



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

*Università degli Studi di Padova*

*Padua Research Archive - Institutional Repository*

Matching with Partners and Projects

*Original Citation:*

*Availability:*

This version is available at: 11577/3308222 since: 2020-02-02T09:25:39Z

*Publisher:*

*Published version:*

DOI: 10.1016/j.jet.2019.104942

*Terms of use:*

Open Access

This article is made available under terms and conditions applicable to Open Access Guidelines, as described at <http://www.unipd.it/download/file/fid/55401> (Italian only)

(Article begins on next page)

# Matching with Partners and Projects <sup>\*</sup>

Antonio Nicolò<sup>†</sup>, Arunava Sen<sup>‡</sup>, Sonal Yadav<sup>§</sup>

August 22, 2019

## Abstract

We propose a model where agents are matched in pairs in order to undertake a project. Agents have preferences over both the partner and the project they are assigned to. These preferences over partners and projects are separable and dichotomous. Each agent partitions the set of partners into friends and outsiders, and the set of projects into good and bad ones. Friendship is mutual and transitive. In addition, preferences over projects among friends are correlated (homophily). We define a suitable notion of the weak core and propose an algorithm, the minimum demand priority algorithm (MDPA) that generates an assignment in the weak core. In general, the strong core does not exist but the MDPA assignment satisfies a limited version of the strong core property when only friends can be members of the blocking coalition. The MDPA is also strategy-proof. Finally we show that our assumptions on preferences are indispensable. We show that the weak core may fail to exist if any of the assumptions of homophily, separability and dichotomous preferences are relaxed.

---

<sup>\*</sup>We are extremely grateful to two reviewers and the editor of the journal for their extensive comments on earlier versions of the paper. We would also like to thank Yann Bramoullé, Jean-Jacques Herings, Michele Lombardi, Jordi Massó, Debasis Mishra, Herve Moulin, Hans Peters, Juan Sebastián Pereya, Ton Storcken, Fedor Sandomirskiy, Francesco De Sinopoli, Attila Tasnádi, William Thomson, Claudio Zoli and especially Ravindra B. Bapat, Sarvesh Bandhu, Marco LiCalzi and Thayer Morill for their comments and suggestions. This paper has also benefitted from the comments of seminar participants at Alicante, Bath, Budapest, Delhi, Glasgow, Maastricht, Nizhny Novgorod, Saint Petersburg, Venice, Verona and York. This paper was earlier circulated with the title “Matching with Homophily”. Nicolò and Yadav thank the financial support of the Università delgi Studi di Padova Strategic Project “Incorporating patients’ preferences in kidney transplant decision protocols” (STPD1159FJ).

<sup>†</sup>University of Padua, Padua, Italy and University of Manchester, UK.

<sup>‡</sup>Indian Statistical Institute, New Delhi, India.

<sup>§</sup>Umeå University, Umeå, Sweden.

# 1 INTRODUCTION

In many situations agents are matched in teams in order to work on a project. Agents have preferences over the project they are asked to work on as well as over the partners they are assigned to work with. Consequently, forming stable teams is important - it ensures that agents do not have opportunities to abandon their assignments and do better for themselves.

A centralized authority matches agents in pairs and assigns them a project. We are interested in mechanisms that satisfy stability and provide incentives to agents to truthfully reveal their preferences. This problem shares some features with roommate matching models since agents have preferences over their potential partners. However it also has common features with one-sided matching models like the object allocation model and the house allocation problem because a project has to be assigned to each pair of agents. In this sense our model is a hybrid version of two classical models. We introduce appropriate notions of the weak and strong cores in our framework. We show that strong core allocations no longer exist in this setting. The contribution of our paper is twofold: we introduce a new model of matching and identify a suitable restriction on preferences such that weak core allocations exist. We propose an algorithm that generates weak core allocations and provides incentives to agents to truthfully reveal their preferences. Our paper can be thought of as an attempt to extend the matching literature to team formation focussing on the case of matching in pairs in a well-defined but restricted preference domain.

Our model is a variant of the roommate problem where agents have preferences over potential roommates *and* available rooms. A relevant application is the assignment of primary school teachers in Italy. In Italian primary schools around one third of the teachers are assigned in pairs to a class<sup>1</sup> of students. The assignment of teachers to classes is done via a centralized mechanism at school level: the head of the school is responsible for matching the teachers in pairs and assigning each pair to a class of students. The two teachers work as a team, sharing the teaching load and doing some classroom activities jointly. It is therefore natural to assume that teachers have preferences over potential teaching partners and over the classes they teach.

Another example is the problem of assigning term papers to students. In several academic institutions, undergraduate or Master’s students are required to take “Project” courses. Students have to undertake some independent (non-coursework) research in these courses. Due to the large number of students, they are often assigned in pairs (and sometimes in

---

<sup>1</sup>Families can choose between two options (DPR, Presidential Decree 89, 2009): a standard program with 27 hours per week (“modules”) and an extended-time program with 40 hours per week (“full-time”). In the latter case two teachers are assigned to a class of pupils and they only teach those children (while in the former case a group of three or more teachers are assigned to a set of classes and usually teach the same subject in every class). According to the Italian Ministry of Education in 2018, there are 130,462 classes of primary schools in Italy and among these 43,804 adopt the extended-time program, around 33.6 percent. Students in Italian primary schools range from 6 to 11 years in age.

larger groups) to particular research topics. Students also benefit from learning from their peers. Grades in these courses depend in very large measure on a joint report prepared by a student team. As a result students have preferences over their assigned partner (often based on whether they “get on” with him/her and on their assessments of their abilities) and the project.

In our model there are a set of agents and a set of projects, and the number of projects is at least as large as the number of pairs of agents. We assume that it is not feasible for an agent to be unassigned. For instance, in the term paper assignment problem, every student must write a term paper for the course with another student. It is not an option for a student to not do a project. It would also be unfair for some students to work on their own while others work in pairs. Similarly dorm rooms are often built for sharing, and fairness would require all agents to share these rooms instead of some agents living on their own while others share. An alternative approach would be to assume a preference restriction: all agents strictly prefer to be assigned a partner and a project than to remain unassigned. We adopt the feasibility approach for convenience. Each agent has a preference ordering over partners and projects.

We investigate stable and strategy proof assignment rules in this setting. For stability, we employ standard notions of the core - the strong core and the weak core.<sup>2</sup> A coalition can block an assignment using either an unassigned project or a project assigned to a pair of agents *both of whom belong to the coalition*. An agent cannot unilaterally evict her partner from the assigned project and use this project with other members of the blocking coalition. A coalition can strongly block if all members of the blocking coalition are no worse-off and at least one member is strictly better off. It can weakly block if all members are strictly better off. Strong and weak core assignments are those that are immune to weak and strong blocking respectively. These concepts differ whenever indifferences occur in agent preferences (as they will in our model).

Neither the strong nor the weak cores exist in our model with an unrestricted domain of preferences. One of the contributions of our paper is to provide a plausible restriction on preferences under which the weak core and some variants of the strong core exist. We describe our preference domain below.

Preferences are *separable* over partners and projects. Also preferences over each component are *dichotomous* i.e. alternatives in each component are partitioned into good and bad sets.<sup>3</sup> The set of possible partners is partitioned into friends (good partners) and outsiders (bad partners), and the set of projects into good and bad projects. Therefore every partner, project pair can be placed into one of four indifference classes.<sup>4</sup> Friendship is mutual and

---

<sup>2</sup>See [Shapley and Scarf \(1974\)](#) and [Roth and Postlewaite \(1977\)](#).

<sup>3</sup>The assumptions of separability and dichotomous preferences are pervasive in the literature on voting, auctions and matching. Further references can be found in [Section 2.3](#).

<sup>4</sup>The four indifference classes are (good partner, good project), (good partner, bad project), (bad partner,

transitive, so that the set of agents can be partitioned into groups of friends. Finally, preferences of friends are correlated (*homophily*).<sup>5</sup> Specifically, we assume a strong alignment in the preference for projects among friends: for any pair of friends, the set of good projects of one of them is weakly contained in the set of good projects of the other one. However, two friends need not have the same set of good projects. One agent can be fussier than the other and like only a subset of the projects liked by the other agent. Notice that homophily is satisfied when the projects can be ordered according to a criterion such as the level of complexity, degree of riskiness etc. and a threshold divides the projects into good and bad ones. Friends evaluate the projects using the same criterion but they may have different thresholds.

In our opinion, our model restrictions in the Italian primary teacher assignment application are believable. Teachers have to be matched in pairs to a “project” (a class) and cannot remain unmatched. Preferences over colleagues and classes are likely to be independent and the fact that teachers divide the set of colleagues and classes into those they like more and those they like less, seems plausible. Teachers typically have correlated preferences over classes<sup>6</sup> and may have common criteria to evaluate them. For instance, teachers may agree that it is more difficult to teach a class in which there are many children from non-native language speaking families. However they are likely to disagree on the critical number of such students that make a class “hard” to teach.

Our main results are as follows. We show that the strong core may fail to exist in our domain. We propose an algorithm, the minimum demand priority algorithm (MDPA) that generates a weak core assignment. We also consider an intermediate core notion which we call the friendship core. In this notion, agents can weakly block but all members of the coalition must be friends. We show that the friendship core exists subject to certain cardinality restrictions.

We also investigate the incentive properties of the MDPA. Since friendship relationships are mutual, we assume that they are common knowledge. However agent preferences about good projects is private information. There is a conceptual difficulty arising from the fact that homophily is a restriction on preference profiles, rather than on individual preferences. Nevertheless we find a convenient way to deal with this issue. We assume that for every group of friends, there is a common ordering over projects in terms of acceptability. Each agent is characterized by a threshold level of acceptability - all projects “less” than the threshold

---

good project) and (bad partner, bad project).

<sup>5</sup>Sociological literature provides ample evidence of the existence of homophily, that is, the tendency of individuals to connect and form close ties with other individuals who are similar to them. See [Cohen \(1977\)](#), [Kandel \(1978\)](#), [Verbrugge \(1983\)](#), [McPherson et al. \(2001\)](#), [Golub and Jackson \(2012\)](#).

<sup>6</sup>Boyd et al. (2013) show that in US teachers exhibit preferences for schools that are closer geographically, are suburban, have a smaller proportion of students in poverty, and, for white teachers, have a smaller proportion of minority students. Here we look at teachers’ preferences once they are already assigned to a school.

project are good for the agent. A profile of thresholds generates a profile of preferences satisfying homophily and strategy-proofness can be then defined in the usual way. We show the MDPA algorithm is strategy-proof.

The key assumptions underlying our preference domain are separability, dichotomous preferences and homophily. We show the weak core fails to exist if any one of the three is dropped, while maintaining the other two. Our preference restrictions are indispensable in this sense.

Earlier in the introduction, we alluded to the fact that our model is a combination of the house allocation model and the roommate problem. We would like to emphasize that our model is distinct from both - in particular, none of our results follow from results in either model.

## 1.1 EXISTING LITERATURE

To the best of our knowledge, our model is distinct from others in the literature. However there are existing models that bear some resemblance to ours. We discuss some of these connections below.

Our model extends the classical roommate problem (see [Roth and Sotomayor \(1992\)](#) for a discussion) by allowing agents to have preferences over roommates *and* rooms. Stable allocations in the roommate problem do not always exist. [Tan et al. \(1991\)](#), [Gudmundsson \(2014\)](#) and [Abizada \(2016\)](#) identify preference restrictions that guarantee the existence of stable allocations. The existence problem in our model is more severe than in the roommate problem. We discuss this issue in [Section 2](#).

Another related model is the “stable activities” (or SA) variant of the roommate problem studied in [Cechlárová and Fleiner \(2005\)](#). In the SA model different kinds of partnerships are possible between every pair of agents. A partnership between agents can be thought of as a common activity such as playing chess, going to the movies together and so on. Each agent has preferences over partner, activity pairs in which respect it is similar to our model. However activities can be replicated arbitrarily in the SA model, i.e. several pairs can be assigned the same activity. In our model, on the other hand, a project can be assigned to only one pair if it is assigned at all. This makes the two problems substantively different. For instance, pairs of agents in our model can block by swapping their projects, but this notion is meaningless in the SA problem. The main result of the [Cechlárová and Fleiner \(2005\)](#) paper establishes an equivalence between the SA problem and the roommates problem. However, it is not possible to extend a stable roommate assignment to a weak core assignment in the partner, project model - see [example 3](#) in [Section 2](#).

[Combe \(2017\)](#) and [Sethuraman and Smilgins \(2016\)](#) consider a model where two distinct sets of agents (men and women) have to be matched in pairs to a common set of objects (houses). Agents’ of each type have preferences on the house and the partner they are

matched with.<sup>7</sup> The cardinality of the the set of agents and houses is assumed to be equal. Both papers show that a stable matching may fail to exist in this setting. Both papers introduce a notion of *house ownership* whereby one of the agents assigned to a house is made the owner of the house. Ownership places restrictions on possible deviations by agents. They show that if agents on one side of the market are given ownership, the classical result of Gale and Shapley applies and a stable match (with respect to the ownership structure), is guaranteed. [Combe \(2017\)](#) goes on to show that even though stable matches exist for one-sided ownership structures, the core may be empty unless further restrictions on preferences are introduced. Our model, is similar to the models considered in these papers, but differs in significant ways. One of these differences is that we assume that there are more projects available than required. This makes blocking much easier and the ownership approach less compelling. Our approaches to the fundamental non-existence issue in these models are quite distinct. Consequently our results and the arguments that undergird them, are also distinct both in substance and flavour.

[Pycia \(2012\)](#) proves a very general existence result on coalition stability. However, his model does not cover ours because of the presence of objects in our model. Since objects do not have preferences over agents they are assigned to, our notions of blocking do not have a counterpart in his setting. In particular, if objects are interpreted as agents, deviating coalitions that involve objects cannot strictly improve. The difference between the models is manifested in the results as well. According to the main result of [Pycia \(2012\)](#), a coalition structure is stable if and only if agents’ preferences are “pairwise-aligned”<sup>8</sup> i.e. every pair of agents rank coalitions that contain both of them, in the same way. Our preference restrictions do not satisfy pairwise alignment, yet the weak core exists.

A special case of the question investigated in [Pycia \(2012\)](#) is the existence of stable threesome matchings. [Alkan \(1988\)](#) considers societies where there are three sets of agents (men, women and children) and a matching consists of distinct triples, each formed by a man, a woman and a child. Agents have preferences over the pairs they are matched with; thus a blocking coalition is a triple, where each member in the coalition strictly improves. [Alkan \(1988\)](#) shows that stable matchings fail to exist, even when the preferences are restricted to be separable. [Biró and McDermid \(2010\)](#) considers a three-sided model with cyclic preferences, where men care only about women, women only about children and children only about men. As discussed in the earlier paragraph, our model cannot be interpreted as a special case of these models.

In a slightly different spirit, [Raghavan \(2018\)](#) considers an allocation problem where agents have to be assigned in pairs to objects. An object is either assigned to a unique pair of agents or not assigned at all. However, unlike in the roommate problem and ours,

---

<sup>7</sup>[Burkett et al. \(2018\)](#) analyze two versions of the random serial dictatorship mechanism in a model where pairs of roommates have to be assigned to rooms and agents care about their roommates and their rooms.

<sup>8</sup>Certain richness assumptions on preferences are made as well.

agents only have preferences over objects and not on potential partners. The paper examines allocation rules from the perspective of strategy-proofness and efficiency.

An important strand in the recent literature on matching is concerned with matching with contracts (see [Ostrovsky \(2008\)](#), [Hatfield and Kominers \(2012\)](#) etc.). Our model can be interpreted as roommate matching problem with contracts. However to the best of our understanding, the results in this literature do not have any bearing on ours. It is possible to reformulate our model to make it consistent with that of [Hatfield and Kominers \(2012\)](#). However their notion of blocking is stronger than ours. On the other hand, their requirement of cross-side complementarity is not satisfied by our preferences. Hence full-substitutability also fails and the existence of weak core assignments cannot be inferred from their result.

Our model is also related distantly to the those on the existence of stable matchings in two-sided matching models where couples are looking for jobs in the same labour market. A convenient example is the one where doctors are seeking internships at hospitals and some of the doctors are couples. (See [Klaus and Klijn \(2005\)](#), [Klaus and Klijn \(2007\)](#), [Klaus et al. \(2007\)](#) and [Khare and Roy \(2018\)](#)). In these models, the identity of the couples is pre-determined. Every agent has an individual preference on the hospitals and a joint preference on the possible hospital duples. Hospitals also have preferences over the students. It is clear that our model is fundamentally different from these models - in our case, the formation of couples is endogenous while there is no counterpart of the hospital agent.

The remainder of the paper is organized as follows. Section 2 presents the model while Section 3 informally describes the MDPA algorithm. Section 4 contains the main results of the paper. Section 5 discusses the indispensability of our preference restrictions. Section 6 concludes. Appendix 1 provides a formal description of the MDPA algorithm. The proofs are contained in Appendices 2 and 3.

## 2 THE MODEL

There is a finite set of agents  $N = \{1, 2, \dots, i, j, \dots, n\}$  and a finite set of projects,  $A = \{a, b, c, d, \dots\}$ . We assume that  $|N| = 2m$  for some integer  $m \geq 2$  and that  $|A| \geq m$ .

An *assignment*  $\sigma$  is a collection of triples  $(i, j, a)$  with the interpretation that agent pair  $(i, j)$  is assigned project  $a$ . We require all agents to be assigned a partner and a project. Each agent is paired with exactly one other agent. Every project is assigned to exactly one pair of agents or left unassigned.

We say that the triple  $(i, j, a)$  is an element of  $\sigma$  if the pair  $(i, j)$  is assigned to the project  $a$  in  $\sigma$ . Finally let  $u^\sigma$  denote the set of unassigned projects in the assignment  $\sigma$  i.e. it is the set of projects  $a$  such that  $(i, j, a) \notin \sigma$  for every  $i, j \in N$ .

Let  $\Sigma$  denote the set of all feasible assignments.



Each agent  $i$  has a preference ordering<sup>9</sup>  $\succsim_i$  over partner, project pairs  $(j, a)$  where  $j \neq i$ . We denote the asymmetric and symmetric components of  $\succsim_i$  by  $\succ_i$  and  $\sim_i$  respectively. A preference profile  $\succsim \equiv (\succsim_1, \dots, \succsim_n)$ .

## 2.1 BLOCKING AND STABILITY

We introduce notions of blocking and stability appropriate for our model.

Let  $\sigma$  be an assignment and let  $S \subset N$  be a coalition such that  $|S|$  is even. The set of projects available to the coalition  $S$  is  $X(S, \sigma)$  where

$$X(S, \sigma) = \{a \in A : (i, j, a) \in \sigma \text{ and } i, j \in S\} \cup u^\sigma.$$

A project  $a$  belongs to  $X(S, \sigma)$  if either  $a$  is unassigned in  $\sigma$  or agents  $i, j \in S$  are matched together and assigned  $a$  in  $\sigma$ .

We say that  $\sigma'$  is  $X(S, \sigma)$ -feasible if each agent in  $S$  is paired with another agent in  $S$  and each such pair is assigned a project from  $X(S, \sigma)$ . Note that feasibility of  $\sigma'$  requires  $|X(S, \sigma)| \geq \frac{|S|}{2}$ .

**DEFINITION 1** *A coalition  $S$  (with  $|S|$  even) strongly blocks  $\sigma$  at preference profile  $\succsim$  if there exists  $\sigma'$  which is  $X(S, \sigma)$ -feasible and  $(k, b) \succ_i (j, a)$  for all  $i \in S$  where  $(i, k, b) \in \sigma'$  and  $(i, j, a) \in \sigma$ . An assignment is in the weak core if it is not strongly blocked by any coalition.*

**DEFINITION 2** *A coalition  $S$  (with  $|S|$  even) weakly blocks  $\sigma$  at preference profile  $\succsim$  if there exists  $\sigma'$  which is  $X(S, \sigma)$ -feasible,  $(k, b) \succsim_i (j, a)$  for all  $i \in S$  where  $(i, k, b) \in \sigma'$ ,  $(i, j, a) \in \sigma$  and  $(k, b) \succ_l (j, a)$  for some  $l \in S$  where  $(l, k, b) \in \sigma'$ ,  $(l, j, a) \in \sigma$ . An assignment is in the strong core if it is not weakly blocked by any coalition.*

The notion of the strong and weak core are well known in the literature. In the weak core, every member of the blocking coalition must strictly improve. In the strong core, no member of the blocking coalition can be worse off and at least one agent must be strictly better off. It is easy to show if the preferences are anti-symmetric (admit no indifference) then the strong and weak cores coincide. In our model, indifferences will be pervasive and the distinction between the two notions of the core will be important.

There is another feature of blocking in our model that we would like to comment on. Suppose  $\sigma$  is an assignment where agents 1, 2 are paired together with project  $a$ , while 3, 4 are paired together with project  $b$ . Let  $S = \{1, 3\}$ . According to our definition of  $X(S, \sigma)$ , projects  $a$  and  $b$  are *not* available to  $S$ . More generally, an assigned project  $a$  is available for blocking only if the coalition  $S$  contains both agents who were assigned to  $a$  in  $\sigma$ . Both agents who have been paired together with a project have “rights” over the project. No agent can

---

<sup>9</sup>An ordering is a binary relation which is complete, reflexive and transitive.

$\succsim_1$	$\succsim_2$	$\succsim_3$	$\succsim_4$
(2,.)	(3,.)	(1,.)	(1,.)
.	.	.	.
(3,.)	(1,.)	(2,.)	(2,.)
.	.	.	.
(4,.)	(4,.)	(4,.)	(3,.)
.	.	.	.

Table 1: Preferences of agents in Example 1.

evict her partner and use their assigned project with other members of the blocking coalition. Of course, 1 and 3 can abandon their partners and block with an unassigned project.

We refer to *blocking via unassigned projects* when an assignment can be blocked by a pair of agents who abandon their current partners and choose an unassigned project in the assignment which strictly improves their welfare from the initial situation.

## 2.2 THE GENERAL NON-EXISTENCE OF THE CORE

Neither the weak nor the strong core exist without restrictions on preferences. This is not surprising in view of the fact that it is well known that the core does not exist in the roommate problem.<sup>10</sup> Recall that the roommate problem is one of forming pairs from a set of agents where each agent has a preference ordering over possible roommates. Non-existence in our model can be obtained from the non-existence in the roommate problem by simply making agents care lexicographically more about partners than projects. Consider the example below.

**EXAMPLE 1** Let  $N = \{1, 2, 3, 4\}$  and  $A = \{x, y, z\}$ . Table 1 specifies the preferences of the agents. Agent 1 prefers to be paired with agent 2 and any project to being paired with agent 3 and any project which in turn is preferred to being paired with agent 4 and any project. The preferences of agents 2, 3 and 4 are interpreted in a similar way.

Assume without loss of generality that agent 3 is matched with agent 4 together with project  $x$ . Therefore 1 and 2 are matched together with some project, say  $z$ . Since 3 is agent 2's favourite partner and 3 is paired with her least favourite partner, agents 2 and 3 can form a blocking coalition with unassigned project  $y$ .

The non-existence problem in our model is however deeper and more pervasive than suggested by the connection to the roommate problem. We provide two examples to demonstrate our claim. In Example 2, the core does not exist even though agent preferences over

---

<sup>10</sup>A discussion can be found in Tan et al. (1991) and Gudmundsson (2014).

$\succ_1^{proj}$	$\succ_2^{proj}$	$\succ_3^{proj}$	$\succ_4^{proj}$	$\succ_5^{proj}$	$\succ_6^{proj}$
$a$	$b$	$c$	$d$	$e$	$f$
$b$	$a$	$b$	$e$	$f$	$e$
$c$	$c$	$a$	$f$	$d$	$d$
$d$	$d$	$d$	$c$	$c$	$c$
$e$	$e$	$e$	$b$	$b$	$b$
$f$	$f$	$f$	$a$	$a$	$a$

Table 2: Preferences of agents over projects in Example 2.

projects are lexicographically dominant. Example 3 which is a small variation of Example 2 shows that the core does not exist in our model even when the associated roommate problem has a non-empty core.

**EXAMPLE 2** Let  $N = \{1, 2, 3, 4, 5, 6\}$  and  $A = \{a, b, c, d, e, f\}$ . We define the following linear order  $>_o$  over projects:  $a >_o b >_o c >_o d >_o e >_o f$ .

Each agent  $i$  has a preference ordering  $\succ_i^{proj}$  over the set  $A$  specified in Table 2. Observe that  $\succ_i^{proj}$  is *single-peaked* with respect to the ordering  $>_o$ . In particular, agents 2 and 3 are left-oriented i.e. any project which lies to the left of the agent's peak is preferred to a project which lies to the right of her peak. Similarly agents 4 and 5 are right-oriented. For notational convenience, we shall denote the peak of  $\succ_i^{proj}$  by  $\tau(i)$ . Thus  $\tau(1) = a$  etc.

Each agent  $i$  has an arbitrary preference ordering  $\succ_i^{part}$  over the set  $N \setminus \{i\}$ .

The two orderings  $\succ_i^{proj}$  and  $\succ_i^{part}$  are combined to generate an ordering  $\succ_i$  over partner,project tuples as follows:  $(j, a) \succ_i (k, b)$  if either  $a \succ_i^{proj} b$  or  $a = b$  and  $j \succ_i^{part} k$ . Thus  $\succ_i$  is lexicographic and  $\succ_i^{proj}$  is the dominant component.

We claim that the core is empty at the profile  $\succ = (\succ_1, \dots, \succ_6)$ .

Consider an arbitrary assignment. We claim that if the assignment is in the weak core at  $\succ$ , then at least one of the projects in  $\{\tau(1), \tau(2)\}$  and one of the projects in  $\{\tau(5), \tau(6)\}$  must be assigned.

Suppose neither  $\tau(1)$  nor  $\tau(2)$  are assigned. Then neither agents 1 nor 2 are getting their first and second ranked projects. They can then form a blocking coalition by choosing one of the unassigned projects  $\tau(1)$  or  $\tau(2)$ . This would give each agent either their first or second ranked project. By a similar argument, one of the projects in  $\{\tau(5), \tau(6)\}$  must be assigned.

Since there are six projects and six agents, exactly three projects must be assigned. Suppose the third project assigned is the unassigned project in  $\{\tau(1), \tau(2)\}$  i.e. the assigned projects are  $\tau(1)$ ,  $\tau(2)$ , and either  $\tau(5)$  or  $\tau(6)$ . Then at least one of the agents  $i \in \{4, 5, 6\}$  is assigned to either  $\tau(1)$  or  $\tau(2)$ . Observe also that agent 3 is not getting her best project since  $\tau(3) \in u^\sigma$ . Now  $(i, 3, \tau(3))$  is a blocking coalition. This is true because single-peakedness implies agent  $i$  strictly prefers  $\tau(3)$  to  $\tau(1)$  or  $\tau(2)$ .

By a symmetric argument, it cannot be the case that the assigned projects are  $\tau(5), \tau(6)$  and either  $\tau(1)$  or  $\tau(2)$ .

The only remaining possibility is that there is one assigned project in  $\{\tau(1), \tau(2)\}$ , one in  $\{\tau(5), \tau(6)\}$  and one in  $\{\tau(3), \tau(4)\}$ . Assume  $\tau(4)$  is the assigned project.

Observe that either agent 1 or 2 is not getting her best project. Suppose 1 is the agent who is not getting her best project. There are two cases to consider. The first is when  $\tau(1)$  is assigned (clearly not to agent 1) and the second is when  $\tau(2)$  is assigned.

In the first case,  $\tau(2)$  is unassigned. Therefore 1 and 2 can block with  $\tau(2)$ . In the second case, there are four possibilities.

- (i) Agent 1 is getting  $\tau(2)$  and agent 3 is not, i.e. agent 3 is getting a project in  $\{\tau(4), \tau(5), \tau(6)\}$ . Since 3 is left-oriented, 1 and 3 block via the unassigned project  $\tau(1)$ .
- (ii) Agent 3 is getting  $\tau(2)$  but agent 1 is not, i.e. agent 1 is getting a project in  $\{\tau(4), \tau(5), \tau(6)\}$ . Then agents 1 and 3 block via the unassigned project  $\tau(3)$ .
- (iii) Neither agent 1 nor agent 3 is getting  $\tau(2)$ . Here agents 1 and 3 block via one of the unassigned projects  $\tau(1)$  or  $\tau(3)$ .
- (iv) Agents 1 and 3 are matched with project  $\tau(2)$ . Hence agent 2 is assigned a project in  $\{\tau(3), \tau(4), \tau(5), \tau(6)\}$ . Since agent 2 is left-oriented, agents 1 and 2 block the assignment via the unassigned project  $\tau(1)$ .

The other case is where agent 2 is not getting her best project. Once again, there are two possibilities. The first is when  $\tau(2)$  is assigned. So by assumption,  $\tau(1)$  is unassigned. Then agents 1 and 2 can block with  $\tau(1)$ . The remaining case is when  $\tau(1)$  is assigned. So 2 and 3 are assigned projects in  $\{\tau(1), \tau(4), \tau(5)\}$ . Then agents 2 and 3 block via the project  $\tau(2)$ .

The last case is when  $\tau(3)$  is the third assigned project instead of  $\tau(4)$ . This case can be dealt with using analogous arguments. These arguments establish that the core does not exist.<sup>11</sup>

In the following example, the associated roommate problem has a non-empty core while the core is empty in our model.

**EXAMPLE 3** The setting is the same as in Example 2. Preferences are also the same, except that agents' preferences over partners ( $\succ_i^{part}$ ) are as specified in Table 3.

---

<sup>11</sup>The model in Raghavan (2018) can be embedded in our setting by assuming that all agents' are indifferent about their partners. Example 2 shows that the weak core does not exist in this case.

$\succ_1^{part}$	$\succ_2^{part}$	$\succ_3^{part}$	$\succ_4^{part}$	$\succ_5^{part}$	$\succ_6^{part}$
2	1	4	3	6	5
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Table 3: Preferences of agents over partners  $\succ_i^{part}$  in Example 3.

Consider the problem where the preferences over projects are ignored, i.e. we consider a pure roommate problem with preferences shown in Table 3. In this problem, the assignment (1, 2), (3, 4), (5, 6) belongs to the core. However we know from Example 2, that the core does not exist.

In the next subsection, we propose some preference restrictions under which a version of the core exists.

### 2.3 THE PREFERENCE DOMAIN

Each agent  $i$  has a preference ordering  $\succsim_i$  over partner, project tuples in  $N \setminus \{i\} \times A$ . We allow for weak preferences. The asymmetric and symmetric components of  $\succsim_i$  are denoted by  $\succ_i$  and  $\sim_i$  respectively.

We assume preferences satisfy separability, dichotomous marginal orderings and homophily. We describe each of them successively.<sup>12</sup>

**DEFINITION 3** *The ordering  $\succsim_i$  is separable if for all  $a \neq b \in A$  and  $j \neq k \in N \setminus \{i\}$ , we have (i)  $(j, a) \succ_i (k, a) \iff (j, b) \succ_i (k, b)$  and  $(j, a) \sim_i (k, a) \iff (j, b) \sim_i (k, b)$  and (ii)  $(j, a) \succ_i (j, b) \iff (k, a) \succ_i (k, b)$  and  $(j, a) \sim_i (j, b) \iff (k, a) \sim_i (k, b)$ .*

Separability implies that marginal preferences over partners and projects are well-defined. We refer to the marginal preferences over partners and projects by  $\succsim_i^{part}$  and  $\succsim_i^{proj}$  respectively. We assume  $\succsim_i^{part}$  and  $\succsim_i^{proj}$  are *dichotomous*.

<sup>12</sup>As we have noted earlier in the Introduction, the assumptions of separability and dichotomous preferences have been used extensively in the literature. A few instances are noted below. Separable preferences have been used in multi-dimensional voting (Border and Jordan (1983), Barberà et al. (1991), Breton and Sen (1999)), in combinatorial auctions (Hart and Nisan (2017), Hart and Reny (2015)), multi-dimensional screening (Carroll (2017)) and matching (Kurino (2014)). Dichotomous preferences have been employed in voting (Bogomolnaia et al. (2005), Gaurav et al. (2017)), auctions (Mishra and Roy (2013)) and matching (Bogomolnaia and Moulin (2004), Roth et al. (2005)).

**DEFINITION 4** Fix a separable ordering  $\succsim_i$ . The induced marginal ordering  $\succsim_i^{part}$  is dichotomous if there exists a set  $P^i(\succsim_i) \subseteq N \setminus \{i\}$  such that (i)  $j \succ_i^{part} k \iff j \in P^i(\succsim_i), k \notin P^i(\succsim_i)$  and (ii)  $j \sim_i^{part} k \iff$  either  $a, b \in P^i(\succsim_i)$  or  $a, b \notin P^i(\succsim_i)$ . The induced marginal ordering  $\succsim_i^{proj}$  is dichotomous if there exists a set  $G^i(\succsim_i) \subseteq A$  such that (iii)  $a \succ_i^{proj} b \iff a \in G^i(\succsim_i), b \notin G^i(\succsim_i)$  and (iv)  $a \sim_i^{proj} b \iff$  either  $a, b \in G^i(\succsim_i)$  or  $a, b \notin G^i(\succsim_i)$ . The ordering  $\succsim_i$  is dichotomous if both its induced marginal orderings  $\succsim_i^{part}$  and  $\succsim_i^{proj}$  are dichotomous.

We refer to the sets  $P^i(\succsim_i)$  and  $G^i(\succsim_i)$  as the set of good partners (friends) and good projects for agent  $i$ . We allow for the possibility of  $P^i(\succsim_i), G^i(\succsim_i) = \emptyset$ . The complement sets of  $P^i(\succsim_i)$  and  $G^i(\succsim_i)$  will be referred to as the set of bad partners and bad projects respectively.

As we have remarked earlier, dichotomous preferences are a natural preference domain and have been used in several matching and allocation models. An immediate consequence of this assumption is that agent preferences have “large” indifference classes.

According to these preferences all partner, project pairs can be placed in one of four indifference classes: (I) both partner and project are good, (II) only the partner is good (III) only the project is good and (IV) neither partner nor project is good. We shall refer to the classes (I), (II), (III) and (IV) as  $(G, G)$ ,  $(G, B)$ ,  $(B, G)$  and  $(B, B)$  respectively. In general, we adopt the convention that the first component in the ordered pairs refers to the type of partner and the second to the type of project.

Separability implies that  $(G, G)$  is the most preferred equivalence class and  $(B, B)$ , the worst. Classes  $(G, B)$  and  $(B, G)$  can be ranked either way. We refer to the case where  $(G, B)$  is ranked above  $(B, G)$  as a *partner dominant* preference. In this case agent  $i$  prefers a good partner, bad project pair over a bad partner, good project pair.

Similarly, a *project dominant* preference is one where  $(B, G)$  is ranked above  $(G, B)$  i.e. agent  $i$  prefers a bad partner, good project pair over a good partner, bad project pair.

We assume the following (i) if  $j \in P^i(\succsim_i)$  then  $i \in P^j(\succsim_j)$  and (ii) if  $i \in P^j(\succsim_j)$  and  $j \in P^k(\succsim_k)$ , then  $i \in P^k(\succsim_k)$ . Thus the friendship relationship is mutual (if  $i$  is  $j$ 's friend, then  $j$  is  $i$ 's friend) and transitive (if  $i$  is  $j$ 's friend and  $k$  is  $i$ 's friend, then  $k$  is  $i$ 's friend).

These assumptions induce a partition on the set of agents  $N$  where each element in the partition is a set of agents who are all friends. We shall refer to each element of this partition as a *friendship component*. These components are labelled  $\{F_1, F_2, \dots, F_L\}$ . Note that the number of components which have an odd number of agents must be even.

The final assumption on preferences is that of *homophily*. Unlike separability and dichotomous preferences, homophily is an assumption on preference profiles.

**DEFINITION 5** Let  $\succsim \equiv (\succsim_1, \dots, \succsim_n)$  be a preference profile where each  $\succsim_i$  is a separable and dichotomous preference ordering. Then  $\succsim$  satisfies homophily if for any  $i, j \in N$  such that  $i, j$  are friends, we have either  $G^i(\succsim_i) \subseteq G^j(\succsim_j)$  or  $G^j(\succsim_j) \subseteq G^i(\succsim_i)$ .

$F_1$	$F_2$
1 $\{x, y, z\}$	4 $\{z\}$
2 $\{x, y, z\}$	
3 $\{x, y, z\}$	

Table 4: Preferences of agents in Example 4.

This induces an alignment in the preference for projects among friends. There cannot be a pair of projects  $a, b$  and a pair of friends  $i, j$  such that  $i$  likes  $a$  but not  $b$ , while  $j$  likes  $b$  but not  $a$ . However  $i$  and  $j$  need not have the same set of good projects, one agent can be more fussy than the other and like only a subset of the projects liked by the other agent.

## 2.4 FURTHER REMARKS ON THE CORE

The next example shows that the strong core may not exist in our preference domain.

**EXAMPLE 4** Let  $N = \{1, 2, 3, 4\}$  and  $A = \{x, y, z\}$ . Table 4 summarizes the friendship components and the sets of good projects for the agents. All agents have partner dominant preferences.

In any assignment, an agent in  $F_1$  is paired with agent 4. Consider an arbitrary assignment, say  $(1, 2, x), (3, 4, z)$ . In this assignment, project  $y$  is unassigned. Agents 1, 3 (or 2, 3) can form a blocking coalition using project  $y$ . Here agent 1 (or 2) remains indifferent by blocking while agent 3 strictly improves. Thus the strong core does not exist.

In view of the non existence of the strong core, we investigate two weaker notions. In the first, agents can weakly block but the blocking coalition must consist entirely of friends. We call this notion *friendship core*. In the second, we consider the existence of a strengthened version of the weak core which we call the *robust weak core*.

**DEFINITION 6** *An assignment is in the  $F_q$ -friendship core (for some  $q \in \{1, \dots, L\}$ ) at  $\succsim$  if no coalition  $S \subseteq F_q$  can weakly block it at  $\succsim$ . An assignment is in the friendship core at  $\succsim$  if it is in the  $F_q$ -friendship core at  $\succsim$  for all  $q \in \{1, 2, \dots, L\}$ .*

We note in Example 4 that the  $F_1$ -friendship core does not exist. An aspect of the example is that  $F_1$  contains an odd number of agents. We will show that our algorithm will generate an assignment in the  $F_q$ -friendship core whenever  $|F_q|$  is even.

The second variant of the core that we consider is the weak core. We show that the weak core exists in our preference domain. In fact, we show that the assignment in the weak core satisfies some stronger properties which we define below.

The profile  $\succsim'$  is *good-set equivalent* to the profile  $\succsim$  if  $P^i(\succsim'_i) = P^i(\succsim_i)$  and  $G^i(\succsim'_i) = G^i(\succsim_i)$  for all  $i$ .

**DEFINITION 7** *An assignment is in the robust weak core at  $\succsim$  if it is in the weak core at every  $\succsim'$  that is good-set equivalent to  $\succsim$ .*

This is a desirable property because it implies that the only information that has to be elicited from the agents in order to propose an assignment in the weak core is their set of good partners and projects. Information regarding the partner or project dominance of agent preferences is not required.

### 3 THE MINIMUM DEMAND PRIORITY ALGORITHM

We describe an algorithm to generate an assignment which we refer to as the *Minimum Demand Priority Algorithm* (MDPA). We provide here an informal description of the algorithm. The formal detailed description can be found in Appendix 1.

Let  $\succ^N$  and  $\succ^A$  be fixed orderings of the sets  $N$  and  $A$  respectively. If agent  $i \succ^N j$ , then agent  $i$  has priority over  $j$ . Let  $\succ^F$  be a fixed ordering over friendship components  $\{F_1, \dots, F_L\}$  where  $F_1 \succ^F F_2 \succ^F \dots \succ^F F_L$ . If  $F_q$ ,  $q \in \{1, \dots, L\}$  has an odd number of agents we remove the last agent in this component according to  $\succ^N$ . The agents who have been removed from their respective components are called residual agents.

We then proceed to assign projects to pairs of agents who belong to the same component, starting from  $F_1$  following the fixed ordering  $\succ^F$  (note that every component is now formed by an even number of friends).

Consider component  $F_1$  and project  $a \in A$ . The demand for  $a$  in  $F_1$  is the cardinality of the set of agents belonging to  $F_1$  who consider project  $a$  good. We select the project with the minimum positive demand in  $F_1$ , breaking ties according to  $\succ^A$ : let  $x \in A$  be this project. If there are at least two agents who demand  $x$  (i.e. for whom  $x$  is good), we assign  $x$  to the pair of agents with highest priority according to  $\succ^N$  in the set of agents who demand  $x$ . This triple is then removed from further consideration in the algorithm. If there is only one agent who demands  $x$ , we provisionally assign  $x$  to the (only) agent who demands it. We call this pair, formed by a project and an agent without a partner, a *waiting pair*.

In the next step, we again compute the demand for each *unassigned* project in  $F_1$ , excluding the agents to whom a project has already been assigned, either definitively or provisionally. We select the project with the minimum positive demand, breaking ties according to the fixed ordering  $\succ^A$ . Suppose this project is  $y$ . If there are at least two agents who demand  $y$ , we assign  $y$  to the pair of agents with the highest priority according to  $\succ^N$  in the set of agents who demand  $y$ . We remove this triple from further consideration. If the demand for  $y$  is exactly one and  $j$  is the agent who considers  $y$  good, then there are two



cases to consider. The first is that a waiting pair already exists. Note that this waiting pair must be  $(i, x)$ . We form the triple  $(i, j, y)$  and remove it from further consideration in the algorithm. The project  $x$  is now *unassigned* and is available for future assignments. The second case is that there is no waiting pair. Here  $(j, y)$  becomes a waiting pair for the next step.

We proceed in this manner to subsequent steps calculating at each step, the demand in  $F_1$  for every unassigned project, excluding agents to whom a project has already been assigned, either definitively or provisionally. Note that at the end of each step there is at most one waiting pair.

It is clear that we must finally reach a step where the demand in  $F_1$  for each unassigned project is zero. The first possibility is that all agents in  $F_1$  have already been matched and this completes the assignment for  $F_1$ . The second possibility is that there exists a set of agents in  $F_1$  who are still unmatched. If this set has even cardinality, we arrange the unmatched agents using  $\succ^N$ , form pairs of consecutive agents and assign projects to each pair in sequence using  $\succ^A$  from the set of unassigned projects. The final possibility is that the set of unmatched agents has odd cardinality. This implies that there exists a waiting pair. The unmatched agent with highest priority using  $\succ^N$  is matched with the waiting pair to form a triple. If there are still unmatched agents, (notice that their number is even), we proceed as above, forming pairs of consecutive agents (according to  $\succ^N$ ) and assign projects to each pair in sequence using  $\succ^A$  from the set of unassigned projects.

We repeat this procedure for every component in sequence till the last one,  $F_L$ . Note that the set of projects we can assign to agents in component  $F_l$ , where  $1 < l \leq L$  is given by the set of remaining projects, calculated by removing all projects previously assigned to pairs in components  $F_1, \dots, F_{l-1}$  from the set  $A$ .

Once we have completed the matching of agents in  $F_L$ , we assign projects to residual agents. We use  $\succ^N$  to form as many pairs of agents with common good projects as possible. We then match the remaining agents in pairs and assign to each pair a project which is good for at least one agent (favouring the agent with the highest priority according to the fixed ordering  $\succ^N$ ). Finally, agents who do not have any good project available, are given projects from the set of remaining projects according to  $\succ^A$ .

We illustrate it with an example.

**EXAMPLE 5** Let  $N = \{1, 2, \dots, 24\}$  and  $A = \{a, b, \dots, z\}$ . Table 5 summarizes the friendship components and the sets of good projects for the agents. The priority order of agents is  $1 \succ^N 2 \succ^N 3 \dots \succ^N 24$  and the priority order of projects is  $z \succ^A y \dots \succ^A a$ .

Step 0: Each component has an odd number of agents. We remove agent 11 from  $F_1$  as 11 is the last agent in  $F_1$  according to  $\succ^N$ . We remove agents 22, 23, 24 from  $F_2, F_3, F_4$  respectively. These agents form the set  $R$  of residual agents.

$F_1$	$F_2$	$F_3$	$F_4$
4 $\{x, y, z, w, q\}$	12 $\{a, b, c, d\}$	<b>23</b> $\{f\}$	<b>24</b> $\{e\}$
2 $\{x, y, z, w\}$	13 $\{a, b, c\}$		
6 $\{x, y, z, w\}$	14 $\{a, b, c\}$		
3 $\{x, y, z\}$	15 $\{a, b\}$		
7 $\{x, y\}$	16 $\{a, b\}$		
8 $\{x, y\}$	17 $\{a, b\}$		
1 $\{x\}$	18 $\{a\}$		
5 $\{x\}$	19 $\{a\}$		
9 $\{x\}$	20 $\{a\}$		
10 $\{x\}$	21 $\{a\}$		
<b>11</b> $\{x, y, z, w, q, e\}$	<b>22</b> $\{a\}$		

Table 5: Preferences of agents in Example 5.

The adjusted components are  $\tilde{F}_1 = F_1 \setminus \{11\}$ ,  $\tilde{F}_2 = F_2 \setminus \{22\}$ . The MDPA assigns projects to  $\tilde{F}_1$ ,  $\tilde{F}_2$ ,  $R$  in sequence.

Step 1: We assign projects to agent in  $\tilde{F}_1$ . Table 6 illustrates the demand for the available projects, the waiting set and the assignment made at every substep of Step 1. In the table, NA indicates that the project is not available as it has been assigned either definitively or provisionally.

In Step 1.0,  $q$  is the least demanded project with demand equal to 1. We provisionally assign  $q$  to 4, the (only) agent who considers this project good. The pair  $(4, q)$  is a waiting pair. In the next step, we again compute the demand for each *unassigned* project in  $F_1$ , excluding the agents to whom a project has already been assigned, either definitively or provisionally. In Step 1.1, the set of unassigned projects is  $A \setminus \{q\}$  and agent 4 is excluded while calculating the demand for unassigned projects. The project with the least demand is  $w$  with demand equal to 2. Now the MDPA assigns  $w$  to the pair  $(2, 6)$ . The set of agents and projects for the next step are updated by removing agents 2, 6 and project  $w$ . Also  $(4, q)$  is still a waiting pair. In Step 1.2, the project with least demand is  $z$  with demand 1 and there exists a waiting pair  $(4, q)$ . So the triple  $(4, 3, z)$  is formed in this step. The project  $q$  is now *unassigned* and available for future assignments.<sup>13</sup> In Step 1.3, project  $y$  is the least demanded with demand equal to 2. So the MDPA forms  $(7, 8, y)$ .<sup>14</sup> In Step 1.4, the least demanded project is  $x$  with demand equal to 4. The set of agents who consider project  $x$  good is  $\{1, 5, 9, 10\}$ . Since 1 and 5 are the agents with the highest and second highest priority

<sup>13</sup>Note that there is no waiting pair formed in Step 1.2.

<sup>14</sup>Since there is no waiting pair in Step 1.2 and the project with least demand in Step 1.3 has demand 2, there is no waiting pair in Step 1.3.

Step 1. $q$	$x$	$y$	$z$	$w$	$q$	Waiting Pair	Assignment
1.0	10	6	4	3	1	$\{(4, q)\}$	Provisional assignment
1.1	9	5	3	2	NA	$\{(4, q)\}$	$(2, 6, w)$
1.2	7	3	1	NA	NA	$\{(4, q)\}$	$(4, 3, z)$
1.3	6	2	NA	NA	0	$\emptyset$	$(7, 8, y)$
1.4	4	NA	NA	NA	0	$\emptyset$	$(1, 5, x)$
1.5	NA	NA	NA	NA	0	$\emptyset$	$(9, 10, v)$

Table 6: Step 1 in Example 5.

Step 2. $q$	$a$	$b$	$c$	$d$	Waiting Pair	Assignment
2.0	10	6	3	1	$\{(12, d)\}$	Provisional assignment
2.1	9	5	2	NA	$\{(12, d)\}$	$(13, 14, c)$
2.2	7	3	NA	NA	$\{(12, d)\}$	$(15, 16, b)$
2.3	5	NA	NA	NA	$\{(12, d)\}$	$(17, 18, a)$
2.4	NA	NA	NA	NA	$\{(12, d)\}$	$(12, 19, d), (20, 21, u)$

Table 7: Step 2 in Example 5.

according to  $\succ^N$  in this set, the MDPA forms  $(1, 5, x)$ . Finally Step 1.5 is the termination step for the component  $\tilde{F}_1$  as the demand for each available project in this step is zero. Note that there is no waiting pair and agents 9, 10 are unassigned. The pair  $(9, 10)$  is assigned the highest ranked available project according to  $\succ^A$ . So we have  $(9, 10, v)$ .

Step 2: We make assignments to agent pairs in  $\tilde{F}_2$ . Table 7 provides the demand for the available projects, the waiting set and the assignment made in every substep of Step 2. In the table, NA indicates that the project is not available as it has been assigned either definitively or provisionally.

In Step 2.0, the project with the least demand is  $d$ , with demand equal to 1. Thus project  $d$  is assigned provisionally to agent 12 and  $(12, d)$  is a waiting pair. Note that in any Step 2. $q$ ,  $q \in \{1, 2, 3\}$ , the least demanded project has demand greater than or equal to two: this project is assigned to the highest and the second highest priority agents (according to  $\succ^N$ ) in the set of agents who consider the project good. So  $(12, d)$  is a waiting pair in any Step 2. $q$ . For instance, the triple  $(15, 16, b)$  is formed by the MDPA in Step 2.2. The termination step for component  $\tilde{F}_2$  is Step 2.4:  $(12, d)$  is a waiting pair and 19, 20, 21 are unassigned. Since  $19 \succ^N 20 \succ^N 21$ , we have  $(12, 19, d)$  and  $(20, 21, u)$ .

Step 3: Partner, project assignments are made for the agents in  $R = \{11, 22, 23, 24\}$ . Let

$A(R)$  be the set of available projects to the residual agents.<sup>15</sup> We use  $\succ^N$  to form as many pairs of agents with common good projects as possible. Note that  $G^{11} \cap A(R) = \{g, e\}$ ,  $G^{22} \cap A(R) = \emptyset$ ,  $G^{23} \cap A(R) = \{f\}$  and  $G^{24} \cap A(R) = \{e\}$ . Also  $11 \succ^N 22 \succ^N 23 \succ^N 24$ . In Step 3.1,  $(11, 24, e)$  is formed by the MDPA. This is because 11 is the highest priority agent in this step and has a common good available project with agent 24. Note that  $G^{11} \cap G^j \cap A(R) = \emptyset$  for  $j \in \{22, 23\}$ . The triple  $(11, 24, e)$  is removed from the algorithm. In Step 3.2, the set of remaining agents is  $\{22, 23\}$ . The set of available projects is  $A(R) \setminus \{e\}$ . Agent 22 is the higher priority agent in  $\{22, 23\}$ . Since  $G^{22} \cap G^{23} \cap A(R) = \emptyset$ , no assignment in this step. Thus we have formed as many pairs as possible where each pair is assigned a project which is good for both agents. We now proceed to the termination substep, where agents 22 and 23 are unassigned.

Step 4: Agent 22 has no good projects among the set of available projects. Agent 23 has a good project among the set of available projects. The MDPA forms  $(22, 23, f)$ . The algorithm terminates at Step 4.

## 4 PROPERTIES OF THE MDPA

In this section we show that the MDPA satisfies several important properties.

### 4.1 ROBUST WEAK CORE

We have already defined and discussed the notion of stability. We show below the MDPA generates an assignment in the robust weak core.

**THEOREM 1** *The MDPA algorithm generates an assignment in the robust weak core at every profile. If  $|F_q|$  is even, it is in the  $F_q$ -friendship core. If  $|F_q|$  is even for all  $q \in \{1, \dots, L\}$ , then it is in the friendship core.*

The proof of Theorem 1 is provided in Appendix 2.

### 4.2 STRATEGY-PROOFNESS

In this section we investigate the strategic properties of the MDPA. We assume that friendship is commonly observable. Since friendship is mutual and transitive, no agent can *individually* manipulate and misreport her set of good partners or friends. We assume that the set of good projects for an agent is private information and can be misreported by an agent if she believes this could be advantageous. However, the assumption of homophily

---

<sup>15</sup> $A(R)$  contains the projects in  $A$  which have not been allocated to agents in  $\tilde{F}_1$  and  $\tilde{F}_2$ .

introduces some complications as it imposes a restriction on preference *profiles*. Therefore individual announcements of sets of good projects may lead to profile announcements that are inconsistent with homophily. Below, we propose a model that satisfactorily deals with this issue.

For any friendship component  $F_q$ , there is a commonly known linear order  $\succ_q^O$  over the set of all projects with the following interpretation: for any  $x, y \in A$ , if  $x \succ_q^O y$  then all agents in  $F_q$  who like  $x$  also like  $y$ .<sup>16</sup> The *threshold* project, say  $x$  is private information of an agent.<sup>17</sup> All projects  $y$  such that  $x \succ_q^O y$  are good. We believe this is a natural assumption that also ensures that any announced profile of threshold projects is consistent with homophily.

We make a minor modification to the MDPA for convenience. We assume that while making assignments in any component the order  $\succ_q^O$  is used to break ties when there are several projects with identical minimum demand. This amounts to a version of the MDPA where a different order on  $A$  is used for each component and for the set  $R$ . The results on robust weak core and the friendship core are not affected by this modification.

We now describe a general mechanism in this setting. Consider an agent  $i \in N$ . A *type* for this agent consists of her threshold project and whether she is partner or project dominant. Let  $\Gamma_i$  denote the set of types of agent  $i$ . Recall that  $\Sigma$  is the set of all feasible assignments.

With a slight abuse of notation, we will refer to the type of agent  $i$  by her preference ordering  $\succsim_i$ .<sup>18</sup>

An assignment rule is a map  $\sigma, \sigma : \times_{i \in N} \Gamma_i \rightarrow \Sigma$ .

**DEFINITION 8** *An assignment rule  $\sigma$  is strategy-proof if there does not exist  $\succsim_i, \succsim'_i \in \Gamma_i$  and  $\succsim_{-i} \in \times_{j \neq i} \Gamma_j$  such that  $\sigma(\succsim'_i, \succsim_{-i}) \succsim_i \sigma(\succsim_i, \succsim_{-i})$ .*

The notion of strategy-proofness is standard: an agent cannot strictly improve by misreporting her type for any possible announcements of types of other agents. Our main result in this section is the following.

**THEOREM 2** *The MDPA algorithm is strategy proof.*

The proof of Theorem 2 can be found in Appendix 3.

---

<sup>16</sup>It may be convenient to think of the order  $\succ_q^O$  as the counterpart of the underlying order on alternatives required in the definition of single peakedness. The order  $\succ_q^O$  is arbitrary but “common” to all agents in the same component. Like in the analysis of single peaked preferences, the mechanism designer is assumed to know the ordering  $\succ_q^O$ .

<sup>17</sup>Note that the private information of an agent consists of more than her threshold project. It is also whether she is partner or project dominant.

<sup>18</sup>Note that  $\succsim_i$  consists of exactly three elements: her set of good partners, her set of good projects and whether she is partner or project dominant. In the model, the set of good partners is fixed. A type for  $i$  can therefore be unambiguously identified with a preference ordering  $\succsim_i$ .

## 5 INDISPENSABILITY OF THE PREFERENCE RESTRICTIONS

The key assumptions underlying our preference domain are as follows: (i) separability of preferences over partners and projects, (ii) marginal component preferences are dichotomous and (iii) homophily. We show below by a series of examples that the weak core fails to exist if any of the assumptions (i), (ii) and (iii) are dropped.<sup>19</sup>

### 5.1 NON-SEPARABLE DICHOTOMOUS DOMAINS WITH HOMOPHILY

In this subsection, we retain all assumptions on preferences except separability. We assume that the sets of good partners for agents are well defined. Also friendship continues to be mutual and transitive. However the set of good projects for an agent depends on her partner. For agent  $i$ , let  $G^i(j)$  be the set of good projects for agent  $i$  when her partner is  $j$ . Note that if agent  $i$ 's preferences over partner, project tuples are separable, we have  $G^i(j) = G^i(k)$  for all  $j, k \in N \setminus \{i\}$ . In this setting, a natural interpretation of homophily is that either  $G^i(j) \subseteq G^j(i)$  or  $G^j(i) \subseteq G^i(j)$  for all agents  $i, j$  who are friends.

**EXAMPLE 6** Let  $N = \{1, 2, 3, 4, 5, 6\}$  and  $A = \{a, b, c, d\}$ . The friendship components are  $F_1 = \{1, 2, 3\}$  and  $F_2 = \{4, 5, 6\}$ . Table 8 shows the sets of good projects for the agents. If the set of good projects is indicated as  $\emptyset$ , all projects are bad for the agent.

Partner	$G^1(j)$	$G^2(j)$	$G^3(j)$	$G^4(j)$	$G^5(j)$	$G^6(j)$
1	-	$\{a\}$	$\{a, b, c, d\}$	$\emptyset$	$\emptyset$	$\emptyset$
2	$\{a, b, c, d\}$	-	$\{a\}$	$\emptyset$	$\emptyset$	$\emptyset$
3	$\{a\}$	$\{a, b, c, d\}$	-	$\emptyset$	$\emptyset$	$\emptyset$
4	$\emptyset$	$\emptyset$	$\emptyset$	-	$\{a\}$	$\{a, b, c, d\}$
5	$\emptyset$	$\emptyset$	$\emptyset$	$\{a, b, c, d\}$	-	$\{a\}$
6	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	$\{a, b, c, d\}$	-

Table 8: Preferences of agents over projects in Example 6.

Each agent  $i$ 's preference ordering  $\succsim_i$  has three indifference classes. The first class consists of tuples where  $i$  is matched to an agent  $j$  in her friendship component and works on a project in  $G^i(j)$ . The second class consists of tuples where  $i$  is matched to  $j$  in her friendship component but assigned a project that does not belong to  $G^i(j)$ . The third class consists of the remaining tuples.

Observe that preferences satisfy homophily (as defined above) and the component preferences over partners and projects are dichotomous. However, preferences are non-separable.

---

<sup>19</sup>A similar approach is followed in [Klaus and Klijn \(2005\)](#).

For instance,  $\{a, b, c, d\}$  is the set of good projects for 1 when her partner is 2, while  $\{a\}$  is the set of good projects for 1 when her partner is 3. Formally,  $(1, 3, a) \succ_i (1, 3, b)$  but  $(1, 2, a) \sim_i (1, 2, b)$ .

In each friendship component, every agent has  $\{a, b, c, d\}$  as the set of good projects with a friend and only  $\{a\}$  with the other one. Consider an arbitrary stable assignment. Since both friendship components have an odd number of agents, at least one of the agents in  $F_1$  must be matched with an agent in  $F_2$ . In fact, there must be exactly one such pair. Suppose this is not true. Then each agent in  $F_1$  is matched to an agent in  $F_2$ . Two agents from the same component can then block using the unassigned project.<sup>20</sup> Without loss of generality, let agents 3 and 6 be matched to each other. So 1 and 2 are matched together as are 4 and 5.

We first argue that the pair  $(1, 2)$  must be assigned  $a$  in any stable assignment. Suppose not i.e.  $(1, 2)$  is assigned a project in  $\{b, c, d\}$ . For agent 2,  $a$  is the only good project when she is matched to agent 1, while all projects are good for her when she is matched to agent 3. So the triple  $(2, 3, x)$  where  $x$  is the unassigned project blocks this assignment. Therefore  $(1, 2)$  is assigned  $a$  in any stable assignment and  $(4, 5)$  is assigned a project in  $\{b, c, d\}$ .

Note that  $G^4(5) = \{a, b, c, d\}$  and  $G^5(4) = \{a\}$ . So any project assigned to the pair  $(4, 5)$  in  $\{b, c, d\}$  is good for 4 and bad for 5 if they are paired together. Without loss of generality, let  $b$  be the project assigned to the pair  $(4, 5)$ . Since  $G^5(6) = \{a, b, c, d\}$ , the unassigned project belongs to  $G^5(6)$ . Thus the pair  $(5, 6)$  will block using the unassigned project.

Hence the weak core is empty.

## 5.2 SEPARABLE, NON-DICHOTOMOUS PREFERENCES WITH HOMOPHILY

In this section, we retain all assumptions on preferences except that of dichotomous marginal component preferences. Due to separability, the marginal component preferences are well-defined. We assume that the marginal preferences over partners are non-dichotomous. In particular, each agent partitions the set of partners into friends, outsiders and enemies.<sup>21</sup> Friendship continues to be mutual and transitive. In addition, the “other” categories are mutual and transitive i.e. if agent  $i$  considers  $j$  to be an outsider (resp. enemy), then  $j$  considers  $i$  as an outsider (resp. enemy) as well. Note that the set of agents can still be partitioned into friendship components. The marginal preferences over projects continue to be dichotomous. The homophily assumption is satisfied for any pair of agents who are friends with each other.

In Example 7, we demonstrate the non-existence of a weak core assignment where preferences are separable, satisfy homophily for every pair of friends and the component preferences

<sup>20</sup>There are 4 projects and 6 agents. So in any assignment, there exists an unassigned project.

<sup>21</sup>The marginal preferences over partners are trichotomous.

over partners are non-dichotomous.

**EXAMPLE 7** Let  $N = \{1, 2, \dots, 12\}$  and  $A = \{a, b, c, d, e, f, g, h, u, x, y, z\}$ . Table 9 summarizes the marginal preferences over partners ( $\succsim_i^{part}$ ) for the agents. The friendship components are  $F_1 = \{1, 2, 3\}$ ,  $F_2 = \{4, 5, 6\}$ ,  $F_3 = \{7, 8, 9\}$  and  $F_4 = \{10, 11, 12\}$ . Table 10 summarizes the sets of good projects for the agents.

Agent	Friends	Outsiders	Enemies
1	{2, 3}	{4, 5, 6, 7, 8, 9}	{10, 11, 12}
2	{1, 3}	{4, 5, 6, 7, 8, 9}	{10, 11, 12}
3	{1, 2}	{4, 5, 6, 7, 8, 9}	{10, 11, 12}
4	{5, 6}	{1, 2, 3, 7, 8, 9}	{10, 11, 12}
5	{4, 6}	{1, 2, 3, 7, 8, 9}	{10, 11, 12}
6	{4, 5}	{1, 2, 3, 7, 8, 9}	{10, 11, 12}
7	{8, 9}	{1, 2, 3, 4, 5, 6}	{10, 11, 12}
8	{7, 9}	{1, 2, 3, 4, 5, 6}	{10, 11, 12}
9	{7, 8}	{1, 2, 3, 4, 5, 6}	{10, 11, 12}
10	{11, 12}	$\emptyset$	{1, 2, 3, 4, 5, 6, 7, 8, 9}
11	{10, 12}	$\emptyset$	{1, 2, 3, 4, 5, 6, 7, 8, 9}
12	{10, 11}	$\emptyset$	{1, 2, 3, 4, 5, 6, 7, 8, 9}

Table 9: Preferences of agents over partners  $\succsim_i^{part}$  in Example 7.

$F_1$	$F_2$	$F_3$	$F_4$
1 {a, b, c}	4 {d, e, f}	7 {g, h, u}	10 {x, y, z}
2 {a, b, c}	5 {d, e, f}	8 {g, h, u}	11 {x, y, z}
3 {a, b, c}	6 {d, e, f}	9 {g, h, u}	12 {x, y, z}

Table 10: Preferences of agents in Example 7.

Each agent  $i$ 's preference ordering  $\succsim_i$  is lexicographic and  $\succsim_i^{part}$  is the dominant component.<sup>22</sup>

In any assignment  $\sigma$ , the set of unassigned projects  $u^\sigma$  is non empty. We will first argue that any assignment in the weak core has exactly one surplus agent from each friendship

<sup>22</sup>Note that  $\succsim_i$  has six indifference classes. The first class consists of tuples where  $i$  is matched with a friend and assigned a good project. The second class consists of tuples where  $i$  is matched with a friend but assigned a bad project. The third class consists of tuples where  $i$  is matched with an outsider and assigned a good project and so on.



component i.e. only one agent from a friendship component is matched to an agent who is not her friend. To see this, we assume for contradiction that at least two agents (say agents 1 and 2) in  $F_1$  are paired with agents who do not belong to  $F_1$  in the assignment.<sup>23</sup> Then agents 1, 2 can block the assignment with an unassigned project. They both strictly improve as they now move to either the first or the second indifference class.

Thus there is only one agent in every friendship component who is matched to an agent who is not her friend.

All agents in any friendship components have the same set of good projects. As a consequence, the two agents in a friendship component who are matched together must either receive a good project or all good projects for these agents are assigned. If this is not true, then these two agents will block with the unassigned good project for them.

In order for the claim in the previous paragraph to hold for all four friendship components, it must be the case that the pair formed in each component is assigned a project that is good for both agents. Thus in any assignment and for every component, there exists an unassigned project which is good for all three agents in the component. Hence there is a good project that is unassigned for the surplus agent in every component.

In view of the underlying symmetry in the example, we can assume without loss of generality that agents 3, 6, 9, 12 are the surplus agents in components  $F_1$ ,  $F_2$ ,  $F_3$  and  $F_4$  respectively. The agents 3, 6 and 9 regard 12 as an enemy. In addition, each of the agents in  $\{3, 6, 9\}$  view the other two agents as outsiders. One of the agents in  $\{3, 6, 9\}$  must be matched with 12. We can assume without loss of generality that this agent is 3. Therefore agents 6 and 9 are matched together. Moreover, one of these agents must receive a bad project since they have no common good projects. Suppose 6 is the agent who is not receiving a good project. By our earlier argument, there exists a good project for agent 6 that is unassigned. Now agents 3 and 6 together with this project will block the assignment.

Hence the weak core is empty.

### 5.3 SEPARABLE DICHOTOMOUS DOMAINS WITHOUT HOMOPHILY

We retain all the assumptions on preferences in Section 2.3 except homophily. We show the non-existence of weak core assignments for a partner dominant preference profile (Example 8) and for a project dominant preference profile (Example 9).

**EXAMPLE 8** Let  $N = \{1, 2, 3, 4\}$  and  $A = \{a, b, c, d, e, f\}$ . The friendship components are  $F_1 = \{1, 2, 3\}$  and  $F_2 = \{4\}$ . The sets of good projects for agents are specified in Table 11. Note that the preferences of agents in  $F_1$  violate homophily since the set inclusion relation does not hold for any pair of agents in  $F_1$ . Consider the preference profile such that all agent preferences are partner dominant.

---

<sup>23</sup>Note that agents 1, 2 are not in the first and second indifference classes in the assignment.

$F_1$	$F_2$
1 $\{a, b\}$	4 $\{g\}$
2 $\{c, d\}$	
3 $\{e, f\}$	

Table 11: Preferences of agents in Example 8.

$F_1$	$F_2$
1 $\{a, b\}$	4 $\{g, h\}$
2 $\{c, d\}$	5 $\{x, y\}$
3 $\{e, f\}$	6 $\{z, u\}$

Table 12: Preferences of agents in Example 9.

In any assignment, one of the agents in  $F_1$  must be matched with agent 4, while the remaining agents are matched with each other. Assume without loss of generality that (1, 2) and (3, 4) are matched with each other. Since agent 3's preferences are partner dominant, agent 3 would prefer to be matched with a good partner. Since the set of projects of agents 1 and 2 are disjoint, at least one of them must be receiving a bad project. Moreover this agent must also have a good project which is unassigned. Agent 3 can then pair with this agent and this unassigned project to block the assignment. Hence the weak core is empty.

**EXAMPLE 9** Let  $N = \{1, 2, 3, 4, 5, 6\}$  and  $A = \{a, b, c, d, e, f, g, h, x, y, z, u\}$ . The friendship components are  $F_1 = \{1, 2, 3\}$  and  $F_2 = \{4, 5, 6\}$ . The sets of good projects for agents are specified in Table 12. Once again the preferences of agents in  $F_1$  and  $F_2$  violate homophily. Consider the preference profile such that all agent preferences are project dominant.

Since there are an odd number of agents in each component, at least one agent in  $F_1$  is matched to an agent in  $F_2$  in any assignment. Assume without loss of generality that agents 3 and 6 are paired together. Since the good projects of agents 3 and 6 are disjoint, at least one of them is not getting a good project. Suppose it is agent 3.

In order for agent 3 not to form a blocking coalition with either agents 1 or 2 via an unassigned project either (i) or (ii) below must hold: (i) all good projects of 1 and 2 are assigned or (ii) agents 1 and 2 are both getting good projects. However there are four projects that are either good for agent 1 or good for agent 2, while the total number of assigned projects is three. Therefore (i) cannot hold.

Suppose (ii) holds. Then agents 1 and 2 must be paired with agents 4 and 5 in  $F_2$ . Moreover both agents 1 and 2 must be getting a good project. Therefore agents 4 and 5 are not getting good projects i.e. they are getting bad partners and bad projects. They will

then block by pairing with each other with an unassigned project. Therefore (ii) cannot hold either.

Hence agent 3 will block with either agent 1 or 2 and hence the weak core is empty.

## 6 CONCLUSION

In this paper, we have investigated a class of matching models in which agents are matched in pairs and each pair is assigned a project. We provide a preference domain restriction that guarantees the existence of the robust weak core. The key assumptions underlying our preference domain are separability, dichotomous marginal component preferences and homophily. We present an algorithm, the MDPA, which generates an assignment in the robust weak core at every preference profile. In addition it is strategy-proof. We also show that if any one of the assumptions underlying our preference domain is dropped while maintaining the others, the weak core fails to exist.

Several questions remain open in our model. The examples used to demonstrate the non-existence of the strong core rely on there being at least one odd component. We conjecture that the strong core is, in fact non-empty. The strong core is also likely to be non-empty if the number of projects is exactly equal to half the number of agents. Our current results on pairs also do not extend in a straightforward manner to teams of general size. We hope to address these issues in future work.

## 7 APPENDIX 1

In this section, we formally describe the MDPA algorithm.

Let  $\succ^N$  and  $\succ^A$  be linear orderings<sup>24</sup> of the sets  $N$  and  $A$  respectively. If agent  $i \succ^N j$ , then agent  $i$  has priority over  $j$ .

Fix an arbitrary preference profile  $\succsim$ . This profile induces friendship components  $\{F_1, \dots, F_L\}$  which are ordered as  $F_1, F_2, \dots, F_L$  (according to the linear ordering  $\succ^F$ ).

Step 0: In each component  $F_q$  where  $|F_q|$  is odd, remove the agent with the lowest priority in  $F_q$  and add this agent to the *Residual set*  $R$ . If  $|F_q|$  is even, then no changes are made and the original component is retained. The adjusted friendship components are  $\{\tilde{F}_1, \dots, \tilde{F}_L\}$ . Each adjusted component has even cardinality.

We shall make assignments in the components  $\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_L$  in sequence. These will be labelled Steps 1 to  $L$  respectively. Each step comprises of an initial step, several intermediate steps and a termination step. For component  $\tilde{F}_q$ , the initial step will be denoted by Step  $q.0$ ,

---

<sup>24</sup>A linear ordering is a reflexive, complete and antisymmetric binary relation

intermediate steps by Step  $q.1, q.2, \dots$  and the termination step by Step  $q.T$ . At the start of the generic step  $q.s$  where  $q \in \{1, \dots, L\}$  and  $s \in \{0, \dots, T\}$ , the algorithm is provided three inputs: (i) the set of available projects denoted by  $A(q.s)$ , (ii) the set of unassigned agents  $N(q.s)$  in component  $\tilde{F}_q$  and (iii) the set of waiting agent, project pairs  $W(q.s)$ . For every step  $q.s$ ,  $|W(q.s)|$  is either 0 or 1.

Step 1.0: Here  $A(1.0) = A$ ,  $N(1.0) = \tilde{F}_1$  and  $W(1.1) = \emptyset$ . The demand for every available project  $a \in A(1.0)$  is given by  $d(a; (1.0)) = \#\{i \in N(1.0) : a \in G^i(\succ_i)\}$ . The set of agents who demand project  $a$  is given by  $S(a; (1.0)) = \{i \in N(1.0) : a \in G^i(\succ_i)\}$ .

Remove all projects with zero demand. Consider the project (projects) with the least demand. In case there is more than one such project, pick the project which is ranked highest according to  $\succ^A$ . Denote the project with the lowest demand (after tie breaking) by  $a$ . There are two cases to consider.

1.  $d(a; (1.0)) = 1$ . Then no assignments are made at Step 1.0 and  $W(1.1) = \{(i, a)\}$  where  $S(a; (1.0)) = \{i\}$ . Also sets  $A(1.1) = A(1.0) \setminus \{a\}$ ,  $N(1.1) = N(1.0) \setminus \{i\}$  and proceed to Step 1.1.
2.  $d(a; (1.0)) \geq 2$ . Assign  $a$  to the pair of agents with the highest and second highest priority in  $S(a; (1.0))$ . Now sets  $A(1.1) = A(1.0) \setminus \{a\}$ ,  $N(1.1) = N(1.0) \setminus \{i, j\}$  where  $i, j$  are the agents who have just been assigned  $a$  in this step and  $W(1.1) = \emptyset$ . Proceed to Step 1.1.

Step 1.1: Step 1.1 repeats Step 1.0 with the sets  $A(1.1)$ ,  $N(1.1)$  and  $W(1.1)$  (determined at the end of Step 1.0) but with an important difference.

As in Step 1.0, consider the project with the least non zero demand with appropriate tie breaking. Suppose this project is  $b$ . Once again there are three possibilities.

1.  $d(b; (1.1)) = 1$  and  $W(1.1) \neq \emptyset$ . Let  $(i, a) \in W(1.1)$  and  $S(b; (1.1)) = \{j\}$ . Then the pair  $(i, j)$  is assigned  $b$ . Also  $A(1.2) = [A(1.1) \cup \{a\}] \setminus \{b\}$ ,  $N(1.2) = N(1.1) \setminus \{j\}$ ,  $W(1.2) = \emptyset$  and proceed to Step 1.2.
2.  $d(b; (1.1)) = 1$  and  $W(1.1) = \emptyset$ . Then no assignments are made at Step 1.1 and  $W(1.2) = \{(j, b)\}$  where  $S(b; (1.0)) = \{j\}$ . Also  $A(1.2) = A(1.1) \setminus \{b\}$ ,  $N(1.2) = N(1.1) \setminus \{j\}$  and proceed to Step 1.2.
3.  $d(b; (1.1)) \geq 2$ . Assign  $b$  to the pair of agents with the highest and second highest priority in  $S(b; (1.1))$ . Now  $A(1.2) = A(1.1) \setminus \{b\}$ ,  $N(1.2) = N(1.1) \setminus \{i, j\}$  where  $i, j$  are the agents who have just been assigned  $a$  in this step and  $W(1.2) = W(1.1)$ . Proceed to Step 1.2.

By construction at any stage,  $W(1.s)$  is either null or consists of one element.

Step 1.2 repeats Step 1.1 with the appropriate sets. Proceeding in this way, there exists a step  $1.T$  where  $d(a; (1.T)) = 0$  for every  $a \in A(1.T)$ . This is called the termination step. Here three cases can arise.

- I.  $N(1.T) = \emptyset$  and  $W(1.T) = \emptyset$ . This means that all agents in  $\tilde{F}_1$  have been assigned a partner and a project.
- II.  $|N(1.T)|$  is even. Note that  $W(1.T) = \emptyset$ . Arrange the agents in  $N(1.T)$  using  $\succ^N$ . Form pairs of consecutive agents proceeding in sequence. Assign projects to pairs in sequence using  $\succ^A$  from  $A(1.T)$ .
- III.  $|N(1.T)|$  is odd. Then  $W(1.T) \neq \emptyset$ . Let  $W(1.T) = \{(i, a)\}$ . Arrange the agents in  $N(1.T)$  using  $\succ^N$ . Suppose  $j$  is the highest priority agent in this set. Form the triple  $(i, j, a)$ . Observe that  $|N(1.T) \setminus \{j\}|$  is even. For the remaining agents, partners and projects are assigned as in (II) above.

This completes the assignment for all agents in the component  $\tilde{F}_1$ . Let the set of projects assigned to pairs in  $\tilde{F}_1$  be  $\Delta(\tilde{F}_1)$ . We repeat the procedure for  $\tilde{F}_2$  with  $A(2.0) = A \setminus \Delta(\tilde{F}_1)$ ,  $N(2.0) = \tilde{F}_2$  and  $W(2.0) = \emptyset$ . This completes the assignment for the agents in component  $\tilde{F}_2$ . Proceeding in this manner, at the end of Step  $L$  we have assigned partners and projects to all agents in  $\tilde{F}_1 \cup \tilde{F}_2 \dots \cup \tilde{F}_L$ .

Step  $L + 1$ : In this step, projects are assigned to agents in the set  $R$ . The set of available projects is  $A(R) = A(\tilde{F}_L) \setminus \Delta(\tilde{F}_L)$ . Without loss of generality (and by suitable relabelling of agents according to  $\succ^N$ ), let  $R = \{1, 2, \dots, 2r\}$  for  $r \geq 0$ . We make assignments for the agents in  $R$  in Steps  $L + 1.1$  through  $L + 1.r^*$ . At each of these steps  $L + 1.k$ , a set of agents  $R_k$  is obtained. The procedure terminates in Step  $L + 1.r^*$  where  $\cup_{k=1}^{r^*} R_k = R$ .

Step  $(L + 1.1)$ : In this step, we match agent 1 with the agent with the lowest index  $t > 1$  in  $R$  such that  $G^1(\succ_1) \cap G^t(\succ_t) \cap A(R)$  is non empty. This pair is assigned a project  $a$  in  $G^1(\succ_1) \cap G^t(\succ_t) \cap A(R)$  (ties are once again broken according to  $\succ^A$ ). In this case,  $R_1 = \{i, t\}$ . If no such agent  $t$  exists, then agent 1 is unmatched at this step and  $R_1 = \{1\}$ .

The set of projects assigned in this step is denoted by  $\Delta_R(1)$  where  $\Delta_R(1) = \emptyset$  if  $R_1 = \{1\}$  and  $\{a\}$  if  $R_1 = \{1, t\}$  and  $a$  is assigned to  $(1, t)$ .

The set of agents remaining for the next step is  $R \setminus R_1$ . The set of projects available for the next step is  $A(R) \setminus \Delta_R(1)$ .

Step  $(L + 1.2)$ : We repeat Step  $(L + 1.1)$  with agent 1 being replaced by the agent with the smallest index in  $R \setminus R_1$  and the set of projects  $A(R)$  replaced by  $A(R) \setminus \Delta_R(1)$ . This

generates possibly another assignment. At the end of this step, we obtain  $R_2$  and the set of available projects  $A(R) \setminus \cup_{k=1}^2 \Delta_R(k)$ .

Proceeding in this manner, there will exist a Step  $(L + 1.r^*)$  where  $\cup_{j=1}^{r^*} R_j = R$ . The set of projects available for assignment in Step  $(L + 1.r^*)$  is  $A(R) \setminus \cup_{k=1}^{r^*} \Delta_R(k)$ .

There are two possibilities.

1. All agents in  $R$  have been assigned a partner and a project. In this case the algorithm terminates.
2. Suppose 1 does not hold. Let  $\bar{R}$  be the set of agents who have not been assigned a partner and a project. Note that  $\bar{R}$  must be even in cardinality. Proceed to Step  $L + 2$ .

Step  $L + 2$ : In this step,  $A(R) \setminus \cup_{k=1}^{r^*} \Delta_R(k)$  is the set of available projects. Partition  $\bar{R}$  into  $\{\bar{R}^1, \bar{R}^2\}$  where  $\bar{R}^1$  consists of agents who have no good projects among the set of available projects and  $\bar{R}^2$  are the remaining agents.

Order the agents in  $\bar{R}^1$  and  $\bar{R}^2$  according to  $\succ^N$ . Assign the highest priority agent in  $\bar{R}^1$  with the highest priority agent in  $\bar{R}^2$  and an available good project of the agent in  $\bar{R}^2$ , the agent with the second highest priority in  $\bar{R}^1$  with the agent with the second highest priority in  $\bar{R}^2$  and so on.

1. If  $|\bar{R}^1| = |\bar{R}^2|$ , then the procedure will terminate with all agents in  $R$  being assigned a project and partner.
2. If  $|\bar{R}^1| > |\bar{R}^2|$ , then an even number of agents in  $\bar{R}^1$  will be left unassigned. They are now paired consecutively and are assigned a project from the available set according to  $\succ^A$ .
3. If  $|\bar{R}^1| < |\bar{R}^2|$ , then an even number of agents in  $\bar{R}^2$  are left unassigned. They are paired consecutively and assigned a good project of the higher priority agent (ties are broken according to  $\succ^A$ ).

This completes the description of the algorithm.

We make several salient observations about the MDPA which we shall use extensively in the proofs.

**OBSERVATION 1** Let  $k, l \in \tilde{F}_q$ . Suppose they are paired together and assigned project  $a$  in the MDPA. If  $a$  is not a good project for at least one agent in  $\{k, l\}$ , then  $k, l$  are present in the termination step  $q.T$  for component  $\tilde{F}_q$ . In Step  $q.T$ , the demand for every available project is zero.

**OBSERVATION 2** Let  $k \in \tilde{F}_q$ . If  $k$  belongs to a waiting pair in some Step  $q.r$  (where  $1 \leq r < T$ ) of the MDPA, then  $k$  always receives a good project in the algorithm.

**OBSERVATION 3** An agent is paired with someone who is not a friend only if she is a residual agent. There can be at most one residual agent in every friendship component.

**OBSERVATION 4** A residual agent who is assigned a project in Step  $L + 1$  always receives a good project. A residual agent who is assigned a bad project in the MDPA must have been assigned in Step  $L + 2$ .

## 8 APPENDIX 2

We provide the proof of Theorem 1.

*Proof of Theorem 1:* Let  $\succsim$  be an arbitrary profile and  $\sigma$  be the assignment generated by the MDPA algorithm at  $\succsim$ . Note that  $\sigma$  is also generated by MDPA at any  $\succsim'$  which is good-set equivalent to  $\succsim$ . We will show that  $\sigma$  cannot be strongly blocked by any coalition  $S$  at  $\succsim'$  which is good-set equivalent to  $\succsim$ . This will establish that  $\sigma$  belongs to the robust weak core at  $\succsim$ .

Assume by way of contradiction that  $S$  blocks  $\sigma$  at  $\succsim'$  which is good-set equivalent to  $\succsim$ . We consider the following exhaustive possibilities: (A)  $|S| = 2$  and (B)  $|S| > 2$ .

Case A: Let  $S = \{k, l\}$ . Suppose  $(k, i, a), (l, j, c) \in \sigma$ . Since  $S$  is a blocking coalition at  $\succsim'$ , there exists  $b \in X(S, \sigma)$  such that  $(k, l, b) \succ'_k (k, i, a)$  and  $(k, l, b) \succ'_l (l, j, c)$ . It follows from the definition of  $X(S, \sigma)$  that  $b$  is unassigned in  $\sigma$  i.e.  $b \in u^\sigma$ .<sup>25</sup>

There are two possibilities to consider: (A.1)  $k, l \in F_q$  for some  $q$  and (A.2)  $k, l \notin F_q$  for any  $q$ .

Case A.1. There are two subcases to consider.

Case A.1.1. Agents  $k, l$  are paired together in  $\sigma$  i.e.  $(k, l, a) \in \sigma$ . Since  $k$  and  $l$  strictly improve, we have  $a \notin G^k(\succsim'_k)$  and  $a \notin G^l(\succsim'_l)$ . Moreover  $b \in G^k(\succsim'_k) \cap G^l(\succsim'_l)$ . By Observation 1, we know that agents  $k, l$  are present in Step  $q.T$  of the algorithm (the termination step for the component  $\tilde{F}_q$ ) and the demand for each available project in this step is zero. Since  $b \in u^\sigma$ ,  $b$  is available in Step  $q.T$  and  $d(b; (q.T)) = 2$  in this step. We have a contradiction.

Case A.1.2. Agents  $k, l$  are not paired together in  $\sigma$  i.e.  $(k, i, a), (l, j, c) \in \sigma$  where  $i \neq l$  and  $j \neq k$ . In the algorithm, there is at most one residual agent from each friendship component. This implies that at least one of the agents among  $k$  and  $l$  is paired with a friend. Assume without loss of generality  $k, i \in F_q$ . Since  $k$  strictly improves, we have  $a \notin G^k(\succsim'_k)$  and  $b \in G^k(\succsim'_k)$ . By Observation 1, we know agent  $k$  is present in Step  $q.T$  of the algorithm and

---

<sup>25</sup>The only case where  $|S| = 2$  and  $u^\sigma \subset X(S, \sigma)$  is when agents  $k, l$  are matched together in  $\sigma$ . Let  $(k, l, a) \in \sigma$  and  $a \in X(S, \sigma)$ . Since agents  $k, l$  strictly improve by blocking, we know  $a$  is a bad project for them. Here  $X(S, \sigma) \setminus \{a\} = u^\sigma$ . So there exists  $b \in u^\sigma$  such that the allocation  $(k, l, b)$  is a strict improvement for agents  $k, l$ .

the demand for each available project in this step is zero. Since  $b \in u^\sigma$ ,  $b$  is available in Step  $q.T$  and  $d(b; (q.T)) \geq 1$  in this step. We have a contradiction.

Case A.2. Suppose  $k \in F_q$  and  $l \in F_p$ . There are two cases.

Case A.2.1: Agents  $k, l$  are paired together in  $\sigma$  i.e.  $(k, l, a) \in \sigma$ . Since agents  $k$  and  $l$  belong to different components, we have  $k, l \in R$ . Also we have  $a \notin G^k(\succ'_k)$ ,  $a \notin G^l(\succ'_l)$  and  $b \in G^k(\succ'_k) \cap G^l(\succ'_l)$ . By Observation 4, we can conclude that  $(k, l, a)$  is being formed in Step  $L + 2$ . Suppose  $k \succ^N l$ . Since  $b$  is unassigned, there will exist a substep of Step  $L + 1$ , say  $L + 1.r$  (with  $r \leq r^*$ ) where  $k$  and  $l$  could have been paired together with project  $b$  which is good for both agents. Therefore we have a contradiction.

Case A.2.2: Agents  $k, l$  are not paired together in  $\sigma$ . Let the partners of agents  $k$  and  $l$  in  $\sigma$  be  $i$  and  $j$  respectively.

There are three possibilities to consider.

(a)  $\succ'_k$  and  $\succ'_l$  are both partner dominant. Since  $k$  and  $l$  are strictly improving, we must have  $i \notin F_q$  and  $j \notin F_p$ . Thus  $k, i, l, j$  all belong to the residual set  $R$ . Also we have  $a \notin G^k(\succ'_k)$ ,  $b \in G^k(\succ'_k)$ ,  $c \notin G^l(\succ'_l)$  and  $b \in G^l(\succ'_l)$ . Since  $k$  and  $l$  are being assigned bad projects in  $\sigma$ , we can conclude that  $(k, i, a)$  and  $(l, j, c)$  are being formed in Step  $L + 2$  (Observation 4). Suppose agent  $k \succ^N l$ . Since  $b$  is unassigned, there will exist a substep of Step  $L + 1$ , say  $L + 1.r$  (with  $r \leq r^*$ ) where  $k$  and  $l$  could have been paired together with project  $b$  which is good for both agents. Therefore we have a contradiction.

(b)  $\succ'_k$  and  $\succ'_l$  are both project dominant.

Suppose one of the pairs  $(k, i)$ ,  $(l, j)$  belongs to the same friendship component. Without loss of generality let  $k, i \in F_q$ . Since agent  $k$  strictly improves, we have  $a \notin G^k(\succ'_k)$  and  $b \in G^k(\succ'_k)$ . We then have a contradiction exactly as in Case A.1.2.

Finally, suppose neither  $k, i$  nor  $l, j$  belong to the same component. Thus  $k, i, l, j$  belong to the residual set. This reduces to Case A.2.2 (a) which we have already dealt with.

(c) One of  $\succ'_k$  and  $\succ'_l$  is partner dominant while the other is project dominant. Suppose  $\succ'_k$  is partner dominant. Since agent  $k$  strictly improves, we conclude that  $i$  and  $k$  cannot belong to the same friendship component. Also  $a \notin G^k(\succ'_k)$  and  $b \in G^k(\succ'_k)$ .

There are two further possibilities. Suppose  $l, j$  do not belong to the same component. Then agents  $k, i, l, j$  are residual agents. Once again, we are back to Case A.2.2 (a). The remaining case is  $l, j \in F_p$  for some  $p$ . Since  $l$  strictly improves, we have  $c \notin G^l(\succ'_l)$  and  $b \in G^l(\succ'_l)$ . Note that this case is equivalent to Case A.1.2, where agent  $k$  is replaced by  $l$  and  $i$  is replaced by  $j$ .

Therefore  $\sigma$  cannot be blocked by  $S$ .

Case B: We consider two subcases: (B.1)  $S \subseteq F_q$  for some  $q$  and (B.2)  $S \not\subseteq F_q$  for any  $q$ .



Case B.1. Since  $S$  is even,  $|S| \geq 4$  and there is at most one residual agent from any friendship component, there is at least one agent  $i \in S$  and a project  $a \in A$  such that  $(i, j, a) \in \sigma$  and  $i, j \in F_q$ . Let agent  $i$  be paired with agent  $k \in S$  and project  $x \in X(S, \sigma)$  in the blocking assignment. Clearly  $a \notin G^i(\succ'_i)$  and  $x \in G^i(\succ'_i)$ .

Suppose  $x \in u^\sigma$ . By Observation 1, agent  $i$  is present in Step  $q.T$  of the algorithm (the termination step for  $\tilde{F}_q$ ) where  $x$  is available i.e.  $d(x; q.T) \geq 1$ . However this is a contradiction to the specification of the termination step.

The remaining possibility is  $x \in X(S, \sigma) \setminus u^\sigma$ . By the definition of  $X(S, \sigma)$ , there exist agent  $l_1, l_2 \in S$  such that  $(l_1, l_2, x) \in \sigma$ . Moreover  $l_1, l_2$  strictly improve by blocking, it must be the case  $x$  is not a good project for either  $l_1$  or  $l_2$ . Therefore  $x$  is available in the termination Step  $q.T$ . Since agent  $i$  is present in this step and  $x \in G^i(\succ'_i)$ , it must be the case that  $d(x; q.T) \geq 1$ . Once again, we have a contradiction to the specification of the termination step.

Case B.2. Suppose there exists an agent in  $S$  who does not belong to the residual set  $R$ . Hence we can find a component  $F_q$  such that  $F_q$  is the highest ranked component according to the ordering  $\succ^F$  which contains an agent  $i \in S$ . This agent  $i$  is not a residual agent and was matched with  $j \in F_q$  in  $\sigma$ . Since  $i$  strictly improves by blocking, it must be the case that  $i$  is assigned a bad project in  $\sigma$ . Moreover  $i$  must be assigned a good project, say  $x$  in the blocking assignment  $\sigma'$ . There are two possibilities regarding  $x$ : either  $x$  is unassigned in  $\sigma$  or  $x$  was assigned to some pair  $(l_1, l_2)$  in  $\sigma$  where  $l_1, l_2 \in S$ . We will argue that  $x$  was available for allocation in the termination step  $q.T$ . If  $x$  is unassigned in  $\sigma$ , then it is certainly available in the termination step  $q.T$ . Consider the second possibility in more detail. It must be the case that both  $l_1$  and  $l_2$  belong to the same component or belong to  $R$ . If they belong to a component different from  $F_q$  or to  $R$ , then  $x$  is available in the termination step  $q.T$ . The only remaining case is when  $l_1, l_2 \in F_q$ . Since agents  $l_1$  and  $l_2$  strictly improve by blocking,  $x$  is a bad project for them. Thus the allocation of  $x$  to the pair  $(l_1, l_2)$  was made in the termination step  $q.T$  and  $x$  was available in this step. We have thus shown in all cases that  $x$  is available in the termination step  $q.T$ . We know that  $i$  is present in Step  $q.T$  i.e.  $d(x; q.T) \geq 1$  (Observation 1). This contradicts the specification of the termination step.

The second possibility is the one where all agents in  $S$  belong to the residual set  $R$ . Consider an arbitrary agent  $i \in S$ . Let  $(i, j, a) \in \sigma$  and  $(i, k, b) \in \sigma'$  where  $\sigma'$  is the blocking assignment. Since  $i$  strictly improves by blocking,  $a$  is a bad project for  $i$  while  $b$  is a good one. Let  $(k, l, c) \in \sigma$  be the MDPA assignment for agent  $k$ , where  $k$  is agent  $i$ 's partner in the blocking assignment  $\sigma'$ . Since  $k$  strictly improves by blocking,  $c$  is a bad project for  $k$  while  $b$  is a good one. So agents  $i, k$  are assigned bad projects in  $\sigma$ . By Observation 4, we can conclude that  $(i, j, a)$  and  $(k, l, c)$  are formed in Step  $L + 2$ . Suppose  $k \succ^N l$ .

Suppose  $b \in u^\sigma$ . There exists a substep of Step  $L + 1$ , say  $L + 1.r$  (with  $r \leq r^*$ ) where

$k$  and  $l$  could have been paired together with project  $b$  which is good for both agents. Thus we have a contradiction.

The remaining possibility is  $b \in X(S) \setminus u^\sigma$ . Then there exists  $m, n \in S$  such that  $(m, n, b) \in \sigma$ . Since  $m, n$  strictly improve by blocking,  $b$  is a bad project for them. By Observation 4, we know that  $(m, n, b)$  is formed in Step  $L + 2$ . So  $b$  is available in every substep of Step  $L + 1$ . Thus there exists a substep of Step  $L + 1$ , say  $L + 1.r$  (with  $r \leq r^*$ ) where  $k$  and  $l$  could have been paired together with project  $b$  which is good for both agents. This is a contradiction.

Thus the MDPA generates an assignment in the robust weak core at every preference profile.

We now show that the MDPA generates an assignment in the  $F_q$ -friendship core whenever  $|F_q|$  is even. We begin with an important preliminary lemma regarding the structure of the MDPA assignment in any component.

Consider a component  $F_q$  with an even number of agents and a preference profile  $\succsim$ . Let  $A'$  be the set of projects available for assignment to  $F_q$  in the MDPA. For convenience, we will denote  $F_q$  by  $J$ . Also with a minor abuse of notation, we will write  $G^i(\succsim_i) \cap A'$  simply as  $G^i(\succsim_i)$ . Construct pairs of sets  $(J_1, A_1), \dots, (J_T, A_T)$  as follows:

$$(1.a) \quad A_1 = \{a \in A' : a \in G^i(\succsim_i) \text{ for all } i \in J\}.$$

$$(1.b) \quad J_1 = \{i \in J : G^i(\succsim_i) = A_1\}.$$

$$(2.a) \quad A_2 = \{a \in A' \setminus A_1 : a \in G^i(\succsim_i) \text{ for all } i \in J \setminus J_1\}.$$

$$(2.b) \quad J_2 = \{i \in J \setminus J_1 : G^i(\succsim_i) = A_1 \cup A_2\}.$$

$\vdots$   
 $\vdots$

$$(t.a) \quad A_t = \{i \in A \setminus (A_1 \cup \dots \cup A_{t-1}) : a \in G^i(\succsim_i) \text{ for all } i \in J \setminus (J_1 \dots \cup J_{t-1})\}.$$

$$(t.b) \quad J_t = \{i \in J \setminus (J_1 \cup \dots \cup J_{t-1}) : G^i(\succsim_i) = A_1 \cup \dots \cup A_{t-1}\}.$$

$\vdots$

The set  $A_1$  consists of the available projects that are good for all agents in  $J$ . The set  $J_1$  consists of those agents whose set of good projects is exactly the set  $A_1$ . Either  $J_1 = J$  or there exist some agents who have good projects not belonging to  $A_1$ . Let  $A_2$  be the set of projects which are good for all agents in  $J \setminus J_1$ . Similarly  $J_2$  are the agents whose set of good projects is exactly the set  $A_1 \cup A_2$ . Note that  $A_1 \cap A_2 = \emptyset$  and  $J_1 \cap J_2 = \emptyset$ . Proceeding in this manner, we obtain sets  $(A_1, J_1), \dots, (A_T, J_T)$  where  $A_1, \dots, A_T$  and  $J_1, \dots, J_T$  are partitions

of  $A'$  and  $J$  respectively. Notice that this procedure is well defined due to the assumption of homophily. The set  $J_T$  is the set of least fussy agents i.e. agents with the largest sets of good projects (amongst available projects) while the set  $A_T$  consists of projects that are good only for the agents in  $J_T$ . By homophily, agents in  $J_T$  like all the projects in  $\cup_{t=1}^T A_t$ . In general, for any  $i \in J_t$ ,  $G^i(\succ_i) = \cup_{r=1}^t A_r$ . Also the projects in  $\cup_{r=t+1}^T A_r$  are not good projects for any agent  $i \in J_t$ .

Denote  $|J_t|$  by  $\alpha_t$  and  $2|A_t|$  by  $\beta_t$  for all  $t \in \{1, \dots, T\}$ . Since  $F_q$  has an even number of agents, the MDPA matches all agents in  $F_q$  with a friend. We refer to agents who are assigned good projects in  $F_q$  as *happy* agents. Likewise agents in  $F_q$  who receive bad projects are referred to as *unhappy* agents.

**LEMMA 1** *Let  $t^* \in \{1, \dots, T\}$  be the smallest integer such that*

$$\sum_{r=t^*}^t \alpha_r \leq \sum_{r=t^*}^t \beta_r \text{ for all } t = t^*, t^* + 1, \dots, T. \quad (1)$$

*In case no such  $t^*$  exists, let  $t^* = T + 1$ . Then*

$$\sum_{r=t}^{t^*-1} \alpha_r > \sum_{r=t}^{t^*-1} \beta_r \text{ for all } t = t^* - 1, \dots, 1. \quad (2)$$

*The number of unhappy agents in the MDPA is  $\gamma = \sum_{r=1}^{t^*-1} (\alpha_r - \beta_r)$ .*

*Proof:* Suppose  $t^*$  is the smallest integer satisfying 1. We will show that 2 is satisfied by induction on  $t \in \{t^* - 1, \dots, 1\}$ . We first show that it holds for  $t^* - 1$ . In the second step we show that if it holds for all  $t' \in \{t^* - 2, \dots, 2\}$ , it also holds for  $t' - 1$ .

Suppose  $\alpha_{t^*-1} \leq \beta_{t^*-1}$ . Adding  $\alpha_{t^*-1}$  and  $\beta_{t^*-1}$  to the LHS and RHS of Equations 1, we obtain

$$\sum_{r=t^*-1}^t \alpha_r \leq \sum_{r=t^*-1}^t \beta_r \text{ for all } t = \{t^* - 1, \dots, T\}. \quad (3)$$

Thus  $t^* - 1$  satisfies 1 contradicting our assumption that  $t^*$  is the smallest integer satisfying 1.

Fix  $t' \in \{t^* - 1, \dots, 2\}$  and suppose

$$\sum_{r=t'}^{t^*-1} \alpha_r > \sum_{r=t'}^{t^*-1} \beta_r \text{ for all } t = \{t^* - 1, \dots, t'\}. \quad (4)$$

but

$$\sum_{r=t'-1}^{t^*-1} \alpha_r \leq \sum_{r=t'-1}^{t^*-1} \beta_r. \quad (5)$$

Pick an arbitrary  $s \in \{t' - 1, \dots, t^* - 1\}$ . Using 4 we have

$$\sum_{r=s}^{t^*-1} \alpha_r > \sum_{r=s}^{t^*-1} \beta_r. \quad (6)$$

We can also rewrite 5 as

$$\sum_{r=t'-1}^{t^*-1} \alpha_r = \sum_{r=t'-1}^s \alpha_r + \sum_{r=s}^{t^*-1} \alpha_r \leq \sum_{r=t'-1}^s \beta_r + \sum_{r=s}^{t^*-1} \beta_r + \sum_{r=t'-1}^{t^*-1} \beta_r. \quad (7)$$

Subtracting 6 from 7, we obtain

$$\sum_{r=t'-1}^s \alpha_r \leq \sum_{r=t'-1}^s \beta_r. \quad (8)$$

Adding 8 for the case  $s = t^* - 1$  to 1, we have

$$\sum_{r=t'-1}^t \alpha_r \leq \sum_{r=t'-1}^t \beta_r \text{ for all } t = t^*, t^* + 1, \dots, T. \quad (9)$$

Combining 9 with 8, we have

$$\sum_{r=t'-1}^t \alpha_r \leq \sum_{r=t'-1}^t \beta_r \text{ for all } t = t' - 1, \dots, T. \quad (10)$$

Since  $t' - 1 < t^*$ , we obtain a contradiction to our initial assumption that  $t^*$  is the smallest integer in  $\{1, \dots, T\}$  satisfying 1.

In our next step, we shall show that all agents in  $\cup_{t=t^*}^T J_t$  are happy.

Let  $\hat{t}_1$  be the largest integer in the set  $\{t^*, t^* + 1, \dots, T\}$  such that  $\sum_{r=\hat{t}_1}^T \alpha_r \leq \sum_{r=\hat{t}_1}^T \beta_r$ . Such an integer must exist in view of the fact that  $\sum_{r=t^*}^T \alpha_r \leq \sum_{r=t^*}^T \beta_r$  (from 1). If  $\hat{t}_1 < T$ ,  $\alpha_T > \beta_T$ , the MDPA will allocate all projects in  $A_T$  (there are  $2\beta_T$  of them) to a subset of  $J_T$  of cardinality  $\alpha_T$ . A surplus of  $\alpha_T - \beta_T$  agents are transferred to the next step where these surplus agents are considered together with agents in  $J_{T-1}$  for projects in  $A_{T-1}$ . If  $\hat{t}_1 < T - 1$ , a surplus of  $\alpha_T + \alpha_{T-1} - \beta_T - \beta_{T-1}$  agents are generated at this step and transferred to the next step. By construction of  $\hat{t}_1$ ,  $\alpha_T + \alpha_{T-1} + \dots + \alpha_{\hat{t}_1} \leq \beta_T + \beta_{T-1} + \dots + \beta_{\hat{t}_1}$ . Hence, at step  $\hat{t}_1$ , all agents in  $\cup_{r=\hat{t}_1}^T J_r$  are all assigned good objects in  $\cup_{r=\hat{t}_1}^T A_r$  and are happy.

We can apply 1 to infer that  $\sum_{r=t^*}^{\hat{t}_1} \alpha_r \leq \sum_{r=t^*}^{\hat{t}_1} \beta_r$ . Therefore there exists a largest integer  $\hat{t}_2$  in the set  $\{t^*, t^* + 1, \dots, \hat{t}_1\}$  such that  $\sum_{r=\hat{t}_2}^{\hat{t}_1} \alpha_r \leq \sum_{r=\hat{t}_2}^{\hat{t}_1} \beta_r$ . Repeating the argument in

the previous paragraph, we can conclude that all agents in  $\cup_{r=\hat{t}_2}^{\hat{t}_1} J_r$  are all assigned good objects in  $\cup_{r=\hat{t}_2}^{\hat{t}_1} A_r$  and are happy. Proceeding in this way, we obtain integers  $\hat{t}_3, \dots, \hat{t}_k = t^*$  and can apply the earlier argument to conclude that all agents in  $\cup_{r=t^*}^T J_t$  are assigned good projects and are happy.

Consider the step where agents in  $J_{t^*-1}$  are allocated projects in  $A_{t^*-1}$  by the MDPA. Applying 2,  $\alpha_{t^*-1} > \beta_{t^*-1}$  so that a surplus of  $\alpha_{t^*-1} - \beta_{t^*-1}$  is transferred to the next step where projects in  $A_{t^*-2}$  are allocated. Again, applying 2,  $\alpha_{t^*-1} + \alpha_{t^*-2} > \beta_{t^*-1} + \beta_{t^*-2}$  and a surplus of  $\alpha_{t^*-1} + \alpha_{t^*-2} - \beta_{t^*-1} - \beta_{t^*-2}$  is transferred to the step where projects in  $A_{t^*-3}$  are allocated. Proceeding in this manner and applying 2 at every step, we conclude that a surplus of  $\gamma = \sum_{r=1}^{t^*-1} \alpha_r - \sum_{r=1}^{t^*-1} \beta_r$  remains in the final step when projects in  $A_1$  are allocated. These are clearly the agents who remain unhappy.

This establishes the lemma. ■

Since  $|F_q|$  is even, all agents in this friendship component are assigned in Step  $q$  and no agent is placed in the residual set  $R$ . According to the MDPA, all agents in  $F_q$  are paired with each other - let  $\sigma$  denote this assignment.

Suppose  $\sigma$  can be weakly blocked by  $S \subseteq F_q$  i.e.  $|S|$  is even and there is an assignment  $\sigma'$  where agents  $S$  are paired with each other with projects in  $X(S, \sigma')$  such that all agents in  $S$  are at least as well-off as under  $\sigma$  and at least one agent is strictly better-off.

According to our notation for the MDPA,  $A(q.0)$  is the set of projects available to  $F_q$  at Step  $q.0$ . Applying Lemma 1 with  $A' = A(q.0)$ , we can infer that there exists  $t^*$  satisfying inequalities 1 and 2. For convenience, we denote the sets  $\cup_{t=t^*}^T A_t$  and  $\cup_{t=1}^{t^*-1} A_t$  by  $\bar{A}$  and  $\underline{A}$  respectively and the sets  $\cup_{t=t^*}^T J_t$  and  $\cup_{t=1}^{t^*-1} J_t$  by  $\bar{J}$  and  $\underline{J}$  respectively. Let  $H^\sigma(\underline{J}) \subset \underline{J}$  denote the set of happy agents in  $\underline{J}$  i.e. the agents who receive good projects in  $\sigma$ .

We make three observations regarding  $\sigma$  below using Lemma 1 and the properties of the MDPA.

**OBSERVATION 5** All agents in  $\bar{J}$  are assigned objects in  $\bar{A}$  and are happy. All unhappy agents in  $F_q$  belong to  $\underline{J}$ . All happy agents in  $\underline{J}$  are assigned projects in  $\underline{A}$ .

**OBSERVATION 6** In view of inequalities 2, all projects in the set  $\underline{A}$  are assigned.

**OBSERVATION 7** A project that has been assigned to two agents in  $F_q$  who are both unhappy cannot be a good project for any agent in  $\underline{J}$ . This follows from the fact that for every  $A_t$  ( $t \in \{1, \dots, t^* - 1\}$ ), the number of agents who demand projects in  $A_t$  strictly exceeds twice the number of projects in  $A_t$ . Thus every project in MDPA is assigned to a pair of agents for whom the project is good.

We now return to the proof. Let  $i_1 \in S$  be an agent who strictly improves by blocking. By Observation 5,  $i_1 \in \underline{J}$ . Let  $(i_1, i_2, x_1) \in \sigma'$ . It follows from the construction of the sets

$(A_t, J_t)$ ,  $t = 1, \dots, T$  that  $i_1 \in \underline{J}$  implies  $x_1 \in \underline{A}$ . This is so because  $i_1$  does not like any project in the set  $\overline{A} = \cup_{t=1}^T A_t$ . According to Observation 6,  $x_1$  was assigned to some pair  $(i_3, i_4)$  in  $\sigma$ . By the definition of weak blocking,  $i_3, i_4 \in S$ . Also Observation 7 implies that  $x_1$  is a good project for  $i_3$  and  $i_4$ . Since  $x_1 \in \underline{A}$ , we have  $i_3, i_4 \in \underline{J}$ . Observe  $i_3, i_4 \in H^\sigma(\underline{J})$ . Thus agents  $i_3, i_4$  must continue to be assigned good projects in  $\sigma'$ . Let  $x_2$  be the project assigned to  $i_3$  in  $\sigma'$ .

Replicating the arguments in the previous paragraph, we infer that there exists  $(i_5, i_6, x_2) \in \sigma$  such that (i)  $i_5, i_6 \in S$ , (ii)  $x_2$  is a good project for  $i_5, i_6$ , (iii)  $i_5, i_6 \in \underline{J}$  and (iv)  $x_2 \in \underline{A}$ . Observe  $i_5, i_6 \in H^\sigma(\underline{J})$ .

Proceeding in this manner, we conclude that there exists a sequence of distinct agents  $i_1, i_3, i_4, i_5, \dots$  all of whom belong to both  $S$  and  $\underline{J}$  who also receive good projects in  $\sigma'$ . Let  $Q$  denote the set of these agents. Note that agents  $i_3, i_4, i_5, \dots$  are chosen successively from  $H^\sigma(\underline{J})$ . Since  $H^\sigma(\underline{J})$  is finite, it must be the case that  $Q = \{i_1\} \cup H^\sigma(\underline{J})$  i.e.  $|Q| > |H^\sigma(\underline{J})|$ .

Applying Lemma 1,  $|H^\sigma(\underline{J})| = |\underline{J}| - \gamma$ . Consider an arbitrary allocation of projects in  $\underline{A}$  to agents in  $\underline{J}$ . By construction, the good projects of agents in  $\underline{J}$  must belong to the set  $\underline{A}$ . Therefore the set of unhappy agents amongst the set of agents  $\underline{J}$  must at least be  $\gamma$  i.e.  $|Q| \leq |S| - \gamma = |H^\sigma(\underline{J})|$ . However this contradicts our earlier conclusion that  $|Q| > |H^\sigma(\underline{J})|$ .

Therefore  $\sigma$  belongs to the  $F_q$ -friendship core. If  $|F_q|$  is even for all  $q$ , it follows immediately that  $\sigma$  belongs to the friendship core. ■

## 9 APPENDIX 3

We provide a proof of Theorem 2.

*Proof of Theorem 2:* Let  $i$  be an agent such that  $i \in F_q$ . Let  $\succsim_i$  be a preference ordering for agent  $i$  and  $\succsim$  be a preference profile. We will show that there does not exist  $\succsim'_i$  such that  $\sigma(\succsim'_i, \succsim_{-i}) \succ_i \sigma(\succsim)$ . Note that the outcome in the MDPA does not depend on whether  $\succsim_i$  is partner or project dominant. We therefore only need to show that  $i$  cannot benefit by misreporting  $G^i(\succsim_i)$  or equivalently her threshold project. In view of the assumptions made on the sets of good projects, this is equivalent to checking that  $i$  cannot benefit by changing her threshold project.

Suppose  $(k, i, a) \in \sigma(\succsim)$ . There are two separate cases to consider: Case 1 where  $i$  is a residual agent and Case 2 where  $i$  is matched with a friend. We consider each case in turn.

Case 1: Note that a misreport by agent  $i$  will not change her residual status as the MDPA algorithm always picks the minimum agent in  $F_q$  according to  $\succ^N$ . Suppose there exists a misreport  $\succsim'_i$  such that  $i$  strictly improves. In both cases,  $i$  is not matched with a friend; therefore  $a \notin G^i(\succsim_i)$  while the misreport yields a project  $b \in G^i(\succsim_i)$ .

Since  $a \notin G^i(\succsim_i)$ , it must be the case that  $i$  is assigned in Step  $L + 2$  (Observation 4) for the profile  $\succsim$ . Suppose  $b$  is assigned in Step  $L + 1$  at  $(\succsim'_i, \succsim_{-i})$  and  $(i, k', b) \in \sigma(\succsim'_i, \succsim_{-i})$ .

Assume  $k' \succ^N i$ . It must be the case that the pair  $(k', i)$  was considered in  $\sigma(\succ)$  in some substep of  $L + 1$ ; otherwise the pair would not be considered in  $(\succ'_i, \succ_{-i})$ . Since  $(i, k', b) \in \sigma(\succ'_i, \succ_{-i})$  is formed in Step  $L + 1$ , we have  $b \in G^{k'}(\succ_{k'})$  (Observation 4). Also  $b \in G^i(\succ_i)$  as agent  $i$  strictly improves by misreporting. So  $b \in G^i(\succ_i) \cap G^{k'}(\succ_{k'})$  and  $i$  would have received a good project in  $\sigma(\succ)$  which is a contradiction. Assume that  $i$  has a higher priority than  $k'$ . Then the pair  $(i, k')$  would have been considered in some substep in Step  $L + 1$  and  $i$  would have received a good project in  $\sigma(\succ)$ . We therefore conclude that  $(i, k', b)$  could not have been formed in Step  $L + 1$  in  $(\succ'_i, \succ_{-i})$  i.e.  $b$  is assigned to  $(k', i)$  in some substep of  $L + 2$ .

It is clear that the set of unassigned agents and available projects in Step  $L + 2$  is the same in  $\succ$  and  $(\succ'_i, \succ_{-i})$ . Moreover the sets  $\bar{R}^1$  and  $\bar{R}^2$  are the same in both cases. Since  $b$  is available in this step,  $i \in \bar{R}^2$ . If  $k' \in \bar{R}^1$ , then  $(i, k', b) \in \sigma(\succ)$  which is a contradiction. Finally if  $k' \in \bar{R}^2$ ,  $i$  and  $k'$  must be consecutive in priority in  $\bar{R}^2$  which implies that  $(i, k', b) \in \sigma(\succ)$ . This is a contradiction.

Case 2: In this case,  $k, i \in \tilde{F}_q$ . Agent  $i$  will continue to be matched to another agent in  $\tilde{F}_q$  for any misreport. Assume without loss of generality that the set of available projects to  $\tilde{F}_q$  is  $\{a_1, a_2, \dots, a_T\}$  and  $a_T \succ_q^O a_{T-1} \succ_q^O \dots \succ_q^O a_1$ .

Consider an arbitrary preference profile  $\succ$ . Then its associated good set profile for component  $\tilde{F}_q$  is  $\{G^j(\succ_j)\}$ ,  $j \in \tilde{F}_q$ . As described earlier, it is equivalent to a profile of threshold projects,  $\{a^j\}$ ,  $j \in \tilde{F}_q$ . This profile generates a demand vector  $d(a_t)$ ,  $t = 1, \dots, T$ . By assumption, the good project profile is homophily consistent and  $d(a_1) \geq d(a_2) \geq \dots \geq d(a_T)$ . The profile also specifies the sets of agents who demand each project  $S(a_1), S(a_2), \dots, S(a_T)$ .

According to the MDPA and as a consequence of homophily (and the tie breaking assumption), projects are considered in the sequence  $a_T, a_{T-1}, \dots, a_2, a_1$ . Abusing notation slightly, let  $d^r(a_t)$  denote the demand for project  $a_t$  when  $a_r$  is considered by the algorithm i.e.  $a_r$  has the least demand and is chosen after tie breaking. Thus  $d^T(a_t) = d(a_t)$  in the demand vector in the first step when project  $a_T$  is being considered. When  $a_r$  is being considered, all projects  $a_t$ ,  $t > r$  have already been considered. Note that these projects have not necessarily been allocated to some pair; however for the purpose of the algorithm we can regard  $d^r(a_t) = 0$  whenever  $t > r$ . Let  $S^r(a_t)$ ,  $t \leq r$  denote the set of agents who have positive demand for project  $a_t$ .

Suppose  $i$ 's threshold project in the profile  $\{G^j(\succ_j)\}$ ,  $j \in \tilde{F}_q$  is  $a_k$ , i.e she has positive demand only for projects  $a_t$  where  $t \leq k$ . We begin with an important observation. Agent  $i$  can manipulate only if she does not receive a good project when being truthful. By Observation 1, we know that agent  $i$  is present in the termination step for  $\tilde{F}_q$  (Step  $q.T$ ) for the profile  $\succ$ . Thus agent  $i$  is present in all substeps of Step  $q$ .<sup>26</sup> We claim that  $d^t(a_t) \geq 3$  for all  $t \leq k$ . Moreover there are at least two agents in each set  $S^t(a_t)$  who are ahead of  $i$

<sup>26</sup>Note that it is possible that  $i$  belongs to a waiting pair in a substep.

in the order  $\succ^N$ . Let  $a_t$  be a project with  $t \leq k$ . If  $i$  belongs to a waiting pair when  $a_t$  is being considered, she will receive a good project (Observation 2). Therefore it must be the case that  $i \in S^t(a_t)$ . If  $d^t(a_t) = 1$ , agent  $i$  will be a waiting pair and by Observation 2, must receive a good project. If  $d^t(a_t) = 2$ ,  $i$  gets  $a_t$ . Hence  $d^t(a_t) \geq 3$  so that project  $a_t$  must be allocated when it is considered. There must therefore be two agents who are ahead of  $i$  in the numerical order in the set  $S^t(a_t)$  who are allocated  $a_t$ .

Agent  $i$  can misreport in only one of two ways: (i) by announcing a threshold  $a_m$  where  $T \geq m > k$ , i.e by expanding the set of good projects and (ii) by announcing a threshold  $a_m$  where  $1 \leq m < k$ , i.e by contracting the set of good projects.

We consider case (i) first. Let  $\hat{d}^r(a_t)$ ,  $t \leq r$  denote the demand of projects when project  $a_r$  is being considered, i.e these are the demands in the various substeps for  $\tilde{F}_q$  when the MDPA is run on the profile where  $i$  misreports. Similarly  $\hat{S}^r(a_t)$  denotes the set of agents who demand  $a_t$  at the misreported profile. We will track the project received by  $i$  in this profile and show that it cannot belong to the set  $\{a_1, a_2, \dots, a_k\}$ .

Consider projects  $a_r$  where  $r > m$ . For such projects,  $\hat{d}^r(a_r) = d^r(a_r)$  and  $\hat{S}^r(a_r) = S^r(a_r)$ . Hence these projects if allocated, are allocated to the same agents in the truthful and misreported profiles. In addition any agent, project pair that is a waiting pair in one of the profiles is also a waiting pair in the other.

We now turn to the case where  $a_m$  is being considered in the misreported profile. By assumption,  $i \in \hat{S}^m(a_m)$ . Let  $\hat{d}_{-i}^m(a_m) = \hat{d}^m(a_m) - 1$  and  $\hat{S}_{-i}^m(a_m) = \hat{S}^m(a_m) \setminus \{i\}$ . Note that  $d^m(a_m) = \hat{d}_{-i}^m(a_m)$  and  $S^m(a_m) = \hat{S}_{-i}^m(a_m)$  by virtue of the argument in the previous paragraph.

There are several cases to consider at this point. If  $\hat{d}_{-i}^m(a_m) = 0$ ,  $(i, a_m)$  is the waiting pair at this step. If there already exists a waiting pair, then  $i$  is allocated  $a_m$  which is, by assumption, a bad project. Otherwise  $(i, a_m)$  is a waiting pair and will be assigned a project later. Note that  $\hat{d}_{-i}^{m-1}(a_{m-1}) = d^{m-1}(a_{m-1})$  and  $\hat{S}_{-i}^{m-1}(a_{m-1}) = S^{m-1}(a_{m-1})$ .

Suppose  $\hat{d}_{-i}^m(a_m) = 1$  or  $\hat{d}_{-i}^m(a_m) \geq 2$  and  $i$  is one of the two highest priority agents in the set  $\hat{S}^m(a_m)$ . Then  $i$  is allocated  $a_m$  which is a bad project. Otherwise, since  $S^m(a_m) = \hat{S}_{-i}^m(a_m)$ ,  $a_m$  will be allocated to the same two agents who were assigned  $a_m$  in the truthful profile. Once again,  $\hat{d}_{-i}^{m-1}(a_{m-1}) = d^{m-1}(a_{m-1})$  and  $\hat{S}_{-i}^{m-1}(a_{m-1}) = S^{m-1}(a_{m-1})$ .

We can therefore conclude the following: if  $i$  is assigned a project at this step, it must be to a bad one; otherwise  $\hat{d}_{-i}^{m-1}(a_{m-1}) = d^{m-1}(a_{m-1})$  and  $\hat{S}_{-i}^{m-1}(a_{m-1}) = S^{m-1}(a_{m-1})$ .

Suppose  $i$  is not assigned a project at Step  $q.a_m$ .<sup>27</sup> Consider Step  $q.a_{m-1}$ . Suppose  $i \notin S(a_{m-1})$ , i.e.  $m - 1 > k$  and  $a_{m-1}$  is not a good project for  $i$ . If  $\hat{d}_{-i}^{m-1}(a_{m-1}) = 0$ ,  $i$  remains waiting with  $a_m$ . If  $\hat{d}_{-i}^{m-1}(a_{m-1}) = 1$ ,  $i$  is allocated  $a_{m-1}$  with the other demander for  $a_{m-1}$ . Since  $a_{m-1}$  is not a good project for  $i$ , misreporting is not beneficial. Similarly

---

<sup>27</sup>Step  $q.a_m$  denotes the step in which project  $a_m$  is the least demanded project that is chosen after tie breaking.



$\hat{D}_{-i}^{m-1}(a_{m-1}) \geq 2$  and  $i$  is one of the two highest priority agents in  $\hat{S}^{m-1}(a_{m-1})$ ;  $i$  is assigned  $a_{m-1}$ . The only remaining case is when  $i$  is not among the two highest agents  $\hat{S}^{m-1}(a_{m-1})$  according to the numerical order. In this case,  $\hat{S}_{-i}^{m-1}(a_{m-1}) = S^{m-1}(a_{m-1})$  implies that the same pair of agents are allocated  $a_{m-1}$  in the misreported as in the truthful profile.

Summarizing, we have reached the same conclusion in stage  $a_{m-1}$  as in stage  $a_m$ . Suppose  $a_{m-1}$  is a bad project for  $i$ . If  $i$  is allocated an project in stage  $a_{m-1}$ , it must be to a bad project. If  $i$  has not been allocated an project, it must be true that  $\hat{d}_{-i}^{m-2}(a_{m-2}) = d^{m-2}(a_{m-2})$  and  $\hat{S}_{-i}^{m-2}(a_{m-2}) = S^{m-2}(a_{m-2})$ .

In fact, the same argument can be replicated for all stages  $a_{m-t}$ ,  $t = 0, \dots, m-k$  till one reaches  $a_k$  where the project been considered by the algorithm in the misreported profile is a good project of  $i$ . In particular, we can conclude that either  $i$  is allocated a bad project or the algorithm reaches stage  $a_k$  with  $\hat{d}_{-i}^k(a_k) = d^k(a_k)$  and  $\hat{S}_{-i}^k(a_k) = S^k(a_k)$ .

At Step  $q.a_k$ , we have already argued that  $d^k(a_k) \geq 3$  and there are two agents other than  $i$  in  $S^k(a_k)$  who are ahead of  $i$  in the priority order  $\succ^N$ . Therefore  $\hat{d}_{-i}^k(a_k) \geq 2$ . Moreover  $\hat{S}_{-i}^k(a_k) = S^k(a_k)$  implies that the same pair of agents who were assigned  $a_k$  in the truthful profile are also allocated  $a_k$  in the misreported profile. The same argument can be applied repeatedly to show that if  $i$  has not been assigned at Step  $q.a_{k+1}$ , she cannot be assigned any of the projects  $a_1, \dots, a_{k-1}, a_k$ . The algorithm then assigns a bad project to  $i$ . This can happen in one of two ways. If  $i$  is waiting with an project  $a_l$ ,  $k < l \leq T$ , after  $a_1$  has been allocated,  $i$  will be allocated  $a_l$ . Otherwise,  $i$  will receive a bad project after allocation of good projects in  $F_q$  has been completed. In any case, manipulation does not succeed.

Finally, consider case (ii) where  $i$  misreports by contracting her good set. Following the earlier argument, it is clear that assignment proceeds in exactly the same manner in the misreported profile as in the truthful profile until Step  $q.a_k$  is reached. This implies that the same agents who were assigned projects  $a_1, \dots, a_k$  are not assigned in Steps  $q.a_{k+1}$  and earlier. Hence at each Step  $q.a_t$ ,  $t = k, k-1, \dots, 1$ ,  $\hat{d}_{-i}(a_t) \geq 2$  and  $\hat{S}_{-i}(a_t)$  contains two agents with higher priority than  $i$ . Therefore these agents are assigned  $a_1, \dots, a_k$  and  $i$  does not receive a good project. This completes the proof.

## REFERENCES

- ABIZADA, A. (2016): "Exchange-stability in roommate problems," *Review of Economic Design*, 1–10.
- ALKAN, A. (1988): "Nonexistence of stable threesome matchings," *Mathematical social sciences*, 16, 207–209.
- BARBERÀ, S., H. SONNENSCHN, L. ZHOU, ET AL. (1991): "Voting by committees," *Econometrica*, 59, 595–609.

- BIRÓ, P. AND E. MCDERMID (2010): “Three-sided stable matchings with cyclic preferences,” *Algorithmica*, 58, 5–18.
- BOGOMOLNAIA, A. AND H. MOULIN (2004): “Random matching under dichotomous preferences,” *Econometrica*, 72, 257–279.
- BOGOMOLNAIA, A., H. MOULIN, AND R. STONG (2005): “Collective choice under dichotomous preferences,” *Journal of Economic Theory*, 122, 165–184.
- BORDER, K. C. AND J. S. JORDAN (1983): “Straightforward elections, unanimity and phantom voters,” *The Review of Economic Studies*, 50, 153–170.
- BRETON, M. L. AND A. SEN (1999): “Separable preferences, strategyproofness, and decomposability,” *Econometrica*, 67, 605–628.
- BURKETT, J., F. X. FLANAGAN, AND A. L. GRIFFITH (2018): “Allocating group housing,” *Social Choice and Welfare*, 50, 581–596.
- CARROLL, G. (2017): “Robustness and separation in multidimensional screening,” *Econometrica*, 85, 453–488.
- CECHLÁROVÁ, K. AND T. FLEINER (2005): “On a generalization of the stable roommates problem,” *ACM Transactions on Algorithms (TALG)*, 1, 143–156.
- COHEN, J. M. (1977): “Sources of peer group homogeneity,” *Sociology of education*, 227–241.
- COMBE, J. (2017): “Matching with Ownership,” .
- GAURAV, A., J. PICOT, AND A. SEN (2017): “The decomposition of strategy-proof random social choice functions on dichotomous domains,” *Mathematical Social Sciences*, 90, 28–34.
- GOLUB, B. AND M. O. JACKSON (2012): “How homophily affects the speed of learning and best-response dynamics,” *The Quarterly Journal of Economics*, 127, 1287–1338.
- GUDMUNDSSON, J. (2014): “When do stable roommate matchings exist? A review,” *Review of Economic Design*, 18, 151–161.
- HART, S. AND N. NISAN (2017): “Approximate revenue maximization with multiple items,” *Journal of Economic Theory*, 172, 313–347.
- HART, S. AND P. J. RENY (2015): “Maximal revenue with multiple goods: Nonmonotonicity and other observations,” *Theoretical Economics*, 10, 893–922.
- HATFIELD, J. W. AND S. D. KOMINERS (2012): “Matching in networks with bilateral contracts,” *American Economic Journal: Microeconomics*, 4, 176–208.

- KANDEL, D. B. (1978): “Homophily, selection, and socialization in adolescent friendships,” *American journal of Sociology*, 84, 427–436.
- KHARE, S. AND S. ROY (2018): “Stability in Matching with Couples having Non-Responsive Preferences,” *Available at SSRN 3288195*.
- KLAUS, B. AND F. KLIJN (2005): “Stable matchings and preferences of couples,” *Journal of Economic Theory*, 121, 75–106.
- (2007): “Paths to stability for matching markets with couples,” *Games and Economic Behavior*, 58, 154–171.
- KLAUS, B., F. KLIJN, AND J. MASSÓ (2007): “Some things couples always wanted to know about stable matchings (but were afraid to ask),” *Review of Economic Design*, 11, 175–184.
- KURINO, M. (2014): “House allocation with overlapping generations,” *American Economic Journal: Microeconomics*, 6, 258–89.
- MCPHERSON, M., L. SMITH-LOVIN, AND J. M. COOK (2001): “Birds of a feather: Homophily in social networks,” *Annual review of sociology*, 27, 415–444.
- MISHRA, D. AND S. ROY (2013): “Implementation in multidimensional dichotomous domains,” *Theoretical Economics*, 8, 431–466.
- OSTROVSKY, M. (2008): “Stability in supply chain networks,” *American Economic Review*, 98, 897–923.
- PYCIA, M. (2012): “Stability and preference alignment in matching and coalition formation,” *Econometrica*, 80, 323–362.
- RAGHAVAN, M. (2018): “Pairwise Allocation Via Partner Trading,” .
- ROTH, A. E. AND A. POSTLEWAITE (1977): “Weak versus strong domination in a market with indivisible goods,” *Journal of Mathematical Economics*, 4, 131–137.
- ROTH, A. E., T. SÖNMEZ, AND M. U. ÜNVER (2005): “Pairwise kidney exchange,” *Journal of Economic theory*, 125, 151–188.
- ROTH, A. E. AND M. A. O. SOTOMAYOR (1992): *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*, 18, Cambridge University Press.
- SETHURAMAN, J. AND A. SMILGINS (2016): “Two-sided matching with objects,” .

- SHAPLEY, L. AND H. SCARF (1974): "On cores and indivisibility," *Journal of mathematical economics*, 1, 23–37.
- TAN, J. J. M. ET AL. (1991): "A necessary and sufficient condition for the existence of a complete stable matching," *Journal of Algorithms*, 12, 154–178.
- VERBRUGGE, L. M. (1983): "A research note on adult friendship contact: a dyadic perspective," *Soc. F.*, 62, 78.