

# Strategic Voting in the Context of Stable-Matching of Teams

Leora Schmerler and Noam Hazon and Sarit Kraus

## Abstract

In the celebrated stable-matching problem, there are two sets of agents  $M$  and  $W$ , and the members of  $M$  only have preferences over the members of  $W$  and vice versa. It is usually assumed that each member of  $M$  and  $W$  is a single entity. However, there are many cases in which each member of  $M$  or  $W$  represents a team that consists of several individuals with common interests. For example, students may need to be matched to professors for their final projects, but each project is carried out by a team of students. Thus, the students first form teams, and the matching is between teams of students and professors.

When a team is considered as an agent from  $M$  or  $W$ , it needs to have a preference order that represents it. A voting rule is a natural mechanism for aggregating the preferences of the team members into a single preference order. In this paper, we investigate the problem of strategic voting in the context of stable-matching of teams. Specifically, we assume that members of each team use the Borda rule for generating the preference order of the team. Then, the Gale-Shapley algorithm is used for finding a stable-matching, where the set  $M$  is the proposing side. We show that the single-voter manipulation problem can be solved in polynomial time, both when the team is from  $M$  and when it is from  $W$ . We show that the coalitional manipulation problem is computationally hard, but it can be solved approximately both when the team is from  $M$  and when it is from  $W$ .

## 1 Introduction

Matching is the process in which agents from different sets are matched with each other. The theory of matching originated with the seminal work of Gale and Shapley [9], and since then intensive research has been conducted in this field. Notably, the theory of matching has also been successfully applied to many real-world applications including college admissions and school matching [1], matching residents to hospitals [17], and kidney exchange [18]. A very common matching problem, which is also the problem that was studied by Gale and Shapley in their original paper, is the *stable-matching* problem. In this problem there are two equally sized disjoint sets of agents,  $M$  and  $W$ , and the members of  $M$  have preferences over only the members of  $W$ , and vice versa. The goal is to find a stable bijection (i.e., matching) from the agents of  $M$  to the agents of  $W$ , where the stability requirement is that no pair of agents prefers a match with each other over their matched partners. Many works have analyzed this setting, and they assume that each member of the sets  $M$  and  $W$  represents a single agent. However, there are many cases in which each member of  $M$  or  $W$  represents more than one individual [14].

For example, suppose that teams of students need to be matched with professors who will serve as their advisors in their final projects. It is common that students form their teams based on friendship connections and common interests and then approach the professors. Therefore, each team is considered to be a single agent for the matching process: the professors may have different preferences regarding which team they would like to mentor, and the teams may have preferences regarding which professor they would like as their mentor. Clearly, even though the team is considered to be a single agent for the matching process, it is still composed of several students, and they may have different opinions regarding the

appropriate mentor for their team. Thus, every team needs a mechanism that aggregates the students' opinions and outputs a single preference order that represents the team for the matching process, and a voting rule is a natural candidate.

Indeed, voters might benefit from reporting rankings different from their true ones, and this problem of manipulation also exists in the context of matching. For example, suppose that there are 4 possible professors, denoted by  $p_1, p_2, p_3$  and  $p_4$  and 4 teams. Now, suppose that one of the students, denoted  $r$ , who is a member of one of the teams, prefers  $p_1$  over  $p_2$ ,  $p_2$  over  $p_3$ , and  $p_3$  over  $p_4$ . It is possible that  $r$  will gain an (unauthorized) access to the preferences of the professors and to the preferences of the other teams. Since the matching algorithm is usually publicly known,  $r$  might be able to reason that  $p_3$  is matched with his team, but if  $r$  votes strategically and misreports his preferences then  $p_2$  will be matched with his team.

In this paper, we investigate the problem of strategic voting in the context of stable-matching of teams. We assume that the members of each team use the Borda rule as a *social welfare function (SWF)*, which outputs a complete preference order. This preference order represents the team for the matching process. The agents then use the Gale-Shapley (GS) algorithm for finding a stable-matching. In the GS algorithm, one set of agents makes proposals to the other set of agents, and it is assumed that  $M$  is the proposing side and  $W$  is the proposed-to side. The proposing side and proposed-to side are commonly referred to as men and women, respectively. Note that the GS algorithm treats the men and women differently. Therefore, every manipulation problem in the context of stable-matching has two variants: one in which the teams are from the men's side, and another one in which the teams are from the women's side. Moreover, we analyze both manipulation by a single voter and coalitional manipulation. In a single voter manipulation, the goal is to find a preference order for a single manipulator such that his team will be matched by the GS algorithm with a specific preferred agent. In the coalitional manipulation setting, there are several voters who collude and coordinate their votes so that an agreed upon agent will be matched with their team.

We begin by studying manipulation from the men's side, and show that the single voter manipulation problem can be solved in polynomial time. We then analyze the coalitional manipulation problem, and show that the problem is computationally hard. However, we provide a polynomial-time algorithm with the following guarantee: given a manipulable instance with  $|R|$  manipulators, the algorithm finds a successful manipulation with at most one additional manipulator. We then study manipulation from the women's side. Manipulation here is more involved, and we propose different algorithms, but with almost the same computational complexity as in manipulation from the men's side. That is, the single voter manipulation problem can be solved in polynomial time, and the coalitional manipulation problem is computationally hard. Indeed, we provide a polynomial-time algorithm with the following guarantee: given a manipulable instance with  $|R|$  manipulators, the algorithm finds a successful manipulation with at most two additional manipulators.

The contribution of this work is twofold. First, it provides an analysis of a voting manipulation in the context of stable-matching of teams, a problem that has not been investigated to date. Second, our work concerns the manipulation of Borda as an SWF, which has scarcely been investigated.

## 2 Related Work

The computational analysis of voting manipulation has been vastly studied in different settings. We refer the reader to the survey provided by Faliszewski and Procaccia [8], and the more recent survey by Conitzer and Walsh [4]. However, most of the works on voting

manipulation analyze the problem with no actual context, and where a voting rule is used to output one winning candidate or a set of tied winning candidates (i.e., a social choice function). In this work, we investigate manipulation of Borda as a SWF, which outputs a complete preference order of the candidates, and analyze it within the context of stable-matching.

Indeed, there are a few papers that investigate the manipulation of SWFs. The first work that directly deals with the manipulation of SWF was by Bossert and Storcken [3], who assumed that a voter prefers one order over another if the former is closer to her own preferences than the latter according to the Kemeny distance. Bossert and Sprumont [2] assumed that a voter prefers one order over another if the former is strictly between the latter and the voter’s own preferences. Built on this definition, their work studies three classes of SWF that are not prone to manipulation (i.e., strategy-proof). Dogan and Lainé [6] characterized the conditions to be imposed on SWFs so that if we extend the preferences of the voters to preferences over orders in specific ways, the SWFs will not be prone to manipulation. Our work also investigates the manipulation of SWF, but we analyze the SWF in the specific context of stable-matching. Therefore, unlike all of the above works, the preferences of the manipulators are well-defined and no additional assumptions are needed. The work that is closest to ours is that of Schmerler and Hazon [19]. They assume that a positional scoring rule is used as a SWF, and study the manipulation of the SWF in the context of negotiation.

The strategic aspects of the GS algorithm have previously been studied in the literature. It was first shown that reporting the true preferences is a weakly dominant strategy for men, but women may have an incentive to misreport their preferences [7, 16, 10]. Teo et al. [22] provided a polynomial-time algorithm for computing the optimal manipulation by a woman. Shen et al. [21] generalized this result to manipulation by a coalition of women. For the proposing side, Dubins and Freedman [7] investigated the strategic actions of a coalition of men, and proved that there is no manipulation that is a strict improvement for every member of the coalition. Huang [13] studied manipulation that is a weak improvement for every member of a coalition of men. Hosseini et al. [11] introduced a new type of strategic action: manipulation through an accomplice. In this manipulation, a man misreports his preferences in behalf of a woman, and Hosseini et al. provided a polynomial time algorithm for computing an optimal accomplice manipulation, and they further generalized this model in [12]. All of these works consider the manipulation of the GS algorithm, while we study the manipulation of Borda as a SWF. Indeed, the output of the SWF is used (as part of the input) for the GS algorithm. As an alternative to the GS algorithm, Pini et al. [15] show how voting rules which are NP-hard to manipulate can be used to build stable-matching procedures, which are themselves NP-hard to manipulate.

### 3 Preliminaries

We assume that there are two equally sized disjoint sets of agents,  $M$  and  $W$ . Let  $k = |M| = |W|$ . The members of  $M$  have preferences over only the members of  $W$ , and vice versa. The preference of each  $m \in M$ , denoted by  $\succ_m$ , is a strict total order over the agents in  $W$ . The preference profile  $\succ_M$  is a vector  $(\succ_{m_1}, \succ_{m_2}, \dots, \succ_{m_k})$ . The preference order  $\succ_w$  and the preference profile  $\succ_W$  are defined analogously. We will refer to the agents of  $M$  as men and to the agents of  $W$  as women.

A matching is a mapping  $\mu : M \cup W \rightarrow M \cup W$ , such that  $\mu(m) \in W$  for all  $m \in M$ ,  $\mu(w) \in M$  for all  $w \in W$ , and  $\mu(m) = w$  if and only if  $\mu(w) = m$ . A stable-matching is a matching in which there is no blocking pair. That is, there is no man  $m$  and woman  $w$  such that  $w \succ_m \mu(m)$  and  $m \succ_w \mu(w)$ . The GS algorithm finds a stable-matching, and

it works as follows. There are multiple rounds, and each round is composed of a proposal phase followed by a rejection phase. In a proposal phase, each unmatched man proposes to his favorite woman from among those who have not yet rejected him (regardless of whether the woman is already matched). In the rejection phase, each woman tentatively accepts her favorite proposal and rejects all of the other proposals. The algorithm terminates when no further proposals can be made. Let  $o(w)$  be the set of men that proposed to  $w$  in one of the rounds of the GS algorithm.

In our setting, (at least) one of the agents of  $M$  ( $W$ ) is a team that runs an election for determining its preferences. That is, there is a man  $\hat{m}$  (woman  $\hat{w}$ ), which is associated with a set of voters,  $V$ . The preference of each  $v \in V$ , denoted by  $\ell_v$ , is a strict total order over  $W$  ( $M$ ). The preference profile  $\mathcal{L}$  is a vector of the preference orders for each  $v \in V$ . The voters use the *Borda* rule as a SWF, denoted by  $\mathcal{F}$ , which is a mapping of the set of all preference profiles to a single strict preference order. Specifically, in the Borda rule, each voter  $v$  awards the candidate that is placed in the top-most position in  $\ell_v$  a score of  $k - 1$ , the candidate in the second-highest position in  $\ell_v$  a score of  $k - 2$ , etc. Then, for the output of  $\mathcal{F}$ , the candidate with the highest aggregated score is placed in the top-most position, the candidate with the second-highest score is placed in the second-highest position, etc. Since ties are possible, we assume that a lexicographical tie-breaking rule is used. Note that the output of  $\mathcal{F}$  is the preference order of  $\hat{m}$  ( $\hat{w}$ ). That is,  $\succ_{\hat{m}} = \mathcal{F}(\mathcal{L})$ , and  $\succ_{\hat{w}}$  is defined analogously.

Recall that the GS algorithm finds a stable matching, given  $\succ_M$  and  $\succ_W$ . Given a man  $m \in M$ , let  $\succ_{M-m}$  be the preference profile of all of the men besides  $m$ , and  $\succ_{W-w}$  is defined analogously. We consider a setting in which the input for the GS algorithm is  $\succ_{M-\hat{m}}, \succ_{\hat{m}}$ , and  $\succ_W$ , and thus  $\mu(\hat{m})$  is the spouse that is the match of  $\hat{m}$  according to the output of the GS algorithm. We also consider a setting in which the input for the GS algorithm is  $\succ_{W-\hat{w}}, \succ_{\hat{w}}$  and  $\succ_M$ , and thus  $\mu(\hat{w})$  is the spouse that is the match of  $\hat{w}$  according to the output of the GS algorithm. In some circumstances, we would like to examine the output of the GS algorithm for different possible preference orders that represent a man  $m \in M$ . We denote by  $\mu_x(m, \succ)$  the spouse that is the match of  $m$  when the input for the GS algorithm is  $\succ_{M-m}, \succ$  (instead of  $\succ_m$ ), and  $\succ_W$ . We define  $\mu_x(w, \succ)$  and  $o_x(w, \succ)$  similarly.

We study the setting in which there exists a manipulator  $r$  among the voters associated with a man  $\hat{m}$  (woman  $\hat{w}$ ), and his (her) preference order is  $\ell_r$ . The preference order that represents  $\hat{m}$  ( $\hat{w}$ ) is thus  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ . We also study the setting in which there is a set of  $R$  manipulators, their preference profile is  $\mathcal{L}_R = \{\ell_1, \ell_2, \dots, \ell_{|R|}\}$ , and the preference order that represents  $\hat{m}$  ( $\hat{w}$ ) is thus  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ . For clarity purposes we slightly abuse notation, and write  $\mu(\hat{m}, \ell_r)$  for denoting the spouse that is the match of  $\hat{m}$  according to the output of the GS algorithm, given that its input is  $\succ_{M-\hat{m}}, \mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ , and  $\succ_W$ . We define  $\mu(\hat{w}, \ell_r)$ ,  $o(\hat{w}, \ell_r)$ ,  $\mu(\hat{m}, \mathcal{L}_R)$ ,  $\mu(\hat{w}, \mathcal{L}_R)$  and  $o(\hat{w}, \mathcal{L}_R)$  similarly.

Let  $s(c, \ell_v)$  be the score of candidate  $c$  from  $\ell_v$ . Similarly, let  $s(c, \mathcal{L})$  be the total score of candidate  $c$  from  $\mathcal{L}$ , i.e.,  $s(c, \mathcal{L}) = \sum_{v \in V} s(c, \ell_v)$ . Similarly,  $s(c, \mathcal{L}, \ell_r) = \sum_{v \in V} s(c, \ell_v) + s(c, \ell_r)$ , and  $s(c, \mathcal{L}, \mathcal{L}_R) = \sum_{v \in V} s(c, \ell_v) + \sum_{r \in R} s(c, \ell_r)$ . Since we use a lexicographical tie-breaking rule, we write that  $(c, \ell) > (c', \ell')$  if  $s(c, \ell) > s(c', \ell')$  or  $s(c, \ell) = s(c', \ell')$  but  $c$  is preferred over  $c'$  according to the lexicographical tie-breaking rule. We define  $(c, \mathcal{L}, \ell) > (c', \mathcal{L}, \ell')$  and  $(c, \mathcal{L}, \mathcal{L}_R) > (c', \mathcal{L}, \mathcal{L}_R)$  similarly. Note that due to space constraints, almost all of the proofs are deferred to the full version of the paper [20].

## 4 Men's Side

We begin by considering the variant in which a specific voter, or a coalition of voters, are associated with an agent  $\hat{m}$ , and they would like to manipulate the election so that a

preferred spouse  $w^*$  will be the match of  $\hat{m}$ .

## 4.1 Single Manipulator

With a single manipulator, the Manipulation in the context of Matching from the Men's side (MnM-m) is defined as follows:

**Definition 1** (MnM-m). *We are given a man  $\hat{m}$ , the preference profile  $\mathcal{L}$  of the honest voters that associate with  $\hat{m}$ , the preference profile  $\succ_{M-\hat{m}}$ , the preference profile  $\succ_W$ , a specific manipulator  $r$ , and a preferred woman  $w^* \in W$ . We are asked whether a preference order  $\ell_r$  exists such that  $\mu(\hat{m}, \ell_r) = w^*$ .*

We show that MnM-m can be decided in polynomial time by Algorithm 1, which works as follows. The algorithm begins by verifying that a preference order exists for  $\hat{m}$ , which makes  $w^*$  the match of  $\hat{m}$ . It thus iteratively builds a temporary preference order for  $\hat{m}$ ,  $\succ_x$  in lines 4-7. Moreover, during the iterations in lines 4-7 the algorithm identifies a set  $B$ , which is the set of women that might prevent  $w^*$  from being  $\hat{m}$ 's match. Specifically,  $\succ_x$  is initialized as the original preference order of  $\hat{m}$ ,  $\succ_{\hat{m}}$ . In each iteration, the algorithm finds the woman  $b$ , which is the match of  $\hat{m}$  given that  $\succ_x$  is the preference order of  $\hat{m}$ . If  $b$  is placed higher than  $w^*$  in  $\succ_x$ , then  $b$  is added to the set  $B$ , it is placed in  $\succ_x$  immediately below  $w^*$ , and the algorithm proceeds to the next iteration (using the updated  $\succ_x$ ).

Now, if  $b = \mu_x(\hat{m}, \succ_x)$  is positioned lower than  $w^*$  in  $\succ_x$ , then no preference order exists that makes  $w^*$  the match of  $\hat{m}$ , and the algorithm returns false. If  $b = w^*$ , then the algorithm proceeds to build the preference order for the manipulator,  $\ell_r$ . Clearly,  $w^*$  is placed in the top-most position in  $\ell_r$ . Then, the algorithm places all the women that are not in  $B$  in the highest available positions. Finally, the algorithm places all the women from  $B$  in the lowest positions in  $\ell_r$ , and they are placed in a reverse order with regard to their order in  $\mathcal{F}(\mathcal{L})$ .

---

**ALGORITHM 1:** Manipulation by a single voter from the men's side

---

```

1  $B \leftarrow \emptyset$ 
2 set  $\succ_x$  to be  $\succ_{\hat{m}}$ 
3  $b \leftarrow \mu_x(\hat{m}, \succ_x)$ 
4 while  $b \succ_x w^*$  do
5   | add  $b$  to  $B$ 
6   | move  $b$  in  $\succ_x$  immediately below  $w^*$ 
7   |  $b \leftarrow \mu_x(\hat{m}, \succ_x)$ 
8 if  $b \neq w^*$  then
9   | return false
  // phase 1:
10  $\ell_r \leftarrow$  empty preference order
11 place  $w^*$  in the highest position in  $\ell_r$ 
12 for each  $w \in W \setminus (B \cup \{w^*\})$  do
13   | place  $w$  in the next highest available position in  $\ell_r$ 
  // phase 2:
14 while  $B \neq \emptyset$  do
15   |  $b \leftarrow$  the least preferred woman from  $B$  according to  $\mathcal{F}(\mathcal{L})$ 
16   | place  $b$  in the highest available position in  $\ell_r$ 
17   | remove  $b$  from  $B$ 
18 if  $\mu(\hat{m}, \ell_r) = w^*$  then
19   | return  $\ell_r$ 
20 return false

```

---

For proving the correctness of Algorithm 1 we use the following known results:

**Theorem 1** (due to [16]). *In the Gale-Shapley matching procedure which always yields the optimal stable outcome for the set of the men agents,  $M$ , truthful revelation is a dominant strategy for all the agents in that set.*

**Lemma 1** (due to [13]). *For man  $m$ , his preference list is composed of  $(P_L(m), \mu(m), P_R(m))$ , where  $P_L(m)$  and  $P_R(m)$  are respectively those women ranking higher and lower than  $\mu(m)$ . Let  $A \subseteq W$  and let  $\pi_r(A)$  be a random permutation from all  $|A|!$  sets. For a subset of men  $S \subseteq M$ , if every member  $m \in S$  submits a falsified list of the form  $(\pi_r(P_L(m)), \mu(m), \pi_r(P_R(m)))$ , then  $\mu(m)$  stays  $m$ 's match.*

An immediate corollary of Theorem 1 is the following.

**Corollary 1.** *Given a man  $m$  with his preference order  $\succ_m$ , let  $w_m = \mu(m)$ . Let  $\succ'_m$  be a preference order for  $m$  such that if  $w_m \succ_m w$  then  $w_m \succ'_m w$ . Then,  $w_m$  is also the match of  $m$  with the preference order  $\succ'_m$ , i.e.,  $\mu_x(m, \succ'_m) = w_m$ .*

We begin the analysis of Algorithm 1 by showing that it is possible to verify (in polynomial time) whether a preference order exists for  $\hat{m}$ , which makes  $w^*$  the match of  $\hat{m}$ . We do so by showing that it is sufficient to check whether  $w^* = \mu_x(\hat{m}, \succ_x)$ , where  $\succ_x$  is the preference order that is built by Algorithm 1 in lines 4-7.

**Lemma 2.** *A preference order  $\succ_t$  for  $\hat{m}$  exists such that  $w^* = \mu_x(\hat{m}, \succ_t)$  if and only if  $w^* = \mu_x(\hat{m}, \succ_x)$ .*

That is, if Algorithm 1 returns false in line 9 then there is no preference order for  $\hat{m}$  that makes  $w^*$  the match of  $\hat{m}$  (and thus no manipulation is possible for  $r$ ).

We now show that the set  $B$ , which is identified by the algorithm in lines 4-7, is a set of woman that might prevent  $w^*$  from being  $\hat{m}$ 's match.

**Lemma 3.** *Given a preference order  $\succ_t$  for  $\hat{m}$ , if there exists  $b \in B$  such that  $b \succ_t w^*$  then  $\mu_x(\hat{m}, \succ_t) \neq w^*$ .*

That is, Algorithm 1 should place the women from  $B$  in the lowest position in  $\ell_r$  and  $w^*$  in the highest position in  $\ell_r$ , so that  $w^*$  will be preferred over every woman  $b \in B$  in  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ .

**Theorem 2.** *Algorithm 1 correctly decides the MnM-m problem in polynomial time.*

## 4.2 Coalitional Manipulation

We now study manipulation by a coalition of voters. The coalitional manipulation in the context of matching from the men's side is defined as follows:

**Definition 2** (coalitional MnM-m). *We are given a man  $\hat{m}$ , the preference profile  $\mathcal{L}$  of the honest voters that associate with  $\hat{m}$ , the preference profile  $\succ_{M-\hat{m}}$ , the preference profile  $\succ_W$ , a coalition of manipulators  $R$ , and a preferred woman  $w^* \in W$ . We are asked whether a preference profile  $\mathcal{L}_R$  exists such that  $\mu(\hat{m}, \mathcal{L}_R) = w^*$ .*

We show that the coalitional MnM-m problem is computationally hard, even with two manipulators. The reduction is from the Permutation Sum problem (as defined by Davies et al. [5]) that is *NP*-complete [24].

**Definition 3** (Permutation Sum). *Given  $q$  integers  $X_1 \leq \dots \leq X_q$  where  $\sum_{i=1}^q X_i = q(q+1)$ , do two permutations  $\sigma$  and  $\pi$  of 1 to  $q$  exist such that  $\sigma(i) + \pi(i) = X_i$  for all  $1 \leq i \leq q$ ?*

**Theorem 3.** *Coalitional MnM-m is NP-Complete.*

Even though coalitional MnM-m is NP-complete, it might still be possible to develop an efficient heuristic algorithm that finds a successful coalitional manipulation. We use Algorithm 2, which is a generalization of Algorithm 1, that works as follows. Similar to

---

**ALGORITHM 2:** Manipulation by a coalition of voters from the men's side

---

```

1  $B \leftarrow \emptyset$ 
2 set  $\succ_x$  to be  $\succ_{\hat{m}}$ 
3  $b \leftarrow \mu_x(\hat{m}, \succ_x)$ 
4 while  $b \succ_x w^*$  do
5   | add  $b$  to  $B$ 
6   | place  $b$  in  $\succ_x$  immediately below  $w^*$ 
7   |  $b \leftarrow \mu_x(\hat{m}, \succ_x)$ 
8 if  $b \neq w^*$  then
9   | return false
10 for each  $r \in R$  do
11   |  $\ell_r \leftarrow$  empty preference order
12   | place  $w^*$  in the highest position in  $\ell_r$ 
13   | for each  $w \in W \setminus (B \cup \{w^*\})$  do
14     | place  $w$  in the next highest available position in  $\ell_r$ 
15   |  $B' \leftarrow B$ 
16   | while  $B' \neq \emptyset$  do
17     |  $b \leftarrow$  the least preferred woman from  $B'$  according to  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ 
18     | place  $b$  in the highest available position in  $\ell_r$ 
19     | remove  $b$  from  $B'$ 
20   | add  $\ell_r$  to  $\mathcal{L}_R$ 
21 if  $\mu(\hat{m}) = w^*$  then
22   | return  $\mathcal{L}_R$ 
23 return false

```

---

Algorithm 1, Algorithm 2 identifies a set  $B$ , which is the set of women that might prevent  $w^*$  from being  $\hat{m}$ 's match. In addition, it verifies that a preference order for  $\hat{m}$  exists, which makes  $w^*$  the match of  $\hat{m}$ . Then, Algorithm 2 proceeds to build the preference order of every manipulator  $r \in R$  similarly to how Algorithm 1 builds the preference order for the single manipulator. Indeed, Algorithm 2 builds the preference order of each manipulator  $r$  in turn, and the order in which the women in  $B$  are placed depends on their order according to  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ . That is, the order in which the woman in  $B$  are placed in each  $\ell_r$  is not the same for each  $r$ , since  $\mathcal{L}_R$  is updated in each iteration. We refer to each of the iterations in Lines 10-20 as a *stage* of the algorithm. We now show that Algorithm 2 is an efficient heuristic that also has a theoretical guarantee. Specifically, the algorithm is guaranteed to find a coalitional manipulation in many instances, and we characterize the instances in which it may fail. Formally,

**Theorem 4.** *Given an instance of coalitional MnM-m,*

1. *If there is no preference profile making  $w^*$  the match of  $\hat{m}$ , then Algorithm 2 will return false.*
2. *If a preference profile making  $w^*$  the match of  $\hat{m}$  exists, then for the same instance with one additional manipulator, Algorithm 2 will return a preference profile that makes  $w^*$  the match of  $\hat{m}$ .*

That is, Algorithm 2 will succeed in any given instance such that the same instance but with one less manipulator is manipulable. Thus, it can be viewed as a 1-additive approximation algorithm (this approximate sense was introduced by Zuckerman et al. [25] when analyzing Borda as a social choice function (SCF)).

In order to prove Theorem 4 we use the following definitions. Let  $D_0 = \{d_0\}$ , where  $(d_0, \mathcal{L}) > (w, \mathcal{L})$ , and  $d_0, w \in B$ . For each  $s = 1, 2, \dots$ , let  $D_s \subseteq B$  be  $D_s = D_{s-1} \cup \{d \in B : d \text{ was ranked above some } d' \in D_{s-1} \text{ according to } \mathcal{F}(\mathcal{L}, \mathcal{L}_R) \text{ in some stage } l, 1 \leq l \leq |R|\}$ . Now, let  $D = \bigcup_{0 \leq s} D_s$ . Note that  $\forall s D_s \neq D_{s-1}$ , and  $s$  does not necessarily equal  $|R|$ . Let  $s_\ell(w)$  be the score of woman  $w$  in  $\mathcal{F}(\mathcal{L}, \mathcal{L}_R)$  after stage  $\ell$ .

The proof of Theorem 4 relies on Lemmata 4-8, and its general intuition is as follows. Consider the women in  $D$ : we show that if there exists a manipulation, then Algorithm 2 is able to determine the votes in  $\mathcal{L}_R$  such that the average score of the women in  $D$  is lower than the score of woman  $w^*$ . Moreover, a successful manipulation requires that  $w^*$  will be ranked higher than any women in  $D$ , and thus the algorithm may use one additional manipulator.

We begin with a basic lemma that clarifies where Algorithm 2 places the women of  $D$ .

**Lemma 4.** *The women in  $D$  are placed in each stage  $l$ ,  $1 \leq l \leq |R|$  in the  $|D|$  lowest positions.*

We now show the relation between the score of  $w^*$  and the average score of the women in  $D$ , when there are  $|R| - 1$  manipulators. In essence, the lemma characterizes the settings in which Algorithm 2 returns false and no manipulation exists.

**Lemma 5.** *Let  $q(D)$  be the average score of women in  $D$  after  $|R| - 1$  stages. That is,  $q(D) = \frac{1}{|D|} \sum_{d \in D} s_{|R|-1}(d)$ . If  $s_{|R|-1}(w^*) < q(D)$  and there are  $|R| - 1$  manipulators, then there is no manipulation that makes  $w^*$  the match of  $\hat{m}$ , and the algorithm will return false.*

The following lemma shows the relation between the score of  $w^*$  and the maximum score of a woman in  $D$ , when there are  $|R|$  manipulators. In essence, the lemma characterizes the settings in which Algorithm 2 finds a successful manipulation.

**Lemma 6.** *Assume that a preference order for  $\hat{m}$  exists, which makes  $w^*$  the match of  $\hat{m}$ . If  $\max_{d \in D} \{s_{|R|}(d)\} < s_{|R|}(w^*)$  and there are  $|R|$  manipulators, then there is a manipulation that makes  $w^*$  the match of  $\hat{m}$ , and Algorithm 2 will find such a manipulation.*

Finally, we show that the highest score of a woman from  $D$  is not much higher than the average score of the women in  $D$ . We thus first show that the scores of the women in  $D$  are dense, as captured by the following definition.

**Definition 4** (due to [25]). *A finite non-empty set of integers  $A$  is called 1-dense if, when sorting the set in a non-increasing order  $a_1 \geq a_2 \geq \dots \geq a_i$  (such that  $\{a_1, \dots, a_i\} = A$ ),  $\forall j, 1 \leq j \leq i - 1, a_{j+1} \geq a_j - 1$  holds.*

**Lemma 7.** *Let  $D$  be as before. Then the set  $\{s_{|R|-1}(d) : d \in D\}$  is 1-dense.*

**Lemma 8.**  $\max_{d \in D} \{s_{|R|}(d)\} \leq q(D) + |D| - 1$ .

Now we can prove the theorem.

*Proof of Theorem 4.* Clearly, if Algorithm 2 returns a preference profile  $\mathcal{L}_R$ , then it is a successful manipulation that will make  $w^*$  the match of  $\hat{m}$ . Suppose that a preference profile exists that makes  $w^*$  the match of  $\hat{m}$  with  $|R| - 1$  manipulators. By Lemma 8,  $\max_{d \in D} \{s_{|R|}(d)\} \leq q(D) + |D| - 1$ . By Lemma 5,  $q(D) + |D| - 1 \leq s_{|R|-1}(w^*) + |D| - 1$ . Since  $|D| \leq k - 1$ ,  $s_{|R|-1}(w^*) + |D| - 1 < s_{|R|-1}(w^*) + k - 1 = s_{|R|}(w^*)$ . Overall,  $\max_{d \in D} \{s_{|R|}(d)\} < s_{|R|}(w^*)$ , and by Lemma 6 the algorithm will find a preference profile that will make  $w^*$  the match of  $\hat{m}$  with  $|R|$  manipulators.  $\square$



## 5 Women's Side

We now consider the second variant, in which a specific voter, or a coalition of voters, are associated with an agent  $\hat{w}$ , and they would like to manipulate the election so that a preferred spouse  $m^*$  will be the match of  $\hat{w}$ . This variant is more involved, since manipulation of the GS algorithm is also possible by a single woman or a coalition of women. Indeed, there are notable differences between manipulation from the women's side and manipulation from the men's side. First, the manipulators from the women's side need to ensure that **two** men are positioned "relatively" high. In addition, the set  $B$ , which is the set of agents that are placed in low positions, is defined differently, and it is not built iteratively. Finally, in manipulation from the women's side, it is not always possible to place all the agents from  $B$  in the lowest positions.

### 5.1 Single Manipulator

With a single manipulator, the Manipulation in the context of Matching from the Women's side (MnM-w) is defined as follows:

**Definition 5** (MnM-w). *We are given a woman  $\hat{w}$ , the preference profile  $\mathcal{L}$  of the honest voters that associate with  $\hat{w}$ , the preference profile  $\succ_M$ , the preference profile  $\succ_{W-\hat{w}}$ , a specific manipulator  $r$ , and a preferred man  $m^* \in M$ . We are asked whether a preference order  $\ell_r$  exists such that  $\mu(\hat{w}, \ell_r) = m^*$ .*

---

**ALGORITHM 3:** Manipulation by a single voter from the women's side

---

```

1 for each  $m_{nd} \in M \setminus \{m^*\}$  do
   | // phase 1:
2   |  $\ell_r \leftarrow$  empty preference order
3   | place  $m_{nd}$  in the highest position in  $\ell_r$ 
4   | place  $m^*$  in the second-highest position in  $\ell_r$ 
5   | if  $(m_{nd}, \mathcal{L}, \ell_r) > (m^*, \mathcal{L}, \ell_r)$  then
6   | | place  $m^*$  in the highest position in  $\ell_r$ 
7   | | place  $m_{nd}$  in  $\ell_r$  in the highest position such that  $(m^*, \mathcal{L}, \ell_r) > (m_{nd}, \mathcal{L}, \ell_r)$ , if such
   | | position exists
8   | | if no such position exists then
9   | | | continue to the next iteration
10  | if  $\mu(\hat{w}, \ell_r) \neq m^*$  or  $m_{nd} \notin o(\hat{w}, \ell_r)$  then
11  | | continue to the next iteration
   | // phase 2:
12  | for each  $m \notin o(\hat{w}, \ell_r)$  do
13  | | place  $m$  in the highest available position in  $\ell_r$ 
   | // phase 3:
14  |  $B^{nd} \leftarrow o(\hat{w}, \ell_r) \setminus \{m^*, m_{nd}\}$ 
15  | while  $B^{nd} \neq \emptyset$  do
16  | |  $b \leftarrow$  the least preferred man from  $B^{nd}$  according to  $\mathcal{F}(\mathcal{L})$ 
17  | | place  $b$  in the highest available position in  $\ell_r$ 
18  | | remove  $b$  from  $B^{nd}$ 
19  | if  $\mu(\hat{w}, \ell_r) = m^*$  then
20  | | return  $\ell_r$ 
21 return false

```

---

Clearly, if  $\mu(\hat{w}) = m^*$  then finding a preference order  $\ell_r$  such that  $\mu(\hat{w}, \ell_r) = m^*$  is trivial. We thus henceforth assume that  $\mu(\hat{w}) \neq m^*$ . The MnM-w problem can be decided

in polynomial-time, using Algorithm 3. The algorithm tries to identify a man  $m_{nd} \in M$ , and to place him and  $m^*$  in  $\ell_r$  such that  $m_{nd}$  is ranked in  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$  as high as possible while  $m^*$  is still preferred over  $m_{nd}$  according to  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ . In addition, the algorithm ensures (at the end of phase 1) that  $\mu(\hat{w}, \ell_r) = m^*$  and  $m_{nd} \in o(\hat{w}, \ell_r)$ . Note that we compute  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$  even though  $\ell_r$  is not a complete preference order, since we assume that all the men that are not in  $\ell_r$  get a score of 0 from  $\ell_r$ . If phase 1 is successful (i.e.,  $\mu(\hat{w}, \ell_r) = m^*$  and  $m_{nd} \in o(\hat{w}, \ell_r)$ ), the algorithm proceeds to phase 2, where it fills the preference order  $\ell_r$  by placing all the men that are not in  $o(\hat{w}, \ell_r)$  in the highest available positions. Finally, in phase 3, the algorithm places all the men from  $o(\hat{w}, \ell_r)$  (except for  $m^*$  and  $m_{nd}$  that are already placed in  $\ell_r$ ) in the lowest positions in  $\ell_r$ , and they are placed in a reverse order with regard to their order in  $\mathcal{F}(\mathcal{L})$ . If  $\mu(\hat{w}, \ell_r) = m^*$  then we are done; otherwise, the algorithm iterates and considers another man.

For proving the correctness of Algorithm 3 we need the following result.

**Lemma 9** (Swapping lemma, due to [23]). *Given a woman  $w \in W$ , let  $\succ'_w$  be a preference order that is derived from  $\succ_w$  by swapping the positions of an adjacent pair of men ( $m_i, m_j$ ) and making no other changes. Then,*

1. *if  $m_i \notin o(w)$  or  $m_j \notin o(w)$ , then  $\mu_x(w, \succ'_w) = \mu(w)$ .*
2. *if both  $m_i$  and  $m_j$  are not one of the two most preferred proposals among  $o(w)$  according to  $\succ_w$ , then  $\mu_x(w, \succ'_w) = \mu(w)$ .*
3. *if  $m_i$  is the second preferred proposal among  $o(w)$  according to  $\succ_w$  and  $m_j$  is the third preferred proposal among  $o(w)$  according to  $\succ_w$ , then  $\mu_x(w, \succ'_w) \in \{\mu(w), m_j\}$ .*
4. *if  $m_i = \mu(w)$  and  $m_j$  is the second preferred proposal among  $o(w)$  according to  $\succ_w$ , then the second preferred proposal among  $o(w)$  according to  $\succ'_w$  is  $m_i$  or  $m_j$ .*

If we use the swapping lemma sequentially, we get the following corollary.

**Corollary 2.** *Given a woman  $w \in W$ , let  $\succ'_w$  be a preference order for  $w$  such that  $\succ_w \neq \succ'_w$ . Let  $m^* \in M$  be the most preferred man among  $o(w)$  according to  $\succ_w$ . That is,  $\mu(w) = m^*$ . Let  $m_{nd} \in M$  be the second most preferred man among  $o(w)$  according to  $\succ_w$ . If  $m_{nd}$  is the most preferred man among  $o(w) \setminus \{m^*\}$  according to  $\succ'_w$ , and  $m^* \succ'_w m_{nd}$ , then  $o(w) = o_x(w, \succ'_w)$  and thus  $\mu_x(w, \succ'_w) = \mu(w) = m^*$ .*

Corollary 2 is the basis of our algorithm. Intuitively, the manipulator needs to ensure that  $m^*$  is among the set of proposals  $o(\hat{w}, \ell_r)$ , and that  $m^*$  is the most preferred men, according to  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ , among this set. That is,  $m^* = \mu(\hat{w}, \ell_r)$ . Thus, the algorithm searches for a man, denoted by  $m_{nd}$ , that serves as the second-best proposal. If such a man exists, then, according to Corollary 2, the position of every man  $m \in o(\hat{w}, \ell_r)$  does not change  $\hat{w}$ 's match (which is currently  $m^*$ ) if  $m_{nd}$  is preferred over  $m$  in  $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ . In addition, the position of every man  $m \notin o(\hat{w}, \ell_r)$  does not change  $\hat{w}$ 's match at all.

**Theorem 5.** *Algorithm 3 correctly decides the MnM-w problem in polynomial time.*

## 5.2 Coalitional Manipulation

Finally, We study manipulation by a coalition of voters from the women's side.

**Definition 6** (coalitional MnM-w). *We are given a woman  $\hat{w}$ , the preference profile  $\mathcal{L}$  of the honest voters that associate with  $\hat{w}$ , the preference profile  $\succ_M$ , the preference profile  $\succ_{W-\hat{w}}$ , a coalition of manipulators  $R$ , and a preferred man  $m^* \in M$ . We are asked whether a preference profile  $\mathcal{L}_R$  exists such that  $\mu(\hat{w}, \mathcal{L}_R) = m^*$ .*

Similar to the single manipulator setting, if  $\mu(\hat{w}) = m^*$  then finding a preference profile  $\mathcal{L}_R$  such that  $\mu(\hat{w}, \ell_R) = m^*$  is trivial. We thus henceforth assume that  $\mu(\hat{w}) \neq m^*$ . The coalitional MnM-w problem is computationally hard, even with two manipulators, and we again reduce from the Permutation Sum problem (Definition 3).

**Theorem 6.** *Coalitional MnM-w is NP-Complete.*

Similar to the coalitional MnM-m, the coalitional MnM-w also has an efficient heuristic algorithm that finds a successful manipulation. We use Algorithm 4, which works as follows. Similar to Algorithm 3, Algorithm 4 needs to identify a man  $m_{nd} \in M$ , such

---

**ALGORITHM 4:** Manipulation by a coalition of voters from the women's side

---

```

1 for each  $m_{nd} \in M \setminus \{m^*\}$  do
   | // phase 1:
2    $gap \leftarrow s(m_{nd}, \mathcal{L}) - s(m^*, \mathcal{L})$ 
3   if  $m_{nd}$  is preferred over  $m^*$  according to the lexicographical tie breaking rule then
4     |  $gap = gap + 1$ 
5     if  $|R| \cdot (k - 1) < gap$  then
6       | continue to the next iteration
7      $\mathcal{L}_R \leftarrow (\ell_1, \dots, \ell_{|R|})$  where each preference order is an empty one
8     if  $|R| \geq gap$  then
9       | place  $m^*$  in the highest position and  $m_{nd}$  in the second highest position, in
10      |  $\max(gap + \lceil (|R| - gap)/2 \rceil, 0)$  preference orders of  $\mathcal{L}_R$ 
11      | place  $m^*$  in the second highest position and  $m_{nd}$  in the highest position in all of the
12      | other preference orders of  $\mathcal{L}_R$ 
13     else
14       | place  $m^*$  in the highest position in each  $\ell_r \in \mathcal{L}_R$ 
15       |  $s_{m_{nd}} \leftarrow |R| \cdot (k - 1) - gap$ 
16       | place  $m_{nd}$  in  $(s_{m_{nd}} \bmod |R|)$  manipulators such that it gets a score of  $\lceil \frac{s_{m_{nd}}}{|R|} \rceil$  from
17       | each manipulator
18       | place  $m_{nd}$  in the other manipulators such that it gets a score of  $\lfloor \frac{s_{m_{nd}}}{|R|} \rfloor$  from each
19       | manipulator
20     if  $\mu(\hat{w}, \mathcal{L}_R) \neq m^*$  or  $m_{nd} \notin o(\hat{w}, \mathcal{L}_R)$  then
21       | continue to the next iteration
22     // phase 2:
23      $B^{nd} \leftarrow o(\hat{w}, \mathcal{L}_R) \setminus \{m^*, m_{nd}\}$ 
24     for each  $r \in R$  do
25       | for each  $m \in M \setminus (B^{nd} \cup \{m^*, m_{nd}\})$  do
26       | | place  $m$  in the next highest available position in  $\ell_r$ 
27       | |  $B' \leftarrow B^{nd}$ 
28       | | while  $B' \neq \emptyset$  do
29       | | |  $b \leftarrow$  the least preferred man from  $B'$  according to  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ 
30       | | | place  $b$  at the highest available position in  $\ell_r$ 
31       | | | remove  $b$  from  $B'$ 
32     if  $\mu(\hat{w}, \mathcal{L}_R) = m^*$  then
33       | return  $\mathcal{L}_R$ 
34 return false

```

---

that  $m_{nd}$  is ranked in  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$  as high as possible while  $m^*$  is still preferred over  $m_{nd}$  according to  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ . In addition, the algorithm needs to ensure that  $\mu(\hat{w}, \mathcal{L}_R) = m^*$  and  $m_{nd} \in o(\hat{w}, \mathcal{L}_R)$ , which is done at the end of phase 1. Indeed, finding such a man  $m_{nd} \in M$ , and placing him and  $m^*$  in every  $\ell_r \in \mathcal{L}_R$  is not trivial. The algorithm considers every  $m \in M \setminus \{m^*\}$ , and computes the difference between the score of  $m$  from  $\mathcal{L}$  and the

score of  $m^*$  from  $\mathcal{L}$ . Clearly, if this gap is too big,  $m$  cannot be  $m_{nd}$  (line 5). Otherwise, there are two possible cases. If there are many manipulators, specifically,  $|R| \geq \text{gap}$ , then the algorithm places  $m^*$  and  $m$  in the two highest positions in every  $\ell_r$  (lines 9-10). On the other hand, if  $|R| < \text{gap}$ , then the algorithm places  $m^*$  in the highest position in every  $\ell_r$ . The algorithm places  $m$  such that he gets a total score of  $|R| \cdot (k - 1) - \text{gap}$  from the manipulators. Moreover, the algorithm tries to place  $m$  in almost the same position in every  $\ell_r$ . If phase 1 is successful, the algorithm proceeds to fill the preference orders of  $\mathcal{L}_R$  iteratively in phase 2. The algorithm first defines the set  $B^{nd}$ , which consists of all the men from  $o(\hat{w}, \mathcal{L}_R)$ , except for  $m^*$  and  $m_{nd}$ . Note that at the beginning of phase 2, in which  $B^{nd}$  is defined, only  $m^*$  and  $m$  are positioned in every  $\ell_r \in \mathcal{L}_R$ . Then, in every  $\ell_r \in \mathcal{L}_R$ , the algorithm places all the men that are not in  $B^{nd}$  (except for  $m^*$  and  $m_{nd}$  that are already placed in  $\mathcal{L}_R$ ) in the highest available positions. The algorithm places all the men from  $B^{nd}$  in the lowest positions in  $\ell_r$ , and they are placed in a reverse order with regard to their current order in  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ . Note that since  $\mathcal{L}_R$  is updated in every iteration, the order in which the men from  $B^{nd}$  are placed in each  $\ell_r$  is not the same for each  $r$ . If  $\mu(\hat{w}, \ell_R) = m^*$  then we are done; otherwise, the algorithm iterates and considers another man. We refer to each of the iterations in Lines 19-26 as a *stage* of the algorithm.

We now show that Algorithm 4 will succeed in any given instance such that the same instance but with two less manipulators is manipulable. That is, unlike the coalitional MnM-m, the coalitional MnM-w admits a 2-additive approximation algorithm. Formally,

**Theorem 7.** *Given an instance of coalitional MnM-w,*

1. *If there is no preference profile making  $m^*$  the match of  $\hat{w}$  exists, then Algorithm 4 will return false.*
2. *If a preference profile making  $m^*$  the match of  $\hat{w}$  exists, then for the same instance with two additional manipulators, Algorithm 4 will return a preference profile that makes  $m^*$  the match of  $\hat{w}$ .*

## 6 Conclusion and Future Work

In this paper, we initiate the analysis of strategic voting in the context of stable matching of teams. Specifically, we assume that the Borda rule is used as a SWF, which outputs an order over the agents that is used as an input in the GS algorithm. Note that in the standard model of manipulation of Borda, the goal is that a specific candidate will be the winner. In our setting, the algorithms need also to ensure that specific candidates will not be ranked too high. Similarly, in the standard model of manipulation of the GS algorithm, the goal is simply to achieve a more preferred match. In our setting, the algorithms for manipulation need also to ensure that a less preferred spouse is matched to a specific agent. Therefore, even though the manipulation of the Borda rule and the manipulation of the GS algorithm have already been studied, our analysis of the manipulation of Borda rule in the context of GS stable matching provides a better understanding of both algorithms.

Interestingly, our algorithms for the single manipulator settings are quite powerful. They provide exact solutions for the single manipulator case, and their generalizations provide approximate solutions to the coalitional manipulation settings, both when the manipulators are on the men's side or on the women's side.

For future work, we would like to extend our analysis and study additional voting rules as SWFs. It is also worth studying the destructive manipulation objective. In addition, it will be interesting to examine a new type of manipulation, which is only relevant in our setting, in which there is a coalition of manipulators, but every manipulator is associated with a different agent.

## Acknowledgments

This research has been partly supported by the Israel Science Foundation under grant 1958/20, by the EU Project TAILOR under grant 952215, and by the Ministry of Science, Technology & Space (MOST), Israel.

## References

- [1] Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, 2005.
- [2] Walter Bossert and Yves Sprumont. Strategy-proof preference aggregation: Possibilities and characterizations. *Games and Economic Behavior*, 85:109–126, 2014.
- [3] Walter Bossert and Ton Storcken. Strategy-proofness of social welfare functions: the use of the kemeny distance between preference orderings. *Social Choice and Welfare*, 9(4):345–360, 1992.
- [4] Vincent Conitzer and Toby Walsh. Barriers to manipulation in voting. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 127–145. Cambridge University Press, 2016.
- [5] Jessica Davies, George Katsirelos, Nina Narodytska, and Toby Walsh. Complexity of and algorithms for borda manipulation. In *The Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 657–662, 2011.
- [6] Onur Dogan and Jean Lainé. Strategic manipulation of social welfare functions via strict preference extensions. In *The 13th Meeting of the Society for Social Choice and Welfare*, page 199, 2016.
- [7] Lester E Dubins and David A Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.
- [8] Piotr Faliszewski and Ariel D Procaccia. Ai’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- [9] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- [10] David Gale and Marilda Sotomayor. Ms. machiavelli and the stable matching problem. *The American Mathematical Monthly*, 92(4):261–268, 1985.
- [11] Hadi Hosseini, Fatima Umar, and Rohit Vaish. Accomplice manipulation of the deferred acceptance algorithm. In *The Thirtieth International Joint Conference on Artificial Intelligence*, pages 231–237, 2021.
- [12] Hadi Hosseini, Fatima Umar, and Rohit Vaish. Two for one & one for all: Two-sided manipulation in matching markets. *arXiv preprint arXiv:2201.08774*, 2022.
- [13] Chien-Chung Huang. Cheating by men in the gale-shapley stable matching algorithm. In *European Symposium on Algorithms*, pages 418–431. Springer, 2006.
- [14] Nitsan Perach and Shoshana Anily. Stable matching of student-groups to dormitories. *European Journal of Operational Research*, 2022.

- [15] Maria Silvia Pini, Francesca Rossi, K Brent Venable, and Toby Walsh. Manipulation complexity and gender neutrality in stable marriage procedures. *Autonomous Agents and Multi-Agent Systems*, 22(1):183–199, 2011.
- [16] Alvin E Roth. The economics of matching: Stability and incentives. *Mathematics of operations research*, 7(4):617–628, 1982.
- [17] Alvin E Roth. The nrmp as a labor market: understanding the current study of the match. *Journal of the American Medical Association*, 275(13):1054–1056, 1996.
- [18] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Kidney exchange. *The Quarterly journal of economics*, 119(2):457–488, 2004.
- [19] Leora Schmerler and Noam Hazon. Strategic voting in negotiating teams. In *International Conference on Algorithmic Decision Theory*, pages 209–223. Springer, 2021.
- [20] Leora Schmerler, Noam Hazon, and Sarit Kraus. Strategic voting in the context of stable-matching of teams. *arXiv preprint arXiv:2207.04912*, 2022.
- [21] Weiran Shen, Yuan Deng, and Pingzhong Tang. Coalitional permutation manipulations in the gale-shapley algorithm. *Artificial Intelligence*, 301:103577, 2021.
- [22] Chung-Piaw Teo, Jay Sethuraman, and Wee-Peng Tan. Gale-shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, 47(9):1252–1267, 2001.
- [23] Rohit Vaish and Dinesh Garg. Manipulating gale-shapley algorithm: Preserving stability and remaining inconspicuous. In *The Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 437–443, 2017.
- [24] Wenci Yu, Han Hoogeveen, and Jan Karel Lenstra. Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard. *Journal of Scheduling*, 7(5):333–348, 2004.
- [25] Michael Zuckerman, Ariel D Procaccia, and Jeffrey S Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.

Leora Schmerler  
Ariel University  
Ariel, Israel  
Email: [leoras@ariel.ac.il](mailto:leoras@ariel.ac.il)

Noam Hazon  
Ariel University  
Ariel, Israel  
Email: [noamh@ariel.ac.il](mailto:noamh@ariel.ac.il)

Sarit Kraus  
Bar-Ilan University  
Ramat-Gan, Israel  
Email: [sarit@cs.biu.ac.il](mailto:sarit@cs.biu.ac.il)