

Strategic Voting in the Context of Stable-Matching of Teams^{*}

Leora Schmerler¹, Noam Hazon¹, and Sarit Kraus²

¹ Department of Computer Science, Ariel University, Israel
{leoras,noamh}@ariel.ac.il

² Department of Computer Science, Bar-Ilan University, Israel
sarit@cs.biu.ac.il

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 [16], and kidney exchange [17]. A very common

^{*} 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.

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 [13].

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 the same computational complexity as in manipulation from the men’s side.

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 [18]. 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, 15]. Teo et al. [21] provided a polynomial-time algorithm for computing the optimal manipulation by a woman. Shen et al. [20] 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 [12] studied manipulation that is a weak improvement for every member of a coalition of men. Hosseini et al. [10] 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 [11]. 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. [14] 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 ℓ_v , is a strict total order over W (M). The preference profile \mathcal{L} is a vector $(\ell_{v_1}, \ell_{v_2}, \dots, \ell_{v_{|V|}})$. 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 ℓ_v a score of $k-1$, the candidate in the second-highest position in ℓ_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 her preference order is ℓ_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 $R = \{r_1, \dots, r_n\}$ of manipulators, their preference profile is $\mathcal{L}_R = \{\ell_{r_1}, \ell_{r_2}, \dots, \ell_{r_n}\}$, and 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 ℓ_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.

Due to space constraint, many proofs are deferred to the full version of the paper [19].

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 ℓ_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, ℓ_r . Clearly, w^* is placed in the top-most position in ℓ_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 ℓ_r , and they are placed in a reverse order with regard to their order in $\mathcal{F}(\mathcal{L})$.

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

Theorem 1 (due to [15]). *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 [12]). *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.*

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
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  $\ell_r$ 
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
```

We begin by showing that the set B , which is identified by the algorithm in lines 4-7, is a set of women that might prevent w^* from being \hat{m} 's match. The intuition is as follows. If $b = \mu_x(\hat{m}, \succ_x)$ in a given iteration does not equal w^* , then changing the order of the women ranking higher (or lower) than w^* in \succ_t will not make w^* the match of \hat{m} due to Lemma 1. Moreover, moving a woman that is above b to a position below w^* (or moving a woman that is below w^* to a position above b) will not make w^* the match of \hat{m} , due to Theorem 1.

Lemma 2. *Given a preference order \succ_t for \hat{m} , if there exists $b \in B$ such that $b \succ_t w^*$ then $\mu_x(\hat{m}, \succ_t) \neq w^*$.*

Using Lemma 2, we show 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 3. *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).

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

Proof. Clearly, the algorithm runs in polynomial time since there are three loops, where the three loops together iterate at most $2k$ times, and the running time of the GS matching algorithm is in $O(k^2)$. In addition, if the algorithm returns a preference order, which is a manipulative vote for the manipulator r , then w^* will be the match of \hat{m} by the GS algorithm. We need to show that if there exists a preference order for the manipulator r that makes w^* the match of \hat{m} , then our algorithm will find such a preference order for r . Assume that a manipulative vote, ℓ_t , exists, which makes w^* the match of \hat{m} . That is, $\mu(\hat{m}, \ell_t) = w^*$. Then, by Lemma 3, the algorithm finds a preference order for \hat{m} that makes w^* his match (i.e., the preference order \succ_x), and thus it does not return false in line 9. We show that Algorithm 1 returns ℓ_r in line 19.

We now proceed to build the preference order ℓ_r . By Theorem 1, since each man should be truthful, then r should position w^* as high as possible, and thus w^* is positioned in the highest position in ℓ_r . Therefore, $s(w^*, \ell_r) \geq s(w^*, \ell_t)$, and consequently, w^* is positioned higher in $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$ than in $\mathcal{F}(\mathcal{L} \cup \{\ell_t\})$ or in the same position. That is, after line 11, $\mu(\hat{m}, \ell_r) = w^*$.

Note that \succ_x is different from $\succ_{\hat{m}} = \mathcal{F}(\mathcal{L})$, but \succ_x does not change the position of all $w \in W \setminus B$. Now, let $w \in W \setminus B$ be such that $w^* \succ_x w$. w is not preferred over w^* in $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$, since $w^* \succ_{\hat{m}} w$ and $s(w^*, \ell_r) > s(w, \ell_r)$. According to the GS algorithm, the women that are positioned below $\mu(m)$ for some man m do not affect m 's match. Thus, placing w in ℓ_r does not change \hat{m} 's match, which is w^* . Let $w \in W \setminus B$ be such that $w \succ_x w^*$, but w^* is preferred over w in $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$. Due to Theorem 1, placing w in ℓ_r does not change \hat{m} 's match, which is w^* . Finally, let $w \in W \setminus B$ be such that $w \succ_x w^*$, and w is also preferred over w^* in $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$. Due to Lemma 1, placing w in ℓ_r does not change \hat{m} 's match, which is w^* .

According to Lemma 2, if $b \in B$ is positioned higher than w^* in $\mathcal{F}(\mathcal{L} \cup \{\ell_r\})$, w^* will not be \hat{m} 's match. We thus show that Algorithm 1 (lines 14-17) can assign scores to all the women $w \in B$ such that ℓ_r is a successful manipulation. According to Lemma 1 the order of the set B in \hat{m} 's preference order does not prevent w^* being the match (as long as each woman of the set B is placed below w^* in \hat{m} 's preference order). For any $w \in B$, if $s(w, \ell_r) \leq s(w, \ell_t)$ then $s(w, \mathcal{L}, \ell_r) \leq s(w, \mathcal{L}, \ell_t)$. Since $(w^*, \mathcal{L}, \ell_t) > (w, \mathcal{L}, \ell_t)$ then $(w^*, \mathcal{L}, \ell_r) > (w, \mathcal{L}, \ell_r)$. Otherwise, let $w \in B$ be a woman such that $s(w, \ell_r) > s(w, \ell_t)$ and let $s = s(w, \ell_r)$. There are s women from B below w in ℓ_r . According to the pigeonhole principle, at least one of the women from B , denoted w' , gets a score of at least s from ℓ_t . That is, $s(w', \ell_t) \geq s(w, \ell_r)$. By the algorithm construction, all of the women $w'' \in B$ that are positioned lower than w in ℓ_r are positioned higher than w in $\mathcal{F}(\mathcal{L})$. That is, $(w', \mathcal{L}) > (w, \mathcal{L})$. However, $(w', \mathcal{L}, \ell_t) < (w^*, \mathcal{L}, \ell_t)$ and thus $(w, \mathcal{L}, \ell_r) < (w^*, \mathcal{L}, \ell_r)$. Overall, after placing the women from B in ℓ_r , $\forall w \in B$, $(w, \mathcal{L}, \ell_r) < (w^*, \mathcal{L}, \ell_r)$. That is, $\mu(\hat{m}, \ell_r) = w^*$. \square

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. The reduction is from the Permutation Sum problem (as defined by Davies et al. [5]) that is *NP*-complete [23].

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 σ and π 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 1, Algorithm 2 identifies a set B , which

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

```

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 woman 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 ℓ_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. [24] when analyzing Borda as a social choice function (SCF)).

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

```

Clearly, if $\mu(\hat{w}) = m^*$ then finding a preference order ℓ_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 ℓ_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 stage 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 ℓ_r is not a complete preference order, since we assume that all the men that are not in ℓ_r get a score of 0 from ℓ_r . If stage 1 is successful (i.e., $\mu(\hat{w}, \ell_r) = m^*$ and $m_{nd} \in o(\hat{w}, \ell_r)$), the algorithm proceeds to stage 2, where it fills the preference order ℓ_r by placing all the men that are not in $o(\hat{w}, \ell_r)$ in the highest available positions. Finally, in stage 3, the algorithm places all the men from $o(\hat{w}, \ell_r)$ (except for m^* and m_{nd} that are already placed in ℓ_r) in the lowest positions in ℓ_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 4 (Swapping lemma, due to [22]). *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 1. *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 $\mu_x(w, \succ'_w) = \mu(w) = m^*$.*

Proof. We construct the preference order \succ'_w by starting from \succ_w and performing a sequence of swaps of two adjacent men till the resulting preference order is \succ'_w . We show that each swap does not change the set of proposals, by repeatedly invoking Lemma 4.

We begin by positioning the most preferred man according to \succ'_w , using swaps of two adjacent men. That is, if m_{st} is the most preferred man according to \succ'_w , we swap pairs $(m, m_{st}), m \in M$, until m_{st} is placed in the first position in \succ'_w . We call these swaps the swaps of m_{st} . We then position the second preferred man using his swaps, and so on. Clearly, this process terminates since the number of men is finite. Let $\succ_w^{(t)}$ be \succ_w after t swaps. That is, $\succ_w^{(0)}$ is \succ_w , $\succ_w^{(1)}$ is \succ_w after one swap, and $\succ_w^{(t+1)}$ is $\succ_w^{(t)}$ after one swap. We show that for every $t \geq 0$, $o_x(w, \succ_w^{(t)}) = o_x(w, \succ_w^{(t+1)})$ and thus $o(w) = o_x(w, \succ_w^{(0)}) = o_x(w, \succ'_w)$. Let (m_i, m_j) be the pair of adjacent men that swap their positions when moving from $\succ_w^{(t)}$ to $\succ_w^{(t+1)}$. That is, $m_i \succ_w^{(t)} m_j$ and $m_j \succ_w^{(t+1)} m_i$. Recall that for every $m_1, m_2 \in M$, if $m_1 \succ'_w m_2$, then all the swaps of m_1 are executed before all the swaps of m_2 . In addition, since m_{nd} is the most preferred man among $o(w) \setminus \{m^*\}$ according to \succ'_w and $m^* \succ'_w m_{nd}$, then the following cases are not possible:

1. $m_i = m^*$ and $m_j \in o_x(w, \succ_w^{(t)})$.

2. $m_i \in o_x(w, \succ_w^{(t)})$ and $m_j = m^*$.
3. $m_i = m_{nd}$ and $m_j \in o_x(w, \succ_w^{(t)})$.
4. $m_i \in o_x(w, \succ_w^{(t)})$ and $m_j = m_{nd}$.
5. $m_i = m^*$ and $m_j = m_{nd}$.

We thus need to consider only the following two cases:

1. $m_i \notin o_x(w, \succ_w^{(t)})$ or $m_j \notin o_x(w, \succ_w^{(t)})$. According to the GS algorithm, a swap of such m_i and m_j cannot change the response of w (either an acceptance or rejection). Therefore, $o_x(w, \succ_w^{(t)}) = o_x(w, \succ_w^{(t+1)})$.
2. $m_i, m_j \in o_x(w, \succ_w^{(t)}) \setminus \{m^*, m_{nd}\}$. We use case 2 of Lemma 4 for this case. Assume to contradiction that $o_x(w, \succ_w^{(t)}) \neq o_x(w, \succ_w^{(t+1)})$. There are two possible cases:
 - (a) There exists a man $o \in o_x(w, \succ_w^{(t)})$ such that $o \notin o_x(w, \succ_w^{(t+1)})$. By case 2 of Lemma 4, $m^* = \mu_x(w, \succ_w^{(t)}) = \mu_x(w, \succ_w^{(t+1)})$. Let \succ^o be $\succ_w^{(t+1)}$ such that o is positioned above m^* . We can construct \succ^o from $\succ_w^{(t+1)}$ by swaps of o . Since $o \notin o_x(w, \succ_w^{(t+1)})$, then by case 1 of Lemma 4, $\mu_x(w, \succ_w^{(t+1)}) = \mu_x(w, \succ^o)$. We now swap m_j and m_i in \succ^o , and thus $m_i \succ^o m_j$ as in $\succ_w^{(t)}$. Let $Pre^o \subset o_x(w, \succ_w^{(t)})$ be the set of proposals that w receives before she receives the proposal o . Note that all the men $o \in Pre^o$ are in the same order in $\succ_w^{(t)}$ and in \succ^o . Therefore, the response of woman w is the same for all the proposals $o \in Pre^o$ and thus $o \in o_x(w, \succ^o)$. Therefore, $m^* \neq \mu_x(w, \succ^o)$, which is a contradiction.
 - (b) There exists a man $o \notin o_x(w, \succ_w^{(t)})$ such that $o \in o_x(w, \succ_w^{(t+1)})$. Using a similar argument to case (a) above (i.e., we now construct \succ^o from $\succ_w^{(t)}$) we get that in this case also $o_x(w, \succ_w^{(t)}) = o_x(w, \succ_w^{(t+1)})$.

□

Corollary 1 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 1, 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.*

Proof. Clearly, the algorithm runs in polynomial time since there are three loops, where the three loops together iterate at most k^2 times, and the running time of the GS matching algorithm is in $O(k^2)$. In addition, if the algorithm returns a preference order, which is a manipulative vote for the manipulator r , then m^* will be the match of \hat{w} by the GS algorithm. We need to show that if there exists

a preference order for the manipulator r that makes m^* the match of \hat{w} , then our algorithm will find such a preference order for r . Assume that a manipulative vote, ℓ_t , exists, which makes m^* the match of \hat{w} . That is, $\mu(\hat{w}, \ell_t) = m^*$. We show that Algorithm 3 returns ℓ_r in line 20. Let $\ell_{r(1)}$ be the preference order ℓ_r after stage 1 of the algorithm. $\ell_{r(2)}$ and $\ell_{r(3)}$ are defined similarly. Note that $\ell_{r(3)}$ is the preference order ℓ_r that is returned by the algorithm in line 20.

Algorithm 3 iterates over all $m_{nd} \in M \setminus \{m^*\}$, and thus there exists an iteration in which m_{nd} is the second preferred proposal among $o(\hat{w}, \ell_t)$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_t\})$. Let $\ell_{t(1)}$ be the preference order ℓ_t where m^* and m_{nd} are placed in the same positions as in $\ell_{r(1)}$. Note that m^* and m_{nd} are placed in $\ell_{r(1)}$ such that m^* is preferred over m_{nd} according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(1)}\})$, and thus m^* is preferred over m_{nd} according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(1)}\})$. In addition, m_{nd} is positioned in $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(1)}\})$ not lower than in $\mathcal{F}(\mathcal{L} \cup \{\ell_t\})$. Therefore, m_{nd} is the most preferred man among $o(\hat{w}, \ell_t) \setminus \{m^*\}$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(1)}\})$. By Corollary 1, $o(\hat{w}, \ell_t) = o(\hat{w}, \ell_{t(1)})$ and $\mu(\hat{w}, \ell_{t(1)}) = m^*$. Thus, m_{nd} is the second preferred proposal among $o(\hat{w}, \ell_{t(1)})$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(1)}\})$.

Let $\ell_{t(2)}$ be the preference order $\ell_{t(1)}$ where the men $m \notin o(\hat{w}, \ell_{t(1)})$ are placed in the highest positions in $\ell_{t(2)}$ without changing the positions of m^* and m_{nd} (similar to the positioning of the men $m \notin o(\hat{w}, \ell_{r(2)})$ after stage 2 of the algorithm). That is, m^* is preferred over m_{nd} according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(2)}\})$ and m_{nd} is the most preferred man among $o(\hat{w}, \ell_{t(1)}) \setminus \{m^*\}$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(2)}\})$. We can thus use (again) Corollary 1 to get that $o(\hat{w}, \ell_{t(1)}) = o(\hat{w}, \ell_{t(2)})$ and $\mu(\hat{w}, \ell_{t(2)}) = m^*$.

Recall that at the end of stage 1 of Algorithm 3, m^* and m_{nd} are placed in $\ell_{r(1)}$ such that m^* is preferred over m_{nd} according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(1)}\})$. In addition, m_{nd} is positioned in $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(1)}\})$ not lower than in $\mathcal{F}(\mathcal{L} \cup \{\ell_{t(2)}\})$, since they are placed in the same position in $\ell_{r(1)}$ and $\ell_{t(2)}$ and the other men in $\ell_{r(1)}$ get a score of 0 from $\ell_{r(1)}$. Specifically, the men $m \in o(\hat{w}, \ell_{t(2)})$ also get a score of 0 from $\ell_{r(1)}$ and thus m_{nd} is the most preferred man among $o(\hat{w}, \ell_{t(2)}) \setminus \{m^*\}$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(1)}\})$. By Corollary 1, $o(\hat{w}, \ell_{t(2)}) = o(\hat{w}, \ell_{r(1)})$ and $\mu(\hat{w}, \ell_{r(1)}) = m^*$. Since in stage 2 of Algorithm 3 we place only men $m \notin o(\hat{w}, \ell_{r(1)})$, then, we can (again) use Corollary 1 to show that $o(\hat{w}, \ell_{t(2)}) = o(\hat{w}, \ell_{r(2)})$, $\mu(\hat{w}, \ell_{r(2)}) = m^*$, and m_{nd} is the second preferred proposal among $o(\hat{w}, \ell_{r(2)})$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(2)}\})$.

We now show that Algorithm 3 (lines 14-18) can assign scores to all the men $m \in B$ such that ℓ_r is a successful manipulation. For any $m \in B$, if $s(m, \ell_{r(3)}) \leq s(m, \ell_{t(2)})$ then $s(m, \mathcal{L}, \ell_{r(3)}) \leq s(m, \mathcal{L}, \ell_{t(2)})$. Since $(m_{nd}, \mathcal{L}, \ell_{t(2)}) > (m, \mathcal{L}, \ell_{t(2)})$ and $s(m_{nd}, \mathcal{L}, \ell_{t(2)}) = s(m_{nd}, \mathcal{L}, \ell_{r(3)})$ then $(m_{nd}, \mathcal{L}, \ell_{r(3)}) > (m, \mathcal{L}, \ell_{r(3)})$. Otherwise, let $m \in B$ be a man such that $s(m, \ell_{r(3)}) > s(m, \ell_{t(2)})$ and let $s = s(m, \ell_{r(3)})$. By the algorithm construction, there are s men from B below m in $\ell_{r(3)}$. According to the pigeonhole principle, at least one of the men from B , denoted m' , gets a score of at least s from $\ell_{t(2)}$. That is, $s(m', \ell_{t(2)}) \geq s(m, \ell_{r(3)})$. By the algorithm construction, all the men $m'' \in B$ that are positioned lower than m in $\ell_{r(3)}$ are positioned higher than m in $\mathcal{F}(\mathcal{L})$. That is, $(m', \mathcal{L}) > (m, \mathcal{L})$. However, $(m', \mathcal{L}, \ell_{t(2)}) < (m_{nd}, \mathcal{L}, \ell_{t(2)})$ and

thus $(m, \mathcal{L}, \ell_{r(3)}) < (m_{nd}, \mathcal{L}, \ell_{r(3)})$. Overall, after placing the men from B in $\ell_{r(3)}$, $\forall m \in B$, $(m, \mathcal{L}, \ell_{r(3)}) < (m_{nd}, \mathcal{L}, \ell_{r(3)})$. That is, m_{nd} is the most preferred man among $o(\hat{w}, \ell_{r(2)}) \setminus \{m^*\}$ according to $\mathcal{F}(\mathcal{L} \cup \{\ell_{r(2)}\})$. In addition $(m_{nd}, \mathcal{L}, \ell_{r(3)}) < (m^*, \mathcal{L}, \ell_{r(3)})$ and thus by Corollary 1, $\mu(\hat{w}, \ell_{r(3)}) = m^*$. \square

5.2 Coalitional Manipulation

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

Definition 5 (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 $\mu(\hat{w}, \mathcal{L}_R) = m^*$ is trivial. We thus henceforth assume that $\mu(\hat{w}) \neq m^*$. The coalitional MnM-w problem is computationally hard, 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 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 stage 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 6). Otherwise, there are two possible cases. If there are many manipulators, specifically, $|R| \geq gap$, then the algorithm places m^* and m in the two highest positions in every ℓ_r (lines 8-10). On the other hand, if $|R| < gap$, then the algorithm places m^* in the highest position in every ℓ_r . The algorithm places m in the second highest position or in the lowest position in every ℓ_r , except for $\ell_{|R|}$; in this preference order the algorithm places m in the highest position such that m^* is preferred over m according to $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$. If stage 1 is successful, the algorithm proceeds to fill the preference orders of \mathcal{L}_R iteratively in stage 2. In every $\ell_r \in \mathcal{L}_R$, the algorithm places all the men that are not in $o(\hat{w}, \mathcal{L}_R)$ in the highest available positions. The algorithm places all the men from $o(\hat{w}, \mathcal{L}_R)$ (except for m^* and m_{nd} that are already placed in \mathcal{L}_R) in the lowest positions in \mathcal{L}_R , and they are placed in a reverse order in each manipulator 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 men from $o(\hat{w}, \mathcal{L}_R) \setminus \{m^*, m_{nd}\}$ may be placed in different order in each ℓ_r .

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

```
1 for each  $m_{nd} \in M \setminus \{m^*\}$  do
  // stage 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      $\mathcal{L}_R \leftarrow \{\ell_{r_1}, \dots, \ell_{r_{|R|}}\}$  where each preference order is an empty one
8     if  $|R| \geq gap$  then
9       place  $m^*$  in in the highest position and  $m_{nd}$  in the second highest
        position, in  $\max(gap + \lceil (|R| - gap)/2 \rceil, 0)$  preference orders of  $\mathcal{L}_R$ 
10      place  $m^*$  in the second highest position and  $m_{nd}$  in the highest position in
        all of the other preference orders of  $\mathcal{L}_R$ 
11     else
12       place  $m^*$  in the highest position in each  $\ell_r \in \mathcal{L}_R$ 
13       place  $m_{nd}$  in the lowest position in  $\lfloor (gap - |R|)/(k - 2) \rfloor$  preference orders
        of  $\mathcal{L}_R \setminus \{\ell_{r_{|R|}}\}$ 
14       place  $m_{nd}$  in the second-highest position in all of the other preference
        orders of  $\mathcal{L}_R \setminus \{\ell_{r_{|R|}}\}$ 
15       in  $\ell_{r_{|R|}}$ , place  $m_{nd}$  in the highest position such that
         $(m^*, \mathcal{L} \cup \mathcal{L}_R) > (m_{nd}, \mathcal{L} \cup \mathcal{L}_R)$ 
16     if  $\mu(\hat{w}, \mathcal{L}_R) \neq m^*$  or  $m \notin o(\hat{w}, \mathcal{L}_R)$  then
17       continue to the next iteration
  // stage 2:
18    $B \leftarrow o(\hat{w}, \mathcal{L}_R) \setminus \{m^*, m_{nd}\}$ 
19   for each  $r \in R$  do
20     for each  $m \notin B$  do
21       | place  $m$  in the next highest available position in  $\ell_r$ 
22      $M^B \leftarrow B$ 
23     while  $M^B \neq \emptyset$  do
24       |  $b \leftarrow$  the least preferred man from  $M^B$  according to  $\mathcal{F}(\mathcal{L} \cup \mathcal{L}_R)$ 
25       | place  $b$  at the highest available position in  $\ell_r$ 
26       | remove  $b$  from  $M^B$ 
27   if  $\mu(\hat{w}, \mathcal{L}_R) = m^*$  then
28     | return  $\mathcal{L}_R$ 
29 return false
```

We now show that Algorithm 4 will succeed in any given instance such that the same instance but with one less manipulator is manipulable. That is, the coalitional MnM-w admits also a 1-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} , then for the same instance with one additional manipulator, Algorithm 4 will return a preference profile that makes m^* the match of \hat{w} .*

6 Conclusion

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 a 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.

References

1. Abdulkadiroğlu, A., Pathak, P.A., Roth, A.E.: The new york city high school match. *American Economic Review* **95**(2), 364–367 (2005)
2. Bossert, W., Sprumont, Y.: Strategy-proof preference aggregation: Possibilities and characterizations. *Games and Economic Behavior* **85**, 109–126 (2014)
3. Bossert, W., Storcken, T.: 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. Conitzer, V., Walsh, T.: Barriers to manipulation in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D. (eds.) *Handbook of Computational Social Choice*, pp. 127–145. Cambridge University Press (2016)

5. Davies, J., Katsirelos, G., Narodytska, N., Walsh, T.: Complexity of algorithms for borda manipulation. In: The Twenty-Fifth AAAI Conference on Artificial Intelligence. pp. 657–662 (2011)
6. Dogan, O., Lainé, J.: Strategic manipulation of social welfare functions via strict preference extensions. In: The 13th Meeting of the Society for Social Choice and Welfare. p. 199 (2016)
7. Dubins, L.E., Freedman, D.A.: Machiavelli and the gale-shapley algorithm. The American Mathematical Monthly **88**(7), 485–494 (1981)
8. Faliszewski, P., Procaccia, A.D.: Ai’s war on manipulation: Are we winning? AI Magazine **31**(4), 53–64 (2010)
9. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1), 9–15 (1962)
10. Hosseini, H., Umar, F., Vaish, R.: Accomplice manipulation of the deferred acceptance algorithm. In: The Thirtieth International Joint Conference on Artificial Intelligence. pp. 231–237 (2021)
11. Hosseini, H., Umar, F., Vaish, R.: Two for one & one for all: Two-sided manipulation in matching markets. arXiv preprint arXiv:2201.08774 (2022)
12. Huang, C.C.: Cheating by men in the gale-shapley stable matching algorithm. In: European Symposium on Algorithms. pp. 418–431. Springer (2006)
13. Perach, N., Anily, S.: Stable matching of student-groups to dormitories. European Journal of Operational Research (2022)
14. Pini, M.S., Rossi, F., Venable, K.B., Walsh, T.: Manipulation complexity and gender neutrality in stable marriage procedures. Autonomous Agents and Multi-Agent Systems **22**(1), 183–199 (2011)
15. Roth, A.E.: The economics of matching: Stability and incentives. Mathematics of operations research **7**(4), 617–628 (1982)
16. Roth, A.E.: The nrm as a labor market: understanding the current study of the match. Journal of the American Medical Association **275**(13), 1054–1056 (1996)
17. Roth, A.E., Sönmez, T., Ünver, M.U.: Kidney exchange. The Quarterly journal of economics **119**(2), 457–488 (2004)
18. Schmerler, L., Hazon, N.: Strategic voting in negotiating teams. In: International Conference on Algorithmic Decision Theory. pp. 209–223. Springer (2021)
19. Schmerler, L., Hazon, N., Kraus, S.: Strategic voting in the context of stable-matching of teams. arXiv preprint arXiv:2207.04912 (2022)
20. Shen, W., Deng, Y., Tang, P.: Coalitional permutation manipulations in the gale-shapley algorithm. Artificial Intelligence **301**, 103577 (2021)
21. Teo, C.P., Sethuraman, J., Tan, W.P.: Gale-shapley stable marriage problem revisited: Strategic issues and applications. Management Science **47**(9), 1252–1267 (2001)
22. Vaish, R., Garg, D.: Manipulating gale-shapley algorithm: Preserving stability and remaining inconspicuous. In: The Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 437–443 (2017)
23. Yu, W., Hoogeveen, H., Lenstra, J.K.: 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)
24. Zuckerman, M., Procaccia, A.D., Rosenschein, J.S.: Algorithms for the coalitional manipulation problem. Artificial Intelligence **173**(2), 392–412 (2009)