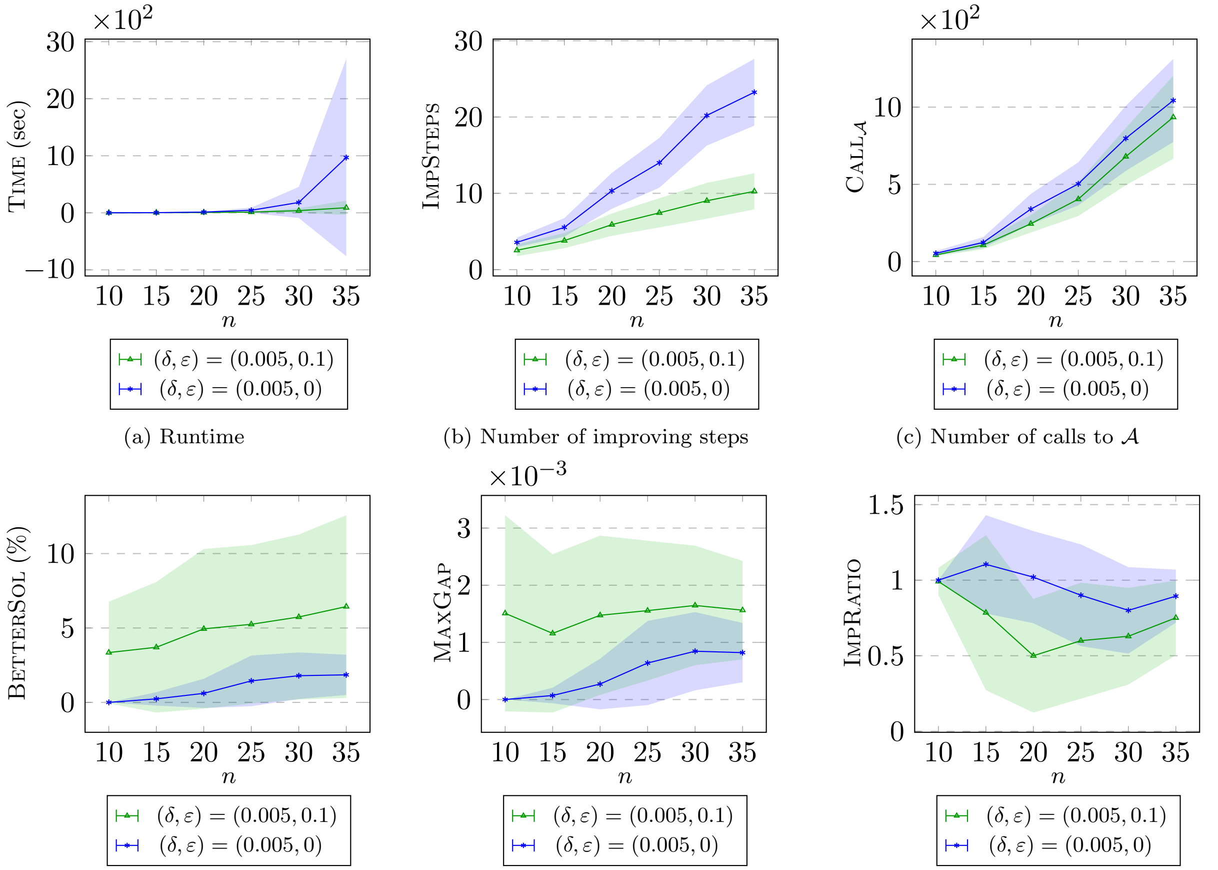
(f) Ratio of improvement of the $(\varepsilon, \mathcal{A})$-LSA compare to the improvement of LSA

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 $\pm$ MAD.

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 obervations 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 running time results in Figure 6a and Figure 7a exhibit more noisy behaviour 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.



(a) Runtime

(b) Number of improving steps

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

(d) Percentage of improving solution compare 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` compare 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.

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 (9) 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 (Erdos, Rényi, et al. 1960), 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 $h = 0.1n$.

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, where both scaling ($\varepsilon = 0.1$) and solving the follower's problem approximately ($\delta = 0.1$) are applied. The isolated effects of either scaling, i.e., $(\delta, \varepsilon) = (0, 0.1)$, or solving the follower's problem approximately, i.e., $(\delta, \varepsilon) = (0.1, 0)$, are relegated to Appendix E.2.

		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 (9) 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	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 (9) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

Comparing Tables 2 and 3, we observe a decrease in the runtime across all instances when 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 1. 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 (Garey and Johnson 1979), 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 (Håstad 1999). 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: scaling the leader's objective function in an adaptive manner, and evaluating the leader's objective function via an inexact follower's response. Specifically, the latter idea implies that we consider approximate solutions to the lower-level problem rather than the follower's fully rational response. Our approach integrates these two ideas within a local search-based algorithm. We observe that applying independently either scaling or an inexact follower's response is generally insufficient; both techniques are essential to ensure efficiency. Moreover, our computational study demonstrates that there is a trade-off between the runtime gains achieved by using $(\varepsilon, \mathcal{A})$-`LSA` instead of LSA, 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 (Thürauf et al. 2024). The results of these supplementary experiments are provided in Appendix E.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.

For practitioners, adopting relaxed solution concepts as proposed here provides guarantees on solution quality with controllable runtime for otherwise intractable bilevel problems. For the research community, 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.

While our study focuses on follower's problems with linear objective functions, many real-world applications involve nonlinear lower-level objective functions, presenting further opportunities for extension. Incorporating continuous upper-level variables and fully addressing coupling constraints also remain important directions for future research. Finally, although our numerical experiments suggest that lower-level computational complexity impacts the performance of our approach, further empirical and theoretical analysis is needed to better understand this phenomena.

References

Aarts, E. and Lenstra, J. K. (2003). *Local Search in Combinatorial Optimization.* Princeton: Princeton University Press.

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

Altner, D. S., Ahuja, R. K., Ergun, Ö., and Orlin, J. B. (2013). "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, pp. 339–367.

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

Arslan, O., Jabali, O., and Laporte, G. (2018). "Exact solution of the evasive flow capturing problem". In: *Operations Research* 66.6, pp. 1625–1640.

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

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

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

Bernhard, K. and Vygen, J. (2008). *Combinatorial optimization: Theory and algorithms.* Springer.

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

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

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

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

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

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

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

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

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

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

Dantzig, G. B. (1957). “Discrete-variable extremum problems”. In: *Operations Research* 5.2, pp. 266–288.

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

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

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

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

Domínguez, L. F. and Pistikopoulos, E. N. (2010). “Multiparametric programming based algorithms for pure integer and mixed-integer bilevel programming problems”. In: *Computers & Chemical Engineering* 34.12, pp. 2097–2106.

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

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

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

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

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

Furini, F., Ljubić, I., Malaguti, E., and Paronuzzi, P. (2020). "On integer and bilevel formulations for the k-vertex cut problem". In: *Mathematical Programming Computation* 12, pp. 133–164.

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

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

Gurobi (2024). *Gurobi Optimization, LLC*. `https://www.gurobi.com`. Accessed: 2024-07-26.

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

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

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

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

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

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

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

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

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

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

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

— (2023). "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, p. 3.

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

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

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

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

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

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

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

Mahdavi Pajouh, F., Boginski, V., and Pasiliao, E. L. (2014). "Minimum vertex blocker clique problem". In: *Networks* 64.1, pp. 48–64.

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

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

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

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

Prokopyev, O. A. and Ralphs, T. K. (2024). "On the Complexity of Finding Locally Optimal Solutions in Bilevel Linear Optimization". In.

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

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

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

Smith, J. C., Prince, M., and Geunes, J. (2013). "Modern network interdiction problems and algorithms". In: *Handbook of Combinatorial Optimization*. Ed. by Pardalos, P. M., Du, D.-Z., and Graham, R. L. New York, NY: Springer New York, pp. 1949–1987.

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

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

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

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

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

Thürauf, J., Kleinert, T., Ljubić, I., Ralphs, T., and Schmidt, M. (2024). *BOBILib: Bilevel Optimization (Benchmark) Instance Library*. URL: https://optimization-online.org/?p=27063.

Vazirani, V. V. (2001). *Approximation algorithms*. Springer Berlin, Heidelberg.

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

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

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

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

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

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

Zenklusen, R. (2010). “Matching interdiction”. In: *Discrete Applied Mathematics* 158.15, pp. 1676–1690.

A Justification of the technical assumptions for Section 1

Recall from Section 1 that the following technical 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 $\|\mathbf{y}\|_1 \leq U$ for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and for all $\mathbf{x} \in \mathcal{X}$.
- **A3:** $\mathbf{a} \in \mathbb{R}^n_+$ and $\mathbf{d} \in \mathbb{R}^m_+$.
- **A4:** $z^* > 0$ and $\mathbf{c}^\top \mathbf{y} \geq 0$ for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and for all $\mathbf{x} \in \mathcal{X}$.

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 (Vicente et al. 1996). Assumption **A1** is standard in bilevel optimization (see, e.g., Tavaslıoğlu et al. (2019) and Yang et al. (2023)), 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 bilevel programming with unbounded follower's feasible sets we refer to the survey by Kleinert et al. (2021).

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 $d_i < 0$ for some $i \in [m]$, then variables y_i and $y^*_i(\mathbf{x})$ are replaced by $1 - y_i$ and $1 - y^*_i(\mathbf{x})$, 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 $\tilde{y}_i$ defined as $\tilde{y}_i = U - y_i$, 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 the leader's objective function, as stated in Assumption **A3**, ensures that the leader's optimal objective function value of [**BP**] is non-negative, i.e., $z^* \geq 0$. If $z^* = 0$, then the optimal solution to [**BP**] must be $(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) = (\mathbf{0}, \mathbf{0})$. Therefore, one can simply assess whether $z^* = 0$ is the optimal objective function value of [**BP**] by verifying that $\mathbf{x} = \mathbf{0}$ belongs to $\mathcal{X}$, and that the corresponding follower's rational response satisfies $y^*_i(\mathbf{x}) = 0$ for all i in $[m]$.

Finally, the second part of Assumption **A4** can be ensure 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 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, such as interdiction problems, naturally use cost vectors with non-negative entries; see, for example, Weninger and Fukasawa (2025). 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 (Vazirani 2001).

Both Assumptions **A4** and **A5** 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 $c^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \geq (1-\delta)c^\top \mathbf{y}^*(\mathbf{x}) + \delta M$, which is aligned with standard definition in the literature (Vavasis 1993). 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 IMPROVE.

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.

B Algorithms and discussions for Section 3

In this section, we divide the discussion into three parts. First, in Section B.1, we describe the weak local search algorithm and illustrate it through the example from Section 3.3. Then, by constructing a particular class of bilevel MILPs, we show that both standard and weak local search require an exponential number of improving steps to converge in the worst case. Second, in Section B.2, 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. Lastly, in Section B.3, we describe a greedy heuristic for the maximum weighted clique problem, which is used in the example discussed in Section 3.3.

B.1 On (weak) local search and its complexity

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 Improve (see Algorithm 1 and our discussion in Section 5.1) 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)
    k ← 0 and x^k ← x^0
    ("answer", x^{k+1}) ← IMPROVE(x^k, N_X, a, d, A, 0)
    while "answer" is "Yes" do
        ("answer", x^{k+1}) ← IMPROVE(x^k, N_X, a, d, A, 0)
        Set x^k := x^{k+1}
        Set k := k + 1
    Return x^k
```

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 unique 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 Improve. That is, $\gamma = 0$; recall Algorithm 1 .

Clearly, $\mathcal{A}$-`LSA` returns a weak local optimal solution within a finite number of calls to Improve, but does not necessarily return a local optimal solution. Without any surprise, 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 bellow.

Example 1 (continued) We consider the example from Section 3.3. Specifically, we fix γ to 0, and the initial leader's feasible decision $\mathbf{x}^0$ to $\mathbf{0}$, i.e., the decision that consists of not interdicting any vertex in the graph G. Also, Improve is chosen such that the first improving solution found during the neighborhood search is selected and returned. Therefore, the maximum weighted clique in the graph G is given by $\mathcal{C} = \{1, 2, 3, 4\}$ and $\omega(\mathcal{C}) = 300$.

Exact follower. $\mathcal{A}$-`LSA` starts by searching for an improving solution in the neighborhood of $\mathbf{x}^0$. When calling Improve on $\mathbf{x}^0$, we obtain $\big(\text{“Yes”}, \mathbf{x}^1\big)$, where $\mathbf{x}^1 \in N_{\mathcal{X}}(\mathbf{x}^0)$ consists to interdict vertex 1, that is $x_1^1 = 1$ and $x_i^1 = 0$ otherwise. The obtained clique $\mathcal{C}^*$ has value $\omega(\mathcal{C}^*) = 220 < 299$. Next, $\mathbf{x}^0$ is replaced by $\mathbf{x}^1$. Then, Improve is called on $\mathbf{x}^1$ to obtained $\big(\text{“No”}, \mathbf{x}^1\big)$. Therefore, the algorithm ends and returns $\mathbf{x}^* = \mathbf{x}^1$.

Inexact follower. If the maximum clique problem is solved using $\mathcal{A}$, as described in Algorithm 4, then the algorithm returns the decision $\mathbf{x}^{\mathcal{A}}$ that consists of interdicting vertex 9. ■

Computational complexity of local search. We demonstrate that any *quadratic binary program* (QBP) can be reformulated as a bilevel MILP. Specifically, we show that 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 (Papp 2016), we show that for bilevel MILPs, both the standard and weak local search methods require 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:

$$[\mathbf{QBP}]: \quad z^*_{QP} := \min_{\mathbf{x} \in \{0,1\}^n} \ \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{b}^\top \mathbf{x}, \tag{12}$$

where $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ 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 (Glover 1975; McCormick 1976). 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:

$$[\mathbf{QP\text{-}BP}]: \quad z^*_{BP} := \min_{\mathbf{x},\mathbf{y}^*(\mathbf{x})} \ \mathbf{b}^\top \mathbf{x} + \sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij} y^*_{ij}(\mathbf{x}) \tag{13a}$$

$$\text{s.t. } \mathbf{x} \in \{0,1\}^n\,, \ \mathbf{y}^*(\mathbf{x}) \in \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}^{BP}(\mathbf{x})} \sum_{i=1}^{n}\sum_{j=1}^{n} y_{ij}, \tag{13b}$$

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

$$\mathcal{Y}^{BP}(\mathbf{x}) := \left\{\mathbf{y} \in [0,1]^{n\times n} : \ y_{ij} \le x_i\,, \ y_{ij} \le x_j \quad \forall\, i,j \in [n]\right\}, \tag{14}$$

as the follower's feasible set in (13b). 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 (14), the decision variables are all binary, i.e., $\mathbf{y} \in \{0,1\}^{n\times n}$, or any mix of binary and continuous variables.

To stay concise, we omit the definitions of the neighborhood and the improving solution for quadratic binary programs. Some of the concepts outlined in Section 3, including local optimality, can also be defined in the context of quadratic binary programming (Papp 2016).

Next, we demonstrate 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. Formally:

Lemma 2. *For any* $\mathbf{x} \in \{0,1\}^n$*,* $\mathbf{x}$ *is a feasible decision to* (12) *if and only if* $(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ *is a feasible decision to* (13)*, where* $\mathbf{y}^*(\mathbf{x})$ *is the optimal solution to the lower-level problem* (13b)*. In addition, any feasible decision* $\tilde{\mathbf{x}}$ *which is in the neighborhood of* $\mathbf{x}$*, i.e.,* $\tilde{\mathbf{x}} \in N_{\{0,1\}^n}(\mathbf{x})$*, is an improving solution to* (12) *if and only if it is an improving solution for* (13)*.*

Lemma 2 implies that local search in bilevel MILPs is at least as challenging as in quadratic binary programming. Additionally, it establishes that a solution $\mathbf{x}$ is locally optimal for [**QBP**] if and only if it is locally optimal for [**QP-BP**]. The proof of Lemma 2 is straightforward and hence, it is omitted for brevity. Next, we demonstrate that in the worst case, local search may require an exponential number of improving steps for the leader. Formally:

Proposition 4. *Given a fixed positive integer* $k \ge 1$*, the standard local search with respect to the k-flip neighborhood function returns a local optimal solution to* (13) *in an exponential number of improving steps for the leader.*

Proof. Using Theorem 2 by Papp (2016), there exists a class of [**QBP**], where local search requires an exponential number of improving steps to converge to a local minimum using the k-flip neighborhood function (recall Definition 4). Then, this class can be reduced to a bilevel program by applying Lemma 2. Additionally, we have that each improving solution to the former problem is an improving solution to the latter problem, and vice versa.

However, in the bilevel MILPs formulation (13b), some elements of the matrix $\mathbf{Q}$ may be negative. Refer to the discussion following Assumption **A3** for the details on handling negative coefficients in the leader's objective function. By applying Lemma 2, we can adapt the proof of Theorem 2 by Papp (2016) to the bilevel context, thereby obtaining the desired result. ■

As a direct consequence of Proposition 4, we make several important observations:

• Recall that the follower's feasible set is given by (14) i.e., the follower's decision variables are all continuous. As a result, applying standard local search to [**C-BP**] may require an exponential number of improving steps for the leader in the worst case.

• We mentioned earlier that the follower's optimal decision variables in [**QP-BP**] are always binary. Therefore, the follower's feasible set $\mathcal{Y}^{BP}(\mathbf{x})$ in (14) can also be represented as a discrete set. As a result, applying standard local search to [**B-BP**] requires an exponential number of improving step to converge in the worst case.

• Consider a simple heuristic that takes as an input a linear 0-1 program, solves its LP relaxation, and then returns a feasible decision by rounding. In general, this procedure does not necessarily return an optimal decision (or even feasible); however, it returns the optimal solution to the considered follower's problem in [**QP-BP**]. Hence, $\mathcal{A}$-`LSA` requires an exponential number of improving steps in the worst case, whenever $\mathcal{A}$ is a naive LP-based heuristic method.

B.2 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 (Garey and Johnson 1979), to verifying whether a given leader's feasible decision is ε-local 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 5. *Given $\varepsilon \geq 0$, there exist both an instance of* [**B-BP**] *and a leader's feasible decision $\mathbf{x} \in \mathcal{X}$ such that the answer to* **SSP** *is "Yes" if and only if $\mathbf{x}^{\varepsilon}$ is ε-locally optimal.*

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

$$\min_{\mathbf{x},\mathbf{y}} \quad \sum_{i=1}^{n} w_i y_i + x_1 + \frac{W+\frac{1}{2}}{1+\varepsilon} x_2 \tag{15a}$$

$$\text{s.t.} \quad \mathbf{x} \in \{0,1\}^2, \tag{15b}$$

$$\mathbf{y} \in \arg\max_{\tilde{y}} \quad -\sum_{i=1}^{n} w_i \tilde{y}_i \tag{15c}$$

$$W - \frac{1}{2}x_1 - W x_2 \leq \sum_{i=1}^{n} w_i \tilde{y}_i \leq \sum_{i=1}^{n} w_i, \tag{15d}$$

$$\tilde{\mathbf{y}} \in \{0,1\}^n. \tag{15e}$$

Solving **SSP** can be reduced to verifying that $(x_1, x_2) = (0,0)$ is ε-local optimal for (15). We first introduce $f\,(x_1, x_2,$ *"answer"*$)$, a function that returns the leader's objective function value of (15) 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 4.

"*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 4: Summary of the leader's objective function values of (15), denoted by f, for different pairs (x_1, x_2) of leader's feasible decisions and the answer to **SSP**.

Next, we show that if *"answer"* is "Yes," then $(x_1, x_2) = (0,0)$ is ε-locally optimal. Indeed, from Table 4, 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.

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**. ■

B.3 Greedy heuristic for the maximum weighted clique problem

To ensure a comprehensive discussion, we include algorithm $\mathcal{A}$, which is use to find a weak local optimal solution for the example in Section 3.3. This method, see Algorithm 4, is a simple greedy heuristic (Wu and Hao 2015).

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

Require: Graph $G = (V, E)$

```
function A(V, E)
    Initialize C ← ∅
    while V ≠ ∅ do
        Select vertex v ∈ V with maximum degree (break ties by highest weight)
        Add v to C: C ← C ∪ {v}
        Update V: V ← {u ∈ V \ {v} | u is adjacent to all vertices in C}
    return 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.

C Proofs for Section 4

In this section, we provide the proofs of our results from Section 4. Specifically, the proof of Proposition 1 is in Section C.1, and the proof of Theorem 1 is in Section C.2. Next, we present the details of the empirical evaluation of the maximum gap associated with Theorem 1 in Section C.3.

C.1 Proof of Proposition 1

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

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right). \tag{16}$$

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, it follows that:

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

where the first inequality holds by (6) 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:

$$\begin{aligned}
(1-\delta)\left(\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}})\right) &= (1-\delta)\left(\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \alpha\mathbf{c}^\top \mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}})\right) \\
&\leq (1-\delta)\,\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \alpha\mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) && \text{by the definition of } \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \\
&\leq \mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) && \text{since } 0 \leq \delta < 1 \text{ and } \mathbf{d} = \alpha\mathbf{c} \\
&\leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right) && \text{by (16)} \\
&= (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \alpha\mathbf{c}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right) \\
&\leq (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \alpha\mathbf{c}^\top \mathbf{y}^*(\mathbf{x})\right) && \text{by the definition of } \mathbf{y}^*(\mathbf{x}) \\
&= (1+\varepsilon)\left(\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^*(\mathbf{x})\right).
\end{aligned}$$

Therefore, by observing that $\frac{1+\varepsilon}{1-\delta} = 1 + \frac{\varepsilon+\delta}{1-\delta}$, it follows that:

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

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

C.2 Proof of Theorem 1

First, for simplicity of our discussion, we assume that the follower's rational response is unique for any given leader's feasible decision. 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.

Next, assume that $\mathcal{A}$ always 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. This assertion is formalized in the following technical lemma.

Lemma 3. *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.,*

$$\left\|\mathbf{y}^{\mathcal{A}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\right\|_1 \leq r, \tag{18}$$

for any leader's feasible decision $\mathbf{x} \in \mathcal{X}$, where $r > 0$ does not depend on $\mathbf{x}$. Then, given $\varepsilon \geq 0$ and the lower bound $\underline{z} > 0$ to [**BP**]*, there exists $\Pi > 0$, defined as:*

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

such that 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 Π-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 linear 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 the linear mapping, together with (18), we have that for any $\mathbf{x} \in \mathcal{X}$:

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

which implies that:

$$-r d_{\max} \leq \mathbf{d}^\top \mathbf{y}^*(\mathbf{x}) - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) \leq r d_{\max}. \tag{19}$$

Given that (19) holds for any leader's feasible decision $\mathbf{x}$, we conclude that:

$$\mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}) \le \mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^*(\mathbf{x}) + rd_{\max}, \tag{20}$$

and, in particular, (19) holds for $\mathbf{x}^{\varepsilon,\mathcal{A}}$. That is:

$$\mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top\mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) \le \mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) + rd_{\max}. \tag{21}$$

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):

$$\mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \le (1+\varepsilon)\left(\mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x})\right). \tag{22}$$

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

$$\begin{aligned}
\mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top\mathbf{y}^*(\mathbf{x}^{\varepsilon,\mathcal{A}}) &\le \mathbf{a}^\top\mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) + rd_{\max} && \text{by (21)}\\
&\le (1+\varepsilon)\left(\mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^{\mathcal{A}}(\mathbf{x})\right) + rd_{\max} && \text{by (22)}\\
&\le (1+\varepsilon)\left(\mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^*(\mathbf{x}) + rd_{max}\right) + rd_{\max} && \text{by (20)}\\
&\le (1+\varepsilon)\left(\mathbf{a}^\top\mathbf{x} + \mathbf{d}^\top\mathbf{y}^*(\mathbf{x})\right) + (2+\varepsilon)rd_{\max}.
\end{aligned}$$

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

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

Hence,

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

which implies that:

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

If we define:

$$\Pi := \varepsilon + \frac{(2+\varepsilon)\,rd_{\max}}{\underline{z}},$$

then we obtain the desired result. That is, $\mathbf{x}^{\varepsilon,\mathcal{A}}$ is Π-locally optimal with respect to $N_{\mathcal{X}}$. ■

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 $\mathbf{b}\,(\delta) \in \mathbb{R}^{q+1}$ as follows:

$$\mathbf{b}\,(\delta) := \begin{pmatrix} \mathbf{f} - \mathbf{L}\mathbf{x} \\ -\,(1-\delta)\,\varphi(\mathbf{x}) \end{pmatrix},$$

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

$$\tilde{\mathbf{F}}_\mathbf{1} := \begin{pmatrix} \mathbf{F}_1 \\ -\mathbf{c}_1 \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{F}}_\mathbf{2} := \begin{pmatrix} \mathbf{F}_2 \\ -\mathbf{c}_2 \end{pmatrix}.$$

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 feasibility problem:

$$\begin{aligned} \min_{\mathbf{y}_1,\mathbf{y}_2,\mathbf{s}} \quad & 0 \\ \text{s.t.} \quad & \tilde{\mathbf{F}}_1\mathbf{y}_1 + \tilde{\mathbf{F}}_2\mathbf{y}_2 + \mathbf{s} = \mathbf{b}\,(0)\,, \\ & \mathbf{y}_1 \in \{0,1\}^{m_1}\,,\ \mathbf{y}_2 \in \mathbb{R}_+^{m_2},\ \mathbf{s} \in \mathbb{R}_+^{q+1}, \end{aligned} \tag{24}$$

where variables $\mathbf{s}$ (referred to as "surplus" 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 $\mathbf{s}^* \in \mathbb{R}_+^{q+1}$ such that $(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x}))$ is the optimal solution of (24).

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

$$\begin{aligned}
\min_{\mathbf{y}_1,\mathbf{y}_2,\mathbf{s}} \quad & 0 \\
\text{s.t.} \quad & \tilde{\mathbf{F}}_1\mathbf{y}_1 + \tilde{\mathbf{F}}_2\mathbf{y}_2 + \mathbf{s} = \mathbf{b}\left(\delta\right) \\
& \mathbf{y}_1 \in \{0,1\}^{m_1}, \ \mathbf{y}_2 \in \mathbb{R}_+^{m_2}, \ \mathbf{s} \in \mathbb{R}_+^{q+1},
\end{aligned} \tag{25}$$

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 surplus 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 corresponding optimal decisions of (25). We rely on a result by Mangasarian and Shiau (1987), which demonstrates that, if (25) 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 (25) is Lipschitz continuous. Formally:

Theorem 5 (Theorem 2.4 by Mangasarian and Shiau (1987))**.** *Assume that* $\tilde{\mathbf{F}}_\mathbf{1} = \mathbf{0}$ *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* (25)*. Then, there exists* $\left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top$*, an optimal solution of* (24)*, and a positive constant* $\mu_1\left(\tilde{\mathbf{F}}_\mathbf{2}\right) > 0$ *that only depends on* $\tilde{\mathbf{F}}_2$ *such that:*

$$\left\|\left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top\right\|_\infty \leq \mu_1\left(\tilde{\mathbf{F}}_\mathbf{2}\right)\left\|\mathbf{b}(0) - \mathbf{b}(\delta)\right\|_1.$$

A direct consequence of Theorem 5 is that whenever $\tilde{\mathbf{F}}_1 = 0$, we have:

$$\begin{aligned}
\left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 &\leq \left\|\left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top\right\|_1 \\
&\leq (m+q+1)\left\|\left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top\right\|_\infty \\
&\leq \mu_1\left(\tilde{\mathbf{F}}_\mathbf{2}\right) \cdot (m+q+1)\,\delta\varphi(\mathbf{x}),
\end{aligned} \tag{26}$$

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

Interestingly, a similar result has been presented earlier by Blair and Jeroslow (1977), where the decision variables are mixed-integer or pure integer. To leverage the result by Blair and Jeroslow (1977), the parameters in (25) 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 surplus vector $\mathbf{s}$ as described earlier can also be assumed to have integer components.

Theorem 6 (Theorem 2.1 from Blair and Jeroslow (1977)). *There exists two positive constants $\mu_2, \mu_3 > 0$ that does not depends on the r.h.s of* (25) *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* (25)*, then there exists an optimal solution $\left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top$ of* (24) *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 constant μ_2 and μ_3 in Theorem 6 only depends 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**. Therefore, together with $\left\|\mathbf{b}(0) - \mathbf{b}(\delta)\right\|_1 = \delta\varphi(\mathbf{x})$ and by the inequality in (26), we have that $\varphi(\mathbf{x}) \leq c_{max} mU$. Next, we define:

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

As a direct consequence of Theorems 5 and 6, 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:

$$\left\|\mathbf{y}^*(\mathbf{x}) - \mathbf{y}^{\mathcal{A}}(\mathbf{x})\right\|_1 \leq \left\| \left(\mathbf{y}_1^{\mathcal{A}}(\mathbf{x}), \mathbf{y}_2^{\mathcal{A}}(\mathbf{x}), \mathbf{s}^{\mathcal{A}}(\mathbf{x})\right)^\top - \left(\mathbf{y}_1^*(\mathbf{x}), \mathbf{y}_2^*(\mathbf{x}), \mathbf{s}^*(\mathbf{x})\right)^\top \right\|_1 \leq \gamma_1\delta + \gamma_2, \tag{27}$$

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 $\mathbf{x}$.

Remark 2. Our definition of δ-approximation is not restricted to the follower's decisions containing solely binary variables. Indeed, there exist approaches, such as first-order methods, that are used to solve large-scale LPs with a given approximation gap (Applegate et al. 2023). Hence, the discussion before applies also whenever all the follower's decision variables are all continuous. ■

From (27), we conclude that r in Lemma 3 can be set to $r = \gamma_1\delta + \gamma_2$. Consequently, Theorem 1 follows from Lemma 3. 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 5 and 6 only provide the existence of a follower's rational response to problem (24). Indeed, without the uniqueness assumption, the follower's decisions obtained in Theorems 5 and 6 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 (24) and (25) 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 $\mathbf{c}_2$). Then, however, Theorem 6 requires that if the right-hand sides equals zero in problems (24) and (25), then the corresponding optimal objective function values must also be equal to zero (Blair and Jeroslow 1977). Hence, to ensure that this additional requirement holds, we need to introduce another variable and an extra constraint into problems (24) and (25), similar to the approach, which is used to justify Assumption **A4** in Appendix A.

C.3 Details for the empirical performance evaluation of 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 $p = 0.5$ (Erdos, Rényi, et al. 1960). 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 $\mathbf{x} = \mathbf{0}$. The follower's problem is solved using the MILP solver by (Gurobi 2024) 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.

D Proofs and additional discussion for Section 5

In Appendix D.1, we provide an example for $(\varepsilon, \mathcal{A})$-$\texttt{LSA}$ applied to the problem from Section 3.3. Then, in Appendix D.2, we provide the proofs for all the results from Section 5.2. In Section D.3, we present an instance of bilevel MILPs, where $(\varepsilon, \mathcal{A})$-$\texttt{LSA}$ is asymptotically sharp. In Section D.4, we extend the discussion from Section D.2 to a follower with mixed-integer decision variables. Finally, in Section D.5, we discuss relevant extensions to our approach.

D.1 Example of $(\varepsilon, \mathcal{A})$-$\texttt{LSA}$

We consider the example from Section 3.3. We explore two variants of $(\varepsilon, \mathcal{A})$-$\texttt{LSA}$: one, where the maximum weighted clique problem is solved exactly, and the other, where it is solved using the greedy search algorithm $\mathcal{A}$, described in Algorithm 4. For both variants, we fix $\varepsilon = 1$ and $\gamma = 0$.

Exact follower. The maximum weighted clique is $\mathcal{C} = \{1, 2, 3, 4\}$ with $\omega(\mathcal{C}) = 300$. Thus, $q_d = \frac{\omega(\mathcal{C})\varepsilon}{4(m+1)(1+\varepsilon)} = \frac{300}{104}$. The original weight vector is $\mathbf{w} := (80, 79, 71, 70, 10, 10, 10, 10, 90, 20, 20, 89)^\top$. After scaling in line 7 of Algorithm 2 (see Section 5.1) and applying the ceiling function to the scaled vector, the vector becomes $\mathbf{w}' := q_d\,(28, 28, 25, 25, 4, 4, 4, 4, 32, 7, 7, 31)^\top$.

The loop at line 10 begins with $\mathbf{x}^0$, where no vertex is interdicted. Then, $\mathbf{x}^1$, which interdicts only vertex 1, is an improving solution in terms of the leader's objective function. The loop continues since the leader's objective function has not been halved. Applying Improve to $\mathbf{x}^1$ returns the answer $\left(\text{“No”}, \mathbf{x}^1\right)$, indicating that $\mathbf{x}^1$ is locally optimal with respect to the scaled leader's objective. The algorithm stops and returns $\mathbf{x}^\varepsilon = \mathbf{x}^1$.

Inexact follower. The clique obtained by calling $\mathcal{A}$ is $\widetilde{\mathcal{C}} = \{9, 10, 11, 12\}$ with $\omega(\widetilde{\mathcal{C}}) = 219$. Thus, $q_d = \frac{\omega(\tilde{\mathcal{C}})\varepsilon}{4(m+1)(1+\varepsilon)} = \frac{219}{104}$. After applying the scaling step of $(\varepsilon, \mathcal{A})$-$\texttt{LSA}$ in line 7, the vector becomes $\mathbf{w}' := q_d\,(38, 38, 34, 34, 5, 5, 5, 5, 43, 10, 10, 43)^\top$. The loop at line 10 begins with $\mathbf{x}^0$, where no vertex is interdicted. Then, $\mathbf{x}^{12}$, which interdicts only vertex 12, is an improving solution. Calling Improve to $\mathbf{x}^{12}$ returns the answer $\left(\text{“No”}, \mathbf{x}^{12}\right)$. The algorithm stops and returns $\mathbf{x}^{\varepsilon,\mathcal{A}} = \mathbf{x}^{12}$.

D.2 Proofs for Section 5.2

Proof of Lemma 1. Given some $\mathbf{x}^{i,k} \in \mathcal{X}$, we assume that the answer returned by Improve is $\left(\text{“Yes”}, \mathbf{x}^{i,k+1}\right)$. Hence, $\mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})$ is an arbitrary neighbor of $\mathbf{x}^{i,k}$ that is an improving solution in terms of the leader's objective function with scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$.

Recall line 6 of Algorithm 2 (see Section 5.1) for the definition of q_a and q_d. We have that:

$$\begin{aligned}
\Delta^{\mathcal{A}}\left(\mathbf{x}^{i,k},\mathbf{x}^{i,k+1},\mathbf{a}',\mathbf{d}'\right) &= \mathbf{a}'^{\top}\mathbf{x}^{i,k} + \mathbf{d}'^{\top}\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k}) - \mathbf{a}'^{\top}\mathbf{x}^{i,k+1} - \mathbf{d}'^{\top}\mathbf{y}^{\mathcal{A}}(\mathbf{x}^{i,k+1}) \\
&= \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil \left(x_j^{i,k} - x_j^{i,k+1}\right) + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil \left(y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1})\right) \\
&= q_a \sum_{j=1}^{n} \left\lceil \frac{a_j}{q_a} \right\rceil \Delta_j^{\mathbf{x},i,k} + q_d \sum_{\ell=1}^{m} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k} > 0, \qquad (28a)
\end{aligned}$$

where the strict positive inequality in (28a) 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:

$$\Delta_j^{\mathbf{x},i,k} := \begin{cases} 1 & \text{if } x_j^{i,k} = 1 \text{ and } x_j^{i,k+1} = 0, \\ 0 & \text{if } x_j^{i,k} = x_j^{i,k+1}, \\ -1 & \text{if } x_j^{i,k} = 0 \text{ and } x_j^{i,k+1} = 1, \end{cases}$$

and,

$$\Delta_\ell^{\mathbf{y},i,k} := \begin{cases} 1 & \text{if } y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) = 1 \text{ and } y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}) = 0, \\ 0 & \text{if } y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) = y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}), \\ -1 & \text{if } y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) = 0 \text{ and } y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}) = 1. \end{cases}$$

It follows from (28a) 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, $k_d = \sum_{\ell=1}^{n} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k}$.

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:

$$\begin{aligned}
|k_a| = \left| \sum_{j=1}^{n} \left\lceil \frac{a_j}{q_a} \right\rceil \Delta_j^{\mathbf{x},i,k} \right| &\leq \sum_{j=1}^{n} \left(\frac{a_j}{q_a} + 1 \right) \qquad (29a) \\
&\leq n \left(\frac{a_{max}}{q_a} + 1 \right) = n \left(\frac{4a_{max} n(1+\varepsilon)}{K\varepsilon} + 1 \right) =: b_a, \qquad (29b)
\end{aligned}$$

where the equality in (29b) follows from the definition of q_a; see line 6 in Algorithm 2. In the same spirit, we can derive a similar bound for k_d, namely:

$$|k_d| = \left| \sum_{\ell=1}^{m} \left\lceil \frac{d_\ell}{q_d} \right\rceil \Delta_\ell^{\mathbf{y},i,k} \right| \leq \sum_{\ell=1}^{m} \left(\frac{d_\ell}{q_d} + 1 \right) \tag{30a}$$

$$\leq m \left(\frac{d_{max}}{q_d} + 1 \right) = m \left(\frac{4 d_{max}(m+1)(1+\varepsilon)}{K\varepsilon} + 1 \right) =: b_d, \tag{30b}$$

where the equality in (30b) 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:

$$\begin{aligned} \min_{k_a \in \mathbb{Z},\, k_d \in \mathbb{Z}} \quad & k_a q_a + k_d q_d \\ & k_a q_a + k_d q_d > 0, \\ & |k_a| \leq b_a, \\ & |k_d| \leq b_d, \end{aligned} \tag{31}$$

which is clearly feasible by simply using (28a). Since the feasible set of (31) is finite, then (31) has at least one optimal solution.

Let (k_a^*, k_d^*) be an optimal solution of problem (31). 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:

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

where we again apply the definitions of q_a and q_d.

Observe that $(0, 0)$ is not a feasible solution for (31). Also note that k_a^* and k_d^* cannot be simultaneously strictly negative. Hence, we need to consider only two cases, as outlined bellow.

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:

$$k_a^* q_a + k_d^* q_d \geq \frac{\varepsilon K}{4(1+\varepsilon)} \cdot \frac{1}{\max\{n, m+1\}}. \tag{32}$$

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 (31). 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:

$$k_a^* q_a + k_d^* q_d = \frac{\varepsilon K}{4(1+\varepsilon)(m+1)} \left(\frac{m+1}{n} k_a^* + \left\lceil \frac{m+1}{n} |k_a^*| \right\rceil \right). \tag{33}$$

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 (33), i.e., (k_a^*, k_d^*) is not a feasible solution of (31). 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 numerator of the rational number. ∎

Proof of Theorem 2. 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.

From Lemma 1, if "answer" is "Yes" in line 11 of Algorithm 2 when calling Improve, then the improving solution that is obtained by Improve 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 Improve between two consecutive iterations i and $i+1$ is:

$$\frac{1}{\Delta} \left(\mathbf{a}'^\top \mathbf{x}^i + \mathbf{d}'^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i) \right) = \frac{1}{\Delta} \left(\sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^i + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^{\mathcal{A}}(\mathbf{x}^i) \right) \tag{35a}$$

$$\leq \frac{1}{\Delta}\left(\sum_{j=1}^{n} q_a\left(\frac{a_j}{q_a}+1\right)x_j^i+\sum_{\ell=1}^{m} q_d\left(\frac{d_\ell}{q_d}+1\right)y_\ell^{\mathcal{A}}(\mathbf{x}^i)\right) \tag{35b}$$

$$\leq \frac{1}{\Delta}\left(\sum_{j=1}^{n}(a_j+q_a)\,x_j^i+\sum_{\ell=1}^{m}(d_\ell+q_d)\,y_\ell^{\mathcal{A}}(\mathbf{x}^i)\right) \tag{35c}$$

$$\leq \frac{1}{\Delta}\left(K+nq_a+(m+1)\,q_d\right), \tag{35d}$$

where (35a) comes from the definitions of the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. Then, (35b) comes from the property of the ceiling function. Also, the leader's and the follower's variables are all binary. Thus, (35d) holds. We use $m+1$ instead of m in the third term for convenience in the derivations bellow.

Additionally, by using the definitions of q_a and q_d as computed in the line 6 of Algorithm 2:

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

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 (35d), 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 10 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 $\mathcal{O}\left(\log K_0\right)$, 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 Improve can be estimated as:

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

In the worst case, each call to Improve, which has a running-time complexity $\mathcal{C}_I$, is followed by a call to $\mathcal{A}$, which has a running-time complexity $\mathcal{C}_{\mathcal{A}}$. Therefore, each call to Improve contributes to the running-time complexity by the order of $\mathcal{O}\left(\mathcal{C}_I+\mathcal{C}_{\mathcal{A}}\right)$, concluding the proof. ■

Proof of Theorem 3. By Theorem 2, $(\varepsilon,\mathcal{A})$-`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. Then:

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

where the transition from the second to the third line 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 15 of Algorithm 2. Then, it follows that:

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

where we simply exploit the properties of the ceiling function.

Furthermore, the subsequent inequality results directly from the stopping criteria that is used within the inner loop in line 10 of Algorithm 2:

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

which we exploit to obtain:

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

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

Proof of Proposition 2 . Fix $i \ge 0$. We show that if the condition (11) is satisfied, then each improving step within the inner loop with respect to the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$ corresponds to an improvement in terms of the original leader's objective function with an inexact follower.

Let $\mathbf{x}^{i,k}$ denote a leader's feasible solution obtained within the inner loop at line 10 of Algorithm 2. Moreover, assume there exists an improving solution $\mathbf{x}^{i,k+1} \in N_{\mathcal{X}}(\mathbf{x}^{i,k})$ in term of the leader's objective function with an inexact follower and with scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$, i.e., that "answer" obtained by IMPROVE is "Yes."

Then, the following sequence of inequalities and equalities holds:

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

where the first inequality holds since $\mathbf{x}^{i,k+1}$ is an improving solution, which reduces the leader's objective function with inexact follower and scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$ by at least Δ; recall Lemma 1. The subsequent inequality are obtained by the definitions of α and β from the discussion in Section 5.2 above Proposition 2. That is, for each $j \in [n]$, $p_j \in \mathbb{Z}_{\geq 0}$ and $\alpha_j \in [0, q_a)$ are defined such that $a_j = p_j q_a - \alpha_j$. Similarly, for each $\ell \in [m]$, $s_\ell \in \mathbb{Z}_{\geq 0}$ and $\beta_\ell \in [0, p_d)$ are defined such that $d_\ell = s_\ell q_d - \beta_\ell$.

As a consequence, a necessary condition for $\mathbf{x}^{i,k+1}$ to be an improving solution in terms of the leader's objective function with an inexact follower is given by:

$$0 < \Delta - \sum_{j=1}^{n} \alpha_j \left(x_j^{i,k} - x_j^{i,k+1} \right) - \sum_{\ell=1}^{m} \beta_\ell \left(y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k}) - y_\ell^{\mathcal{A}}(\mathbf{x}^{i,k+1}) \right),$$

which holds if the following inequality is satisfied:

$$\Delta_x^* \sum_{j=1}^{n} \alpha_j + \Delta_y^{\mathcal{A}} \sum_{\ell=1}^{m} \beta_\ell < \Delta.$$

As a consequence, if (11) holds, then every improving step in term of the scaled leader's objective function leads to an improving step in term of the original leader's objective function. That is, the algorithm returns a weak local optimal solution to [**B-BP**]. ■

Proof of Theorem 4. 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 10 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 Improve is "Yes." Then, the following holds:

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

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:

$$0 < \sum_{\ell=1}^{m} y_\ell^*(\mathbf{x}^{i,k}) - \sum_{\ell=1}^{m} y_\ell^*(\mathbf{x}^{i,k+1}), \tag{39}$$

which holds since $q_d \left\lceil \frac{\alpha}{q_d} \right\rceil$ does not depend on ℓ. Multiplying both sides of (39) 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 is true as well. That is, 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 $\mathbf{d}'$.

Moreover, the outlined proof is independent of the values that are taken by q_a and q_d. Therefore, the relation holds at each iteration i, which concludes the proof. ■

D.3 Sharpness of $(\varepsilon, \mathcal{A})$-`LSA`

In this section, we construct an instance of the knapsack interdiction problem where the bound $\varepsilon > 0$, derived for the solution obtained by $(\varepsilon, \mathcal{A})$-`LSA` in Theorem 3, is asymptotically achieved whenever the follower's problem is solved exactly. The standard linear 0-1 knapsack problem involves selecting a subset of items to maximize the total value. In contrast, the knapsack interdiction problem introduces a leader, who aims to interdict certain items, forcing the follower to solve the knapsack problem with the remaining items. For more details, we refer to Caprara et al. (2016).

Thus, the knapsack interdiction problem can be formulated as the following bilevel program:

$$\min_{\mathbf{x},\mathbf{y}^*(\mathbf{x})} \mathbf{c}^\top \mathbf{y}^*(\mathbf{x})$$
$$\text{s.t. } \mathbf{x} \in \mathcal{X},\ \mathbf{y}^*(\mathbf{x}) \in \operatorname{argmax}\left\{\mathbf{c}^\top \mathbf{y} \ : \ \mathbf{y} \in \mathcal{Y}(\mathbf{x})\right\},$$

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:

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

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, denoted as $N^{(2)}_{\mathcal{X}}$. 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, $q_d = \frac{K\varepsilon}{4n(1+\varepsilon)}$, and:

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

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

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

We enter the inner loop at line 10 of Algorithm 2 with starting feasible solution $\mathbf{x}^0$ and call Improve with the scaled vector $\mathbf{c}'$. Then, the 2-flip neighborhood of $\mathbf{x}^0$ is given by:

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

The answer returned by Improve is "No" 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:

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

which implies that the worst-case bound ε obtained by $(\varepsilon, \mathcal{A})$-`LSA` in Theorem 3 is in the order of ε (asymptotically), assuming $0 < \varepsilon < 1$. Thus, $(\varepsilon, \mathcal{A})$-`LSA` can be considered "sharp" in this sense.

D.4 Proofs and further discussion 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 $\|\mathbf{d}\|_1 > 0$. Indeed, if $\mathbf{d} = 0$, 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 Improve between two iterations i and $i+1$ in $(\varepsilon, \mathcal{A})$-`LSA`; see Lemma 4. We leverage this observation to derive the running-time 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 6.
- We provide a sufficient condition for $\mathbf{x}^{\varepsilon,\mathcal{A}}$ to be weakly local optimal; see Proposition 7.

Lemma 4. *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* Improve *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* Improve *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 5 of Algorithm 2. At each call of Improve, if "answer" is "Yes", then the improving solution, which is returned by the sub-procedure reduces the leader's objective function value with an inexact follower and with scaled vectors by at least γ.

Therefore, the maximum number of calls to Improve between iterations i and $i+1$ is given by:

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

where $\mathbf{a}'$ and $\mathbf{d}'$ are the scaled vectors obtained at line 7 of Algorithm 2. An upper bound for τ can be derived as follows:

$$\begin{aligned}
\tau &= \frac{\mathbf{a'}^\top \mathbf{x}^i + \mathbf{d'}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i)}{\gamma} && \text{(42a)}\\
&= \frac{\sum_{j=1}^n q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j^i + \sum_{\ell=1}^m q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil y_\ell^*(\mathbf{x}^i)}{\gamma} && \text{(42b)}\\
&\leq \frac{\mathbf{a}^\top \mathbf{x}^i + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^i) + nq_a + q_d\,(m+1)\,U}{\gamma} && \text{(42c)}\\
&= \left(K + \frac{K\varepsilon}{4\,(1+\varepsilon)} + \frac{K\varepsilon}{4\,(1+\varepsilon)}\right) \frac{mq_d + \sum_{\ell=1}^m d_\ell}{U q_d^2 \sum_{\ell=1}^m \left\lceil \frac{d_\ell}{q_d} \right\rceil}, && \text{(42d)}
\end{aligned}$$

where (42a) comes from the definition of the scaled vectors $\mathbf{a}'$ and $\mathbf{d}'$. Additionally, (42c) is obtained by the property of the ceiling function. Specifically,

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

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

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

and the other terms in (42c) can be derived in a similar manner. Finally, (42d) is obtained by the definitions of K, q_a, q_d, and γ; see lines 6 and 8 of Algorithm 2.

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

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

which is then exploited, along with (42), to derive the following sequence of inequalities:

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

$$\leq \frac{K}{Uq_d}\left(\frac{2+3\varepsilon}{2\left(1+\varepsilon\right)}\right)(1+1)$$
$$\leq \frac{K}{Uq_d}\left(\frac{2+3\varepsilon}{1+\varepsilon}\right)$$
$$\leq \frac{4U\left(m+1\right)\left(1+\varepsilon\right)K}{KU\varepsilon}\left(\frac{2+3\varepsilon}{1+\varepsilon}\right)$$
$$\leq \frac{4\left(m+1\right)\left(2+3\varepsilon\right)}{\varepsilon},$$

where the passage from the third line to the fourth one is possible by using the definition of q_d; see line 6 of Algorithm 2. Hence, the maximum number of calls only depends 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 4, the maximum number of calls to IMPROVE (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 10 of Algorithm 2. Therefore, the maximum number of iteration is in the order of $\mathcal{O}\left(\log \tilde{K}_0\right)$, and the total number of calls of IMPROVE 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. ■

Since $(\varepsilon,\mathcal{A})$-`LSA` is guaranteed to terminate, even when the follower's response contains continuous variables, we show that the solution obtained is, in fact, weakly ε-local optimal. The result bellow is analogous to Theorem 3 in Section 5.2.

Proposition 6. *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. Let $\mathbf{x}^{\varepsilon,\mathcal{A}}$ be the leader's feasible decision obtained by calling $(\varepsilon,\mathcal{A})$-`LSA`, and let $\mathbf{x}\in N_{\mathcal{X}}(\mathbf{x}^{\varepsilon,\mathcal{A}})$ be a leader's feasible decision in its neighborhood. Let q_a, q_d and γ denote the values obtained before $(\varepsilon,\mathcal{A})$-`LSA` terminates in the last iteration i_f; recall lines 6 and 8 from Algorithm 2. Then:

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

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

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 "answer," which is returned by IMPROVE, is "No." Consequently, the gap between $\mathbf{x}^{\varepsilon,A}$ and $\mathbf{x}$ with respect to the scaled vectors satisfies $\Delta^{\mathcal{A}}(\mathbf{x}^{\varepsilon,A}, \mathbf{x}, \mathbf{a}', \mathbf{d}') \le \gamma$.

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

$$\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) \le \sum_{j=1}^{n} q_a \left\lceil \frac{a_j}{q_a} \right\rceil x_j + \sum_{\ell=1}^{m} q_d \left\lceil \frac{d_\ell}{q_d} \right\rceil \left(y_\ell^{\mathcal{A}}(\mathbf{x}) + U \left(m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d} \right)^{-1} \right) \tag{45a}$$

$$\le \sum_{j=1}^{n} q_a \left(\frac{a_j}{q_a} + 1 \right) x_j + \sum_{\ell=1}^{m} q_d \left(\frac{d_\ell}{q_d} + 1 \right) \left(y_\ell^{\mathcal{A}}(\mathbf{x}) + U \left(m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d} \right)^{-1} \right) \tag{45b}$$

$$\le \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + nq_a + mUq_d + U \frac{\sum_{\ell=1}^{m} d_\ell + mq_d}{m + \frac{\sum_{\ell=1}^{m} d_\ell}{q_d}} \tag{45c}$$

$$\le \mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}) + nq_a + (m+1)Uq_d, \tag{45d}$$

where (45a) and (45b) follow by the definition of γ and by the properties of the ceiling function, respectively; recall line 8 of Algorithm 2.

In addition, according to the stopping criteria of the loop in line 10 of Algorithm 2, together with the previous inequality, we have that:

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

Therefore, the following inequalities are satisfied:

$$\begin{aligned} \frac{\mathbf{a}^\top \mathbf{x}^{\varepsilon,\mathcal{A}} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x}^{\varepsilon,\mathcal{A}}) - \mathbf{a}^\top \mathbf{x} - \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})} &\le \frac{nq_a + (m+1)Uq_d}{\mathbf{a}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y}^{\mathcal{A}}(\mathbf{x})} \\ &\le \frac{nq_a + (m+1)Uq_d}{\frac{K}{2} - nq_a - (m+1)Uq_d} = \varepsilon, \end{aligned}$$

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 the neighborhood can be efficiently searched, i.e., if* Improve *is a polynomial-time algorithm, then* $(\varepsilon, \mathcal{A})$-`LSA` *is a polynomial-time algorithm that finds an* ε*-local optimal solution to* [**C-BP**] *.*

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 6. 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* Improve *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* (10)*.*

In Proposition 2, we establish a sufficient condition under which $(\varepsilon, \mathcal{A})$-`LSA` is guaranteed to return a weak local optimal solution. This condition is based on a better comprehension of the scaling step within Algorithm 2; recall our discussion before Proposition 2. A similar condition can be derived whenever the follower's decision variables are also continuous. We assume that $\alpha, \beta, \Delta_x^*$ and $\Delta_y^{\mathcal{A}}$ are given as in the discussion preceding Proposition 2. Then:

Proposition 7. *Let* $\varepsilon > 0$*,* $N_{\mathcal{X}}$ *be a neighborhood function,* $\mathcal{A}$ *be an algorithm that returns a feasible solution to the follower's problem* (1c) *for any leader's feasible decision. Assume that* K*,* γ*,* q_a *and* q_d *are the parameters obtained at the last iteration* i_f *in Algorithm 2 before it terminates. Let* $\mathbf{x}^{\varepsilon,\mathcal{A}}$ *be the solution obtained by* $(\varepsilon, \mathcal{A})$-`LSA`*. If the following condition holds:*

$$\Delta_x^* \sum_{j=1}^{n} \alpha_j + \Delta_y^{\mathcal{A}} \sum_{\ell=1}^{m} \beta_\ell \leq \gamma, \tag{47}$$

then $\mathbf{x}^{\varepsilon,\mathcal{A}}$ *is weakly local optimal with respect to* $N_{\mathcal{X}}$ *and* $\mathcal{A}$*.*

Proof of Proposition 7. The proof is essentially the same as the one for Proposition 2. The only distinction is that we use γ as the minimum improving gap. ■

As a final comment, if the follower's variables are all continuous, then the lower-level problem is an LP, which can be efficiently solved exactly. Hence, if, first, the follower's decision are all continuous, and, second, the neighborhood can be searched efficiently, then Proposition 7 is a sufficient condition under which $(\varepsilon, \mathcal{A})$-`LSA` returns a local optimal solution in polynomial time.

D.5 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 (Moore and Bard 1990). 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 (Fischetti et al. 2005).

General integrality at the upper level. Next, we consider another extension, where the leader's decision variables are generalized to integer values, i.e., $\mathbf{x} \in \{1, \ldots, u\}$, 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.

Exact neighborhood. In single-level combinatorial optimization, a neighborhood function is defined as *exact* if any local optimal solution with respect to that neighborhood is also globally optimal (Orlin et al. 2004). For example, any local optimal solution to the minimum spanning tree problem, with respect to the 2-flip neighborhood function is guaranteed to be globally optimal. It has also been established that an ε-local optimal solution, defined with respect to an exact neighborhood, does not necessarily constitute an ε-approximate global solution. Nevertheless, the

ε-local search algorithm designed for single-level combinatorial optimization is guaranteed to return an approximate (global) solution whenever the neighborhood function is exact (Orlin et al. 2004).

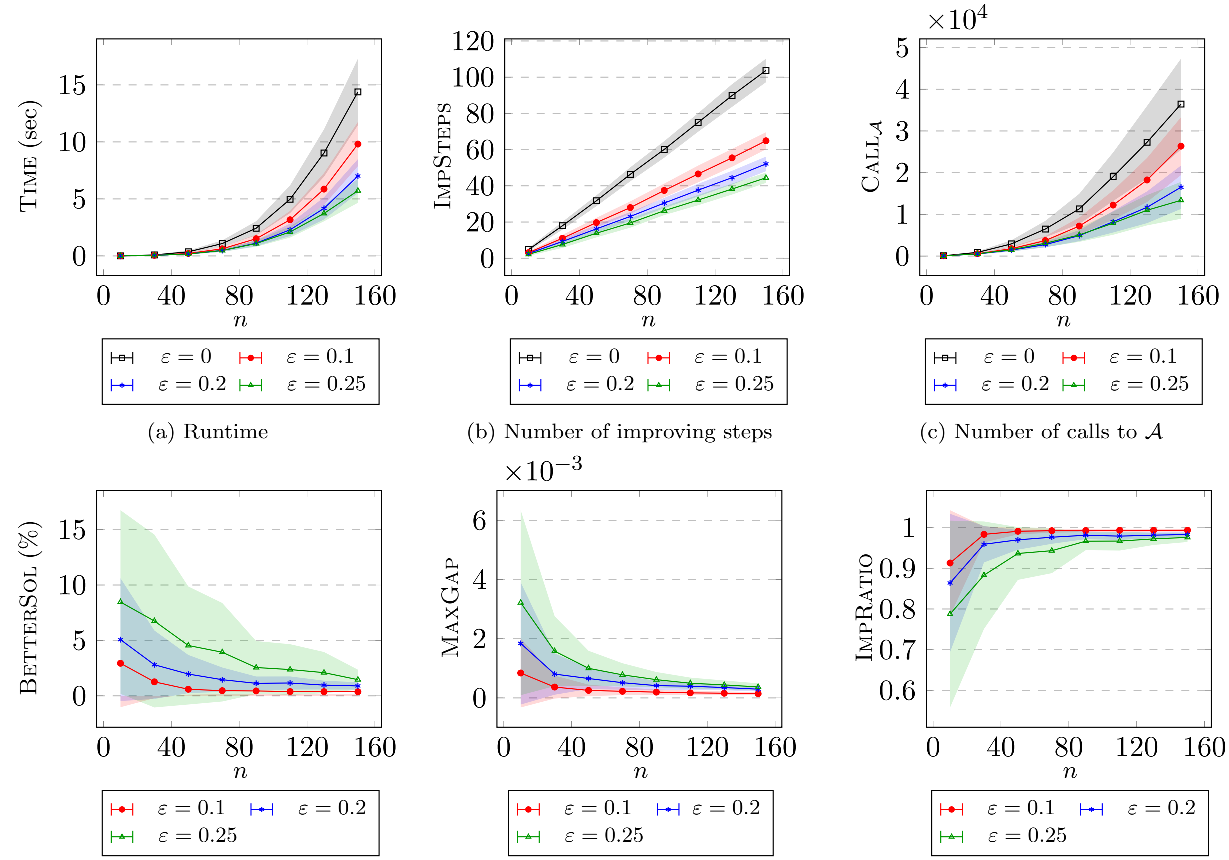
An analogous result can be shown to hold for the $(\varepsilon, \mathcal{A})$-`LSA` when the follower's problem is solved to optimality; for brevity, we omit the formal statement and proof. In contrast, the question of whether these property persists when the follower's problem is solved only approximately remains open and presents an interesting avenue for future research.

E Additional illustrations and discussions for Section 6

In this section, we present a complete analysis of our computational experiments, which were previously omitted for conciseness. In Section E.1, we include the additional figures for the knapsack interdiction problem. Then, in Section E.2, we provide the supplementary tables for the maximum weighted clique interdiction problem.

E.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 8 shows the results for the 2-flip neighborhood function, while Figure 9 presents those for the 3-flip neighborhood function.



(a) Runtime

(b) Number of improving steps

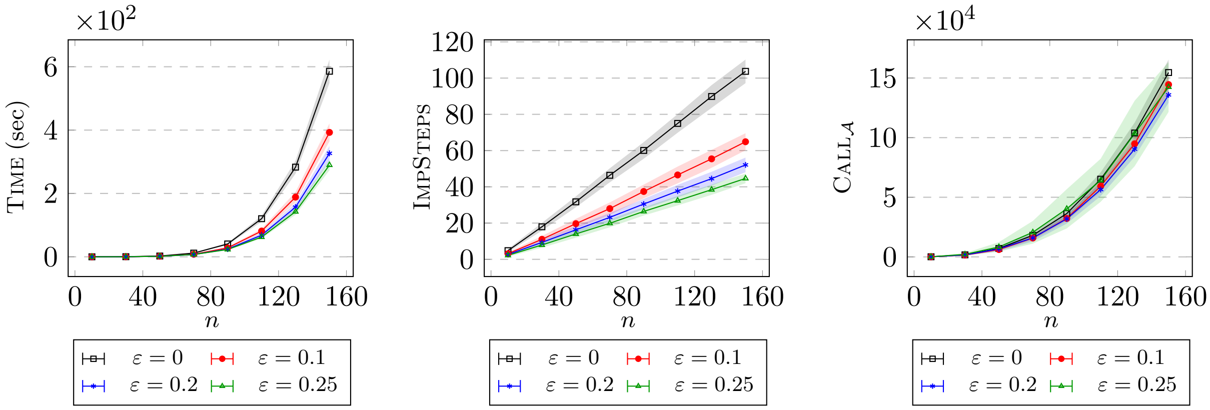
(c) Number of calls to $\mathcal{A}$

(d) Percentage of improving solution compare 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 compare to the improvement of LSA

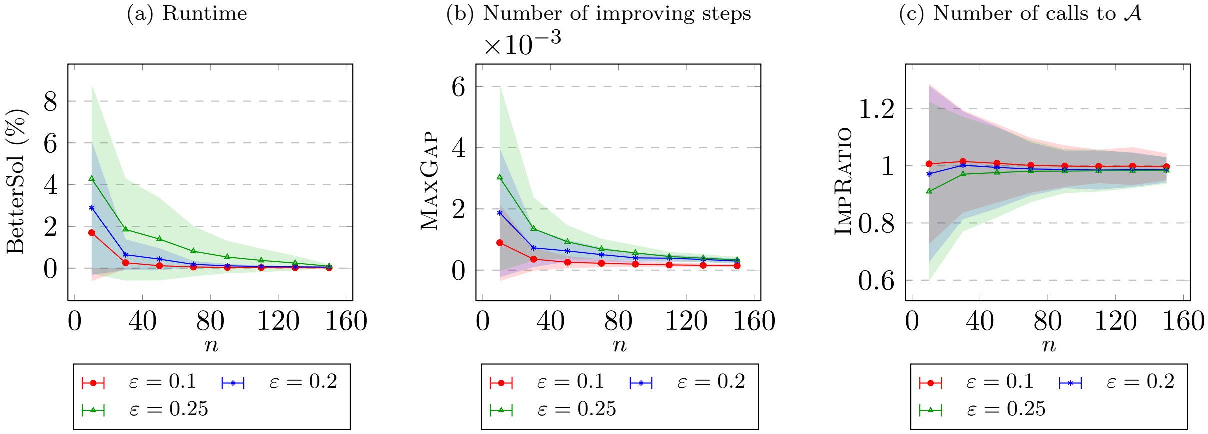
Figure 8: **Continuous follower - exact follower ($\delta = 0$) - 2-flip neighborhood ($k = 2$):** comparison of the efficiency and the performance for the knapsack interdiction problem; see Section 6.2.1. Recall that $\varepsilon = \delta = 0$ corresponds to LSA, while $\varepsilon > 0$ corresponds to $(\varepsilon, \mathcal{A})$-`LSA`. Each line shows the average (Avg), with the shaded region indicating Avg $\pm$ MAD.



(a) Runtime

(b) Number of improving steps

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



(d) Percentage of improving solution compare 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` compare to the improvement of LSA

Figure 9: **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.

E.2 Maximum weighted clique interdiction problem

Below are the missing tables from Section 6.3. Specifically, Table 5 shows the results for the weak local search, where the lower-level problem is solved approximately, while Table 6 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 5: $(\varepsilon, \mathcal{A})$-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 (9) 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 6: $(\varepsilon, \mathcal{A})$-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 (9) in Section 6.3. Moreover, the leader interdicts 10% of the vertices, i.e., $h = 0.1n$.

E.3 Additional computational experiments on non-interdiction instances

We extend our computational experiments on some non-interdiction instances from the library compiled by Thürauf et al. (2024) 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 key characteristics are summarized in Table 7). If an instance does not satisfy our assumptions (notably Assumption **A4**), then we apply the transformation described in Appendix A.

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 7: 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 Thürauf et al. (2024).

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 for both approaches. Our computations are restricted to the 2-flip neighborhood function. As a side observation, none of the instances in Table 7 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.

	Time (sec)			ImpSteps		Call$_{\mathcal{A}}$		ImpRatio	
Instance	0	0.25	relaxation	0	0.25	0	0.25	0	0.25
cov1075-0-100	40.06	39.40	1.38	1	1	1,289	1,289	1.41	1.41
glass-sc-0-100	342.01	352.19	285.82	3	3	6,073	6,073	2.73	2.73
iis-100-0-cov-0-100	59.30	56.18	83.24	7	7	1,294	1,294	11.29	11.29
iis-bupa-cov-0-100	844.14	810.32	223.00	5	5	15,617	15,617	2.81	2.81
lseu-0.500000	10.10	7.87	0.26	7	5	821	715	16.86	11.09
lseu-0.900000	1.50	0.31	0.25	1	0	51	45	2.72	0.00
mad-0-100	98.53	54.14	0.42	5	3	4,953	6,414	30.00	20.00
mas74-0-100	20.61	20.32	0.02	17	17	3,507	3,507	0.02	0.02
p0201-0.500000	98.60	92.81	0.15	1	2	11	14	0.42	0.42
p0282-0.500000	192.34	536.07	0.05	1	7	2,209	5,173	0.10	0.59
p0282-0.900000	4.55	4.60	0.08	10	10	523	523	15.11	15.11
p0548-0.500000	12,913.21	927.48	0.12	16	2	53,099	5,799	1.41	1.11

Table 8: **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 7. Runtime for solving the single-level relaxation is also reported.

We observe from Table 8 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.

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 higher improvement ratio (see the ImpRatio column).

References

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

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

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

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

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

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

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

Gurobi (2024). *Gurobi Optimization, LLC*. `https://www.gurobi.com`. Accessed: 2024-07-26.

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

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

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

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

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

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

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

Thürauf, J., Kleinert, T., Ljubić, I., Ralphs, T., and Schmidt, M. (2024). *BOBILib: Bilevel Optimization (Benchmark) Instance Library*. URL: https://optimization-online.org/?p=27063.

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

Vazirani, V. V. (2001). *Approximation algorithms*. Springer Berlin, Heidelberg.

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

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

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

Yang, J., Shi, X., and Prokopyev, O. A. (2023). “Exact solution approaches for a class of bilevel fractional programs”. In: *Optimization Letters* 17.1, pp. 191–210.