Leadership in Congestion Games:
Multiple User Classes and Non-Singleton Actions

Alberto Marchesi[*] , **Matteo Castiglioni**[*] and **Nicola Gatti**

Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy

{alberto.marchesi, matteo.castiglioni, nicola.gatti}@polimi.it

Abstract

We study the problem of finding *Stackelberg equilibria* in games with a massive number of players. So far, the only known game instances in which the problem is solved in polynomial time are some particular *congestion games*. However, a complete characterization of hard and easy instances is still lacking. In this paper, we extend the state of the art along two main directions. First, we focus on games where players' actions are made of multiple resources, and we prove that the problem is NP-hard and *not* in Poly-APX unless P = NP, even in the basic case in which players are symmetric, their actions are made of only two resources, and the cost functions are monotonic. Second, we focus on games with singleton actions where the players are partitioned into *classes*, depending on which actions they have available. In this case, we provide a dynamic programming algorithm that finds an equilibrium in polynomial time, when the number of classes is fixed and the leader plays pure strategies. Moreover, we prove that, if we allow for leader's mixed strategies, then the problem becomes NP-hard even with only four classes and monotonic costs. Finally, for both settings, we provide mixed-integer linear programming formulations, and we experimentally evaluate their scalability on both random game instances and worst-case instances based on our hardness reductions.

1 Introduction

In the last years, Stackelberg games and their corresponding *Stackelberg equilibria* (SEs) received a lot of attention in the artificial intelligence literature, thanks to their successful applications in real-world settings, *e.g.*, in security domains [Tambe, 2011]. In a Stackelberg game, there is a special player, called *leader*, who has the ability to commit to a (potentially mixed) strategy beforehand, while the other players, who act as *followers*, decide how to play after observing the commitment [Von Stengel and Zamir, 2010]. As most

_____
[*]Equal Contribution

of the works in the literature, see, *e.g.*, [Conitzer and Sandholm, 2006; Paruchuri *et al.*, 2008; Conitzer and Korzhyk, 2011], we focus on *optimistic* SEs (OSEs), which assume that the followers break ties in favor of the leader, maximizing her utility. Few works also analyze *pessimistic* SEs, where the followers break ties so as to minimize the leader's utility. However, the pessimistic assumption usually results in computational problems that are much harder than their optimistic counterparts, as showed, *e.g.*, by Basilico *et al.* [2017] and Coniglio *et al.* [2017; 2018].

It is well-known that, while finding an OSE in two-player normal-form games is easy [Conitzer and Sandholm, 2006], the problem becomes much harder in games with multiple followers playing a *Nash equilibrium* (NE). For instance, Basilico *et al.* [2016; 2019] show that computing an OSE is NP-hard and not in Poly-APX unless P = NP in normal-form games with only two followers, and De Nittis *et al.* [2018] show the same result for polymatrix games with an arbitrary number of followers. Moreover, finding an OSE is NP-hard also in normal-form games with multiple followers playing sequentially [Conitzer and Sandholm, 2006], while the problem becomes easy if they play simultaneously in a correlated manner [Conitzer and Korzhyk, 2011].

Recently, Marchesi *et al.* [2018] and Castiglioni *et al.* [2018] studied the problem of computing OSEs in *congestion games* (CGs). In a CG, the players' actions are subsets of a given set of shared resources, and each resource has a cost that depends on the number of players using it (a.k.a. *congestion*). Most of the computational studies on CGs, see, *e.g.*, [Fabrikant *et al.*, 2004; Ieong *et al.*, 2005; Ackermann *et al.*, 2008], focus on finding pure-strategy NEs. Instead, Marchesi *et al.* [2018] and Castiglioni *et al.* [2018] apply the Stackelberg paradigm to CGs, assuming that the followers play a pure-strategy NE after observing the leader's mixed-strategy commitment. These works focus on *singleton* CGs, where actions are made of only one resource. Specifically, Marchesi *et al.* [2018] show that an OSE can be found in polynomial time when costs are *monotonic* in the congestion and players are *symmetric* (*i.e.*, they share the same set of actions), even when the number of followers is non-fixed. Castiglioni *et al.* [2018] show that the same holds even with non-monotonic costs if we restrict the leader to pure strategies. These results provide some examples of multi-follower Stackelberg games where finding an equilibrium is computa-

tionally tractable. On the other hand, Castiglioni *et al.* [2018] also show that, if either the costs are non-monotonic or players are not symmetric, then the problem becomes NP-hard and not in Poly-APX, unless P = NP.

Identifying new classes of multi-follower Stackelberg games where SEs can be computed in time polynomial in the number of players is crucial for real-world problems, enabling the adoption of SEs in massive applications, such as, *e.g.*, resource-sharing systems with premium (prioritized) users, cybersecuirty with multiple treats, and influence maximization in social networks. Despite this, a complete characterization of hard and easy game instances is still lacking.

Original Contributions

In this paper, we extend the state of the art on the computational problem of finding OSEs in CGs, identifying new cases in which it is solvable in polynomial time and others where it is not. First, we show that having actions made of only one resource is necessary to have efficient (polynomial-time) algorithms. Indeed, we prove that finding an OSE is NP-hard and not in Poly-APX unless P = NP, even if players' actions contain only two resources, costs are monotonic, and players are symmetric. Then, we introduce and study singleton CGs in which the players are partitioned into *classes*, with followers of the same class sharing the same set of actions. These are a generalization of singleton CGs with symmetric players, capturing the common case in which users can be split into (usually few) different classes, such as, *e.g.*, users with different priorities. For these games, we provide a dynamic programming algorithm that computes an OSE in polynomial time, when the number of classes is fixed and the leader is restricted to play pure strategies. On the other hand, we prove that, if the leader is allowed to play mixed strategies, then the problem becomes NP-hard even with only four classes and monotonic costs. Finally, for both settings, we design *mixed-integer linear programming* (MILP) formulations for computing OSEs, and we experimentally evaluate them on a testbed containing both randomly generated game instances and worst-case instances based on our hardness reductions. [1]

2 Related Works

In this work, we apply the Stackelberg paradigm to CGs following the approach of Von Stengel and Zamir [2010] and Conitzer and Sandholm [2006], *i.e.*, we treat the leader as a special player who seeks for an *optimal* (in terms of her utility) *strategy to commit to*. In the literature, there are a number of works that apply the Stackelberg paradigm to CGs following different approaches. Even if these works address settings that are are substantially different from ours, it is worth discussing how their results relate to our work.

There are some works, such as, *e.g.*, [Roughgarden, 2004; Swamy, 2007; Sharma and Williamson, 2009; Fotakis, 2010], which study CGs where the leader is an authority whose objective is to minimize the inefficacy (in terms of followers' social welfare) of the NE reached by the followers (*i.e.*, minimize the price of anarchy). This setting is fundamentally different form ours, as we assume that the leader looks

---

[1] The proofs of all the results are in [Marchesi *et al.*, 2019].

for a strategy to commit to that minimizes her own cost, while she is not concerned with the maximization of followers' social welfare. Let us remark that our approach leads to what is usually called OSE, while the Stackelberg strategies analyzed in these works are not OSEs according to the classical definitions [Conitzer and Sandholm, 2006; Von Stengel and Zamir, 2010].

Moreover, there are other works, such as, *e.g.*, [Leme *et al.*, 2012; De Jong and Uetz, 2014; Correa *et al.*, 2015], which apply the Stackelberg paradigm to CGs following yet another approach. They assume that the players play sequentially in a predefined order, reaching a *subgame perfect equilibrium* (SPE) in the extensive-form extension of the original CG where each player plays after observing the actions performed by the preceding players. This is different from our setting in two fundamental ways: (i) we assume that the followers play simultaneously, rather than sequentially; and (ii) these works study the inefficiency (in terms of followers' social welfare) of SPEs, rather than the computational problem of finding an optimal leader's strategy. Furthermore, we remark that an OSE is an SPE of a particular extensive-form extension of the original CG, known as mixed extension [Von Stengel and Zamir, 2010]. In this extended game, the leader first commits to a mixed strategy (having a continuum of actions), and, then, the followers observe it and play simultaneously, reaching an NE. This is different from the extensive-form extension studied in the work by Leme *et al.* [2012] and its follow-ups, where only pure-strategy commitments are possible and the followers play sequentially.

3 Preliminaries

We study *Stackelberg congestion games* (SCGs) with one leader and multiple followers playing an NE in the CG resulting from the leader's commitment. An SCG is a tuple (N, R, A, c_ℓ, c_f), defined as follows. $N = F \cup \{\ell\}$ is a finite set of players, the leader ℓ and the followers $p \in F$. R is a finite set of resources. $A = \{A_p\}_{p \in N}$, where $A_p \subseteq 2^R$ is the set of player p's actions, with each action $a_p \in A_p$ being a non-empty subset of resources, *i.e*, $a_p \subseteq R$. Finally, $c_\ell = \{c_{i,\ell}\}_{i \in R}$ and $c_f = \{c_{i,f}\}_{i \in R}$ are, respectively, the leader's and followers' cost functions, with $c_{i,\ell}, c_{i,f} : \mathbb{N} \to \mathbb{Q}$ being the costs of resource i as a function of its congestion. As usual, we assume $c_{i,\ell}(0) = c_{i,f}(0) = 0$ for every $i \in R$. Moreover, we let $n := |N|$ and $r := |R|$ be the number of players and resources, respectively.

Let $\sigma_p \in \Delta_p$ be a player p's strategy, which is a probability distribution over actions A_p, with $\sigma_p(a_p)$ denoting the probability of playing action $a_p \in A_p$. A strategy $\sigma_p \in \Delta_p$ is *pure* when it prescribes to play a single action $a_p \in A_p$ with probability $\sigma_p(a_p) = 1$; otherwise, σ_p is *mixed*. With an abuse of notation, given $\sigma_p \in \Delta_p$, let $\sigma_p(i) := \sum_{a_p \ni i} \sigma_p(a_p)$ be the probability of selecting resource $i \in R$ when σ_p is played. We denote with $\sigma = (\sigma_\ell, a)$ a *strategy profile*, *i.e.*, a tuple of players' strategies, in which the leader plays a (potentially) mixed strategy $\sigma_\ell \in \Delta_\ell$ and the followers play pure strategies that determine a *followers' action profile* $a = (a_p)_{p \in F}$, which belongs to the set $A_F = \times_{p \in F} A_p$.

Given a followers' action profile $a \in A_F$, we let $\nu_i^a :=$

$|\{p \in F \mid i \in a_p\}|$ be the number of followers selecting resource $i \in R$ in a. For ease of presentation, we define the *followers' configuration* induced by a as the vector $\nu^a \in \mathbb{N}^r$ whose i-th component is ν_i^a. Furthermore, for $\sigma_\ell \in \Delta_\ell$, $c_{i,f}^{\sigma_\ell} : \mathbb{N} \to \mathbb{Q}$ encodes the followers' expected cost of resource $i \in R$ given σ_ℓ, as a function of the number x of followers selecting i, *i.e.*, $c_{i,f}^{\sigma_\ell}(x) = \sigma_\ell(i)c_{i,f}(x+1) + (1-\sigma_\ell(i))c_{i,f}(x)$. Intuitively, the followers who select resource i experience a congestion incremented by one whenever the leader chooses i, which happens with probability $\sigma_\ell(i)$. When player $p \in N$ plays an action $a_p \in A_p$, she pays the sum of the costs of resources $i \in a_p$. Thus, given a strategy profile $\sigma = (\sigma_\ell, a)$, $c_p^\sigma := \sum_{i \in a_p} c_{i,f}^{\sigma_\ell}(\nu_i^a)$ is the cost of follower $p \in F$ in σ, and $c_\ell^\sigma := \sum_{a_\ell \in A_\ell} \sigma_\ell(a_\ell) \sum_{i \in a_\ell} c_{i,\ell}(\nu_i^a + 1)$ is the leader's one.

In an SCG, after observing a leader's strategy $\sigma_\ell \in \Delta_\ell$, the followers play a new CG with resource costs $c_{i,f}^{\sigma_\ell}$, for $i \in R$. Specifically, we assume that the followers play an NE in pure strategies [Nash, 1951]. Formally, given $\sigma = (\sigma_\ell, a)$, a is a (pure-strategy) NE for σ_ℓ if, for every $p \in F$ and $a_p' \in A_p$, $c_p^\sigma \leq c_p^{\sigma'}$, where $\sigma' = (\sigma_\ell, a')$ and $a' \in A_F$ is obtained from a by replacing a_p with a_p'. In words, in an NE, there is no follower who has an incentive to unilaterally deviate from a_p by playing another action a_p'. Given $\sigma_\ell \in \Delta_\ell$, we let E^{σ_ℓ} be the set of NEs in the followers' game for σ_ℓ. [2]

Different subclasses of SCGs can be defined by making additional assumptions on their elements. For instance, one possibility is to require that players' cost functions be *(weakly) monotonic* in the resource congestion, *i.e.*, for every resource $i \in R$, it must be the case that $c_{i,\ell}(x) \leq c_{i,\ell}(x + 1)$ and $c_{i,f}(x) \leq c_{i,f}(x + 1)$ for all $x \in \mathbb{N}$.

Another possibility is to restrict the structure of the players' action sets A_p. Along this direction, we address *Stackelberg singleton congestion games* (SSCGs), which are SCGs with players' actions required to be singletons, *i.e.*, $|a_p| = 1$ for every $p \in N$ and $a_p \in A_p$. Thus, when studying such games, we identify actions with resources, assuming $A_p \subseteq R$ for all $p \in N$. We also use $\sigma_\ell \in \Delta_\ell$ as if it were directly defined over resources in A_ℓ, with $\sigma_\ell(i)$ being the probability of playing resource $i \in A_\ell$. Moreover, given $\sigma = (\sigma_\ell, a)$, we have $c_p^\sigma = c_{a_p,f}^{\sigma_\ell}(\nu_{a_p}^a)$ and $c_\ell^\sigma = \sum_{i \in A_\ell} \sigma_\ell(i)c_{i,\ell}(\nu_i^a + 1)$.

Finally, we can consider different kinds of players' structures. Here, we focus on two notable cases. In the first one, all players share the same set of actions, *i.e.*, $A_p = A \subseteq 2^R$ for all $p \in N$. We refer to these games as *symmetric*. Instead, in the second case, there exists a finite set $\mathcal{T} = \{1, \dots, T\}$ of *followers' classes*, with followers of the same class sharing the same set of actions. We say that these games are $\mathcal{T}$-*class*. Specifically, in a $\mathcal{T}$-class SCG, we can partition the followers into T disjoint sets, *i.e.*, $F = \bigcup_{t \in \mathcal{T}} F_t$, so that, for each $t \in \mathcal{T}$, $A_p = A_t \subseteq 2^R$ for all $p \in F_t$. We also let $|F_t| = n_t$

be the number of followers of class $t \in \mathcal{T}$. When studying these games, given a followers' action profile $a \in A_F$, we let $\nu_{t,i}^a := |\{p \in F_t \mid i \in a_p\}|$ be the number of followers of class $t \in \mathcal{T}$ selecting resource $i \in R$ in a. Moreover, we define the *followers' configuration of class t induced by a* as the vector $\nu_t^a \in \mathbb{N}^r$ whose i-th component is $\nu_{t,i}^a$. [3]

Observe that $\mathcal{T}$-class SSCGs can be fully analyzed using followers' configurations, rather than action profiles. This is because only the number of followers of each class selecting each resource is significant, and, thus, a followers' action profile $a \in A_F$ can be equivalently represented with the followers' configurations $\{\nu_t^a\}_{t \in \mathcal{T}}$. Thus, we can directly use the vector $\nu_t \in \mathbb{N}^r$ with $\sum_{i \in R} \nu_{t,i} = n_t$ to denote a followers' configuration of class $t \in \mathcal{T}$. Moreover, given $\{\nu_t\}_{t \in \mathcal{T}}$, we let $\nu \in \mathbb{N}^r$ be such that $\nu_i := \sum_{t \in \mathcal{T}} \nu_{t,i}$ for $i \in R$. Notice that, given a strategy profile $\sigma = (\sigma_\ell, \{\nu_t\}_{t \in \mathcal{T}})$, $\{\nu_t\}_{t \in \mathcal{T}}$ is an NE for σ_ℓ if, for every class $t \in \mathcal{T}$, and resources $i \in A_t : \nu_{t,i} > 0$ and $j \in A_t$, it holds $c_{i,f}^{\sigma_\ell}(\nu_i) \leq c_{j,f}^{\sigma_\ell}(\nu_j + 1)$.

In conclusion, in an SCG, an OSE prescribes the leader an optimal strategy to commit to, assuming that the followers play a pure-strategy NE minimizing her cost in the CG resulting from the commitment. Formally, $\sigma = (\sigma_\ell, a)$ is an OSE if it solves the problem: $\min_{\sigma_\ell' \in \Delta_\ell} \min_{a' \in E^{\sigma_\ell'}} c_\ell^{\sigma' = (\sigma_\ell', a')}$.

4 Complexity Results on SCG

First, we study the problem of computing an OSE in SCGs, and we prove it is intractable even when the game is symmetric, cost functions are monotonic, and players' actions are made of only two resources. Thus, non-singleton actions make the problem considerably harder than in the singleton case, which admits a polynomial-time algorithm in symmetric games with monotonic costs [Marchesi *et al.*, 2018].

In particular, we show that the problem is NP-hard and not in Poly-APX unless P = NP, *i.e.*, the leader's cost in an OSE cannot be efficiently approximated up to any polynomial factor in the size of the input. Our results are based on a reduction from 3SAT, a well-know NP-complete problem. Intuitively, given $\epsilon > 0$, we map any 3SAT instance to an SCG that admits an OSE σ with $c_\ell^\sigma = \epsilon$ if and only if 3SAT is satisfiable, otherwise $c_\ell^\sigma = 1$.

Theorem 1. *Computing an OSE in symmetric SCGs is NP-hard, even when cost functions are monotonic and players' actions have cardinality at most two.*

By letting $\epsilon = \frac{1}{2^I}$, where I is the game size, the reduction used for Theorem 1 also shows the following:

Theorem 2. *The problem of computing an OSE in symmetric SCGs is not in Poly-APX unless P = NP, even when costs are monotonic and players' actions have cardinality two.*

In conclusion, we provide some side results that deepen our analysis on how non-singleton actions impact on the complexity of finding an OSE in SCGs. The following theorem shows that our intractability results hold even in SCGs where only the followers have non-singleton actions.

---

[2] Let us remark that we can safely restrict the attention to pure-strategy NEs, as one is guaranteed to exist in every CG [Rosenthal, 1973]. Moreover, it is natural to assume that the followers will end up playing pure-strategy NEs, since they are reachable by *best-response dynamics* [Monderer and Shapley, 1996], where players continuously change their actions, one at a time, playing a best response to the current players' configuration.

[3] Let us remark that symmetric SCGs are a special case of $\mathcal{T}$-class SCGs with only one class, *i.e.*, $\mathcal{T} = \{1\}$, and leader's action set equal to the followers', *i.e.*, $A_\ell = A_1$.

Theorem 3. *The problem of computing an OSE in SCGs is* NP-*hard and not in Poly-*APX *unless* P = NP, *even when leader's actions are singletons, costs are monotonic, and followers are symmetric with actions of cardinality at most two.*

Since the OSEs of the games used in our reduction prescribe the leader to play a pure strategy, we have that:

Observation 1. *The results in Theorems 1, 2, and 3 hold even if we restrict the leader to play pure strategies.*

Finally, let us consider the case in which only the leader has non-singleton actions. We have the following observation.

Observation 2. *In SCGs with symmetric followers having singleton actions, an OSE can be found in polynomial time if we restrict the leader to play pure strategies.* [4]

A polynomial-time algorithm enumerates the leader's pure strategies, and, for each of them, it computes an NE minimizing the leader's cost in the resulting followers' symmetric singleton CG, which can be done in polynomial time using dynamic programming, as shown by Ieong *et al.* [2005].

5 Complexity Results on $\mathcal{T}$-class SSCG

We switch the attention to SSCGs, *i.e.*, games where players' actions are singletons. In particular, we focus on $\mathcal{T}$-class SSCGs, a generalization of symmetric SSCGs. As shown by Castiglioni *et al.* [2018], finding an OSE in symmetric SSCGs with non-monotonic costs is NP-hard, while the problem becomes easy if: (a) we assume that the leader can only play pure strategies [Castiglioni *et al.*, 2018], or (b) we force players' costs be monotonic [Marchesi *et al.*, 2018].

Here, first we show that, under condition (a), computing an OSE is easy also in $\mathcal{T}$-class SSCGs with a fixed number of classes. Next, we prove that, if condition (a) does not hold, then the problem is NP-hard in $\mathcal{T}$-class SSCGs even if we enforce (b) and there are only four followers' classes.

Let us start providing a polynomial-time algorithm for computing OSEs in $\mathcal{T}$-class SSCGs with a fixed number of classes and the leader restricted to pure strategies. Our algorithm iterates over leader's pure strategies, *i.e.*, resources $i \in A_\ell$, and, for each of them, it finds an NE minimizing the leader's cost in the followers' game for $\sigma_\ell \in \Delta_\ell : \sigma_\ell(i) = 1$. Such NE is computed with an extension of the *dynamic programming* algorithm proposed by Ieong *et al.* [2005] for finding an optimal NE in symmetric singleton CGs without leadership. We extend the algorithm to $\mathcal{T}$-class singleton CGs without leadership, which are defined as their Stackelberg counterpart, except for the fact that there is no leader and all players experience the same resource costs $c = \{c_i\}_{i \in R}$, with $c_i : \mathbb{N} \to \mathbb{Q}$ defined as usual.

We define $\mathcal{O}(h, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T)$ as the cost of an optimal (according to some optimality criterion) NE for a $\mathcal{T}$-class singleton CG restricted to i resources $\{1, 2, \ldots, i\} \subseteq R$ and B_t players for each class $t \in \mathcal{T}$, where M_t is the largest cost experienced by a player of class t and V_t is the smallest cost a player of class t would get by switching to another resource. When the optimality criterion is the minimization of the sum of players' costs, we have:

---

[4]We leave as an open problem the study of the computational complexity when the leader is allowed to play mixed strategies.

Lemma 4. $\mathcal{O}(i, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T)$ *satisfies the following recursive equation:*

$$\mathcal{O}(i, B_1, \ldots, B_T, M_1, \ldots, M_T, V_1, \ldots, V_T) =$$
$$\min_{\substack{p_t \in \{0, \ldots, B_t\} \; \forall t \in \mathcal{T} \\ m_t \in \{0, \ldots, M_t\} \; \forall t \in \mathcal{T} \\ v_t \in \{1, \ldots, V_t\} \; \forall t \in \mathcal{T}}} \mathcal{O}(i - 1, p_1, \ldots, p_T, m_1, \ldots, m_T, v_1, \ldots, v_T) + bc_i(b)$$

$$\text{s.t.} \quad b = \sum_{t \in \mathcal{T}} (B_t - p_t) \tag{1a}$$

$$B_t = p_t \qquad\qquad \forall t \in \mathcal{T} : i \notin A_t \tag{1b}$$

$$m_t \leq M_t \qquad\qquad \forall t \in \mathcal{T} \tag{1c}$$

$$v_t \geq V_t \qquad\qquad \forall t \in \mathcal{T} \tag{1d}$$

$$c_i(b) \leq M_t \qquad \forall t \in \mathcal{T} : B_t - p_t > 0 \tag{1e}$$

$$c_i(b+1) \geq V_t \qquad \forall t \in \mathcal{T} : i \in A_t \tag{1f}$$

$$c_i(b) \leq v_t \qquad \forall t \in \mathcal{T} : B_t - p_t > 0 \tag{1g}$$

$$c_i(b+1) \geq m_t \qquad \forall t \in \mathcal{T} : i \in A_t. \tag{1h}$$

Remark 1. *Lemma 4 holds also for other optimality criteria, e.g., the minimization of the cost of a given resource.*

Thus, we can conclude the following:

Theorem 5. *In $\mathcal{T}$-class singleton CGs without leadership, an optimal (given an optimality criterion) NE can be found in $O(n^{6T} r^{4T+1})$. In $\mathcal{T}$-class SSCGs, an OSE can be found in $O(n^{6T} r^{4T+2})$ if we restrict the leader to play pure strategies.*

Corollary 5.1. *In $\mathcal{T}$-class SSCGs with a fixed number of classes, an OSE can be found in polynomial time if we restrict the leader to play pure strategies.*

Now, we prove the hardness result, using a reduction from K-PARTITION, an NP-compete variant of PARTITION with an additional size constraint [Castiglioni *et al.*, 2018].

Theorem 6. *Computing an OSE in $\mathcal{T}$-class SSCGs is* NP-*hard, even when cost functions are monotonic and $|\mathcal{T}| = 4$.*

6 Mathematical Programming Formulations

In this section, we provide two MILP formulations for finding OSEs in SCGs. The first one is specifically tailored to $\mathcal{T}$-class SSCGs, while the second one works for general SCGs.

Let us start with $\mathcal{T}$-class SSCGs. For ease of presentation, let $V(i) := \{1, \ldots, v_i^{\max}\}$ be the set of possible congestion levels induced by the followers on resource $i \in R$, with $v_i^{\max} := \sum_{t \in \mathcal{T} : i \in A_t} n_t$. Moreover, let $V(t) := \{1, \ldots, n_t\}$ for $t \in \mathcal{T}$. For every class $t \in \mathcal{T}$, resource $i \in A_t$, and value $v \in V(t)$, let us introduce the binary variable q_{tiv}, which is equal to 1 if and only if v followers of class t select resource i. Furthermore, for every $i \in R$ and $v \in V(i)$, let the binary variable y_{iv} be equal to 1 if and only if v followers select resource i. These variables represent followers' configurations. Specifically, for $t \in \mathcal{T}$, $\nu_t \in \mathbb{N}^r$ is such that $\nu_{t,i} = \sum_{v \in V(t)} v \, q_{tiv}$ for all $i \in R$, while $\nu \in \mathbb{N}^r$ is such that $\nu_i = \sum_{v \in V(i)} v \, y_{iv}$ for all $i \in R$. For every $i \in R$, let $\alpha_i \in [0, 1]$ be equal to $\sigma_\ell(i)$, and, for $v \in V(i)$, let the auxiliary variable z_{iv} be equal to the bilinear term $y_{iv} \alpha_i$. Finally, let $M > \max\{c_{i,f}(v) \mid i \in R, v \in \{1, \ldots, v_i^{\max} + 2\}\}$.

The complete MILP formulation reads as follows:

$$\min \quad \sum_{i \in R} \sum_{v \in V(i)} c_{i,\ell}(v+1)\, z_{iv} \tag{2a}$$

$$\text{s.t.} \quad \sum_{v \in V(t)} q_{tiv} \leq 1 \qquad\qquad \forall t \in \mathcal{T}, \forall i \in A_t \tag{2b}$$

$$\sum_{v \in V(i)} y_{iv} \leq 1 \qquad\qquad \forall i \in R \tag{2c}$$

$$\sum_{i \in A_t} \sum_{v \in V(t)} v\, q_{tiv} = n_t \qquad\qquad \forall t \in \mathcal{T} \tag{2d}$$

$$\sum_{t \in \mathcal{T}: i \in A_t} \sum_{v \in V(t)} v\, q_{tiv} = \sum_{v \in V(i)} v\, y_{iv} \qquad \forall i \in R \tag{2e}$$

$$\sum_{v \in V(j)} \Big(y_{jv} c_{j,f}(v+1) + z_{jv}\big(c_{j,f}(v+2) - c_{j,f}(v+1)\big) \Big) \geq$$
$$\sum_{v \in V(i)} \Big(y_{iv} c_{i,f}(v) + z_{iv}\big(c_{i,f}(v+1) - c_{i,f}(v)\big) \Big) +$$
$$- M\Big(1 - \sum_{v \in V(t)} q_{tiv}\Big) \qquad \forall t \in \mathcal{T}, \forall i \neq j \in A_t \tag{2f}$$

$$z_{iv} \leq \alpha_i \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{2g}$$

$$z_{iv} \leq y_{iv} \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{2h}$$

$$z_{iv} \geq \alpha_i + y_{iv} - 1 \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{2i}$$

$$z_{iv} \geq 0 \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{2j}$$

$$\sum_{i \in R} \alpha_i = 1 \tag{2k}$$

$$\alpha_i \geq 0 \qquad\qquad \forall i \in R \tag{2l}$$

$$\alpha_i = 0 \qquad\qquad \forall i \notin A_\ell \tag{2m}$$

$$q_{tiv} \in \{0,1\} \qquad\qquad \forall t \in \mathcal{T}, i \in A_t, v \in V(t) \tag{2n}$$

$$y_{iv} \in \{0,1\} \qquad\qquad \forall i \in R, \forall v \in V(i). \tag{2o}$$

Function (2a) is the leader's expected cost. Constraints (2b) ensure that at most one variable q_{tiv} be equal to 1 for each class $t \in \mathcal{T}$ and resource $i \in A_t$, and, thus, the number of followers of class t on each resource is uniquely determined (note that $\sum_{v \in V(t)} q_{tiv} = 0$ if no follower of class t selects resource i). Constraints (2c) ensure that at most one variable y_{iv} be equal to 1 for each resource $i \in R$, which guarantees that the congestion level of each resource is uniquely determined. Constraints (2d) guarantee that the followers' configuration be well-defined, *i.e.*, for all $t \in \mathcal{T}$, exactly n_t followers of class t are present. Constrains (2e) ensure that the congestion level on resource $i \in R$ be equal to the sum of the congestion levels induced by all classes. Constraints (2f) force the followers' configurations defined by the q_{tiv} variables be an NE for the leader's strategy identified by the α_i variables. This follows from the fact that, being $z_{iv} = y_{iv}\alpha_i$ and $z_{jv} = y_{jv}\alpha_j$, the right-hand side is the cost incurred by the followers of class $t \in \mathcal{T}$ selecting resource $i \in A_t$, while the left-hand side is the cost they would incur after deviating to $j \neq i \in A_t$. Note that, for each $t \in \mathcal{T}$, the constrains are active only if there is at least one follower of class t selecting i. Finally, Constraints (2g)–(2j) are McCormick envelope constraints [McCormick, 1976] which guarantee $z_{iv} = y_{iv}\alpha_i$ when $y_{iv} \in \{0,1\}$.

Next, we extend Formulation (2) to deal with general SCGs. Letting $v_i^{\max} := |\{p \in F \mid \exists a_p \in A_p : i \in a_p\}|$ be the maximum number of followers who can select resource $i \in R$, we define $V(i) := \{1, \ldots, v_i^{\max}\}$. For every follower $p \in F$ and action $a_p \in A_p$, we introduce the binary variable $x_{p\,a_p}$, which is equal to 1 if and only p

plays a_p. Moreover, for every $a_\ell \in A_\ell$, let $\alpha_{a_\ell} \in [0,1]$ be equal to $\sigma_\ell(a_\ell)$. All the variables already defined in Formulation (2) are used here with the same meaning. Finally, let $M > \sum_{i \in R} \max\{c_{i,f}(v) \mid i \in R, v \in \{1, \ldots, v_i^{\max} + 2\}\}$.

The complete MILP formulation reads as follows:

$$\min \quad \sum_{i \in R} \sum_{v \in V(i)} c_{i,\ell}(v+1)\, z_{iv} \tag{3a}$$

$$\text{s.t.} \quad \sum_{a_p \in A_p} x_{p\,a_p} = 1 \qquad\qquad \forall p \in F \tag{3b}$$

$$\sum_{v \in V(i)} y_{iv} \leq 1 \qquad\qquad \forall i \in R \tag{3c}$$

$$\sum_{v \in V(i)} v\, y_{iv} = \sum_{p \in F} \sum_{a_p \in A_p : i \in a_p} x_{p\,a_p} \qquad \forall i \in R \tag{3d}$$

$$\sum_{i \in a_p' \setminus a_p} \sum_{v \in V(i)} \Big(y_{iv} c_{i,f}(v+1) + z_{iv}\big(c_{i,f}(v+2) - c_{i,f}(v+1)\big) \Big) \geq$$
$$\sum_{i \in a_p \setminus a_p'} \sum_{v \in V(i)} \Big(y_{iv} c_{i,f}(v) + z_{iv}\big(c_{i,f}(v+1) - c_{i,f}(v)\big) \Big) +$$
$$- M\Big(1 - x_{p\,a_p}\Big) \qquad \forall p \in F, a_p \neq a_p' \in A_p \tag{3e}$$

$$z_{iv} \leq \sum_{a_\ell \in A_\ell : i \in a_\ell} \alpha_{a_\ell} \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{3f}$$

$$z_{iv} \leq y_{iv} \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{3g}$$

$$z_{iv} \geq \sum_{a_\ell \in A_\ell : i \in a_\ell} \alpha_{a_\ell} + y_{iv} - 1 \qquad \forall i \in R, \forall v \in V(i) \tag{3h}$$

$$z_{iv} \geq 0 \qquad\qquad \forall i \in R, \forall v \in V(i) \tag{3i}$$

$$\sum_{a_\ell \in A_\ell} \alpha_{a_\ell} = 1 \tag{3j}$$

$$\alpha_{a_\ell} \geq 0 \qquad\qquad \forall a_\ell \in A_\ell \tag{3k}$$

$$x_{p\,a_p} \in \{0,1\} \qquad\qquad \forall p \in F, \forall a_p \in A_p \tag{3l}$$

$$y_{iv} \in \{0,1\} \qquad\qquad \forall i \in R, \forall v \in V(i). \tag{3m}$$

Function (3a), Constraints (3c), and Constraints (3e)–(3k) have the same meaning as their counterparts in Formulation (2). Note that, in this case, $z_{iv} = y_{iv} \sum_{a_\ell \in A_\ell : i \in a_\ell} \alpha_{a_\ell}$, where the summation represents the probability $\sigma_\ell(i)$ and σ_ℓ is identified by the α_{a_ℓ} variables. Constraints (3b) ensure that each follower selects exactly one action. Constraints (3d) guarantee that the followers' configuration represented by the variables y_{iv} be well-defined. Note that, Constraints (3e) are enforced for each follower $p \in F$ here, and they are active only for the action $a_p \in A_p$ that she plays. Constraints (3e) do not account for the costs of resources $i \in a_p \cup a_p'$, since they do not change when deviating from a_p to a_p'.

7 Experimental Results

There is no standard testbed for our games and there is no evidence that some structured games are more representative than others. Thus, we test the MILP formulations proposed in Section 6 on randomly generated games, which represent instances of average-case complexity, and games based on the reductions provided in the proofs of Theorems 1 and 6, which, instead, represent worst-case complexity instances.

All 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. Each instance is solved on a single core within a time limit of 3600 seconds. We use GUROBI 7.0 (with Python interface) as MILP solver.
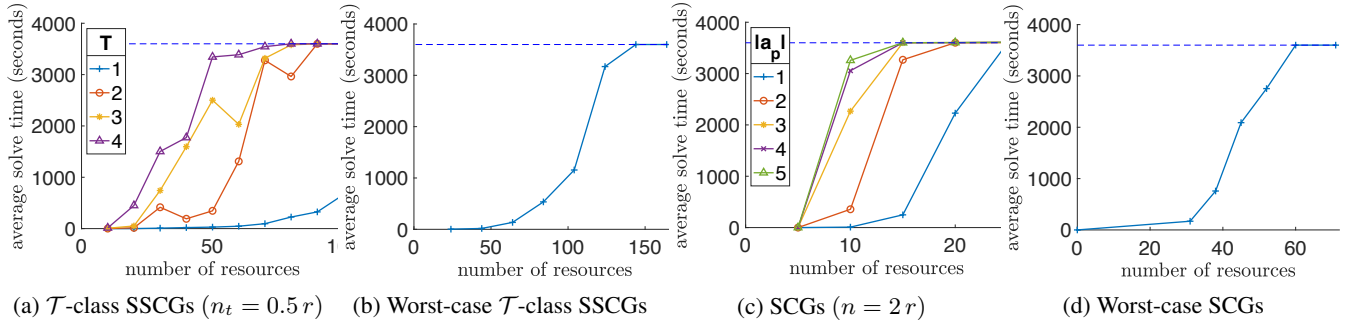
Figure 1: Computing times of Formulations (2) and (3) on randomly generated game instances and worst-case instances.

(a) $\mathcal{T}$-class SSCGs ($n_t = 0.5\,r$) (b) Worst-case $\mathcal{T}$-class SSCGs (c) SCGs ($n = 2\,r$) (d) Worst-case SCGs

Random Game Instances

For $\mathcal{T}$-class SSCGs, we generate random game instances with $r \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ resources and $T \in \{1, 2, 3, 4\}$ classes, with $n_t \in \{0.2\,r, 0.5\,r, r\}$ followers per class $t \in \mathcal{T}$ and $|A_t| = 0.5\,r$ actions per class $t \in \mathcal{T}$. Cost functions are randomly generated by sampling uniformly from $\{1, \ldots, nrT\}$. For general SCGs, we generate instances with $r \in \{5, 10, 15, 20, 25\}$ resources and $n \in \{r, 2\,r, 3\,r\}$ followers, with $|a_p| \in \{1, 2, 3, 4, 5\}$ resources per action $a_p \in A_p$ and $|A_p| = 0.5\,r$ actions per player $p \in N$. Cost functions are randomly generated by sampling uniformly from $\{1, \ldots, nr\}$. We build a testbed with 20 game instances per combination of the parameters.

Figure 1(a) displays the average computing times for Formulation (2) with $0.5\,r$ followers per class. The formulation scales quite well in practice. Symmetric games ($T = 1$) are quickly solved up to $r = 100$. Moreover, we are able to solve within the time limit games with up to four classes, 40 resources, and 160 players (40 players per class). Let us notice that the dynamic programming algorithm presented in Theorem 5 can be employed in this setting to find an OSE, if we restrict the leader to play pure strategies. However, preliminary experiments show that its scalability is extremely limited with respect to that of our formulation, as it finds a solution within the time limit only for games with less than 10 resources, while our formulation scales on much bigger games and it also works for mixed-strategy commitments.

Figure 1(c) shows the average computing times for Formulation (3) with $2\,r$ followers. We can conclude that, as expected, game instances with non-singleton actions are much harder to solve than singleton games. Here, the largest game instances we can solve within the time limit feature actions of cardinality two, 15 resources, and 30 players. [5]

Worst-Case Instances

We test Formulation (3) on instances built according to the reduction in Theorem 1. Specifically, we generate these games from random 3SAT instances with $|U| \in \{4, 5, 6, 7, 8, 9\}$ variables and $|C| = k|U|$ clauses, where $k \approx 4.26$ is the phase transition parameter determining generally hard-to-solve 3SAT instances [Cheeseman *et al.*, 1991]. We test 10 random instances for each number of variables. Furthermore, we experiment Formulation (2) on instances based on the re-

duction in Theorem 6. We generate these games from random K-PARTITION instances with $|S| \in \{20, 40, \ldots, 160\}$ integers, $s_i \in [2, 100]$ for all $s_i \in S$, and $K = \frac{|S|}{2}$. We test 10 random instances for each value of $|S|$.

Figures 1(b) and 1(d) show the computing times for $\mathcal{T}$-class SSCGs and SCGs, respectively. Surprisingly, the results we obtain are comparable to those for random games, empirically showing that, for the games we study, random instances are not easier to solve than structured ones, as instead it is the case, *e.g.*, in normal-form games [Sandholm *et al.*, 2005].

8 Discussion and Future Works

This paper studies the problem of computing Stackelberg equilibria in multi-follower Stackelberg games with a massive number of players, focusing on congestion games. Our results shed light on the boundary between hard and easy game instances, significantly advancing the state of the art.

Our analysis about games with non-singleton actions shows that, in oder to compute an equilibrium in time polynomial in the number of players, singleton actions are necessary (Theorems 1 and 2). This answers a question left open by Castiglioni *et al.* [2018]. Surprisingly, our negative result holds even in games where each action is made of at most two resources. Future works could address non-singleton games with players' actions enjoying specific structures, as it is the case, *e.g.*, in games played on graphs [Werneck *et al.*, 2000].

Our findings about congestion games with multiple classes of players substantially improve known results on symmetric (*i.e.*, single-class) games. Our dynamic programming algorithm can compute optimal leader's pure-strategy commitments in time polynomial in the number of players, even when costs are arbitrary functions and the players are split into a (fixed) number of different classes (Theorem 5).

Future extensions of this work may investigate how adopting a different solution concept for the followers' game, such as, *e.g.*, the *correlated equilibrium* [Aumann, 1974], affects the computation of equilibria. This has been recently studied for polynomial-type games [Castiglioni *et al.*, 2019].

---

[5]Additional experimental results are in [Marchesi *et al.*, 2019].

References

[Ackermann *et al.*, 2008] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *J ACM*, 55(6):25, 2008.

[Aumann, 1974] R.J. Aumann. Subjectivity and correlation in randomized strategies. *J MATH ECON*, 1(1):67–96, 1974.

[Basilico *et al.*, 2016] N. Basilico, S. Coniglio, and N. Gatti. Methods for Finding Leader-Follower Equilibria with Multiple Followers: (Extended Abstract). In *AAMAS*, pages 1363–1364, 2016.

[Basilico *et al.*, 2017] N. Basilico, S. Coniglio, N. Gatti, and A. Marchesi. Bilevel programming approaches to the computation of optimistic and pessimistic single-leader-multi-follower equilibria. *SEA*, 75:1–14, 2017.

[Basilico *et al.*, 2019] N. Basilico, S. Coniglio, N. Gatti, and A. Marchesi. Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games. *EJCO*, 2019.

[Castiglioni *et al.*, 2018] M. Castiglioni, A. Marchesi, N. Gatti, and S. Coniglio. Leadership in singleton congestion games: What is hard and what is easy. *CoRR*, abs/1808.10209, 2018.

[Castiglioni *et al.*, 2019] M. Castiglioni, A. Marchesi, and N. Gatti. Be a leader or become a follower: The strategy to commit to with multiple leaders. In *IJCAI*, 2019.

[Cheeseman *et al.*, 1991] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *IJCAI*, pages 331–337, 1991.

[Coniglio *et al.*, 2017] S. Coniglio, N. Gatti, and A. Marchesi. Pessimistic leader-follower equilibria with multiple followers. In *IJCAI*, pages 171–177, 2017.

[Coniglio *et al.*, 2018] S. Coniglio, N. Gatti, and A. Marchesi. Computing a pessimistic leader-follower equilibrium with multiple followers: the mixed-pure case. *CoRR*, abs/1808.01438, 2018.

[Conitzer and Korzhyk, 2011] V. Conitzer and D. Korzhyk. Commitment to correlated strategies. In *AAAI*, pages 632–637, 2011.

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

[Correa *et al.*, 2015] J. Correa, J. De Jong, B. De Keijzer, and M. Uetz. The curse of sequentiality in routing games. In *WINE*, pages 258–271, 2015.

[De Jong and Uetz, 2014] J. De Jong and M. Uetz. The sequential price of anarchy for atomic congestion games. In *WINE*, pages 429–434. Springer, 2014.

[De Nittis *et al.*, 2018] G. De Nittis, A. Marchesi, and N. Gatti. Computing the optimal strategy to commit to in polymatrix games. In *AAAI*, pages 82–90, 2018.

[Fabrikant *et al.*, 2004] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure nash equilibria. In *STOC*, pages 604–612, 2004.

[Fotakis, 2010] D. Fotakis. Stackelberg strategies for atomic congestion games. *THEOR COMPUT SYST*, 47(1):218–249, 2010.

[Ieong *et al.*, 2005] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *AAAI*, pages 489–494, 2005.

[Leme *et al.*, 2012] R.P. Leme, V. Syrgkanis, and E. Tardos. The curse of simultaneity. In *ITCS*, pages 60–67, 2012.

[Marchesi *et al.*, 2018] A. Marchesi, S. Coniglio, and N. Gatti. Leadership in singleton congestion games. In *IJCAI*, pages 447–453, 2018.

[Marchesi *et al.*, 2019] A. Marchesi, M. Castiglioni, and N. Gatti. Leadership in congestion games: Multiple user classes and non-singleton actions (extended version). *CoRR*, abs/1905.13108, 2019.

[McCormick, 1976] G. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *MATH PROGRAM*, 10(1):147–175, 1976.

[Monderer and Shapley, 1996] D. Monderer and L.S. Shapley. Potential games. *GAME ECON BEHAV*, 14(1):124–143, 1996.

[Nash, 1951] J. Nash. Non-cooperative games. *ANN MATH*, pages 286–295, 1951.

[Paruchuri *et al.*, 2008] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: an efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*, pages 895–902, 2008.

[Rosenthal, 1973] R.W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *INT J GAME THEORY*, 2(1):65–67, 1973.

[Roughgarden, 2004] T. Roughgarden. Stackelberg scheduling strategies. *SIAM J COMPUT*, 33(2):332–350, 2004.

[Sandholm *et al.*, 2005] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding nash equilibria. In *AAAI*, pages 495–501, 2005.

[Sharma and Williamson, 2009] Y. Sharma and D.P. Williamson. Stackelberg thresholds in network routing games or the value of altruism. *GAME ECON BEHAV*, 67(1):174–190, 2009.

[Swamy, 2007] C. Swamy. The effectiveness of stackelberg strategies and tolls for network congestion games. In *SODA*, pages 1133–1142, 2007.

[Tambe, 2011] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[Von Stengel and Zamir, 2010] B. Von Stengel and S. Zamir. Leadership games with convex strategy sets. *GAME ECON BEHAV*, 69(2):446–457, 2010.

[Werneck *et al.*, 2000] R. Werneck, J. Setubal, and A. da Conceicao. Finding minimum congestion spanning trees. *J EXP ALG*, 5:11, 2000.