A Tutorial on Solving Single-Leader-Multi-Follower Problems using SOS1 Reformulations

Didier Aussel, Cécile Egea, and Martin Schmidt

Abstract. In this tutorial we consider single-leader-multi-follower games in which the models of the lower-level players have polyhedral feasible sets and convex objective functions. This situation allows for classic KKT reformulations of the separate lower-level problems, which lead to challenging single-level reformulations of MPCC type. The main contribution of this tutorial is to present a ready-to-use reformulation of this MPCC using SOS1 conditions. These conditions are readily available in all modern MILP solvers that then solve the single-leader-multi-follower problem to optimality. After formally stating the problem class under consideration as well as deriving its reformulations, we present explicit Python code that shows how these techniques can be realized using the MILP solver Gurobi. Finally, we also show the effect of the SOS1-based reformulation using the real-world example of industrial eco-park modeling.

1. Introduction

For many practical applications it is necessary to model non-cooperative behavior of multiple agents, which usually leads to a Nash game (see Nash (1950)) or to its generalized version; see, e.g., Facchinei and Kanzow (2010). However, these game-theoretic models are only applicable in situations in which the competitive agents act simultaneously. If the agents are ordered in a hierarchical way, the resulting model is a so-called multi-leader-multi-follower (MLMF) game—a problem class that dates back to the seminal publications by von Stackelberg (1934, 1952). In such games, the set of agents is split in two groups, the *leaders* and the *followers*, both interacting in a non-cooperative way. The leaders, who usually represent the authorities or the most influencing agents, need to take into account the reactions of the followers, whose problems and, thus, decisions depend on those of the leaders. This setup is also often referred to as a bilevel game and the general solution concept on both levels is that of a (generalized) Nash equilibrium. In other words, the MLMF game is a (generalized) Nash game among Stackelberg leaders. Note that alternative approaches to Nash equilibrium interactions have been recently considered in Allevi et al. (2024) and Aussel and Chaipunya (2024). Obviously, these models are extremely challenging—both in theory and in practice. We refer to Aussel and Svensson (2020) for a recent overview. The easiest (but still challenging) instantiation of such models is a bilevel optimization problem, i.e., a single-leader-single-follower (SLSF) model. This field made an enormous progress in the last years and decades; see, e.g., the recent books by Dempe et al. (2015) and Dempe and Zemkoho (2020), the recent survey by Kleinert et al. (2021a) for SLSF games, Aussel and Svensson (2020) for single-leader-multi-follower (SLMF) games, and Aussel et al. (2021) and Aussel and Svensson (2018) for MLMF games.

In this tutorial we focus on the intermediate setting in which there is a single leader but multiple followers, i.e., we consider SLMF games. In particular, we

2020 *Mathematics Subject Classification.* 90-01, 90-08, 91A06, 91A10, 91A65.

Key words and phrases. Single-leader-multi-follower games, Single-level reformulations, SOS1, Industrial eco-parks.

present and explain a powerful technique to obtain a single-level reformulation of the SLMF game that uses so-called special-ordered-sets of type 1 (SOS1), which date back to Beale and Tomlin (1970) and which have been used for the first time in bilevel optimization in Fortuny-Amat and McCarl (1981). The key idea is to use the Karush–Kuhn–Tucker (KKT) conditions of the lower-level players and reformulate the nonlinear and nonconvex KKT complementarity conditions by using SOS1 conditions. This approach received increasing attention in the last years in bilevel optimization (Kleinert and Schmidt 2023) since it does not require to determine correct big-M values, which is needed if one re-writes the KKT complementarity conditions using additional binary variables; see, e.g., Kleinert et al. (2020) and Pineda and Morales (2019) for the drawbacks of the big-M approach. An alternative approach, proposed in Leyffer and Munson (2010), consists in a penalization method based on the complementarity conditions of the followers. The advantages of the SOS1 approach are the following:

(i) It is easy to implement. We will present Python code that shows how to implement the resulting single-level reformulation of a given SLMF game and then solve it with Gurobi.
(ii) The SOS1 functionality is widely available. Not only Gurobi, but also other solvers for mixed-integer optimization such as CPLEX or SCIP support this modeling technique. We choose one specific combination of programming language (Python) and solver (Gurobi) here to be as specific as possible. However, the mathematical concepts can, of course, also be implemented with other programming languages and solvers as well.
(iii) There is no need for computing big-M values or a penalization parameter.
(iv) In combination with other ready-to-use techniques from bilevel optimization, the approach is competitive with other classic techniques such as the big-M approach; see, e.g., Kleinert et al. (2021b) and Kleinert and Schmidt (2023).

The remainder of this tutorial is structured as follows. In Section 2, we formally present the problem statement and derive a first single-level reformulation based on the KKT conditions of the lower-level players. The SOS1-based reformulation of the single-level KKT reformulation is then briefly discussed in Section 3. This technique is then applied in Section 4 to two academic examples. Here, we also present Python code that shows how easy the implementation is. Afterward, in Section 5, we apply the SOS1 technique to a real-world SLMF game in the area of a circular industrial economy to show the applicability of the technique also for more realistic and larger instances. A comparison with an implementation of the Leyffer–Munson method on this application is also provided. Finally, we conclude in Section 6.

2. Problem Statement and Single-Level Reformulation

As announced in the introduction, our aim is to propose a reformulation scheme based on the SOS1 approach to solve SLMF games. These problems, in their optimistic version, are given by

$$\min_{x,y^1,\dots,y^N} \quad F(x,y) \tag{1a}$$
$$\text{s.t.} \quad G(x,y) \geq 0, \tag{1b}$$
$$y := (y^\nu)_{\nu\in[N]} \text{ solves } \mathrm{GNEP}(x), \tag{1c}$$

where $x \in \mathbb{R}^{n_0}$ is the vector of the leader's decisions and $y^\nu \in \mathbb{R}^{n_\nu}$ is the variable vector of follower $\nu \in [N] := \{1,\dots,N\}$. The vector $y = (y^\nu)_{\nu\in[N]} \in \mathbb{R}^{n_f}$ collects all decisions of all followers, i.e., $n_f = \sum_{\nu\in[N]} n_\nu$. Hence, we have $F : \mathbb{R}^{n_0} \times \mathbb{R}^{n_f} \to \mathbb{R}$ and $G : \mathbb{R}^{n_0} \times \mathbb{R}^{n_f} \to \mathbb{R}^{m_0}$, where m_0 is the number of upper-level constraints. In what follows, we use the classic notation $y^{-\nu} = (y^\mu)_{\mu\in[N]\setminus\{\nu\}}$ from game theory that collects all followers' decisions except for those of player ν. Let us comment a bit more in detail on two important aspects of Model (1). First, the optimistic version of the bilevel game is considered here, which is formalized by the fact that the upper-level player also optimizes over the lower-level variables y in case that there are any multiplicities in the lower-level's set of equilibria $\mathrm{GNEP}(x)$. For a more detailed discussion of the differences between the optimistic and pessimistic version of SLMF games, we refer to Aussel and Svensson (2020). Moreover, let us note that we allow for so-called coupling constraints G, which are upper-level constraints that explicitly depend on lower-level variables y. Furthermore, $\mathrm{GNEP}(x)$ stands for the set of generalized Nash equilibria of the non-cooperative game among the N followers, where the optimization problem of the νth player is given by

$$\min_{y^\nu} \quad f_\nu(y^\nu, x, y^{-\nu}) \tag{2a}$$

$$\text{s.t.} \quad D^\nu y^\nu \geq e - D^{\nu,0}x - \sum_{\mu \neq \nu} D^{\nu,\mu} y^\mu, \tag{2b}$$

$$E^0 x + \sum_{\mu=1}^{N} E^\mu y^\mu \geq g \tag{2c}$$

with $D^\nu \in \mathbb{R}^{m_\nu \times n_\nu}$, $D^{\nu,0} \in \mathbb{R}^{m_\nu \times n_0}$, $D^{\nu,\mu} \in \mathbb{R}^{m_\nu \times n_\mu}$ for all $\mu \neq \nu$, and $e \in \mathbb{R}^{m_\nu}$. Moreover, we have $E^0 \in \mathbb{R}^{m \times n_0}$, $E^\nu \in \mathbb{R}^{m \times n_\nu}$, and $g \in \mathbb{R}^m$. The problems (2) define a GNEP because the feasible set of every player explicitly depends on the decisions on the other players. This is different to classic Nash games in which only the objective functions depend on the other players' decisions. However, let us remark that the approach proposed in this tutorial is also applicable to NEPs in the lower level as well. In Problem (2), (2b) is the private constraint of player ν, which may depend on the leader's decision x and the decisions of all other lower-level players $\mu \in [N] \setminus \{\nu\}$, and (2c) is the shared constraint of the GNEP at the lower level. For later reference, we denote the feasible set of the νth lower-level player by $\Omega_\nu(x, y^{-\nu})$. In this spirit, we also define $\Omega_{-\nu} = \prod_{\mu \in [N]\setminus\{\nu\}} \Omega_\mu(x, y^{-\mu})$. The so-called shared constraint set is then given by

$$\Omega := \left\{(x, y) \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_f} : G(x, y) \geq 0 \text{ and } y^\nu \in \Omega_\nu(x, y^{-\nu}) \text{ for all } \nu \in [N]\right\}.$$

Its projection onto the space of the leader variables is denoted by

$$\Omega_x := \{x \in \mathbb{R}^{n_0} : \exists y \text{ with } (x, y) \in \Omega\}.$$

The class of SLMF games we tackle is quite large since the only restrictions are the following:

- The functions F and G are continuous on $\mathbb{R}^{n_0} \times \mathbb{R}^{n_f}$.
- For any $\nu \in [N]$, any $x \in \Omega_x$, and any $y^{-\nu} \in \Omega_{-\nu}$, the function $f(\cdot, x, y^{-\nu})$ is convex and continuously differentiable on $\mathbb{R}^{n_\nu}$.
- The feasible set of each follower is polyhedral, i.e., it is defined by affine-linear functions.

For any follower $\nu \in [N]$, the corresponding KKT conditions are given by

$$\nabla_\nu f_\nu(y^\nu, x, y^{-\nu}) - (D^\nu)^\top \lambda^\nu - (E^\nu)^\top \delta^\nu = 0, \tag{3a}$$

$$D^\nu y^\nu - e + D^{\nu,0} x + \sum_{\mu \neq \nu} D^{\nu,\mu} y^\mu \geq 0, \tag{3b}$$

$$E^0 x + \sum_{\mu=1}^{N} E^\mu y^\mu - g \geq 0, \tag{3c}$$

$$\lambda^\nu, \delta^\nu \geq 0, \tag{3d}$$

$$(\lambda^\nu)^\top \left(D^\nu y^\nu - e + D^{\nu,0} x + \sum_{\mu \neq \nu} D^{\nu,\mu} y^\mu \right) = 0, \tag{3e}$$

$$(\delta^\nu)^\top \left(E^0 x + \sum_{\mu=1}^{N} E^\mu y^\mu - g \right) = 0. \tag{3f}$$

The KKT conditions of the lower-level problems are both necessary and sufficient for all players. In particular, we do not need any further constraint qualifications for the lower-level problems since the constraint sets are all polyhedral. Moreover, we assume that all linear problems are stated so that the linear independence constraint qualification (LICQ) holds. In the linearly constrained case considered here, this means that the row vectors of all active constraints are linearly independent. This can can be ensured a priori in the linear case by, w.l.o.g., assuming that the respective constraint matrices have full row rank; see, e.g., the seminal textbooks by Nocedal and Wright (2006) or Bertsekas (2016) for an introduction to constraint qualifications. As a further consequence of the LICQ, all Lagrangian multipliers of all KKT points are uniquely determined.

Remark 1. *Taking a closer look at Constraint* (3f) *reveals that every lower-level player* ν *has an own dual variable* δ^ν *for the shared primal constraint. For many applications, this is a problem since the dual variables of shared constraints often define prices, which leads to economic ambiguities if every player sees a different price. As a remedy, Rosen (1965) introduced the concept of normalized or variational equilibria of a GNEP in which all the dual variables of the shared constraint need to be the same—hence leading to a well-defined price in the respective economic applications. Here, we are interested in variational equilibria of the GNEP in the lower-level problem and we thus need to impose* $\delta^\nu = \delta$ *for all* $\nu \in [N]$*. Hence,* (3f) *needs to be replaced by*

$$\delta^\top \left(E^0 x + \sum_{\mu=1}^{N} E^\mu y^\mu - f \right) = 0.$$

Replacing the GNEP in the lower level by the concatenation of all KKT conditions of the lower-level players, we obtain the single-level reformulation

$$
\begin{aligned}
\min_{x,y,\lambda,\delta} \quad & F(x,y) && \text{(4a)}\\
\text{s.t.} \quad & G(x,y) \le 0, && \text{(4b)}\\
& \nabla_\nu f_\nu(y^\nu, x, y^{-\nu}) - (D^\nu)^\top \lambda^\nu - (E^\nu)^\top \delta^\nu = 0, \quad \nu \in [N], && \text{(4c)}\\
& D^\nu y^\nu - e + D^{\nu,0} x + \sum_{\mu \ne \nu} D^{\nu,\mu} y^\mu \ge 0, \quad \nu \in [N], && \text{(4d)}\\
& E^0 x + \sum_{\mu=1}^{N} E^\mu y^\mu - g \ge 0, && \text{(4e)}\\
& \lambda^\nu, \delta^\nu \ge 0, \quad \nu \in [N], && \text{(4f)}\\
& (\lambda^\nu)^\top \left(D^\nu y^\nu - e + D^{\nu,0} x + \sum_{\mu \ne \nu} A^{\nu,\mu} y^\mu \right) = 0, \quad \nu \in [N], && \text{(4g)}\\
& \delta^\top \left(E^0 x - \sum_{\mu=1}^{N} E^\mu y^\mu - g \right) = 0. && \text{(4h)}
\end{aligned}
$$

Here and in what follows, we use the abbreviations $y := (y^\nu)_{\nu\in[N]}$, $\lambda := (\lambda^\nu)_{\nu\in[N]}$, and $\delta := (\delta^\nu)_{\nu\in[N]}$.

Note that the single-level reformulation leads to an optimistic solution of the single-level-multi-follower problem.

Theorem 2. *Let (x^*, y^*) be a global optimal solution of the single-leader-multi-follower game (1). Then, there exists λ^* and δ^* so that $(x^*, y^*, \lambda^*, \delta^*)$ is a global optimal solution of the single-level reformulation (4).*

On the other hand, if $(x^*, y^*, \lambda^*, \delta^*)$ is a global optimal solution of the single-level reformulation (4), then (x^*, y^*) is a global optimal solution of the single-leader multi-follower game (1).

The proof is straightforward or can be deduced from Theorem 3.3.8 in Aussel and Svensson (2020).

Remark 3. *We finally discuss some potential generalizations of the above setting.*

(1) *The setup and the main result can be generalized to convex instead of polyhedral feasible sets if Slater's constraint qualification is satisfied since the corresponding KKT conditions of the followers are still necessary and sufficient. However, in the light of the results by Aussel and Svensson (2019) and Dempe and Dutta (2012), one needs to be careful with the respective Lagrangian multipliers to actually obtain the correctness of Theorem 2.*

(2) *The constraints of the lower-level problems would stay linear if one allows for multilinear terms. Hence, the classic KKT reformulation can still be applied. However, the resulting single-level reformulation would then inherit these multilinear terms. Since one optimizes over all follower variables in the single-level reformulation, this would then lead to nonconvex nonlinearities. The analogue applies to the lower-level objective functions, where we, however, loose one degree of the multilinear polynomial by taking the respective gradient in the KKT conditions.*

3. SOS1-Based Reformulation

The main burden in Problem (4) are the two complementarity constraints (4g) and (4h). These two nonlinear and nonconvex constraints can be modeled using

SOS1-type constraints and can thus be solved using Gurobi or CPLEX to global optimality without choosing any big-Ms; see, e.g., Kleinert and Schmidt (2023).

To this end, we introduce auxiliary variables $s^\nu_{i_\nu} \geq 0$ for all $\nu \in [N]$ and all $i_\nu \in [m_\nu]$ and $t_j \geq 0$ for all $j \in [m]$ as well as the following SOS1 conditions:

$$\lambda^\nu_{i_\nu} \text{ and } s^\nu_{i_\nu} = \left(D^\nu y^\nu - e + D^{\nu,0} x + \sum_{\mu \neq \nu} A^{\nu,\mu} y^\mu \right)_{i_\nu} \quad \text{are SOS1}$$

for all $\nu \in [N]$ and $i_\nu \in [m_\nu]$ as well as

$$\delta_j \text{ and } t_j = \left(E^0 x - \sum_{\nu=1}^{N} E^\nu y^\nu - g \right)_j \quad \text{are SOS1}$$

for all $j \in [m]$.

Let us recall that a SOS1-type constraint defines a set of variables for which at most one variable in the set may take a value other than zero.

Remark 4. *If for some of these components, we have good (and, of course, provably correct) upper bounds ("big-Ms") for the dual variables $\lambda^\nu_{i_\nu}$ and δ_j as well as for the corresponding primal expressions, we can also use the classic big-M-like reformulation for these components.*

4. Academic Examples

Our aim in this section is to show, via two simple examples, how simple the implementation of the SOS1 approach is for the computation of solutions of single-leader-multi-follower games. While the first example is completely linear, the second one also contains nonlinearities. In the spirit of a tutorial and for each example, the single-level/MPCC reformulation of the single-leader-multi-follower game will be given, then the Python code corresponding to the numerical resolution of the MPCC will be fully described, and the optimization results will be presented.

4.1. **Example #1.** Let us start with a very simple and completely linear example of a single-leader-multi-follower game. Three agents are here considered here. Thus, we have one leader and two followers. The leader's problem is given by

$$\begin{aligned} \min_{x, y_1, y_2, y_3} \quad & 2x + y_1 + y_2 - y_3 \\ \text{s.t.} \quad & x \geq 1, \\ & y \text{ solves GNEP}(x), \end{aligned}$$

where the GNEP in the lower level consists of two players—the first follower has a 2-dimensional optimization problem while the second follower solves the 1-dimensional problem:

Player 1		Player 2	
$\min_{y_1,y_2}$	$x - 2y_1 - y_2$	$\min_{y_3}$	$x + y_1 + y_2 + y_3$
s.t.	$y_1 \geq y_2 + y_3$	s.t.	$y_3 \geq x$
	$y_1 \leq x$		

This bilevel problem admits only one optimal solution $(\bar{x}, \bar{y}_1, \bar{y}_2, \bar{y}_3) = (1, 1, 0, 1)$. The corresponding KKT conditions of the first and second follower are given as follows:

KKTs of Player 1	KKTs of Player 2
$-2 - \lambda_1^1 + \lambda_2^1 = 0$	$1 - \lambda_1^2 = 0$
$-1 + \lambda_1^1 = 0$	$y_3 - x \geq 0$
$y_1 - y_2 - y_3 \geq 0$	$\lambda_1^2(y_3 - x) = 0$
$x - y_1 \geq 0$	$\lambda_1^2 \geq 0$
$\lambda_1^1(y_1 - y_2 - y_3) = 0$	
$\lambda_2^1(x - y_1) = 0$	
$\lambda_1^1, \lambda_2^1 \geq 0$	

Thanks to Theorem 2, the single-level reformulation is equivalent to SLMF game (4.1) and reads

$$\begin{aligned}
\min_{x,y_1,y_2,y_3} \quad & 2x + y_1 + y_2 - y_3 \\
\text{s.t.} \quad & x \geq 1, \\
& -2 - \lambda_1^1 + \lambda_2^1 = 0, \\
& -1 + \lambda_1^1 = 0, \\
& y_1 - y_2 - y_3 \geq 0, \\
& x - y_1 \geq 0, \\
& \lambda_1^1(y_1 - y_2 - y_3) = 0, \\
& \lambda_2^1(x - y_1) = 0, \\
& \lambda_1^1, \lambda_2^1 \geq 0, \\
& 1 - \lambda_1^2 = 0, \\
& y_3 - x \geq 0, \\
& \lambda_1^2(y_3 - x) = 0, \\
& \lambda_1^2 \geq 0.
\end{aligned}$$

Finally, if we use the non-negative slack variables

$$s_1^1 = y_1 - y_2 - y_3, \quad s_2^1 = x - y_1, \quad s_1^2 = y_3 - x,$$

we obtain the SOS1-based single-level reformulation

$$\begin{aligned}
\min_{x,y,\lambda,s} \quad & 2x + y_1 + y_2 - y_3 \\
\text{s.t.} \quad & x \geq 1, \\
& -2 - \lambda_1^1 + \lambda_2^1 = 0, \\
& -1 + \lambda_1^1 = 0, \\
& y_1 - y_2 - y_3 \geq 0, \\
& x - y_1 \geq 0, \\
& \text{SOS1}(\lambda_1^1, s_1^1), \\
& \text{SOS1}(\lambda_2^1, s_2^1), \\
& s_1^1 = y_1 - y_2 - y_3, \quad s_1^1 \geq 0, \\
& s_2^1 = x - y_1, \quad s_2^1 \geq 0, \\
& \lambda_1^1, \lambda_2^1 \geq 0, \\
& 1 - \lambda_1^2 = 0, \\
& y_3 - x \geq 0, \\
& \text{SOS1}(\lambda_1^2, s_1^2),
\end{aligned}$$

$$s_1^2 = y_3 - x, \quad s_1^2 \geq 0,$$
$$\lambda_1^2 \geq 0$$

with $y = (y_1, y_2, y_3)^\top$, $\lambda = (\lambda_1^1, \lambda_2^1, \lambda_1^2)^\top$, and $s = (s_1^1, s_2^1, s_1^2)^\top$.

Using the Python interface of Gurobi, this model can be implemented as follows:

```
# Import the Gurobi interface
from gurobipy import *

# Create an empty model
model = Model("example-1")

# Build all variables
x = model.addVar(name="x", lb=1)
y1 = model.addVar(name="y1", lb=-GRB.INFINITY, ub=GRB.INFINITY)
y2 = model.addVar(name="y2", lb=-GRB.INFINITY, ub=GRB.INFINITY)
y3 = model.addVar(name="y3", lb=-GRB.INFINITY, ub=GRB.INFINITY)
lambda11 = model.addVar(name="lambda11")
lambda12 = model.addVar(name="lambda12")
lambda21 = model.addVar(name="lambda21")
s11 = model.addVar(name="s11")
s12 = model.addVar(name="s12")
s21 = model.addVar(name="s21")

# Build the upper-level objective function
model.setObjective(2*x + y1 + y2 - y3, GRB.MINIMIZE)

# Add lower-level dual feasibility constraints
model.addConstr(-2 - lambda11 + lambda12 == 0)
model.addConstr(-1 + lambda11 == 0)
model.addConstr(1 - lambda21 == 0)

# Add lower-level primal feasibility constraints
model.addConstr(y1 - y2 - y3 >= 0)
model.addConstr(x - y1 >= 0)
model.addConstr(y3 - x >= 0)

# Add slack variable defining constraints
model.addConstr(s11 == y1 - y2 - y3)
model.addConstr(s12 == x - y1)
model.addConstr(s21 == y3 - x)

# Add SOS1 conditions
model.addSOS(GRB.SOS_TYPE1, [lambda11, s11])
model.addSOS(GRB.SOS_TYPE1, [lambda12, s12])
model.addSOS(GRB.SOS_TYPE1, [lambda21, s21])

# Solve it
model.optimize()
```

After solving the model, we can get the solution values via

```
# Print the solution
print("       x = " + str(x.X))
print("      y1 = " + str(y1.X))
print("      y2 = " + str(y2.X))
print("      y3 = " + str(y3.X))
print("lambda11 = " + str(lambda11.X))
print("lambda12 = " + str(lambda12.X))
print("lambda21 = " + str(lambda21.X))
print("     s11 = " + str(s11.X))
print("     s12 = " + str(s12.X))
print("     s21 = " + str(s21.X))
```

and obtain

```
       x = 1.0
      y1 = 1.0
      y2 = 0.0
      y3 = 1.0
lambda11 = 1.0
lambda12 = 3.0
lambda21 = 1.0
     s11 = 0.0
     s12 = 0.0
     s21 = 0.0
```

4.2. **Example #2.** The second academic example also involves one leader and two followers but the objective function of one of the followers is nonlinear now. Here, the leader's problem is given by

$$\begin{aligned} \min_{x,y_1,y_2} \quad & -2x + y_1 + 3y_2 \\ \text{s.t.} \quad & x \geq -1, \\ & x \leq 1 \\ & y \text{ solves } \mathrm{GNEP}(x), \end{aligned}$$

where the GNEP in the lower level consists of two players, each of them solving a 1-dimensional problem:

Player 1		Player 2	
$\min_{y_1}$	$y_1 + 2y_2 + 10$	$\min_{y_2}$	$y_1 y_2$
s.t.	$y_1 + y_2 \geq x$	s.t.	$y_2 \geq 0$

One can easily see that the best-response functions of the followers, i.e., the optimal solutions of each follower given the leader's decision as well as the decisions of all other followers, are given by

$$R_1(x, y_2) = \{x - y_2\}, \quad R_2(x, y_1) = \begin{cases} \mathbb{R}^+, & \text{if } y_1 = 0, \\ \{0\}, & \text{if } y_1 > 0, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Since $(y_1, y_2) \in \mathrm{GNEP}(x)$ is equivalent to $y_1 \in R_1(x, y_2)$ and $y_2 \in R_2(x, y_1)$, we get

$$\mathrm{GNEP}(x) = \begin{cases} \{(x, 0), (0, x)\}, & \text{if } x > 0, \\ \{(0, 0)\}, & \text{if } x = 0. \end{cases}$$

Thus, the SLMF game (4.2) admits only one optimal (optimistic) solution, which is given by $(x, y_1, y_2) = (1, 1, 0)$.

In order to describe the applications of the SOS1 approach on this example, let us describe first the KKT conditions of both followers:

KKTs of Player 1	KKTs of Player 2
$1 - \lambda_1^1 = 0$	$y_1 - \lambda_1^2 = 0$
$y_1 + y_2 - x \geq 0$	$y_2 \geq 0$
$\lambda_1^1 \geq 0$	$\lambda_1^2 \geq 0$
$\lambda_1^1(y_1 + y_2 - x) = 0$	$\lambda_1^2 y_2 = 0$

For this example, we only need an auxiliary slack variable for the first leader:

$$s_1^1 = y_1 + y_2 - x, \quad s_1^1 \geq 0.$$

With this, we obtain the single-level reformulation

$$\begin{aligned} \min_{x,y,\lambda,s} \quad & -2x + y_1 + 3y_2 \\ \text{s.t.} \quad & x \geq -1, \\ & x \leq 1, \\ & 1 - \lambda_1^1 = 0, \\ & y_1 + y_2 - x \geq 0, \\ & \lambda_1^1 \geq 0, \\ & s_1^1 = y_1 + y_2 - x, \quad s_1^1 \geq 0, \\ & \text{SOS1}(\lambda_1^1, s_1^1), \\ & y_1 - \lambda_1^2 = 0, \\ & y_2 \geq 0, \\ & \lambda_1^2 \geq 0, \\ & \text{SOS1}(\lambda_1^2, y_2) \end{aligned}$$

with $y = (y_1, y_2)^\top$, $\lambda = (\lambda_1^1, \lambda_1^2)^\top$, and $s = s_1^1$.

The required Python code for solving the single-level reformulation with Gurobi reads

```
# Import the Gurobi interface
from gurobipy import *

# Create an empty model
model = Model("example-2")

# Build all variables
x = model.addVar(name="x", lb=-1, ub=1)
y1 = model.addVar(name="y1", lb=-GRB.INFINITY, ub=GRB.INFINITY)
y2 = model.addVar(name="y2")
lambda11 = model.addVar(name="lambda11")
lambda21 = model.addVar(name="lambda21")
s11 = model.addVar(name="s11")

# Build the upper-level objective function
model.setObjective(-2*x + y1 + 3*y2, GRB.MINIMIZE)
```

```
    # Add lower-level dual feasibility constraints
    model.addConstr(1 - lambda11 == 0)
    model.addConstr(y1 - lambda21 == 0)

    # Add lower-level primal feasibility constraints
    model.addConstr(y1 + y2 - x >= 0)

    # Add slack variable defining constraints
    model.addConstr(s11 == y1 + y2 - x)

    # Add SOS1 conditions
    model.addSOS(GRB.SOS_TYPE1, [lambda11, s11])
    model.addSOS(GRB.SOS_TYPE1, [lambda21, y2])

    # Solve it
    model.optimize()
```

and printing the solution via

```
    # Print the solution
    print("       x = " + str(x.X))
    print("      y1 = " + str(y1.X))
    print("      y2 = " + str(y2.X))
    print("lambda11 = " + str(lambda11.X))
    print("lambda21 = " + str(lambda21.X))
    print("     s11 = " + str(s11.X))
```

leads to

```
       x = 1.0
      y1 = 1.0
      y2 = 0.0
lambda11 = 1.0
lambda21 = 1.0
     s11 = 0.0
```

5. A Real-World Application: Industrial Eco-parks

Our aim in this section is to apply the above described SOS1 approach to a reasonably large application case and to compare the obtained results with an alternative method, namely the penalization approach proposed by Leyffer and Munson (2010). An interesting example of such an application is the problem of optimally designing industrial eco-parks (IEP). An industrial eco-park consists of a network of plants of different companies built in order to share one or more resources. More precisely, when companies are located close to each other, the IEP structure aims to explore the possibility for the companies to use waste resources coming from the other companies for their plant processes. An exemplary implementation of such an IEP is the industrial park of Kalendburg in Denmark in which companies are sharing water, vapor, ammonia, and more than 30 other different resources. The main targets for implementing an IEP are, on the one hand, to attract the participation of companies to the IEP by reducing the production cost for each of them and, on the other hand, to reduce the total needs of resources of the industrial park. Thus, in the concept of industrial eco-parks the term “eco” indicates

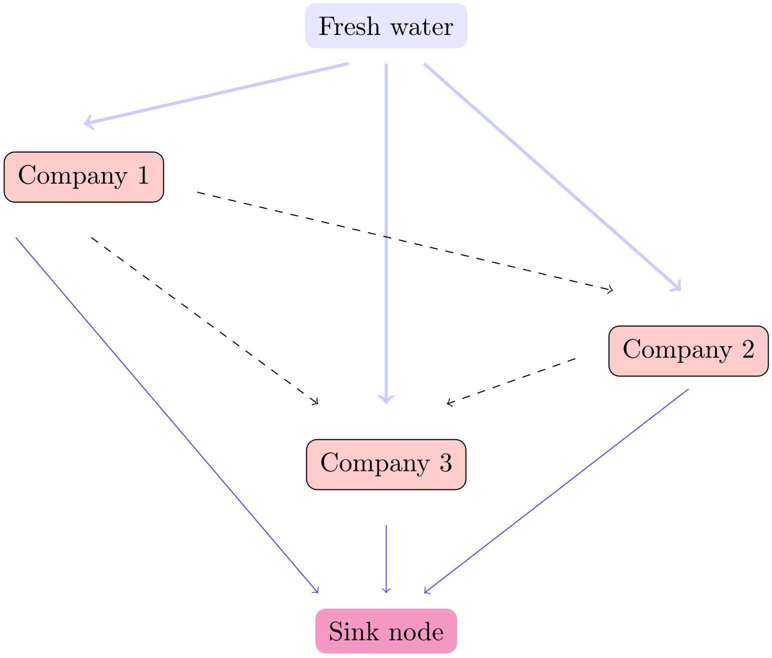

FIGURE 1. Example of a connection graph of an IEP with three companies/processes

both economic and ecological concerns. For many years, the optimal design of IEPs has been done by using multi-objective optimization (Ehrgott 2005), where the components of the vectorial objective function are the cost functions of the companies and a function evaluating the total ecological impact of the production; see, e.g., Boix et al. (2015, 2012). More recently, in Aussel et al. (2023), Ramos et al. (2016), and Salas et al. (2020), an alternative approach has been proposed. It is a game-theoretical approach based on a single-leader-multi-follower model in which the followers are the companies that aim to minimize their cost and the unique leader is the designer/manager of the eco-park, who wants to determine the optimal implementation of an IEP, i.e., the interconnections and fluxes between companies, which allow to better reduce the ecological impact. Industrial eco-parks are well adapted examples for the purpose of the comparative analysis of this section because (i) the constraints of the companies (the followers) are naturally expressed by affine-linear functions if so-called regeneration units are not considered and (ii) because the size of the problem is quite large as soon as a realistic number of companies/processes is involved.

In order to simplify the presentation of the example and to focus on its numerical treatment, the only resource which is exchanged in our example is water. Hence, by implementing an industrial eco-park, the leader creates a graph of water connections in which the companies/processes are the vertices and the edges are the connecting pipes. Water is bought by the companies at a "fresh water node" and finally rejected (also subject to payment) to the sink node, which is where the water that will not be used anymore leaves the system. Whenever one company uses the waste water of another one, then an additional connection is implemented between both companies and the associated transfer cost is equally shared by both companies.

After modeling the problem of optimally designing an industrial eco-park in Subsection 5.1 and describing the associated MPCC reformulation in Subsection 5.2,

TABLE 1. Meaning and unit of all constants

Symbol	Meaning	Unit
M_i	Contaminant load of process i	g/h
β	Polluted water discharge cost	\$/ton
δ	Polluted water pumping cost	\$/ton
$C_{i,\text{out}}$	Maximum contaminant concentration allowed in outlet of processes i	ppm
$C_{i,\text{in}}$	Maximum contaminant concentration allowed in inlet of processes i	ppm

we present and compare the implementation of the SOS1 and the Leyffer–Munson approach in Subsections 5.3 and 5.4.

5.1. **Modeling.** We use the notation from Salas et al. (2020) to model the optimal design problem of an IEP in which companies only exchange water. The model is an SLMF game, with the leader being in charge of the design as well as of the regulatory structure (e.g., the state), and the followers are the companies. Each company is assumed to handle only one process but the modeling can be easily extended to several processes per company. The leader minimizes the total use of fresh resources while the companies aim to minimize their production cost. Note that their production level is assumed to be fixed. Moreover, we define I_P as the set of all companies and I as the union of the companies and the sink node.

Further following Salas et al. (2020), each company controls the amount of polluted water it will send to the other companies or to the sink node of the water network. Recall that the production level of each company is assumed to be constant. Thus, the model is built for a single hour. For any $(i, j) \in I_P^2$, let us denote by

- $F_{i,j}$ the water flux going from company i to company j and by
- $F_{i,0}$ the water flux going from company i to sink node.

Thus, the variable of company i is the vector $F_{i,\boldsymbol{\cdot}}$ composed of all the fluxes exiting from company i. Now, the manager of the IEP, i.e., the leader, decides about the implementation of connections between companies and this decision is represented by the binary matrix y given by

$$y_{i,j} = \begin{cases} 1, & \text{the implementation of a pipe between } i \text{ and } j \text{ is decided,} \\ 0, & \text{otherwise,} \end{cases}$$

for all $(i, j) \in I_P^2$. Note that we will later always set $y_{i,i} = 0$ for all $i \in I_P$.

The optimization problem of each company $i \in I_P$ can then be expressed as a parameterized linear problem in which the parameters are the binary design variables y of the leader and the fluxes $F_{-i,\boldsymbol{\cdot}}$ of the other companies. The meaning and the unit of each of the constant terms used in the following model that are not explained explicitly in the text are given in Table 1.

Formally, the problem of company i reads

$$\begin{aligned}
\min_{F_{i,\bullet}} \quad & \frac{c\, M_i}{C_{i,\text{out}}} + \sum_{k \in I_P} \left[\left(\frac{c\, C_{k,\text{out}}}{C_{i,\text{out}}} - c + \delta \right) F_{k,i} + \delta F_{i,k} \right] + \beta F_{i,0} \\
\text{s.t.} \quad & F_{i,j} \geq 0, \quad j \in I, \\
& K y_{i,j} - F_{i,j} \geq 0, \quad j \in I, \\
& M_i + \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} - C_{i,\text{out}} \sum_{j \in I} F_{i,j} = 0, \\
& C_{i,\text{in}} \sum_{j \in I} F_{i,j} - \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} \geq 0, \\
& M_i - \sum_{k \in I_P} (C_{i,\text{out}} - C_{k,\text{out}}) F_{k,i} \geq 0.
\end{aligned}$$

Here and in what follows, c is the cost of fresh water. Note that the objective functions of the followers as well as all their constraints are linear. Moreover, note that in the variables defined above, the amount of fresh water reaching each process/company is not given. Indeed, as observed by Salas et al. (2020), taking into account the contaminant mass balance and since the production level is fixed, the amount of fresh water needed for company i can be described as a function of the other variables:

$$z_i(F_{-i}) = \frac{1}{C_{i,\text{out}}} \left(M_i + \sum_{k \in I_P} (C_{k,\text{out}} - C_{i,\text{out}}) F_{k,i} \right). \tag{5}$$

For more details on the formulas used, we refer the interested reader to Ramos et al. (2016) and Salas et al. (2020). However, let us explain a bit more that the constant value K is chosen in such a way that $F_{i,j}$ is forced to be zero if $y_{i,j}$ is zero, and such that it guarantees the non-negativity of $F_{i,j}$ otherwise. Thus K has to be larger than all of the feasible values of $F_{i,j}$ and a natural choice is

$$K = \sum_{i \in I_P} \frac{M_i}{C_{i,\text{out}}},$$

which corresponds to the sum of the natural resources used by companies if none of them participates in the eco-park.

The leader aims to design the eco-park, i.e., the leader decides about the topology of the IEP network in order to minimize the total amount of fresh water used by the companies, which is a proper measure of the overall ecological impact. This leads to the following optimistic SLMF game

$$\begin{aligned}
\min_{y,F} \quad & \sum_{i \in I_P} \frac{1}{C_{i,\text{out}}} \left(M_i + \sum_{k \in I_P} (C_{k,\text{out}} - C_{i,\text{out}}) F_{k,i} \right) \\
\text{s.t.} \quad & \beta \left(F_{i,0} - \frac{M_i}{C_{i,\text{out}}} \right) + \sum_{k \in I_P} \left[\left(\delta - c + \frac{c\, C_{k,\text{out}}}{C_{i,\text{out}}} \right) F_{k,i} + \delta\, F_{i,k} \right] \leq 0, \quad i \in I_P, \\
& y_{i,i} = 0, \quad i \in I_P, \\
& F \in \text{GNEP}(y).
\end{aligned}$$

The leader only has to guarantee that the production cost of each company will not increase when participating to the IEP, i.e., the resulting production cost in case the IEP is implemented is not larger than the one the company would face in a "stand-alone situation", i.e., if not participating in the eco-park. Here, as above, GNEP(y) stands for the set of generalized Nash equilibria between the companies acting as followers in the lower level. This set is non-empty for at least one binary design matrix y—namely the one corresponding to the stand-alone situation, which is given $y_{i,j} = 0$ for all $(i,j) \in I_P^2$.

5.2. **MPCC Reformulation.** As already explained in Section 2, the first step to solve an SLMF game is to build a reformulation as an MPCC by replacing the computation of a parameterized Nash equilibrium between the followers by the computation of solutions to the associated and concatenated Karush–Kuhn–Tucker conditions. For the case of the IEP, let

$$\Gamma = \begin{pmatrix} (\eta_{i,j})_{j \in I, i \in I_P} \\ (\theta_{i,j})_{j \in I, i \in I_P} \\ (\kappa_i)_{i \in I_P} \\ (\lambda_i)_{i \in I_P} \\ (\mu_i)_{i \in I_P} \end{pmatrix}$$

be the vector of the Lagrangian multipliers. With this notation at hand, the MPCC reformulation is thus given by

$$\begin{aligned}
\min_{y,F,\Gamma} \quad & \sum_{i \in I_P} z_i(F_{-i}) \\
\text{s.t.} \quad & \beta(F_{i,0} - \frac{M_i}{C_{i,\text{out}}}) + \sum_{k \in I_P} \left[\left(\delta - c + \frac{cC_{k,\text{out}}}{C_{i,\text{out}}} \right) F_{k,i} + \delta F_{i,k} \right] \leq 0, \quad i \in I_P, \\
& y_{i,i} = 0, \quad i \in I_P, \\
& (F, \Gamma) \in (\text{KKT})_i, \quad i \in I_P,
\end{aligned}$$

where, for any $i \in I_P$, the set $(\text{KKT})_i$ is given by all points satisfying

$$\begin{aligned}
\delta - \eta_{i,j} + \theta_{i,j} - \kappa_i C_{i,\text{out}} - \lambda_i C_{i,\text{in}} &= 0, \quad j \in I_P, \\
\beta - \eta_{i,0} + \theta_{i,0} - \kappa_i C_{i,\text{out}} - \lambda_i C_{i,\text{in}} &= 0, \\
\eta_{i,j} F_{i,j} &= 0, \quad j \in I, \\
\theta_{i,j}(K y_{i,j} - F_{i,j}) &= 0, \quad j \in I, \\
\lambda_i \left(C_{i,\text{in}} \sum_{j \in I} F_{i,j} - \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} \right) &= 0, \\
\mu_i \left(M_i - \sum_{k \in I_P} (C_{i,\text{out}} - C_{k,\text{out}}) F_{k,i} \right) &= 0, \\
\eta_{i,j} &\geq 0, \quad j \in I, \\
\theta_{i,j} &\geq 0, \quad j \in I, \\
\lambda_i &\geq 0, \\
\mu_i &\geq 0, \\
F_{i,j} &\geq 0, \quad j \in I, \\
K y_{i,j} - F_{i,j} &\geq 0, \quad j \in I, \\
M_i + \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} - C_{i,\text{out}} \sum_{j \in I} F_{i,j} &= 0, \\
C_{i,\text{in}} \sum_{j \in I} F_{i,j} - \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} &\geq 0, \\
M_i - \sum_{k \in I_P} (C_{i,\text{out}} - C_{k,\text{out}}) F_{k,i} &\geq 0.
\end{aligned}$$

Our aim in the forthcoming subsections 5.3 and 5.4 is to see how the SOS1 as well as the Leyffer–Munson approach can be compared on a reasonably large application case of designing an IEP.

The test IEP used in these two subsections is composed of 15 companies and the values of the constant terms of the model are given in Appendix A. All models have been implemented using Python 3.10.9 and have been solved using Gurobi version 10.0.2 on a DELL 5310 Latitude with an i5 Intel Core CPU with 16 GB RAM, and a Intel(R) UHD Graphics GPU.

5.3. **The SOS1 Approach.** We now reformulate the MPCC to get rid of the nonlinear and nonconvex KKT complementarity constraints. The idea is, as shown in Section 3, to re-write them using SOS1 constraints. Thus, for each inequality constraints $g_{i,k}(y, F) \geq 0$ of the optimization problem of company i, one introduces a new real-valued variable $s_{i,k}$ and the additional constraint

$$s_{i,k} = g_{i,k}(y, F).$$

Moreover, the corresponding KKT complementarity constraint $\mu_{i,k}\, g_{i,k}(y, F) = 0$ of the system $(\mathrm{KKT})_i$ is replaced by the SOS1 condition $\mathrm{SOS1}(s_{i,k}, \mu_{i,k})$.

As an example, in $(\mathrm{KKT})_i$, the complementarity constraint

$$\mu_i \left(M_i - \sum_{k \in I_P} (C_{i,\mathrm{out}} - C_{k,\mathrm{out}}) F_{k,i} \right) = 0$$

is replaced by

$$s_{i,k} = M_i - \sum_{k \in I_P} (C_{i,\mathrm{out}} - C_{k,\mathrm{out}}) F_{k,i} \quad \text{and} \quad \mathrm{SOS1}(s_{i,k}, \mu_i).$$

The resulting optimization problem with 15 companies has 1275 continuous and 240 binary variables as well as 1530 constraints—including 510 SOS1 conditions. Gurobi solves the problem to global optimality in 3 seconds. The resulting IEP design is given in Figure 2.

The white nodes represent the companies participating in the eco-park and the red node is the sink node (SN). The edges represent the built connections between the respective companies in the IEP. The amount of fresh, i.e., unpolluted, resource is not displayed. However, the optimal value for the designed IEP shows that, compared to the stand-alone situation, 32.72 % of the fresh-water resource is saved thanks to the implementation of the IEP structure.

5.4. **The Penalization Method by Leyffer and Munson and a Comparison.** To assess the performance of the SOS1 reformulation, we solve the same IEP problem using the Leyffer–Munson reformulation; see Leyffer and Munson (2010). This method is based on a penalization technique in which the original objective function is extended to also penalize the violation of the KKT complementarity constraints, which are then deleted from the set of constraints. By doing so, the new objective function is nonlinear while the new constraint set is now convex.

Starting from the MPCC reformulation of Subsection 5.2, the notation

$$q_i(y, F_i, F_{-i}) = \begin{pmatrix} (F_{i,j})_{j \in I} \\ (K y_{i,j} - F_{i,j})_{j \in I} \\ C_{i,\mathrm{in}} \sum_{j \in I} F_{i,j} - \sum_{k \in I_P} C_{k,\mathrm{out}} F_{k,i} \\ M_i - \sum_{k \in I_P} (C_{i,\mathrm{out}} - C_{k,\mathrm{out}}) F_{k,i} \end{pmatrix}$$

is used for $i \in I_P$.

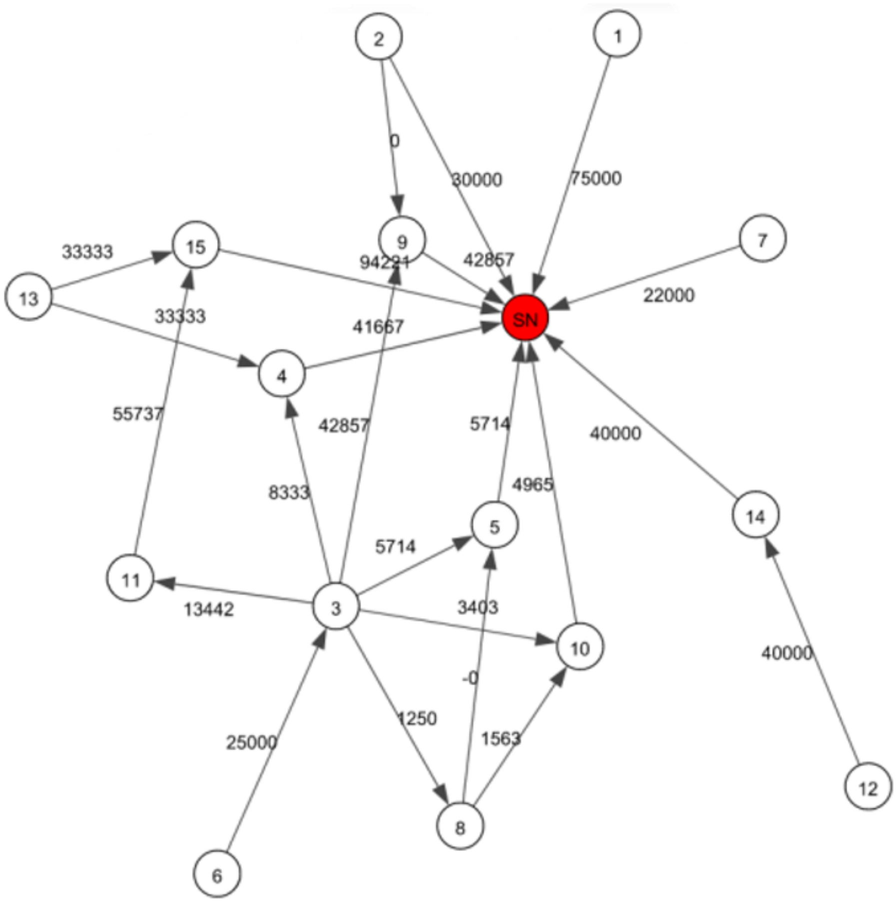

FIGURE 2. The optimal IEP design for 15 participating companies

We introduce slack variables $s := (s_i)_{i \in I_P}$ and reformulate the KKT complementarity conditions as

$$q_i(y, F_i, F_{-i}) - s_i = 0,$$
$$0 \leq \Gamma_i \perp s_i \geq 0,$$

where Γ_i is the subvector of Γ referring to company i and only containing Lagrangian multipliers for inequality constraints. Using the Leyffer–Munson approach, the IEP

Table 2. Gaps and total runtime (in minutes) of solving depending on ρ and warmstart. We set the penalty parameter $\rho = 10^n$

Warmstart	$n=1$	$n=2$	$n=3$	$n \in \{4,5,6\}$	$n=7$	Total time
yes	1.98 %	3.02 %	1.32 %	1.32 %	✓	48
no	1.98 %	4.24 %	4.77 %	1.32 %	✗	52

problem is now formulated as

$$\begin{aligned}
\min_{y,\Gamma,s} \quad & \sum_{i \in I_P} z_i(F_{-i}) + \rho \sum_{i \in I} s_i^\top \Gamma_i \\
\text{s.t.} \quad & \beta(F_{i,0} - \frac{M_i}{C_{i,\text{out}}}) + \sum_{k \in I_P} \left[\left(\delta - c + \frac{cC_{k,\text{out}}}{C_{i,\text{out}}} \right) F_{k,i} + \delta F_{i,k} \right] \leq 0, \quad i \in I_P, \\
& y_{i,i} = 0, \quad i \in I_P, \\
& q_i(y, F_i, F_{-i}) = s_i, \quad i \in I_P, \\
& \Gamma_i \geq 0, \quad i \in I_P, \\
& s_i \geq 0, \quad i \in I_P, \\
& \delta - \eta_{i,j} + \theta_{i,j} - \kappa_i C_{i,\text{out}} - \lambda_i C_{i,\text{in}} = 0, \quad i, j \in I_P, \\
& \beta - \eta_{i,0} + \theta_{i,0} - \kappa_i C_{i,\text{out}} - \lambda_i C_{i,\text{in}} = 0, \quad i \in I_P, \\
& M_i + \sum_{k \in I_P} C_{k,\text{out}} F_{k,i} - C_{i,\text{out}} \sum_{j \in I} F_{i,j} = 0, \quad i \in I_P,
\end{aligned}$$

which can then be solved with state-of-the-art solvers for mixed-integer nonlinear optimization problems (MINLPs). We do this by using Gurobi again, for which we have to change its NonConvex parameter to 2 so that Gurobi is able to handle the given nonconvex MINLP.

It is now important to notice that the penalization coefficient ρ plays a fundamental role since it must be chosen sufficiently large for the violation to be forced to be zero in an optimum, ensuring the equivalence between the above model and the initial formulation of the IEP problem. Since a provably correct threshold for the parameter being large enough is usually not known in advance, one often tries to increase the penalty parameter ρ iteratively. This, however, requires the solution of multiple MINLPs and can thus be computationally expensive.

For the specific IEP problem at hand, we conducted a sensitivity analysis on the value of ρ, i.e., we tested values $\rho = 10^n$ for some $n \in \mathbb{N}$. Table 2 lists the total runtimes for all problems. We set the maximum runtime to 8 minutes for each ρ and try to solve the model without and with warmstart, i.e., we then use the solution of the last run as an initialization for the current run. For almost all runs, we are not able to solve the problem to global optimality within the time limit. In these cases, the respective optimality gaps are given in the table.

We note that, with or without the warmstart, the solver takes around 50 minutes to solve all problems, which is significantly longer than the runtime of the SOS1 method. Moreover, not using warmstarts leads to points for $n = 7$ that violate KKT complementarity conditions. Hence, the sum of the KKT complementarity constraints in the extended objective function is positive and the obtained point is, consequently, not a solution of the SLMF game. Using warmstarts, we get a solution of the SLMF game for $n = 7$.

It is also interesting to note that the Leyffer–Munson reformulation of the IEP with 15 companies generates a problem with 1515 real-valued variables and 240 binary variables, thus around 250 continuous variable more than for the SOS1 formulation. Of course, it does not contain any SOS1 conditions.

The comparison of the SOS1 and the Leyffer–Munson method on this specific IEP example shows that the SOS1 can be more efficient and, more importantly, does not need the determination of a specific parameter, which is required for ρ for the penalty approach. Hence, the penalization approach faces a similar difficulty as the determination of the M-parameter in the "big-M" method.

6. Conclusion

Single-leader-multi-follower games naturally appear in many decision making processes including non-cooperative and hierarchical aspects. Recent results from bilevel optimization paved the way for a ready-to-use single-level and SOS1-based reformulation of these models that can be solved to global optimality by modern branch-and-cut solvers such as Gurobi. We present two academic examples including Python code that exemplify the usage of the SOS1 technique. Moreover, we discussed the application of this reformulation for solving a real-world problem from industrial eco-park modeling.

One of the main advantages of the SOS1-based reformulation is that there is no need for deriving provably correct big-M values or a sufficiently large penalty parameter, which is required if KKT complementarity constraints are re-written using further binary variables for modeling the corresponding disjunction or if the Leyffer–Munson penalty approach is used. However, the SOS1-based reformulation might lead to slower performance. Hence, an important topic for future research is on how to strengthen the resulting single-level reformulation, e.g., by generalizing the valid inequalities given in Audet et al. (2007a,b) and Kleinert et al. (2021b) to the case of SLMF games. For the eco-park example, however, we see that the SOS1 approach leads to a mixed-integer linear model (instead of a mixed-integer nonlinear one that needs to be solved in the Leyffer–Munson approach) and the runtimes are thus significantly faster for the SOS1 approach for this specific application.

Appendix A. Appendix - Calibration of the IEP Text Example

The meaning, unit, and values of the constants in the test example for the IEP model with 15 companies are given in Table 3.

Table 3. Specific data for the IEP problem used in the numerical experiments

	Meaning	Unit	Value
P	number of companies	—	15
M_i	contaminant load of process i	g/h	(6)
c	fresh water cost	\$/ton	0.01
β	polluted water discharge cost	\$/ton	0.22
δ	polluted water pumping cost	\$/ton	0.01
$C_{i,\text{out}}$	maximum contaminant concentration allowed in outlet of processes i	ppm	(8)
$C_{i,\text{in}}$	maximum contaminant concentration allowed in inlet of processes i	ppm	(7)

The values of the contaminant load are given by

$$\begin{aligned} M = (&7500, 6000, 5000, 30000, 4000, 2500, 2200, \\ &500, 30000, 4000, 2000, 2000, 5000, 30000, 13000) \end{aligned} \tag{6}$$

and the maximum contaminant concentrations are given by

$$C_{\bullet,\text{in}} = 10^3 \times (0, 0, 50, 80, 400, 20, 50, 80, 100, 400, 30, 25, 25, 50, 100) \tag{7}$$

as well as by

$$\begin{aligned} C_{\bullet,\text{out}} = 10^3 \times (&100, 200, 100, 800, 800, 100, 100, \\ &400, 800, 1000, 60, 50, 75, 800, 200). \end{aligned} \tag{8}$$

References

Allevi, E., D. Aussel, R. Riccardi, and D. Scopelliti (2024). "Single-Leader-Radner-Equilibrium: a new approach for a class of bilevel problems under uncertainty." In: *Journal of Optimization Theory and Applications* 200, pp. 344–370. DOI: 10.1007/s10957-023-02339-5.

Audet, C., J. Haddad, and G. Savard (2007a). "Disjunctive Cuts for Continuous Linear Bilevel Programming." In: *Optimization Letters* 1.3, pp. 259–267. DOI: 10.1007/s11590-006-0024-3.

Audet, C., G. Savard, and W. Zghal (2007b). "New Branch-and-Cut Algorithm for Bilevel Linear Programming." In: *Journal of Optimization Theory and Applications* 134.2, pp. 353–370. DOI: 10.1007/s10957-007-9263-4.

Aussel, D., G. Bouza, S. Dempe, and S. Lepaul (2021). "Genericity analysis of multi-leader-disjoint-followers game." In: *SIAM Journal on Optimization* 31.3, pp. 2055–2079. DOI: 10.1137/20M1356476. URL: https://doi.org/10.1137/20M1356476.

Aussel, D., K. Cao Van, and D. Salas (2023). "Optimal design of exchange water networks with control inputs in Eco-Industrial Parks." In: *Energy Economics* 120, p. 106480. DOI: 10.1016/j.eneco.2022.106480.

Aussel, D. and P. Chaipunya (2024). *Variational and quasi-variational inequalities under local reproducibility: solution concept and applications.* URL: https://arxiv.org/abs/2402.02115.

Aussel, D. and A. Svensson (2018). "Some remarks about existence of equilibria, and the validity of the EPCC reformulation for multi-leader-follower games." In: *Journal of Nonlinear and Convex Analysis* 19.7, pp. 1141–1162.

– (2019). "Towards Tractable Constraint Qualifications for Parametric Optimisation Problems and Applications to Generalised Nash Games." In: *Journal of Optimization Theory and Applications* 182, pp. 404–416. DOI: 10.1007/s10957-019-01529-4.

– (2020). "A Short State of the Art on Multi-Leader-Follower Games." In: *Bilevel Optimization: Advances and Next Challenges.* Ed. by S. Dempe and A. Zemkoho. Cham: Springer International Publishing, pp. 53–76. DOI: 10.1007/978-3-030-52119-6_3.

Beale, E. M. L. and J. A. Tomlin (1970). "Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables." In: *Proceedings of the Fifth International Conference on Operational Research.* Ed. by J. Lawrence. Tavistock Publications, 447—454.

Bertsekas, D. P. (2016). *Nonlinear Programming.* Athena scientific Belmont.

Boix, M., L. Montastruc, A.-P. Catherine, and S. Domenech (2015). "Optimization methods applied to the design of eco-industrial parks: a literature review." In: *Journal of Cleaner Production* 87, pp. 303–317. DOI: 10.1016/j.jclepro.2014.09.032.

Boix, M., L. Montastruc, L. Pibouleau, C. Azzaro-Pantel, and D. Serge (2012). "Industrial water management by multiobjective optimization: from individual to collective solution through eco-industrial parks." In: *Journal of Cleaner Production* 22(1), 85—97. DOI: 10.1016/j.jclepro.2011.09.011.

Dempe, S. and J. Dutta (2012). "Is bilevel programming a special case of a mathematical program with complementarity constraints?" In: *Mathematical Programming* 131.1-2, pp. 37–48. DOI: 10.1007/s10107-010-0342-1.

Dempe, S., V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova (2015). *Bilevel Programming Problems*. Springer. DOI: 10.1007/978-3-662-45827-3.

Dempe, S. and A. Zemkoho, eds. (2020). *Bilevel Optimization: Advances and Next Challenges*. Cham: Springer International Publishing. DOI: 10.1007/978-3-030-52119-6.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer. DOI: 10.1007/3-540-27659-9.

Facchinei, F. and C. Kanzow (2010). "Generalized Nash Equilibrium Problems." In: *Annals of Operations Research* 175.1, pp. 177–211. DOI: 10.1007/s10479-009-0653-x.

Fortuny-Amat, J. and B. McCarl (1981). "A Representation and Economic Interpretation of a Two-Level Programming Problem." In: *The Journal of the Operational Research Society* 32.9, pp. 783–792. DOI: 10.1057/jors.1981.156.

Kleinert, T., M. Labbé, I. Ljubić, and M. Schmidt (2021a). "A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization." In: *EURO Journal on Computational Optimization* 9, p. 100007. DOI: 10.1016/j.ejco.2021.100007.

Kleinert, T., M. Labbé, F. Plein, and M. Schmidt (2020). "There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization." In: *Operations Research* 68.6, pp. 1716–1721. DOI: 10.1287/opre.2019.1944.

– (2021b). "Closing the Gap in Linear Bilevel Optimization: A New Valid Primal-Dual Inequality." In: *Optimization Letters* 15, 1027—1040. DOI: 10.1007/s11590-020-01660-6.

Kleinert, T. and M. Schmidt (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*. DOI: 10.1007/s10287-023-00435-5.

Leyffer, S. and T. Munson (2010). "Solving multi-leader–common-follower games." In: *Optimization Methods & Software* 25.4-6, pp. 601–623. DOI: 10.1080/10556780903448052.

Nash, J. F. (1950). "Equilibrium points in n-person games." In: *Proceedings of the National Academy of Sciences* 36.1, pp. 48–49. DOI: 10.1073/pnas.36.1.48.

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization*. 2nd. Berlin: Springer. DOI: 10.1007/978-0-387-40065-5.

Pineda, S. and J. M. Morales (2019). "Solving Linear Bilevel Problems Using Big-Ms: Not All That Glitters Is Gold." In: *IEEE Transactions on Power Systems*. DOI: 10.1109/TPWRS.2019.2892607.

Ramos, M., M. Boix, D. Aussel, L. Montastruc, and S. Domenech (2016). "Water integration in Eco-Industrial Parks Using a Multi-Leader-Follower Approach." In: *Computers & Chemical Engineering* 87, pp. 190–207. DOI: 10.1016/j.compchemeng.2016.01.005.

Rosen, J. B. (1965). "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games." In: *Econometrica* 33.3, pp. 520–534. URL: http://www.jstor.org/stable/1911749.

Salas, D., K. C. Van, D. Aussel, and L. Montastruc (Aug. 2020). "Optimal design of exchange networks with blind inputs and its application to Eco-industrial parks." In: *Computers & Chemical Engineering* 143, p. 107053. DOI: 10.1016/j.compchemeng.2020.107053.

von Stackelberg, H. (1934). *Marktform und Gleichgewicht*. Springer.

– (1952). *Theory of the market economy*. Oxford University Press.

(D. Aussel) Lab. PROMES UPR CNRS 8521, University of Perpignan via Domitia, Tecnodud, Perpignan 66100, France
Email address: aussel@univ-perp.fr

(C. Egea) INSA Rouen Normandie, 685 Avenue de l'Université, Saint-Etienne-du-Rouvray 76800, France
Email address: cecile.egea@insa-rouen.fr

(M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier, Germany
Email address: martin.schmidt@uni-trier.de