

Computing Stackelberg Equilibria of Large General-Sum Games*

Avrim Blum[†] Nika Haghtalab[‡] MohammadTaghi Hajiaghayi[§] Saeed Seddighin[§]

Abstract

We study the computational complexity of finding Stackelberg Equilibria in general-sum games, where the set of pure strategies of the leader and the followers are exponentially large in a natural representation of the problem.

In *zero-sum* games, the notion of a Stackelberg equilibrium coincides with the notion of a *Nash Equilibrium* Korzhyk et al. (2011b). Finding these equilibrium concepts in zero-sum games can be efficiently done when the players have polynomially many pure strategies or when (in addition to some structural properties) a best-response oracle is available Ahmadinejad et al. (2016); Dudík et al. (2017); Kalai and Vempala (2005). Despite such advancements in the case of zero-sum games, little is known for general-sum games.

In light of the above, we examine the computational complexity of computing a Stackelberg equilibrium in large general-sum games. We show that while there are natural large general-sum games where the Stackelberg Equilibria can be computed efficiently if the Nash equilibrium in its zero-sum form could be computed efficiently, in general, structural properties that allow for efficient computation of Nash equilibrium in zero-sum games are not sufficient for computing Stackelberg equilibria in general-sum games.

1 Introduction

Recent years have witnessed significant interest in Stackelberg games and their equilibria. A *Stackelberg game* models an interaction between two players, a *leader* and a *follower*, where the leader's goal is to *commit* to a randomized strategy that yields the highest utility, given that the follower responds by choosing an action that is best for itself. Such a pair of strategies is called a *Stackelberg equilibrium (SE)*. The interest in these games is driven, in part, by their applications to security (Tambe, 2011) and their adoption by major security agencies such as the US Coast Guard, the Federal Air Marshals Service, and the Los Angeles Airport Police.

Standard approaches for finding a Stackelberg equilibrium, such as the Multiple LPs approach of Conitzer and Sandholm (2006), run in time polynomial in the number of pure strategies of the leader and follower. As Stackelberg games and their applications have become more prevalent, they are increasingly used to model complex scenarios where one or both players' strategy sets are *exponentially large in a natural representation of the problem*, in which case existing approaches

*This work was supported in part by the National Science Foundation under grants CCF-1733556 and CCF-181501, NSF CAREER award CCF-1053605, NSF BIGDATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant.

[†]Toyota Technological Institute, Chicago, IL 60637, avrim@ttic.edu

[‡]Microsoft Research, Cambridge, MA 02142, nika.haghtalab@microsoft.com

[§]University of Maryland, College Park, MD 20742, {hajiagha,saeedrez}@cs.umd.edu

are not computationally feasible. In this work, we consider such “large” games and ask whether there are computationally efficient algorithms for finding their Stackelberg equilibria.

Of course, such algorithms cannot exist without some assumptions on the problem structure. Here, we review the common assumptions and approaches for computing minimax-optimal solutions in large *zero-sum* games, where minimax strategies, Nash equilibria, and Stackelberg equilibria all coincide. Computing these equilibrium concepts in 2-player zero-sum games has received significant attention (Immorlica et al., 2011; Ahmadi et al., 2016; Behnezhad et al., 2016; Garg et al., 2011; Xu, 2016; Wang and Shroff, 2017; Dudík et al., 2017; Von Neumann and Morgenstern, 1945; Hannan, 1957; Freund and Schapire, 1995; Littlestone and Warmuth, 1994; Conitzer and Sandholm, 2006). For large zero-sum games, two structural assumptions that have proven useful in computing a Nash equilibrium are the ability to efficiently optimize a linear function over the strategy space of each player (Ahmadi et al., 2016) and the ability to compute the best-response of each player against a mixed strategy of the other combined with a decomposability property of the action set (Dudík et al., 2017).

In general-sum games, however, Stackelberg, Nash, and Minimax equilibria diverge (See Appendix A for an illustration). In general-sum games the leader can benefit from committing to a mixed strategy and obtain a more favorable Stackelberg equilibrium than any Nash equilibrium. From the algorithmic perspective, a Stackelberg equilibrium in a general-sum game can be computed efficiently when the game is small. That is, there are algorithms, such as the Multiple LPs approach of Conitzer and Sandholm (2006), that run in time $\text{poly}(|S_L|, |S_F|)$ where S_L and S_F are the set of pure strategies of the leader and follower, respectively. While this method is an efficient approach for computing a Stackelberg equilibrium in small games, it becomes computationally inefficient in many natural scenarios where the set of actions of the leader or follower is exponential in a natural representation of the game. Examples of such settings include games inspired by applications to security, where either the actions of the leader or the follower represent sets of edges in a graph. As opposed to the zero-sum case for which existence of certain structural properties are known to lead to efficient computation of the equilibrium concepts, computation of Stackelberg equilibrium in large general-sum games has remained mostly unexplored.

1.1 Our Results and Contributions

In light of the above, we examine the computational complexity of computing Stackelberg equilibria in large general-sum games. Specifically, we consider two classes of general-sum games, both of which demonstrate structural properties that under the zero-sum assumption would lead to efficient algorithms for computing the minimax optimal strategies. For the first class of games, we give an efficient algorithm for computing a Stackelberg Equilibrium. In the second class of games, we show that even approximating the Stackelberg equilibrium is NP-Hard. This drives home the main message of this work, that is *while there are natural large general-sum games where the Stackelberg Equilibria can be computed efficiently if the Nash equilibrium in its zero-sum form could be computed efficiently, in general, structural properties that allow for efficient computation of Nash equilibrium in zero-sum games are not sufficient for computing Stackelberg equilibria in general-sum games.*

In more details, the two classes of games we work with are as follow.

Incentive Games In Section 3, we introduce a class of games called INCENTIVE GAMES. In these games, the actions of the leader can be described as two-part actions, the first part of the

action is an element of a set and the second part of the action is a set of incentives to the follower for playing certain actions.

As a motivating example consider a taxation scenario. In this setting, a government agency (e.g., IRS) takes the role of the leader and a taxpayer is the follower. A number of investments, indicated by the set E , are available to the taxpayer. Each investment e has a return of c_e to the taxpayer. The taxpayer invests in a package of investments $S \subseteq E$ that has the highest net payoff. The government agency is interested in taxing these investments in order to maximize the tax revenue. To do so, the agency allocates 1 unit of taxes¹ between these investments. There are two types of taxation mechanisms. First is taxing an individual investment e by some amount x_e . The second is to provide tax relief v_S for a package of options the taxpayer has invested in. Examples of the second type of taxation mechanism include United States federal residential renewable energy tax credit that offers a tax break to individuals who have invested in home electric power storage, e.g., batteries, and home-generated renewable energy, e.g., solar panels, but no tax break to those who have invested in the former without the latter (U.S. Department of Energy, 2019; EnergySage, 2019). The tax revenue and the net payoff the taxpayer respectively receive from individual taxes x_e and combinatorial tax reliefs v_S when the taxpayer invests in investments S are $\sum_{e \in S} x_e - v_S$ and $\sum_{e \in S} (-x_e + c_e) + v_S$. It is not hard to see that these tax breaks play an essential role in the design of tax systems. Not only they increase the total tax revenue obtainable by a tax system (See Example 3.1 for an illustration) but they can also be used to incentivize the taxpayers to take actions that are more beneficial to the government.

More generally, we consider Stackelberg games and we consider a family of sets $\S \subseteq \{0, 1\}^E$ and leader and follower element payoffs, C_e and c_e , respectively, for all $e \in E$. A pure strategy of the leader is to choose $e \in E$ and a vector of *incentives* $\vec{v} \in [0, 1]^{|\S|}$, such that $\|\vec{v}\|_0 \in \text{poly}(|E|)$.² A pure strategy of the follower is to choose one set $S \in \S$. The payoff of the leader and follower are defined, respectively, by

$$\begin{aligned} U_L((e, \vec{v}), S) &= 1_{e \in S} - v_S + C_e, \\ U_F((e, \vec{v}), S) &= -1_{e \in S} + v_S + \sum_{e' \in S} c_{e'}, \end{aligned}$$

that is, the players receive non-zero-sum utilities from their individual choices, i.e., C_e and $\sum_{e \in S} c_{e'}$, and zero-sum utilities from choosing actions that intersect, i.e., $\pm 1_{e \in S}$, and from the incentives provided on the followers actions sets, i.e., $\pm v_S$.

We first note that when c_e and C_e are set to 0 for all $e \in E$, this game is zero-sum and can be efficiently solved when each player can compute its best-response to any choice of mixed strategy of the other player, i.e., optimize a linear function over the strategy space of the other player using existing results (Ahmadinejad et al., 2016; Dudík et al., 2017; Kalai and Vempala, 2005).

When c_e and C_e are non-zero, we show that the leader can obtain a higher payoff equilibrium if it could make additional commitments in the form of incentives for the follower, i.e., can play non-zero \vec{v} . An interesting aspect of this game is that it is derived from a simple Stackelberg game model (where $\vec{v} = \vec{0}$) by adding zero-sum payoffs that *only benefit the follower*. Yet, the leader's payoff in the Stackelberg equilibrium of the new game is much higher than its payoff in the

¹More generally instead of 1 unit of tax one can consider any other fixed upperbound for the total amount of taxes used by the system. Note that any modern tax system is design with such an upperbound in mind to ensure the welfare of society and avoid financial or social crises caused by excessive taxes.

²The sparsity requirement is such that the leader can communicate its strategy to the follower efficiently.

original game. Moreover, as we show in Theorem 1 there is a polynomial time algorithm for finding the Stackelberg equilibrium of such games when the leader can optimize a linear function over the actions of the follower, which is a similar condition to the ones used for computing Stackelberg equilibria in large zero-sum games (Ahmadinejad et al., 2016; Dudík et al., 2017; Kalai and Vempala, 2005).

Permuted Matching Game In Section 4, we introduce a non-zero-sum game called PERMUTED MATCHING. In this game, there is a graph $G = (V, E)$ and a permutation $\pi : E \rightarrow E$. The set of pure strategies of the leader and follower is the set of all matchings in G . The goal of the leader is to maximize the intersection of its matching with the π -transformation of the matching of the follower. On the other hand, the goal of the follower is to maximize the intersection of the two matchings, with no regards to π . More formally, for $S \subseteq E$ we define $\pi(S) = \{e \in S | \pi(e)\}$. Then the utility of the leader and follower are defined, respectively, by

$$U_L(M_1, M_2) = |M_1 \cap \pi(M_2)| \quad U_F(M_1, M_2) = |M_1 \cap M_2|.$$

It is not hard to see that, in this game, the problem of finding a best response for a player reduces to computing maximum weighted matching of G and can be solved in polynomial time. This would have been sufficient for getting a polynomial time algorithm for finding a Stackelberg equilibrium had the game been a zero-sum (Ahmadinejad et al., 2016; Dudík et al., 2017; Kalai and Vempala, 2005). In a sharp contrast, however, we show that computing a Stackelberg equilibrium of this general-sum game is APX-hard, even though, we can compute player's best-response efficiently.

We obtain this hardness result via two reductions. First, we define the following computational problem:

π -TRANSFORMATION-IDENTICAL-MATCHING: Given a graph G and a mapping $\pi : E(G) \rightarrow E(G)$ over the edges of G , find a matching M of G that maximizes $|M \cap \pi(M)|$.

We next show that computing an approximate Stackelberg equilibrium of the PERMUTED MATCHING game is at least as hard as computing an approximate solution for the π -TRANSFORMATION-IDENTICAL-MATCHING problem. The crux of the argument is that if in an instance of the π -TRANSFORMATION-IDENTICAL-MATCHING problem there exists a matching which is almost identical to its π -transformation, then a Stackelberg equilibrium of the PERMUTED MATCHING game is closely related to that matching. Thus, any solution for the PERMUTED MATCHING game can be turned into a solution for π -TRANSFORMATION-IDENTICAL-MATCHING with almost the same quality. In the second step, we reduce the π -TRANSFORMATION-IDENTICAL-MATCHING problem to the MAXIMUM 3D MATCHING problem, which is known to be APX-hard (Petranc, 1994).

We note that our results strengthen the existing hardness results of Letchford and Conitzer (2010); Li et al. (2016) that showed that computing Stackelberg equilibrium is *NP-hard*³. Our APX-hardness result shows that one cannot even approximate the Stackelberg equilibria of large games within *an arbitrary constant factor*, even when best-response can be efficiently computed.

1.2 Related Work

There is an extensive body of work investigating the complexity of solving Security games, which is a special case of computing Stackelberg equilibria (see *e.g.* Tambe (2011); Behnezhad et al.

³Interestingly, it is not hard to show that player best-response can also be computed efficiently in the games used by Letchford and Conitzer (2010); Li et al. (2016), although this was not central to their results.

(2017); Xu et al. (2014); Basilico et al. (2009); Letchford and Vorobeychik (2011); Xu (2016)).

zero-sum games Several algorithms have been proposed for finding the Stackelberg equilibria of a special case of security games called the spatio-temporal security games (Behnezhad et al., 2017; Xu et al., 2014). These games are zero-sum by definition, where Stackelberg equilibria, Nash equilibria, and Minimax equilibria all coincide. In comparison, our work focuses on general-sum games.

Smaller general-sum games Several works have introduced polynomial time algorithms for computing Stackelberg equilibria in games where *only one player's* strategy set is exponentially large (Kiekintveld et al., 2009; Xu, 2016). A common approach used in this case is the Multiple LPs approach of Conitzer and Sandholm (2006) that runs in $\text{poly}(|S_L|, |S_F|)$. In this approach one creates a separate Linear Program for every action $y \in S_F$ of the follower, where the variables represent the probability assigned to the actions of the leader, the objective maximizes the expected payoff of the leader, and the constraints assure that action y is the best-response of the follower. This method can be implemented efficiently even when the leader's strategy set is exponentially large, e.g., when a separation oracle can be implemented efficiently. In comparison, our main computational result focus on settings where both the leader and follower have exponentially many strategies.

Existing hardness results Letchford and Conitzer (2010) studied the computational complexity of extensive form games and proved a closely related hardness result.

They showed that computing Stackelberg equilibrium of a game is weakly NP-hard using a reduction from Knapsack. Interestingly, one can efficiently compute player best-response in their setting. In comparison, our hardness result improves over these results by showing that Stackelberg equilibria are hard to approximate within arbitrary constant factor even when player best-response can be computed efficiently.

A number of works have investigated the relationship between the Stackelberg equilibria and Nash equilibria of security games and have shown that computing a Stackelberg equilibrium is at least as hard as computing a Nash equilibrium of general-sum games. Korzhyk et al. (2011a) studied a special class of general-sum Stackelberg Security games where any Stackelberg Equilibrium is also a Nash equilibrium. This shows that computing Stackelberg equilibria is harder than computing Nash equilibria. Li et al. (2016) studied Bayesian Stackelberg Games, where there is additional uncertainty about the attacker and show that computing the Stackelberg equilibrium is hard, and introduce an exponential time algorithm for computing the Nash equilibria. In comparison, our work shows that Stackelberg equilibria are hard to approximate even when players best-response is easy to compute. That is, we show a gap between the computational complexity of approximating Stackelberg equilibrium of a general-sum game and that of its corresponding zero-sum variant.

2 Preliminaries

Throughout this paper, we study Stackelberg equilibria of large games. Our emphasis is on two player games and therefore we denote the players by L (leader) and F (follower). Let S_L and S_F be the set of actions (pure strategies) of players L and F. For a pair of pure strategies $x \in S_L$ and

$y \in S_F$, we denote the payoffs of players L and F by $U_L(x, y)$ and $U_F(x, y)$, respectively. Similarly, for a pair of mixed strategies X and Y we denote the payoffs by

$$\begin{aligned} U_L(X, Y) &= \mathbb{E}_{x \sim X, y \sim Y}[U_L(x, y)] \\ U_F(X, Y) &= \mathbb{E}_{x \sim X, y \sim Y}[U_F(x, y)]. \end{aligned}$$

In Stackelberg games, the leader commits to a (possibly mixed) strategy X and plays this strategy. The follower then plays a best response against X , $b(X)$, according to her payoff function. Since the follower goes second its best-response is a deterministic action $b(X) = \max_y U_F(X, y)$. In case there is more than one best response for the follower, we assume she plays the one that maximizes the payoff of the leader. A pair of strategies X and y are in Stackelberg equilibrium if y is a best response of the follower against X and X maximizes the payoff of the leader, subject to the follower playing a best response.

3 Incentive Games

In this section, we discuss a class of Stackelberg games where the leader has the ability to make additional commitments in the form of additional incentives to the follower. Recall that a natural scenario that can be addressed by this Stackelberg model is *taxation*. In this case the leader can set taxes on individual investments but can also provide tax breaks on bundles of investments that the tax payer has invested in. We first show how these additional combinatorial incentives can improve the leader's payoff significantly and then show polynomial time algorithms for computing a Stackelberg equilibrium in this model.

Let us first recall the definition of INCENTIVE GAMES. In this model, we consider a set of elements E , a family of its subsets $\mathcal{S} \subseteq \{0, 1\}^E$ and rewards C_e and c_e for all $e \in E$. The set of pure strategies of the leader is $S_L = E \times [0, 1]^{|\mathcal{S}|}$. That is, each action of the leader has two parts, the first part is an element $e \in E$ and the second part is a vector of *incentives* $\vec{v} \in [0, 1]^{|E|}$. We assume that the leader is restricted to playing incentive vectors $\|\vec{v}\|_0 \in \text{poly}(|E|)$.⁴ The follower's strategy set is $S_F = \mathcal{S}$. The leader and follower payoffs are as follows.

$$U_L((e, \vec{v}), S) = 1_{e \in S} - v_S + C_e, \text{ and } U_F((e, \vec{v}), S) = -1_{e \in S} + v_S + \sum_{e' \in S} c_{e'}, \quad (1)$$

that is, the players receive non-zero-sum utilities from their individual choices, i.e., C_e and $\sum_{e \in S} c_{e'}$, and zero-sum utilities from choosing actions that intersect, i.e., $\pm 1_{e \in S}$, and from the incentives provided on the followers actions sets, i.e., $\pm v_S$. For ease of exposition and by the linearity of the payoffs, we denote a mixed strategy of the leader by (\vec{x}, \vec{V}) , where x_e is the probability with which the first part of the leader's action is e and V_S is the expected incentive provided on action S in the second part of the leader's action. Note that in this case, the expected utilities of the leader and follower are

$$U_L((\vec{x}, \vec{V}), S) = \sum_{e \in S} x_e - V_S + \sum_{e \in E} x_e C_e \quad (2)$$

$$U_F((\vec{x}, \vec{V}), S) = \sum_{e \in S} (-x_e + c_e) + V_S. \quad (3)$$

⁴The sparsity requirement is such that the leader can communicate its strategy to the follower efficiently.

Let us first consider a variation of INCENTIVE GAMES where the leader cannot provide additional incentives to the follower, i.e., $S_L = E \times \vec{0}$. The only difference between these games is that INCENTIVE GAMES are amended by allowing zero-sum non-negative payments \vec{v} that benefit the follower solely. One might wonder if the commitment to make additional payments \vec{v} to the follower can ever be beneficial to the leader. This is exactly what we demonstrate in the next example. That is, by allowing the leader to make additional zero-sum payoffs that only benefit the follower, we can obtain Stackelberg equilibria that have much higher payoff to the leader.

Example 3.1. Consider a graph instance in Figure 1, E is the set of all edges, \S is the set of all s - t paths, there are no edge payoff to the leader, i.e., $C_e = 0$ for all $e \in E$, and the edge payoff to the follower, c_e , are the negative of the edges costs that are denoted below each edge. That is, this is an instance where the follower is responding by choosing a shortest path with respect to the edge weights that correspond to the probability with which the leader plays them. Note that, since there are many

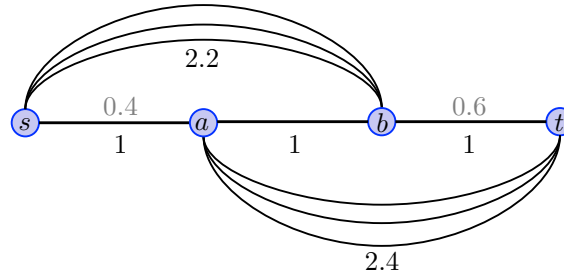


Figure 1: An example where additional commitment increases the leader's payoff in the Stackelberg equilibrium. The follower's cost for each edge is denoted *below* the edge, i.e., $-c_{sa} = -c_{ab} = -c_{bt} = 1$ and $-c_{ab} = 2.2$ and $-c_{at} = 2.4$ for all the edges between s and b , and between a and t . The mixed strategy of the leader is denoted in gray *above* the edges.

parallel edges ab and at , the leader's optimal strategy (with or without additional commitment) is only supported on edges sa , ab , and bt . It is not hard to see that without any additional commitment, the Stackelberg equilibrium involves the leader playing edge sa with probability $x_{sa} = 0.4$ and edge bt with probability $x_{bt} = 0.6$, and all other edges with probability $x_e = 0$. Note that in such a mixed strategy the follower chooses path a, b, t and the leader's payoff is 0.6. On the other hand, when the leader commits to providing additional incentive (or discount in the cost of a path) of $v_{sabt} = 0.2$ on path s, a, b, t , the follower best responds by choosing path s, a, b, t and the leader's payoff is 0.8.

Our main theorem in this section shows that there is a polynomial time algorithm for finding the Stackelberg equilibrium of this modified game.

Theorem 1. *There is a polynomial time algorithm for finding a Stackelberg equilibrium if one can solve the following problem in polynomial time: Given \vec{x} and value W , return $S \in \S$, such that $\sum_{e \in S} (-x_e + c_e) \leq -W$, or return "None" if no such $S \in \S$ exists.⁵*

At a high level, we show that a Stackelberg equilibrium, (\vec{x}^*, \vec{V}^*) , can be found by finding the optimal solution $(\vec{x}, \vec{0})$ (with no additional incentives) that involves maximizing the follower's payoff of the best response, and then providing enough incentive on one of the follower's actions. In

⁵An example of a game where this linear program can be solved efficiently is the shortest path game in Example 3.1.

particular, we choose to provide incentive on the specific $S \in \S$ that constitutes the best response of the follower to the mixed strategy $(\vec{0}, \vec{0})$.

For the first step of this proof, we consider the following LP, which can be efficiently solved by the separation oracle given in Theorem 1,

$$\begin{aligned} \max_{\vec{x}, W} \quad & W + \sum_{e \in E} x_e C_e \\ \forall S \in \S, \quad & \sum_{e \in S} (-x_e + c_e) \leq -W. \end{aligned} \tag{4}$$

Let \vec{x}^*, W^* be the solution to the above LP. Furthermore, let $S^* = \arg \max_{S \in \S} \sum_{e \in S} c_e$, and consider the incentive commitments $V_{S^*}^* = -W^* - \sum_{e \in S^*} (-x_e^* + c_e)$, and $V_0^* = 0$ for all $S \neq S^*$. That is, we provide enough incentive on set S^* such that it becomes the best response for the follower.

To prove Theorem 1, we first prove a lower bound on the incentive needed to make an action the best response of the follower.

Lemma 2. *Let (\vec{x}', \vec{V}') be any mixed strategy of the leader and let $S' = b(\vec{x}', \vec{V}')$ be the corresponding best response of the follower. Let $W' = -\max_{S \in \S} \sum_{e \in S} (-x'_e + c_e)$. We have,*

$$V_{S'}' \geq -W' - \sum_{e \in S'} (-x'_e + c_e).$$

Proof. Let $S'' = \arg \max_{S \in \S} \sum_{e \in S} (-x'_e + c_e)$ be one of the actions of the follower that corresponds to value $-W'$. Since all incentives are non-negative, we have that

$$U_F((\vec{x}', \vec{V}'), S'') = \sum_{e \in S''} (-x'_e + c_e) + V_{S''}' \geq -W'.$$

Therefore, S' is a best response to (\vec{x}', \vec{V}') only if

$$\sum_{e \in S'} (-x'_e + c_e) + V_{S'}' = U_F((\vec{x}', \vec{V}'), S') \geq U_F((\vec{x}', \vec{V}'), S'') \geq -W'.$$

This completes the proof. \square

proof of Theorem 1. Let (\vec{x}^*, W^*) be the solution to Equation 4. Let $S^* = \arg \max_{S \in \S} \sum_{e \in S} c_e$, and let $V_{S^*}^* = -W^* - \sum_{e \in S^*} (-x_e^* + c_e)$, and $V_S^* = 0$ for all $S \neq S^*$. It is clear that $b(\vec{x}^*, \vec{V}^*) = S^*$. Here, we show that (\vec{x}^*, \vec{V}^*) is indeed the optimal leader strategy.

For any leader strategy (\vec{x}', \vec{V}') , let $S' = b(\vec{x}', \vec{V}')$ be the follower's best response. Moreover, let $W' = -\max_{S \in \S} \sum_{e \in S} (-x'_e + c_e)$. We have

$$U_L((\vec{x}^*, \vec{V}^*), S^*) = \sum_{e \in S^*} x_e^* + \sum_{e \in E} x_e^* C_e^* - V_{S^*}^* \tag{5}$$

$$= \sum_{e \in S^*} x_e^* + \sum_{e \in E} x_e^* C_e^* + W^* + \sum_{e \in S^*} (-x_e^* + c_e) \tag{6}$$

$$= \sum_{e \in E} x_e^* C_e^* + W^* + \sum_{e \in S^*} c_e \tag{7}$$

$$\geq \sum_{e \in E} x'_e C'_e + W' + \sum_{e \in S'} c_e, \quad (8)$$

where the second equation is by the definition of $V_{S^*}^*$ and the last inequality follows by the fact that (\vec{x}', W') form a valid solution for the LP in Equation 4, for which (\vec{x}^*, W^*) is the optimal solution and the fact that S^* is chosen to maximize $\sum_{e \in S^*} c_e$.

Using Lemma 2 on the value of $V_{S'}'$, we have

$$U_L((\vec{x}', \vec{V}'), S') = \sum_{e \in S'} x'_e + \sum_{e \in E} x'_e C'_e - V_{S'}' \quad (9)$$

$$\leq \sum_{e \in S'} x'_e + \sum_{e \in E} x'_e C'_e + W' + \sum_{e \in S'} (-x'_e + c_e) \quad (10)$$

$$= \sum_{e \in E} x'_e C'_e + W' + \sum_{e \in S'} c_e. \quad (11)$$

Equations 8 and 11 complete the proof. \square

4 The Permuted Matching Game

In this section, we introduce a large but structured general-sum Stackelberg game, called PERMUTED MATCHING, and examine the computational complexity of computing its Stackelberg equilibrium. We show two sets of results for this game. In Section 4.1, we show that this problem is APX-hard. This implies that unlike zero-sum games, finding a Stackelberg equilibrium is computationally hard even if best-response oracles are provided. In Section B, we give an efficient $\frac{1}{12}$ approximation for this problem.

The PERMUTED MATCHING game is defined as follows. Consider the leader and follower, L and F. Consider a multigraph $G = (V, E)$ and a one-to-one mapping (permutation) $\pi : E \rightarrow E$. Note that π may take different values on parallel edges of a multi-graph. In the remainder of this section, we refer to a multi-graph G as a *graph*. In PERMUTED MATCHING, the set of pure strategies of both players is the set of all matchings in G . Given matchings M_L and M_F played by the leader and follower, respectively, we define

$$U_L(M_L, M_F) = |M_L \cap \pi(M_F)|, \text{ and } U_F(M_L, M_F) = |M_L \cap M_F|,$$

where for a set $S \subseteq E$, we define $\pi(S) = \{e \in S | \pi(e)\}$. Note that G and π are fixed and known to both players.

Let us highlight an important aspect of our hardness result in advance. As the next observation shows, the strategy space of the players in PERMUTED MATCHING, though large, is very structured. At a high level, the reward of each player is a linear function of the action of the other and each player can efficiently optimize a linear function over the strategy space of the other player, for example, each player can compute a best-response to a mixed strategy of the other.

Observation 1. *There is a polynomial time algorithm such that for every vector $\vec{w} \in [0, 1]^{|E|}$ finds a strategy of the players whose corresponding representation vector \vec{v} maximizes $\vec{v} \cdot \vec{w}$.*

Sketch. This problem reduces to computing a maximum weighted matching of a graph with edge weights w_e for all $e \in E$, which can be performed efficiently Corman (2009). \square

In a zero-sum game, existence of such a structure leads to efficient algorithms for computing the Nash or Stackelberg equilibria Ahmadinejad et al. (2016); Dudík et al. (2017); Kalai and Vempala (2005). On the other hand, our APX-hardness result for the PERMUTED MATCHING game shows that existence of this structure does not necessarily lead to efficient algorithms for computing Stackelberg equilibria in general-sum games. With this in mind, we present our hardness results next.

4.1 Hardness of approximation

In this section, we show that it is impossible to approximate a Stackelberg equilibrium of the PERMUTED MATCHING game in polynomial time within an arbitrarily small constant factor unless $P=NP$.

Before we proceed to the proof, we define an auxiliary problem and show a hardness result for this problem. Then, we take advantage of this hardness result and show that computing a Stackelberg equilibrium of PERMUTED MATCHING is APX-hard. We call the intermediary problem π -TRANSFORMATION-IDENTICAL-MATCHING and define it as follows:

π -TRANSFORMATION-IDENTICAL-MATCHING

INPUT: An unweighted undirected graph $G = (V, E)$,
and a permutation $\pi : E \rightarrow E$.

OUTPUT: A matching M of G that maximizes $|M \cap \pi(M)|$.

For an instance I of π -TRANSFORMATION-IDENTICAL-MATCHING, we denote by $\text{Opt}(I)$ the optimal solution to I and refer to the value of this solution by $\text{Val}(I)$.

We show that π -TRANSFORMATION-IDENTICAL-MATCHING has a *hard gap* at *gap location* 1. That is, it is NP-hard to decide whether for a given graph G with n vertices and a function π , the solution of the π -TRANSFORMATION-IDENTICAL-MATCHING problem is exactly equal to $n/2$ or at most $(1 - \epsilon)n/2$ for some $\epsilon > 0$.

Lemma 3. *There exists an $\epsilon > 0$ such that it is NP-hard to decide whether the solution of the π -TRANSFORMATION-IDENTICAL-MATCHING problem is exactly equal to $n/2$ or less than $(1 - \epsilon)n/2$ where n is the number of the vertices of the input graph.*

We defer the proof of 3 to the end of this section and first show how this lemma can be used to prove the main result of this section.

Theorem 4. *Computing a Stackelberg equilibrium of PERMUTED MATCHING is APX-hard.*

Proof. More generally, we show that approximating a Stackelberg equilibrium of the PERMUTED MATCHING game has a hard gap at gap location 1. This immediately implies a hardness of approximation. We show this by a reduction from the π -TRANSFORMATION-IDENTICAL-MATCHING problem. Suppose we are given an instance $I = \langle G, \pi \rangle$ of the π -TRANSFORMATION-IDENTICAL-MATCHING problem and wish to decide for some $\epsilon' > 0$, whether the solution of this problem achieves a value that is exactly $n/2$ or is bounded above by $(1 - \epsilon')n/2$ where n is the size of G . Based on I , we construct an instance $\text{Cor}(I)$ of the PERMUTED MATCHING game with the same graph G and permutation π and seek to find a Stackelberg equilibrium in this game. Note that by definition, $\text{Val}(I)$

is equal to $n/2$ if and only if G contains a perfect matching that is identical to its π -transformation. Otherwise, $\text{Val}(I)$ is at most $(1 - \epsilon')n/2$ and thus any matching of G shares no more than $(1 - \epsilon')n/2$ edges with its π -transformation.

Since for small enough ϵ' , it is NP-hard to distinguish the two cases (Lemma 3), we show that it is NP-hard to approximate a Stackelberg equilibrium of the leader in $\text{Cor}(I)$. If $\text{Val}(I) = n/2$, then there exists a perfect matching in G that is identical to its π -transformation. Thus, if both players play this matching in $\text{Cor}(I)$, they both get a payoff of $n/2$. Notice that $n/2$ is the maximum possible payoff for any player in this game, therefore, such a strategy pair is a Stackelberg equilibrium. Hence, in case $\text{Val}(I) = n/2$, the leader achieves a payoff of $n/2$ in a Stackelberg equilibrium of the corresponding PERMUTED MATCHING game.

Now, suppose for $\epsilon < \epsilon'/13$ we have a $1 - \epsilon$ approximation solution for $\text{Cor}(I)$. If $\text{Val}(I) = n/2$, then the payoff of the leader in an exact solution of $\text{Cor}(I)$ is $n/2$ and therefore a $1 - \epsilon$ approximation solution guarantees a payoff of at least $n(1 - \epsilon)/2$ for the leader. Let the strategies of the leader and follower be X and y in such a solution. Therefore, $U_L(X, y) \geq n(1 - \epsilon)/2$. Notice that X may be a mixed strategy, but we can assume w.l.g that y is a pure strategy since there always exists a best response for the follower which is pure. Also, let y^* be the π -transformation of strategy y . Let for two matchings x and y , $\text{common}(x, y)$ denote the number of edges that x and y have in common and define $\text{dist}(x, y) = |x| + |y| - 2\text{common}(x, y)$. Recall that the payoff of the leader in this game can be formulated as $\mathbb{E}_{x \sim X}[\text{common}(x, y^*)]$. Since this value is at least $n(1 - \epsilon)/2$ we have:

$$\mathbb{E}_{x \sim X}[\text{common}(x, y^*)] = U_L(X, y) \geq n(1 - \epsilon)/2$$

and thus

$$\begin{aligned} \mathbb{E}_{x \sim X}[\text{dist}(x, y^*)] &= \mathbb{E}_{x \sim X}[|x| + |y^*| - 2\text{common}(x, y^*)] \\ &\leq \mathbb{E}_{x \sim X}[n - 2\text{common}(x, y^*)] \\ &= n - 2\mathbb{E}_{x \sim X}[\text{common}(x, y^*)] \\ &\leq n - 2n(1 - \epsilon)/2 \\ &= n\epsilon. \end{aligned} \tag{12}$$

Inequality (12) shows that y^* is very similar (in expectation) to a random matching drawn from strategy X . This intuitively implies that pure strategies of X should have a considerable amount of edges in common. It follows from the definition that for three matchings x , y , and z we have $\text{dist}(x, z) \leq \text{dist}(x, y) + \text{dist}(y, z)$. Therefore, we have

$$\begin{aligned} \mathbb{E}_{x \sim X, x' \sim X}[\text{dist}(x, x')] &\leq \mathbb{E}_{x \sim X, x' \sim X}[\text{dist}(x, y^*) + \text{dist}(y^*, x')] \\ &= \mathbb{E}_{x \sim X, x' \sim X}[\text{dist}(x, y^*) + \text{dist}(x', y^*)] \\ &= \mathbb{E}_{x \sim X}[\text{dist}(x, y^*)] + \mathbb{E}_{x' \sim X}[\text{dist}(x', y^*)] \\ &= 2\mathbb{E}_{x \sim X}[\text{dist}(x, y^*)] \\ &\leq 2\epsilon n \end{aligned} \tag{13}$$

Recall that the payoff of the follower is determined by the number of edges his matching shares with that of the leader. Moreover, since $U_L(X, y) \geq n(1 - \epsilon)/2$, this implies that $\mathbb{E}_{x \sim X}[|x|] \geq n(1 - \epsilon)/2$. What Inequality (13) implies is that if the follower plays X instead of y , he gets a payoff of at least $\mathbb{E}_{x \sim X}[|x|] - 2\epsilon n \geq (1 - 5\epsilon)n/2$ against X . In other words $U_F(X, X) \geq (1 - 5\epsilon)n/2$. Since y is a best

response of the follower against the leader's strategy, we have $U_F(X, y) \geq U_F(X, X) \geq (1 - 5\epsilon)n/2$ and thus

$$\begin{aligned}\mathbb{E}_{x \sim X}[\text{common}(x, y)] &= U_F(X, y) \\ &\geq U_F(X, X) \\ &= \mathbb{E}_{x \sim X, x' \sim X}[\text{common}(x, x')] \\ &\geq (1 - 5\epsilon)n/2.\end{aligned}$$

Hence

$$\begin{aligned}\mathbb{E}_{x \sim X}[\text{dist}(x, y)] &= \mathbb{E}_{x \sim X}[|x| + |y| - 2\text{common}(x, y)] \\ &\leq \mathbb{E}_{x \sim X}[n - 2\text{common}(x, y)] \\ &= n - 2\mathbb{E}_{x \sim X}[\text{common}(x, y)] \\ &\leq n - 2(1 - 5\epsilon)n/2 \\ &\leq 5\epsilon n.\end{aligned}\tag{14}$$

Combining Inequalities (12) and (14) yields

$$\text{dist}(y, y^*) \leq \mathbb{E}_{x \sim X}[\text{dist}(x, y)] + \mathbb{E}_{x \sim X}[\text{dist}(x, y^*)] \leq 6\epsilon n.$$

Therefore, we have $\text{common}(y, y^*) \geq |y^*| - 6\epsilon n$ and since $|y^*| \geq (1 - \epsilon)n/2$ we have $\text{common}(y, y^*) \geq (1 - 13\epsilon)n/2 > (1 - \epsilon')n/2$. If $\text{Val}(\mathbf{l}) \neq n/2$, then $\text{Val}(\mathbf{l})$ is bounded by $(1 - \epsilon')n/2$. Therefore, $\text{common}(y, y^*) > (1 - \epsilon')n/2$ holds if and only if $\text{Val}(\mathbf{l}) = n/2$. Thus, an approximation solution for $\text{Cor}(\mathbf{l})$ within a factor $(1 - \epsilon) > (1 - \epsilon'/13)$ can be used to decide if the solution of \mathbf{l} is $n/2$ or bounded by $(1 - \epsilon')n/2$. This implies a hard gap for the π -TRANSFORMATION-IDENTICAL-MATCHING problem at gap location 1. \square

All that remains is to prove the statement of Lemma 3 and that completes the proof Theorem 4.

proof of Lemma 3. We show this lemma by a reduction from the MAXIMUM 3D MATCHING problem. In the MAXIMUM 3D MATCHING problem, we are given a hypergraph G whose vertices are divided into three parts A , B , and C . Every hyper-edge of G is a triple (a, b, c) of the vertices such that $a \in A$, $b \in B$, and $c \in C$ hold. A matching in this graph is a subset of the hyper-edges that do not share any vertices. The goal of the problem is to find a matching with the maximum number of hyper-edges. Petrank (1994) showed that the MAXIMUM 3D MATCHING problem has a hard gap at gap location 1.

We show via a reduction that π -TRANSFORMATION-IDENTICAL-MATCHING is harder than the MAXIMUM 3D MATCHING problem. This implies a similar hardness result for the π -TRANSFORMATION-IDENTICAL-MATCHING problem. To this end, suppose we are given an instance \mathbf{l} of the MAXIMUM 3D MATCHING problem. Let the hyper-graph of this instance be G and its vertices lie in three parts A , B , and C such that every hyper-edge of the graph contains a vertex of each part. We construct a bipartite graph G' and a function π based on G as follows:

- G' contains two independent parts X and Y , each of which contains an endpoint of every edge of the graph.

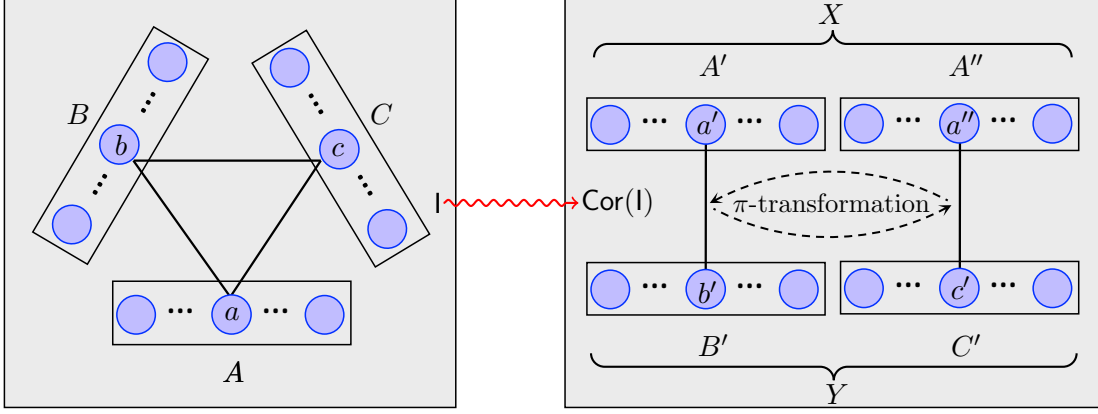


Figure 2: The figure on the left shows an instance I of the MAXIMUM 3D MATCHING problem and the figure on the right shows the corresponding instance $\text{Cor}(I)$ of the π -TRANSFORMATION-IDENTICAL-MATCHING problem. Solid segments show the edges of the graphs and dashed segments specify the corresponding π -transformations of the edges.

- $X = A' \cup A''$ where A' and A'' are two copies of A . That is, for every vertex $a \in A$, we put two vertices a' and a'' in X (in A' and A'' respectively).
- $Y = B' \cup C'$ where B' is a copy of B and C' is a copy of C . That is, for every vertex $b \in B$ and $c \in C$ we put a vertex b' and c' in Y (in part B' and C' , respectively).
- For every edge (a, b, c) of G , we put two edges (a', b') and (a'', c') in G' . Moreover, we set $\pi((a', b')) = (a'', c')$ and $\pi((a'', c')) = (a', b')$. Note that, there may be multiple edges between two vertices of multi-graph G with different π values.

Now, we argue that for every 3-dimensional matching M of G , there exists a matching M' of G' such that $|M' \cap \pi(M')| = 2|M|$ and viceversa. To this end, suppose M is a 3-dimensional matching of G . Now, we set $M' = \{(a, b, c) \in E(G) | (a', b')\} \cup \{(a, b, c) \in E(G) | (a'', c')\}$. Since M is a matching, no two edges of M' share a vertex and thus M' is also a matching. Moreover, for every edge in M' , its π transformation is also included in M' and therefore $M' = \pi(M')$. Hence $|M' \cap \pi(M')| = |M'| = 2|M|$. A similar argument shows that for any matching M' of G' such that $|M' \cap \pi(M')| = 2k$, there exists a matching of size k in G . Therefore, the problem of finding a maximum 3-dimensional matching of G reduces to finding a matching of G' that shares the maximum number of edges with its π -transformation. Since the MAXIMUM 3D MATCHING problem has a hard gap at gap location 1, so does the π -TRANSFORMATION-IDENTICAL-MATCHING problem. Figure 2 describes the reduction mentioned above. \square

References

AmirMahdi Ahmadinejad, Sina Dehghani, MohammadTaghi Hajiaghay, Brendan Lucier, Hamid Mahini, and Saeed Seddighin. 2016. From Duels to Battlefields: Computing Equilibria of Blotto and Other Games. In *Thirtieth AAAI Conference on Artificial Intelligence*.

- Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems- Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 57–64.
- Soheil Behnezhad, Sina Dehghani, Mahsa Derakhshan, MohammadTaghi HajiAghayi, and Saeed Seddighin. 2016. Faster and Simpler Algorithm for Optimal Strategies of Blotto Game. *arXiv preprint arXiv:1612.04029* (2016).
- Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, and Aleksandrs Slivkins. 2017. A Polynomial Time Algorithm for Spatio-Temporal Security Games. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. ACM, 697–714.
- Vincent Conitzer and Tuomas Sandholm. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM Conference on Economics and Computation (EC)*. ACM, 82–90.
- Thomas H Cormen. 2009. *Introduction to algorithms*. MIT press.
- Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. 2017. Oracle-Efficient Online Learning and Auction Design. In *Proceedings of the 58th Symposium on Foundations of Computer Science (FOCS)*.
- EnergySage. 2019. Using the solar investment tax credit for energy storage. (2019). <https://www.energysage.com/solar/solar-energy-storage/energy-storage-tax-credits-incentives/>
- Yoav Freund and Robert E Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*. Springer, 23–37.
- Jugal Garg, Albert Xin Jiang, and Ruta Mehta. 2011. Bilinear games: Polynomial time algorithms for rank based subclasses. In *International Workshop on Internet and Network Economics*. Springer, 399–407.
- James Hannan. 1957. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games* 3 (1957), 97–139.
- Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. 2011. Dueling algorithms. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 215–224.
- Adam Kalai and Santosh Vempala. 2005. Efficient algorithms for online decision problems. *J. Comput. System Sci.* 71, 3 (2005), 291–307.
- Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. 2009. Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems- Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 689–696.

- Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2011a. Security games with multiple attacker resources. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 273.
- Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. 2011b. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *J. Artif. Intell. Res. (JAIR)* 41 (2011), 297–327.
- Joshua Letchford and Vincent Conitzer. 2010. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 83–92.
- Joshua Letchford and Yevgeniy Vorobeychik. 2011. Computing randomized security strategies in networked domains. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.
- Yuqian Li, Vincent Conitzer, and Dmytro Korzhyk. 2016. Catcher-Evader Games. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 329–337. <http://www.ijcai.org/Abstract/16/054>
- Nick Littlestone and Manfred K Warmuth. 1994. The weighted majority algorithm. *Information and computation* 108, 2 (1994), 212–261.
- Erez Petrank. 1994. The hardness of approximation: Gap location. *Computational Complexity* 4, 2 (1994), 133–157.
- Milind Tambe. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.
- U.S. Department of Energy. 2019. Residential Renewable Energy Tax Credit. (2019). <https://www.energy.gov/savings/residential-renewable-energy-tax-credit>
- John Von Neumann and Oskar Morgenstern. 1945. *Theory of games and economic behavior*. Princeton University Press Princeton, NJ.
- Sinong Wang and Ness Shroff. 2017. Security Game with Non-additive Utilities and Multiple Attacker Resources. *arXiv preprint arXiv:1701.08644* (2017).
- Haifeng Xu. 2016. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 497–514.
- Haifeng Xu, Fei Fang, Albert Xin Jiang, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. 2014. Solving zero-sum security games in discretized spatio-temporal domains. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

	L	R	Strategy	Row player's payoff
U	1, 1	10, 0	Stackelberg: $(\frac{1}{2}, \frac{1}{2}) \times (0, 1)$	7.5
D	0, 0	5, 1	Nash: $(1, 0) \times (1, 0)$	1
			Minimax: $(1, 0) \times (\frac{1}{2}, \frac{1}{2})$	5.5

Table 1: A general-sum game matrix where Nash, Minimax, and Stackelberg equilibria are different.

A Stackelberg, Nash, Minimax Equilibria in General-Sum Games

In general-sum games, Stackelberg, Nash, and Minimax equilibria diverge. Let us demonstrate this by the game matrix in Figure 1. In the Stackelberg equilibrium, the row player (leader) commits to mixed strategy $(\frac{1}{2}, \frac{1}{2})$ and the column player (follower) responds by $(0, 1)$, i.e., playing action R deterministically. In this case, the leader gets utility 7.5. In the Nash equilibrium, however, both players move simultaneously and play strategy $(1, 0)$. In this case, the row player gets utility 1. In the Minimax setting, each player maximizes its own utility assuming the worst-case.⁶ In this case, the row player plays $(1, 0)$ and the column player plays $(\frac{1}{2}, \frac{1}{2})$, with the row player gaining utility 5.5.

B A 1/12 Approximate Solution for The Permuted Matching Game

In Section 4.1, we showed that approximating a Stackelberg Equilibrium of the PERMUTED MATCHING game within an arbitrarily small error is NP-hard. We complement this result by presenting a polynomial time algorithm for computing a 1/12-approximately optimal solution. In other words, our algorithm provides a strategy X for the leader that achieves at least a 1/12 fraction of the payoff of a Stackelberg equilibrium against the best response of his opponent.

We first devise a greedy algorithm for finding two matchings x and x' such that x shares as many edges as possible with the π -transformation of x' , i.e., maximizing $|x \cap \pi(x')|$. To this end, we begin with two empty matchings x and x' and iteratively choose two edges e and e' of the graph such that $\pi(e') = e$, e is not adjacent⁷ to any edge of x , and e' is not adjacent to any edge of x' . We then update the matchings by adding e to x and e' to x' .

Algorithm 1: Maximizing the number of edges shared between x and $\pi(x')$

Data: A graph G and a permutation π over the edges of G .

Result: Two matchings x and x' .

1 $x \leftarrow \emptyset, x' \leftarrow \emptyset;$

2 **while** there exist edges e and e' s.t.

• $e = \pi(e')$

• e does not share any vertices with x and e' does not share any vertices with x'

3 **do** Add e to x and e' to x' ;

4 **Return** (x, x') ;

⁶This is equivalent to each player playing their side of a Nash equilibrium when they perceive the game as a zero-sum game.

⁷two edges are adjacent if they share a common vertex.

We show that $U_L(x, x')$ for x, x' computed by Algorithm 1 is at least $1/4$ of the leader's payoff in a Stackelberg equilibrium.

Observation 2. *Let x and x' be the matchings determined by Algorithm 1. $|x \cap \pi(x')|$ is at least $1/4$ fraction of the optimal payoff the leader gets in any Stackelberg Equilibria.*

Proof. Recall that the payoff of the leader is the number of edges his matching has in common with the π -transformation of the follower's matching. This value is always upper bounded by $\max_{y, y'} |y \cap \pi(y')|$. Assume to the contrary that a pair of matchings y, y' exists such that $|y \cap \pi(y')| > 4|x \cap \pi(x')|$. Note that by the choices of Algorithm 1, we have $|x| = |x'| = |x \cap \pi(x')|$. Moreover, there are at least $4|x \cap \pi(x')|$ edges in y' whose π -transformations appear in y . Notice that for every edge e' in x' , there are at most four such edges in y' that either share an end-point with e' or their π -transformations share an endpoint with $\pi(e')$. Thus, there exists an edge e' in y such that neither e' shares a vertex with the vertices of x' nor $\pi(e')$ shares an endpoint with the vertices of x . This is a contradiction, as these edges could have been added to x and x' by Algorithm 1. Therefore, $\max_{y, y'} |y \cap \pi(y')| \leq 4|x \cap \pi(x')|$. This completes the proof. \square

Let $\epsilon > 0$ be an arbitrarily small positive number and X be a strategy of the leader that plays matching x with probability $1/3 - \epsilon$ and matching x' with probability $2/3 + \epsilon$. Moreover, let y be a best response of the follower against X . Based on Observation 2, we show that $U_L(X, y)$ is at least $(1 - 3\epsilon)/12$ fractional of the optimal payoff the leader can get in any SE.

Theorem 5. *Let X and y be the strategies of the leader and the follower defined above. $U_L(X, y)$ is at least $(1 - 3\epsilon)/12$ fraction of the optimal payoff the leader achieves in any Stackelberg equilibrium.*

Proof. Without loss of generality, y is a pure strategy. We first show that y contains all edges of x' . Assume on the contrary that an edge e' of x' does not appear in y . If we add this edge to y and remove the edges of y that share an endpoint with e' then the payoff of the follower changes as follows: Because X plays x' with probability $2/3 + \epsilon$ then e' appears in his strategy with probability $2/3 + \epsilon$ and thus the payoff of the follower is increased by $2/3 + \epsilon$. Moreover, none of the crossing edges of e' with edges of y are in x' and thus appear in the leader's strategy with probability at most $1/3 - \epsilon$. This incurs a loss of at most $2(1/3 - \epsilon) < 2/3 + \epsilon$. Thus, this change improves the payoff of the follower which shows y is not a best response against X which is a contradiction.

Note that X plays x with probability $1/3 - \epsilon$ and y contains every edge of x' . Let U be the maximum payoff of the leader in any Stackelberg equilibrium. We have

$$\begin{aligned} U_L(X, y) &\geq (1/3 - \epsilon)U_L(x, x') = (1/3 - \epsilon)|x \cap \pi(x')| \\ &= \frac{1}{3}(1 - 3\epsilon)|x \cap \pi(x')| \geq \frac{1}{12}(1 - 3\epsilon)U, \end{aligned}$$

where the last step holds by Observation 2. \square