Bilevel Programming Approaches to the Computation of Optimistic and Pessimistic Single-Leader-Multi-Follower Equilibria

Nicola Basilico[1], Stefano Coniglio[2], Nicola Gatti[3], and Alberto Marchesi[4]

1 **Dipartimento di Informatica, University of Milan, Milano, Italy**
nicola.basilico@unimi.it

2 **Department of Mathematical Sciences, University of Southampton, Southampton, UK**
s.coniglio@soton.ac.uk

3 **Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy**
nicola.gatti@polimi.it

4 **Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy**
alberto.marchesi@polimi.it

Abstract

We study the problem of computing an equilibrium in leader-follower games with a single leader and multiple followers where, after the leader's commitment to a mixed strategy, the followers play simultaneously in a noncooperative way, reaching a Nash equilibrium. We tackle the problem from a bilevel programming perspective. Since, given the leader's strategy, the followers' subgame may admit multiple Nash equilibria, we consider the cases where the followers play either the best (optimistic) or the worst (pessimistic) Nash equilibrium in terms of the leader's utility. For the optimistic case, we propose three formulations which cast the problem into a single level mixed-integer nonconvex program. For the pessimistic case, which, as we show, may admit a supremum but not a maximum, we develop an *ad hoc* branch-and-bound algorithm. Computational results are reported and illustrated.

1998 ACM Subject Classification G.1.6 [Optimization] Nonlinear programming

Keywords and phrases Stackelberg games, Nash equilibria, Game theory, Bilevel and nonlinear programming, Branch-and-bound

Digital Object Identifier 10.4230/LIPIcs.SEA.2017.31

1 Introduction

In recent years, *leader-follower* (or *Stackelberg*) *games* have attracted a growing interest not just in game theory, but also in areas such as transportation science, security science, and combinatorial optimization. Such games model the interaction between rational agents (or players) in the context of sequential decision making. Considering, for simplicity, the two-player case, these games address situations where one agent plays first (the *leader*) and the other agent (the *follower*) plays second, after observing the *mixed strategy* the leader has committed to (a probability distribution over his/her actions). The algorithmic task is to compute an *equilibrium* (often called *solution* to the game), that is, a set of mixed strategies, one per player, with the property that no player could obtain a larger utility by deviating from the equilibrium, provided that the other players act as the equilibrium prescribes.

In game theory, rich applications can be found in, among others, the security domain [2, 9]. A defender, whose aim is to protect a set of valuable targets from the attackers, plays first, as leader, while the attackers, acting as followers, observe the leader's defensive strategy and play second. Among different applications in combinatorial optimization, we mention interdiction problems [5, 13], toll setting problems [10], and network routing problems [1].

Most of the game-theoretical investigations on leader-follower games have, to the best of our knowledge, chiefly addressed the case of a single-follower. In that setting, it is known that the single follower can play, w.l.o.g., only a pure strategy (a probability distribution where a single action is played with probability 1), i.e., that there is always a pure strategy by which he/she can maximize his/her utility, and that computing an equilibrium is easy with complete information [17], while it becomes NP-hard for Bayesian games [7]. Algorithms are proposed in [7]. For what concerns games with more than two players, some works have investigated the case with multiple leaders and a single follower, see [11]. For the problem involving a single leader and multiple followers (the one on which we focus in this paper), only few results are available. It is known, for instance, that an equilibrium can be found in polynomial time if the followers play a correlated equilibrium [6], whereas the problem is NP-hard [7] if they play sequentially one at a time.

In this paper, we focus on the single-leader multi-follower case where the followers play *simultaneously* and *noncooperatively*, thus reaching a Nash Equilibrium (NE). We refer to an equilibrium in such games as to a *Single-Leader-Multi-Follower Nash Equilibrium* (SLMFNE). Computing such an equilibrium naturally amounts to solving a *bilevel programming* problem. Since, for a given mixed strategy of the leader, multiple NE might arise in the followers' subgame, we consider two cases: the *optimistic* one, where the followers play a NE which maximizes the leader's utility, and the *pessimistic* case, where the followers play a NE which minimizes it. Solving these two problems allows us to compute the tightest range of values the leader's utility can take independently of which NE is selected. We remark that, although the concept of SLMFNE has already been considered in the literature from a theoretical perspective, see, in particular, [17], no algorithmic methods to compute such an equilibrium are known.

The paper is organized as follows. The definition of the problem, its bilevel programming nature, and some of its properties are described in Section 2. We tackle the optimistic case in Section 3, where we construct three exact mixed-integer nonconvex mathematical programming formulations for it. The pessimistic case is addressed in Section 4, where we propose an *ad hoc* branch-and-bound method for its solution. Computational results are illustrated in Section 5, while Section 6 draws some concluding remarks.[1]

2 The problem

Consider a game with n players, with index set $N = \{1, \dots, n\}$. For each $p \in N$, let A_p be his/her set of actions, with $m_p := |A_p|$, and let the vector $x_p \in [0,1]^{m_p}$, subject to $\sum_{a \in A_p} x_p^a = 1$, be the player's *strategy vector* (or strategy, for short). For each player $p \in N$, each component x_p^a of x_p corresponds to the probability by which action $a \in A_p$ is played. We call x_p a vector of *pure strategies* if $x_p \in \{0,1\}^{m_p}$, or of *mixed strategies* if $x_p \in [0,1]^{m_p}$. Throughout the paper, if not stated otherwise, we assume that all strategies are mixed. We

[1] An extended abstract of a preliminary version of this work appeared in [4].

also denote the collection of strategies of the different players, which forms a so-called *strategy profile*, by $x = (x_1, \ldots, x_n)$.

We consider *normal form games* (a reference can be found in [16]). They are characterized by, for each $p \in N$, a multidimensional utility (or payoff) matrix $U_p \in \mathbb{Q}^{m_1 \times \cdots \times m_n}$, whose components $U_p^{a_1,\ldots,a_n}$ correspond to the utility obtained by player p when all the players play actions $a_1, \ldots, a_n$. For a strategy profile $x = (x_1, \ldots, x_n)$, the expected utility of player p is the multilinear function:

$$u_p(x_1, \ldots, x_n) = \sum_{a_1 \in A_1} \cdots \sum_{a_n \in A_n} U_p^{a_1,\ldots,a_n} x_1^{a_1} \cdots x_n^{a_n}.$$

With n players, this is an nth-degree polynomial.

According to the standard definition, a strategy profile $x = (x_1, \ldots, x_n)$ is a Nash Equilibrium (NE) if, for each player $p \in N$ and for each strategy profile x' with $x'_q = x_q$ for all $q \in N \setminus \{p\}$ with, possibly, $x'_p \neq x_p$, the following inequality holds:

$$u_p(x_1, \ldots, x_n) \geq u_p(x'_1, \ldots, x'_n).$$

Intuitively, this is the same as imposing that, for each $p \in N$ and assuming the other players in $N \setminus \{p\}$ played as prescribed by the strategy profile x, player p would be unable to improve his/her utility when deviating from x_p by playing any other strategy $x'_p \neq x_p$.

In the remainder of the paper and for ease of notation, we consider the case of two followers (thus, with $n = 3$), assuming, w.l.o.g., $m_1 = m_2 = m_3 = m$. The nth player (player 3) takes the role of leader. The extension to the case of any $n > 3$ is, although notationally more involved, not difficult.

2.1 Problem definition

Computing a SLMFNE amounts to solving a so-called *bilevel programming problem with two followers*. Let, for each $p \in N$:

$$\Delta_p := \{x_p \in [0,1]^m \colon \sum_{a \in A_p} x_p^a = 1\}.$$

In the Optimistic case, we can compute a SLMFNE (O-SLMFNE) by solving:

$$\text{(O-SLMFNE)} \quad \max_{\substack{(x_1,x_2,x_3)\in \\ \Delta_1\times\Delta_2\times\Delta_3}} \sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_3^{ijk} x_1^i x_2^j x_3^k \tag{1a}$$

$$\text{s.t.} \quad x_1 \in \operatorname*{argmax}_{x_1\in\Delta_1} \left\{ \sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_1^{ijk} x_1^i x_2^j x_3^k \right\} \tag{1b}$$

$$x_2 \in \operatorname*{argmax}_{x_2\in\Delta_2} \left\{ \sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_2^{ijk} x_1^i x_2^j x_3^k \right\}. \tag{1c}$$

Due to Constraints (1b)–(1c), the second level problems call for a pair (x_1, x_2) of followers' strategies forming a NE in the followers' subgame induced by the $x_3 \in \Delta_3$ chosen by the leader in the first level. Observe that, due to the definition of NE, the pair (x_1, x_2) is a NE for the given x_3 if and only if, at the same time, x_1 maximizes player 1's utility when assuming that player 2 would play x_2, and x_2 maximizes player 2's utility when assuming that player 1 would play x_1. Subject to those constraints, the first level calls for a triple (x_1, x_2, x_3) maximizing the leader's utility.

The problem is optimistic as, assuming that the second level admits many NE (x_1, x_2) for the chosen x_3, it calls for a pair (x_1, x_2) which, together with x_3, maximizes the leader's utility. Notice that, while any triple $(x_1, x_2, x_3) \in \Delta_1 \times \Delta_2 \times \Delta_3$ is a *feasible* solution to the problem as long as the pair (x_1, x_2) is a NE in the subgame induced by x_3, Problem (1a)–(1c) calls for a triple (x_1, x_2, x_3) which is *optimal*—as, if not, the leader would prefer to change his/her strategy and (x_1, x_2, x_3) would not be a SLMFNE.

In the Pessimistic case, computing a SLMFNE (P-SLMFNE) calls for a solution to the following problem:

$$\text{(P-SLMFNE)} \max_{\substack{(x_1,x_2,x_3)\in \\ \Delta_1\times\Delta_2\times\Delta_3}} \min_{\substack{(x_1,x_2)\in \\ \Delta_1\times\Delta_2}} \sum_{i\in A_1} \sum_{j\in A_2} \sum_{k\in A_3} U_3^{ijk} x_1^i x_2^j x_3^k \tag{2a}$$

$$\text{s.t.} \quad \text{Constraints (1b), (1c).} \tag{2b}$$

This problem differs from its optimistic counterpart as, due to the assumption of pessimism, the leader here maximizes the *minimum* value taken by his/her utility over all pairs (x_1, x_2) which are NE in the followers' subgame induced by x_3.

2.2 Some properties of the problem

We observe that, as it is often the case in bilevel problems, the difference in leader's utility between an optimistic and a pessimistic SLMFNE can be arbitrarily large. Consider, for some $\lambda > 0$, a game with $n = 3$, $A_1 = \{i_1, i_2\}, A_2 = \{j_1, j_2\}$, $A_3 = \{k_1\}$, and utilities:

	j_1	j_2
i_1	1,1,λ	0,0,0
i_2	0,0,0	1,1,0

k_1

The unique O-SLMFNE in this game is (i_1, j_1, k_1), corresponding to a leader's utility of λ, while the unique P-SLMFNE is (i_2, j_2, k_1), with a leader's utility of 0. It follows that, for $\lambda \to \infty$, their difference in terms of leader's utility tends to ∞.

Differently from the O-SLMFNE case, where, as shown in [17], an equilibrium is always guaranteed to exist, this is not the case for P-SLMFNE—a behavior which can be observed in many pessimistic bilevel problems [18]. Consider a game with $n = 3$, $A_1 = \{i_1, i_2\}$, $A_2 = \{j_1, j_2\}$, $A_3 = \{k_1, k_2\}$. The matrices reported in the following are the utility matrices for, respectively, the case where the leader plays action k_1 with probability 1, action k_2 with probability 1, or the strategy vector $x_3 = (1 - \rho, \rho)$ for some $\rho \in (0, 1)$ (the latter matrix is the convex combination of the first two with weights x_3):

	j_1	j_2
i_1	1,1,0	2,2,5
i_2	$\frac{1}{2}$,$\frac{1}{2}$,1	1,1,0

k_1

	j_1	j_2
i_1	0,0,0	2,2,10
i_2	$\frac{1}{2}$,$\frac{1}{2}$,1	0,0,0

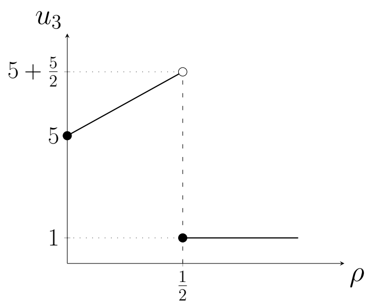
k_2

	j_1	j_2
i_1	$1-\rho$,$1-\rho$,0	2,2,$5+5\rho$
i_2	$\frac{1}{2}$,$\frac{1}{2}$,1	$1-\rho$,$1-\rho$,0

$x_3 = (1 - \rho, \rho)$

In the optimistic case, (i_1, j_2, k_2) is the unique SLMFNE (it is the only pure one and, as it achieves the leader's largest utility in U_3, mixed strategies cannot yield a better utility).

In the pessimistic case, if the leader played k_2, the followers would respond, among the two NE (i_2, j_1) and (i_1, j_2), with (i_2, j_1) (so to minimize the leader's utility). If the leader played $x_3 = (1-\rho, \rho)$, we would obtain the subgame in the third matrix. For $\rho < \frac{1}{2}$, (i_1, j_2) is the unique NE, giving the leader a utility of $5 + 5\rho$. For $\rho \geq \frac{1}{2}$, we have again the two NE (i_1, j_2) and (i_2, j_1), with a utility of, respectively, $5 + 5\rho$ and 1. Since the latter is selected in the pessimistic case, we conclude that the game admits no pessimistic SLMFNE. This is because the leader's utility (and the optimization problem), when written as a function of ρ, achieves a supremum at $\rho = \frac{1}{2}$, but not a maximum. See the following graph for an illustration.



From a combinatorial perspective, the hardness and inapproximability (in polynomial time, up to within a constant factor) of the problem of computing a SLMFNE is a direct consequence of the NP-hardness and inapproximability of computing, in a two-player game, a NE which maximizes the sum of the players' utilities [8]. Indeed, the latter problem is directly reduced to the computation of a SLMFNE for the case where the leader has a single action and his/her utility is equal to plus (in the optimistic case) or minus (in the pessimistic case) the sum of the utilities of the followers.

3 Optimistic case

We propose three Mixed-Integer NonLinear Programming (MINLP) formulations to compute a SLMFNE in the optimistic case.

3.1 MINLP-I

To obtain a first single level formulation for the problem, we proceed as follows, applying a standard reformulation [16] involving complementarity constraints.

Let, for all $i \in A_1$ and $j \in A_2$, $\tilde{U}_1^{ij} := \sum_{k \in A_3} U_1^{ijk} x_3^k$ and $\tilde{U}_2^{ij} = \sum_{k \in A_3} U_2^{ijk} x_3^k$ be the matrices of the followers' subgame, parameterized by x_3. For (x_1, x_2) to be a NE, x_1 must be an optimal solution to the Linear Program (LP):

$$\max_{x_1 \in \Delta_1} \left\{ \sum_{i \in A_1} \sum_{j \in A_2} \tilde{U}_1^{ij} x_1^i x_2^j \right\},$$

where $\tilde{U}_1^{ij} x_1^i x_2^j$ is a linear function of x_1. Since the LP is feasible and bounded for any $x_2 \in \Delta_2$, we have, by complementary slackness, that $x_1 \in \Delta_1$ is optimal if and only if there is a scalar v_1 such that, for all $i \in A_1$:

$$\big(v_1 - \sum_{j \in A_2} \tilde{U}_1^{ij} x_2^j\big) x_1^i = 0$$

$$v_1 \geq \sum_{j \in A_2} \tilde{U}_1^{ij} x_2^j.$$

Applying a similar reasoning to x_2, we obtain that $x_2 \in \Delta_2$ is optimal if and only if there is a scalar v_2 such that, for all $j \in A_2$:

$$\big(v_2 - \sum_{i \in A_1} \tilde{U}_2^{ij} x_1^i\big) x_2^j = 0$$

$$v_2 \geq \sum_{i \in A_1} \tilde{U}_2^{ij} x_1^i.$$

We conclude that (x_1, x_2) is a NE if and only if there are $v_1, v_2 \geq 0$ such that x_1 and x_2 simultaneously satisfy these four conditions.

After substituting for $\tilde{U}_1$ and $\tilde{U}_2$ their linear expressions in x_3, we obtain, for player 1 and for all $i \in A_1$, constraints:

$$\left(v_1 - \sum_{j \in A_2} \sum_{k \in A_3} U_1^{ijk} x_2^j x_3^k\right) x_1^i = 0$$

$$v_1 \geq \sum_{j \in A_2} \sum_{k \in A_3} U_1^{ijk} x_2^j x_3^k$$

and, for player 2 and for all $j \in A_2$, constraints:

$$\left(v_2 - \sum_{i \in A_1} \sum_{k \in A_3} U_2^{ijk} x_1^i x_3^k\right) x_2^j = 0$$

$$v_2 \geq \sum_{i \in A_1} \sum_{k \in A_3} U_2^{ijk} x_1^i x_3^k.$$

By imposing them *in lieu* of the two second level argmax constraints of Problem (1), that is, Constraints (1b)–(1c), we obtain a continuous single level formulation with nonconvex trilinear terms.[2][3]

3.2 MINLP-II

What we propose now is aimed at achieving a formulation which can be solved more efficiently.

Since each term of the complementarity constraints we introduced is bounded, we can apply a simple reformulation. Letting $s_1 \in \{0,1\}^m$ and $s_2 \in \{0,1\}^m$ be the *antisupport vectors* of x_1 and x_2, it suffices to impose, for all $i \in A_1$:

$$x_1^i \leq 1 - s_1^i$$

$$v_1 - \sum_{j \in A_2} \sum_{k \in A_3} U_1^{ijk} x_2^j x_3^k \leq M s_1^i$$

and, for all $j \in A_2$:

$$x_2^j \leq 1 - s_2^j$$

$$v_2 - \sum_{i \in A_1} \sum_{k \in A_3} U_2^{ijk} x_1^i x_3^k \leq M s_2^j,$$

where M is an upper bound on the entries of U_1, U_2. This way, while still retaining the original trilinear objective function, only bilinear constraints are needed.

---

[2] Note that strong duality can be employed in place of complementary slackness. Preliminary experiments, though, suggest that the second option is computationally preferable.

[3] We remark that, in this form, the problem correspond to a Mathematical Program with Equilibrium Constraints (MPEC). The interested reader can find more references to this type of problems in [12].

3.3 MINLP-III

Ultimately, we aim to solve the problem with spatial branch-and-bound techniques, such as those implemented in SCIP. The main strategy of such methods to handle nonlinearities is to isolate "simple" nonlinear terms (bilinear or trilinear in our case) by shifting them into a new (so-called *defining*) constraint to which convex envelopes are applied.

We propose to anticipate this reformulation, so to be able to derive some valid constraints. First, we introduce:

(i) variable y_{23}^{jk} and constraint $y_{23}^{jk} = x_2^j x_3^k$ for all $j \in A_2, k \in A_3$,

(ii) variable y_{13}^{ik} and constraint $y_{13}^{ik} = x_1^i x_3^k$ for all $i \in A_1, k \in A_3$,

(iii) variable z^{ijk} and constraint $z^{ijk} = x_1^i y_{23}^{jk}$ for all $i \in A_1, j \in A_2, k \in A_3$.

Then, by substituting each bilinear and trilinear term with the newly introduced variables, we obtain a formulation which is linear everywhere, except for the defining constraints.

We now observe that, by definition, the matrix $\{y_{23}^{jk}\}_{jk \in A_2 \times A_3}$ is the outer product of the stochastic vectors x_2 and x_3 and, as such, is a stochastic matrix itself. The same holds for the tensor $\{z^{ijk}\}_{ijk \in A_1 \times A_2 \times A_3}$, which is the outer product of the vectors x_1, x_2, x_3 and, as such, is a stochastic tensor. This implies the validity of the following three constraints:

$$\sum_{i \in A_1} \sum_{k \in A_3} y_{13}^{ik} = 1$$

$$\sum_{j \in A_2} \sum_{k \in A_3} y_{23}^{jk} = 1$$

$$\sum_{i \in A_1} \sum_{j \in A_2} \sum_{k \in A_3} z^{ijk} = 1.$$

The final single level mixed-integer nonconvex formulation that we propose for O-SLMFNE thus reads:

$$\max \sum_{i \in A_1} \sum_{j \in A_2} \sum_{k \in A_3} U_3^{ijk} z^{ijk}$$

$$\begin{array}{ll|l}
\text{s.t. } v_1 - \sum_{j \in A_2} \sum_{k \in A_3} U_1^{ijk} y_{23}^{jk} \leq M s_1^i & \forall i \in A_1 & \\
v_2 - \sum_{i \in A_1} \sum_{k \in A_3} U_2^{ijk} y_{13}^{ik} \leq M s_2^j & \forall j \in A_2 & \sum_{i \in A_1} \sum_{k \in A_3} y_{13}^{ik} = 1 \\
x_1^i \leq 1 - s_1^i & \forall i \in A_1 & \sum_{j \in A_2} \sum_{k \in A_3} y_{23}^{jk} = 1 \\
x_2^j \leq 1 - s_2^j & \forall j \in A_2 & \\
v_1 \geq \sum_{j \in A_2} \sum_{k \in A_3} U_1^{ijk} y_{23}^{jk} & \forall i \in A_1 & \sum_{i \in A_1} \sum_{j \in A_2} \sum_{k \in A_3} z^{ijk} = 1 \\
v_2 \geq \sum_{i \in A_1} \sum_{k \in A_3} U_2^{ijk} y_{13}^{ik} & \forall j \in A_2 & (x_1, x_2, x_3) \in \Delta_1 \times \Delta_2 \times \Delta_3 \\
y_{13}^{ik} = x_1^i x_3^k & \forall (i,k) \in A_1 \times A_3 & y_{13}, y_{23} \in [0,1]^{m \times m} \\
y_{23}^{jk} = x_2^j x_3^k & \forall (j,k) \in A_2 \times A_3 & z \in [0,1]^{m \times m \times m} \\
& & s_1, s_2 \in \{0,1\}^m \\
z^{ijk} = x_1^i y_{23}^{jk} & \forall (i,j,k) \in A_1 \times A_2 \times A_3 & v_1, v_2 \text{ free.}
\end{array}$$

4 Pessimistic case

For P-SLMFNE, we introduce an *ad hoc* method based on branch-and-bound. For simplicity, we first describe it for the case where the followers are restricted to pure strategies, with the extension to the mixed case following next.

4.1 Basic enumerative idea and outcome configurations

The key ingredient of the method is what we call *outcome configuration.* Given a strategy vector $x_3 \in \Delta_3$, we call a pair (S^+, S^-), with $S^+ \subseteq A_1 \times A_2$ and $S^- = A_1 \times A_2 \setminus S^+$, *outcome configuration* if, for the given x_3, it satisfies two constraints:

(i) all pairs of actions in S^+ correspond to a NE in the followers' subgame;
(ii) all pairs in S^- do not.

Given a pair (S^+, S^-), a strategy vector $x_3 \in \Delta_3$ such that i) and ii) are satisfied guarantees that, if the leader played x_3, the set of NE in the followers' subgame would coincide with S^+. Since, given an outcome configuration (S^+, S^-), the leader's utility at each NE in S^+ is a (linear) function of x_3, we must then look (due to the pessimistic setting) for an x_3 which iii) maximizes the smallest value taken by the leader's utility over S^+.

By constructing (we will propose a more efficient method later on) all the pairs (S^+, S^-) in $2^{A_1 \times A_2}$ and computing (if it exists) the corresponding x_3, a P-SLMFNE (if it exists) is obtained by choosing the triple (S^+, S^-, x_3) which gives the leader the largest utility (and, then, selecting the NE in S^+ which minimizes the leader's utility).

Let us discuss how to compute x_3 for a given (S^+, S^-). By definition of NE, constraints i) can be expressed as the following set of inequalities, which are linear in x_3:

$$\sum_{k \in A_3} U_1^{ijk} x_3^k \geq \sum_{k \in A_3} U_1^{i'jk} x_3^k \quad \forall (i,j) \in S^+, i' \in A_1 \setminus \{i\} \tag{3a}$$

$$\sum_{k \in A_3} U_2^{ijk} x_3^k \geq \sum_{k \in A_3} U_2^{ij'k} x_3^k \quad \forall (i,j) \in S^+, j' \in A_2 \setminus \{j\}. \tag{3b}$$

Given some sufficiently small $\epsilon > 0$, Constraints ii) can be (approximately) written as the following disjunction:

$$\bigvee_{i' \in A_1 \setminus \{i\}} \left(\sum_{k \in A_3} U_1^{ijk} x_3^k + \epsilon \leq \sum_{k \in A_3} U_1^{i'jk} x_3^k \right) \quad \bigvee$$
$$\bigvee_{j' \in A_2 \setminus \{j\}} \left(\sum_{k \in A_3} U_2^{ijk} x_3^k + \epsilon \leq \sum_{k \in A_3} U_2^{ij'k} x_3^k \right) \qquad \forall (i,j) \in S^-. \tag{4}$$

For every $(i,j) \in S^-$, the disjunction imposes the existence of an action i' of player 1 giving him/her a utility larger than that obtained when playing i by, at least, some $\epsilon > 0$, assuming the other player played j (or *vice versa* for player 2 and an action j').

Note that each term of the disjunction should be strict, as the disjunction represents the complement of a polytope. The approximation with ϵ has the role of preventing x_3 from reaching one of the breakpoints of the leader's utility function (see the illustration in Section 2), where the pessimistic problem always achieves a supremum but not a maximum.

We can cast Constraints (4) in terms of Mixed-Integer Linear Programming (MILP) by introducing a binary variable per term of the disjunction, with a constraint requiring its sum to be 1.[4] After introducing the binary variables $y_{iji'} \in \{0,1\}$, for $i' \in A_1 \setminus \{i\}$, and

[4] Due to considering polytopes, the extended LP formulation of Balas [3] could be used. Nevertheless, we have found the MILP approach computationally affordable.

$y_{ijj'} \in \{0,1\}$, for $j' \in A_2 \setminus \{j\}$, the reformulation reads:

$$\sum_{k \in A_3} U_1^{ijk} x_3^k + \epsilon \leq \sum_{k \in A_3} U_1^{i'jk} x_3^k + M(1 - y_{iji'}) \qquad \forall i' \in A_1 \setminus \{i\} \tag{5a}$$

$$\sum_{k \in A_3} U_1^{ijk} x_3^k + \epsilon \leq \sum_{k \in A_3} U_2^{ij'k} x_3^k + M(1 - y_{ijj'}) \qquad \forall j' \in A_2 \setminus \{j\} \tag{5b}$$

$$\sum_{i' \in A_1 \setminus \{i\}} y_{iji'} + \sum_{j' \in A_2 \setminus \{j\}} y_{ijj'} = 1, \tag{5c}$$

where M is a (previously introduced) upper bound on the entries of U_1, U_2.

A solution satisfying i) and ii) which is also optimal in the sense of iii) is then found by solving the following MILP subproblem:

$$\max_{\substack{\eta \text{ free} \\ x_3 \in \Delta_3}} \left\{ \begin{array}{ll} \eta : \quad \eta \leq \sum_{k \in A_3} U_3^{ijk} x_3^k & \forall (i,j) \in S^+ \\ \qquad \text{Constraints (3), (4).} & \end{array} \right\} \tag{6}$$

4.2 Branch-and-Bound approach

We now propose an alternative method which does not require to carry out the complete enumeration of the elements of $2^{A_1 \times A_2}$.

Given a strategy vector $x_3 \in \Delta_3$, we call a pair (S^+, S^-) *relaxed outcome configuration* if $S^- \subseteq (A_1 \times A_2) \setminus S^+$, differently from the previous definition where $S^- = (A_1 \times A_2) \setminus S^+$.

Notice that, when $S^- \subset (A_1 \times A_2) \setminus S^+$, it is not always the case that, when the leader plays a solution x_3 to Subproblem (6), the only NE in the followers' subgame are those in S^+. Indeed, due to $S^+ \cup S^- \subset A_1 \times A_2$, the followers' subgame may admit another NE $(i', j') \in A_1 \times A_2 \setminus S^+ \setminus S^-$ which provides the leader with a strictly smaller utility than that he/she would receive from the pairs of NE in S^+. Since, whenever this is the case, (i', j') would be part of the P-SLMFNE corresponding to x_3, the correctness of the method would be lost.

To verify whether this is the case, i.e., whether one such (i', j') exists, it suffices to carry out an operation which we refer to as *feasibility check*. We solve the followers' subgame (in the pessimistic sense) for the given x_3, and compare the NE (i', j') thus found to the worst one in S^+ (when working with pure strategies only, this check can be done in $O(m^2)$). If $(i', j') \notin S^+$, we resort to branching. More precisely, to account for the case where (i', j') is a NE, we introduce a *left node* (S_L^+, S_L^-) with $S_L^+ := S^+ \cup \{(i', j')\}$ and $S_L^- := S^-$, whereas, to account for the case where (i', j') is not a NE, we introduce a *right node* (S_R^+, S_R^-) with $S_R^+ := S^+$ and $S_R^- := S^- \cup \{(i', j')\}$.

We remark that, as a consequence of $S^- \subseteq (A_1 \times A_2) \setminus S^+$, optimal solutions to Subproblem (6) always yield an upper bound on the utility the leader would obtain with a strategy x_3 by which all pairs of followers' actions in S^+ constitute a NE, while all pairs in S^- do not.

At the root node, we solve the optimistic problem, obtaining a triple $(x_1 = e_i, x_2 = e_j, x_3)$. If, by feasibility check, we find a pair $(i', j') = (i, j)$ (or a different one, but yielding the same utility), the problem is solved. If not, we create two nodes: (S_L^+, S_L^-) with $S_L^+ = \{(i', j')\}, S_L^- = \emptyset$, and (S_R^+, S_R^-) with $S_R^+ = \emptyset, S_R^- = \{(i', j')\}$. Since Subproblem (6) is not well–defined for (S_R^+, S_R^-) as, in it, $S_R^+ = \emptyset$, whenever $S^+ = \emptyset$ in a (S^+, S^-) pair, we solve, *in lieu* of Subproblem (6), one of the formulations we proposed for O-SLMFNE with the further addition of Constraints (4) for all pairs in S^-. This way, we can find an upper bound also for the case where S^+ is empty.

4.3 Extension to the unrestricted case

To extend the method to the unrestricted mixed case, assume now that the components of each pair (S^+, S^-) are, rather than pairs of actions, disjoint sets of pairs of strategy vectors $(\bar{x}_1, \bar{x}_2)$ of the followers, with $(\bar{x}_1, \bar{x}_2) \in \Delta_1 \times \Delta_2$.

Constraints i) should now impose all strategy vectors in $(\bar{x}_1, \bar{x}_2) \in S^+$ to be NE. Since the utility obtained with any strategy vector x_p is a convex combination with weights x_p of those obtained with pure strategies, it suffices to impose, for each follower, that $(\bar{x}_1, \bar{x}_2)$ should yield a utility at least as large as that obtained with any pure strategy. We arrive at the following linear (in x_3) inequalities:

$$\sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_1^{ijk}\bar{x}_1^i\bar{x}_2^j x_3^k \geq \sum_{j\in A_2}\sum_{k\in A_3} U_1^{i'jk}\bar{x}_2^j x_3^k \quad \forall(\bar{x}_1,\bar{x}_2)\in S^+, i'\in A_1 \tag{7a}$$

$$\sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_2^{ijk}\bar{x}_1^i\bar{x}_2^j x_3^k \geq \sum_{i\in A_1}\sum_{k\in A_3} U_2^{ij'k}\bar{x}_1^i x_3^k \quad \forall(\bar{x}_1,\bar{x}_2)\in S^+, j'\in A_2. \tag{7b}$$

We can apply a similar argument when stating constraints ii) which, in the mixed case, impose that all strategy vectors in S^- are not NE. Indeed, it suffices to require the existence of an action $i' \in A_1$ or of an action $j' \in A_2$ providing the respective player with a strict improvement to his/her utility. Let $\pi(x_p) := \{a\}$ if $x_p^a = 1$, $\emptyset$ otherwise. For a given, sufficiently small, $\epsilon > 0$, the constraints can be (approximately) written as the following disjunction:

$$\begin{aligned}&\bigvee_{i'\in A_1\setminus\pi(\bar{x}_1)}\left(\sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_1^{ijk}\bar{x}_1^i\bar{x}_2^j x_3^k + \epsilon \leq \sum_{j\in A_2}\sum_{k\in A_3} U_1^{i'jk}\bar{x}_2^j x_3^k\right) \quad \bigvee \\ &\bigvee_{j'\in A_2\setminus\pi(\bar{x}_2)}\left(\sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_2^{ijk}\bar{x}_1^i\bar{x}_2^j x_3^k + \epsilon \leq \sum_{i\in A_1}\sum_{k\in A_3} U_2^{ij'k}\bar{x}_1^i x_3^k\right) \forall(\bar{x}_1,\bar{x}_2)\in S^-,\end{aligned} \tag{8}$$

which can be rewritten as a MILP as previously discussed for the restricted case.

A strategy vector x_3 satisfying the previous constraints which is also optimal in the sense of iii) is found by solving the following MILP:

$$\max_{\substack{\eta \text{ free}\\ x_3\in\Delta_3}} \left\{ \begin{array}{ll} \eta : & \eta \leq \sum_{i\in A_1}\sum_{j\in A_2}\sum_{k\in A_3} U_3^{ijk}\bar{x}_1^i\bar{x}_2^j x_3^k \qquad \forall(\bar{x}_1,\bar{x}_2)\in S^+ \\ & \text{Constraints (7), (8)} \end{array} \right\}. \tag{9}$$

The initialization of the search tree and the solution of nodes with $S^+ = \emptyset$ can be carried out as for the restricted case. Differently from that case though, we cannot perform feasibility check in $O(m^2)$ by inspection. For it, we resort to solving one of the formulations we gave for O-SLMFNE with a given, fixed x_3, after changing the sign of the objective function into a minus (so to consider the pessimistic case).

We observe that, differently from the restricted case, the search tree might not be finite when mixed strategies are considered. This is due to fact that a game may contain uncountably many triples (x_1, x_2, x_3) where (x_1, x_2) is a NE in the followers' subgame. As a consequence, the branch-and-bound algorithm might not terminate. Note, though, that, by halting the method after any finite amount of iteration of computing time, one would nevertheless be able to obtain a lower and an upper bound to the problem.

Table 1 Results obtained when computing an O-SLMFNE with the three proposed MINLP formulations.

	MINLP–I					MINLP–II					MINLP–III				
m	Time	Gap	UB	LB	Sol	Time	Gap	UB	LB	Sol	Time	Gap	UB	LB	Sol
4	3600	46.5	100	53.5	80	3600	11.0	100	89.0	100	1	0.0	91.4	91.4	100
6	3600	93.2	100	6.8	10	3600	53.0	100	47.0	80	413	10.7	92.8	82.9	90
8	3600	91.0	100	9.0	10	3600	53.8	100	46.2	70	987	0.7	96.8	96.1	100
10	3600	99.0	100	1.0	0	3600	55.0	100	45.0	70	2147	22.1	99.0	77.1	80
12	3600	99.0	100	1.0	0	3600	64.7	100	35.3	60	3242	7.7	99.7	92.0	100
14	3600	99.0	100	1.0	0	3600	62.3	100	37.7	50	3240	11.8	99.8	88.1	100
16	3600	99.0	100	1.0	0	3600	92.3	100	7.7	10	3243	18.3	99.9	81.6	100
18	3600	99.0	100	1.0	0	3600	93.1	100	6.9	10	2887	7.1	99.9	92.8	100
20	3600	99.0	100	1.0	0	3600	79.2	100	20.8	30	3265	15.2	99.9	84.8	100
25	3600	99.0	100	1.0	0	3600	84.3	100	15.7	40	3600	15.7	100.0	84.3	100
30	3600	99.0	100	1.0	0	3600	99.0	100	1.0	0	3600	14.5	100.0	85.5	100
35	3600	99.0	100	1.0	0	3600	99.0	100	1.0	0	3600	16.7	100.0	83.3	100
40	3600	99.0	100	1.0	0	3600	99.0	100	1.0	0	3400	14.4	100.0	85.6	100
45	3600	99.0	100	1.0	0	3600	99.0	100	1.0	0	3600	22.9	100.0	77.1	90
Avg	3600	94.3	100	5.7	7	3600	74.6	100	25.4	37	2659	12.7	98.5	85.9	97

5 Computational results

We consider a testbed of instances constructed with GAMUT, a widely adopted suite of game generators [14], of class `Uniform Random Games`. We assume the same number of actions m for each agent, with $U_1, U_2, U_3 \in [1, 100]^{m\times m\times m}$, and construct 10 different instances per value of m. We experiment on games with $n = 3$ players, with $m = 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45$ actions. In the upcoming Tables 1 and 2, we report, for each value of m and on average over the 10 corresponding game instances, the following four values:

1. computing time (Time), also including instances for which the time limit is reached,
2. optimality gap (Gap, defined as $\frac{UB-LB}{UB}100\%$),
3. upper bound (UB),
4. lower bound (LB),

plus a fifth value which, rather than an average, reports, for a given value of m, the:

5. percentage of games for which a feasible solution is found (Sol).

The experiments are run on a UNIX machine with a total of 32 cores working at 2.3 GHz, equipped with 128 GB of RAM, within a time limit of 3600 seconds per game, on a single thread. We assume, throughout the section, that leader and followers are entitled to mixed strategies, both in the optimistic and pessimistic cases.

5.1 Optimistic case

We experiment with the three MINLP formulations we proposed in Section 3, solving them with the spatial branch-and-bound solver SCIP 3.2.1. The results are reported in Table 1.

The results confirm that reformulating the complementarity constraints via binary variables yields (as expected) a considerable improvement, reducing the gap from 94.3% (MINLP-I) to 74.6% (MINLP-II), on average. The reformulation with additional constraints carried out to obtain MINLP-III allows for a very substantial improvement, bringing the gap down to 12.7%, on average. Although the UB is not substantially improved by MINLP-III, reaching

Table 2 Results obtained when computing a P-SLMFNE with the proposed branch-and-bound algorithm, for $\epsilon = 0.1, 1, 5$.

	$\epsilon = 0.1$					$\epsilon = 1$					$\epsilon = 5$				
m	Time	Gap	UB	LB	Sol	Time	Gap	UB	LB	Sol	Time	Gap	UB	LB	Sol
4	2525	25.9	88.5	65.6	100	362	0.0	83.0	83.0	100	37	5.6	84.9	80.2	100
6	3600	41.6	89.9	52.5	100	3281	34.2	88.2	58.0	90	2247	11.5	85.7	75.8	100
8	3600	49.3	96.2	48.8	100	3600	42.7	95.4	54.6	100	3600	28.8	90.1	64.1	100
10	3600	56.9	98.3	42.4	100	3600	51.1	97.5	47.7	100	3600	46.1	94.9	51.1	100
12	3600	58.8	97.9	40.3	90	3600	60.0	97.3	38.9	80	3600	53.2	96.2	45.1	90
14	3600	58.1	98.2	41.1	100	3600	53.1	98.1	46.0	100	3600	52.6	97.6	46.3	100
16	3600	70.5	98.1	28.9	70	3600	65.5	97.7	33.7	80	3600	70.2	97.3	29.0	70
Avg	3446	51.6	95.3	45.7	94	3092	43.8	93.9	51.7	93	2898	38.3	92.4	56.0	94

an average of 98.5 (as opposed to 100 with both MINLP-I and MINLP-II), MINLP-III yields LBs of substantially better quality. MINLP-I yields an average LB of 5.7, with feasible solutions found only for 7% of the instances. MINLP-II achieves an average LB of 25.4, with feasible solutions found for 37% of the instances. With MINLP-III, we arrive at a much larger average LB of 85.9 (remember that, due to the way the games are constructed, the leader's utility is upper bounded by 100), with feasible solutions found in 97% of the cases.

5.2 Pessimistic case

For the experiments with our branch-and-bound algorithm, we use GUROBI 6.5.1 for the solution of the MILP Subproblem (9), while employing SCIP 3.2.1 for the solution of nodes (S^+, S^-) with $S^+ = \emptyset$ and to perform the feasibility check. The tree search procedure is implemented in Python. To select the next node to process, we adopt a "worst bound" policy. This leads to always exploring a sequence of nodes (S^+, S^-) with $S^- = \emptyset$ until a leaf node is reached, thus quickly obtaining a feasible solution. The results are reported in Table 2 for games with up to $m = 16$ actions, obtained with different values of ϵ, namely, $\epsilon = 0.1, 1, 5$.[5]

The table shows that the algorithm finds solutions of better quality for larger values of ϵ, in spite of the fact that, the larger ϵ, the poorer the solution should be, as a consequence of larger portions of the leader's feasible region Δ_3 being discarded after a branching operation. This result is due to the fact that, with a larger ϵ, fewer nodes are created and, thus, leaf nodes are reached much faster, resulting in more feasible solutions being found in the time limit. Note that, as the size of the games increases and the algorithm becomes less effective, the method seems to be less affected by the choice of ϵ.

From a gap of, on average, 51.6% obtained with $\epsilon = 0.1$, we achieve one of 43.8% with $\epsilon = 1$, and one of 38.3% with $\epsilon = 5$. While the UB is not much affected by ϵ, we register a larger improvement in the LB, which goes from, on average, 45.7 with $\epsilon = 0.1$ to 51.7 with $\epsilon = 1$ to 56.0 with $\epsilon = 5$. As the table shows, the problem becomes much harder to solve for larger values of m, with an average gap which, from the 65.5%-70.5% range for $m = 16$, considering the three values of ϵ, reaches 99% for $m = 18$ (not shown in the table).

Notwithstanding the problem being a nonconvex pessimistic bilevel program, we remark that the branch-and-bound algorithm we proposed manages to find, with $\epsilon = 5$, solutions in 70% of the cases with an average (approximate, due to ϵ) gap of ∼70% for games with

[5] We omit the results for larger values of m as, for them, the algorithm often fails to find a feasible solution.

$m = 16$ actions (4096 payoffs in each of the $n = 3$ matrices U_1, U_2, U_3). Such games are not much smaller, in terms of the number of outcomes, than those solved in papers concerned with the computation of an optimal NE in the single level case, see [15], which is a special case of the problem of computing a SLMFNE obtained when x_3 is restricted to a constant.

6 Concluding remarks

We have considered the problem of computing leader-follower equilibria in games with two or more followers, assuming that, after witnessing the leader's commitment to a mixed strategy, the followers play a mixed-strategy Nash equilibrium. We have proposed three mixed-integer nonconvex mathematical programming formulations for the optimistic case, and an *ad hoc* branch-and-bound method for the pessimistic one. Computational experiments have revealed that, with the last of the three formulations for the optimistic case, we can obtain average gaps smaller than 13% for games with up to 45 actions per player while, with our branch-and-bound algorithm for the pessimistic case, we obtain average (approximate) gaps below 70% for games with up to 16 actions. Future work includes the design of primal heuristics to be embedded in the branch-and-bound algorithm for the pessimistic problem, as well as the construction of dual bounds via the adoption of relaxed optimality conditions.

References

1 E. Amaldi, A. Capone, S. Coniglio, and L. G. Gianoli. Network optimization problems subject to max-min fair flow allocation. *IEEE COMMUN LETT*, 17(7):1463–1466, 2013.

2 B. An, J. Pita, E. Shieh, M. Tambe, C. Kiekintveld, and J. Marecki. Guards and Protect: Next generation applications of security games. *ACM SIGecom Exchanges*, 10(1):31–34, 2011.

3 E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *DISCRETE APPL MATH*, 89(1):3–44, 1998.

4 N. Basilico, S. Coniglio, and N. Gatti. Methods for Finding Leader-Follower Equilibria with Multiple Followers: (Extended Abstract). In *Proc. of AAMAS*, pages 1363–1364. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

5 A. Caprara, M. Carvalho, A. Lodi, and G.J. Woeginger. Bilevel knapsack with interdiction constraints. *INFORMS J COMPUT*, 28(2):319–333, 2016.

6 V. Conitzer and D. Korzhyk. Commitment to correlated strategies. In *Proc. of AAAI*, 2011.

7 V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC*, pages 82–90, 2006.

8 V. Conitzer and T. Sandholm. New complexity results about nash equilibria. *GAME ECON BEHAV*, 63(2):621–641, 2008.

9 C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *Proc. of AAMAS*, pages 689–696, 2009.

10 M. Labbé and A. Violin. Bilevel programming and price setting problems. *ANN OPER RES*, 240(1):141–169, 2016.

11 S. Leyffer and T. Munson. Solving multi-leader–common-follower games. *OPTIM METHOD SOFTW*, 25(4):601–623, 2010.

12 Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

13 J. Matuschke, S.T. McCormick, G. Oriolo, B. Peis, and M. Skutella. Protection of flows under targeted attacks. *OPER RES LETT*, 45(1):53–59, 2017.

14 E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proc. of AAMAS*, pages 880–887. IEEE Computer Society, 2004.

15 T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-Integer Programming Methods for Finding Nash Equilibria. In *Proc. of AAAI*, pages 495–501, 2005.

16 Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2008.

17 B. von Stengel and S. Zamir. Leadership games with convex strategy sets. *GAME ECON BEHAV*, 69:446–457, 2010.

18 A.B. Zemkoho. Solving ill-posed bilevel programs. *SET-VALUED ANAL*, 24(3):423–448, 2016.