



A simple matching domain with indifferences and a master list

Rohan Chowdhury¹

Received: 22 September 2020 / Accepted: 27 January 2022 / Published online: 18 February 2022
© The Author(s) 2022

Abstract

We present a two-sided one-to-one market setting where one side of the market is objectively ranked. Constraints prevent agents on the other side from matching with agents who are ranked higher than an agent-specific threshold. The domain allows for indifferences, with preferences of one side being derived from a master preference list, while agents on the other side have general preferences. We analyze matchings with respect to two standard properties: *stability* and *efficiency*. Having met these standard goals, the designer facing this problem may additionally care about a third criterion: the *size* of the matching. We find that for the proposed domain all stable matchings have the same size. We characterize the set of stable and efficient matchings for the problem. Our results show that every matching that is both stable and efficient is hierarchical by nature in a precise sense.

Keywords Assignment · Matching · Indifferences · Stable-size · Serial dictatorship

JEL Classifications C78 · D47 · D82

1 Introduction

Consider a two-sided one to one market setting, where agents on one side of the market (say “Projects”) are grouped into indifference classes that are exogenously ranked, and agents on the other side (say “Firms”) prefer higher ranked projects. However, each firm face internal feasibility constraints that prevent her from matching with projects ranked above an individual threshold. Examples include—(1) the refugee matching problem: where *benevolent hosts* with a limited number of beds to spare, are willing to shelter *refugee families* in their homes in exchange for a monetary reward (which is increasing in family size) from the government; (2) firms competing for large-scale

✉ Rohan Chowdhury
r.chowdhury.1@research.gla.ac.uk

¹ Adam Smith Business School, University of Glasgow, Glasgow G12 8QQ, UK

governmental projects over which there is generally a consensus as to which ones are more desirable, but firm-specific capacity constraints (technological, deadline, budget, etc.) prevent it from going after very top projects; (3) the slot allocation problem: where *slots* are arranged on a line with earlier slots ranked higher, *users* arrive over time and wish to be served as early as possible once they do, for instance, researchers wishing to use a supercomputer, ships wanting to load/unload cargo at a port, etc. Being indifferent between different options is a ubiquitous phenomenon in our society, and the above examples are no exception.

These observations collectively inform our assumptions on the preference domain. We assume that projects are exogenously ranked (but several projects can have the same rank, thus giving rise to indifference classes), and each firm f faces a threshold t_f . This firm f prefers projects of higher rank, but projects of rank above t_f are not acceptable to her. Preferences of firms thereby have a nested structure (formally presented in Sect. 2). Each project j can have arbitrary (non-strict) preferences over firms but prefers any firm to remaining unmatched. Taking preferences of both sides into account, firms must be matched to projects in a one-to-one fashion.

We pursue two important design goals that are standard in the literature. The first is *stability*, which ensures that agents would accept the matching proposed by the designer and would not try to individually renegotiate it. To define it, we rely on the notion of *strong blocking*: two agents form a pair and block a matching if each agent in the pair strictly prefers the other over their current match.¹ The second is *efficiency*, i.e., the matching cannot be Pareto improved upon at any preference profile. Having met these two goals, in the above examples, the designer may additionally wish to improve welfare by maximizing as much as possible the *size* (i.e., the total number of matched pairs) of the proposed matching.

In our domain with indifferences, all stable matchings have the same size (Proposition 1). To the best of our knowledge, it is the first non-trivial non-strict preference domain in the literature where this fact holds. This implies that, in the space of stable and efficient matchings, the designer is not constrained by welfare considerations regarding the matching size. The absence of this constraint has an important practical implication. This is because, finding a stable matching that maximizes size is a computationally hard problem in domains with indifferences, even under severe restrictions on preferences.²

We propose two assignment rules, both of which always generate a stable and efficient matching for any given problem. The rules are in the spirit of the classical serial dictatorship rule, where an ordering of projects (in the queue) is used to determine the allocation of firms: the first project in the queue picks his best firm and leaves with that firm, from what remains the second project in the queue picks, from what remains the third project in the queue picks and so on.³ The first rule, which we call the *decreasing*

¹ This notion is called *weak stability* in the literature, but we refer to it simply as *stability*. Irving (1994) proposes two other natural definition of stability in the presence of indifferences: *strong stability* and *super stability*. These are stronger notions, but contrary to weak stability, they are not guaranteed to exist.

² For additional details on these restrictions, see Manlove et al. (2002), Irving et al. (2008). For a brief discussion see the second paragraph in Sect. 1.2.

³ The rule therefore starts with the set of all firms, and at every step a firm is eliminated.

refined priority rule, adds two additional tweaks to this procedure. Firstly, the priority (queue) order over projects used by the rule is always aligned with the exogenous ranking over the indifference classes of projects: the projects in the top indifference class appear first in the queue order, followed next by projects in the second-from-top indifference class, and so on (see, Definition 2). This ensures stability. Secondly, the rule starts with the set of matchings instead of the set of firms. Following the queue order, at every step, the rule eliminates all⁴ matchings where the project being assigned does not get his best firm (one of his best firms if there are many), from the set of matchings available at that step. This refinement ensures that the resulting matching is efficient, in the presence of indifferences. This is elaborated further in Example 1 and the discussion thereafter.

Stability and efficiency do not however characterize the decreasing refined priority rule (Proposition 4). Nevertheless, we show that every matching that is both stable and efficient can be generated by a decreasing (non-refined) priority rule. Therefore, in our proposed domain, stable and efficient matchings are arguably hierarchical by nature: they can be thought of as an outcome of projects being arranged in a queue order, with each project receiving one of their best firms in turn from the set of remaining firms (Proposition 6).

Motivated by the above observation, we propose another class of rules that we call the *Pareto improved decreasing priority rule*. Stability and efficiency completely characterize this class (Proposition 8). The rule works in two steps. In the first step, a matching is constructed using a queue order over projects, that is aligned with the exogenous ranking over projects (just as discussed above, thereby ensuring stability). Then in the second step, it tests for the presence of Pareto improvement (PI) cycles, a notion introduced by Erdil and Ergin (2017). The absence of PI-cycles in a stable matching guarantees that it is efficient: in our domain, a stable matching can be Pareto dominated if and only if projects can form a trading cycle, where every project and firm involved gets weakly better off, with at least one of them getting strictly better (Proposition 7).

We proceed to discuss the strengthening of the stability notion to strong stability. We show that *strongly stable* matchings (those where there are not even *weak blocking* pairs where only one member of the pair strictly benefits) exist rarely, and imply a very specific structure on both preferences and the strongly stable matchings themselves. If they do exist, then at that corresponding profile, the set of all strongly stable matchings is exactly equal to the set of all matchings that can be generated by some decreasing refined priority rule (Proposition 9).

1.1 Outline of the paper

In the following sub-section, we discuss the connection of this work with the existing literature. In Sect. 2 we present the formal model and the result that stable matchings have the same size. Section 3 focuses on stable and efficient matchings and presents the formal definitions for candidate assignment rules to find them. Section 4 discusses the

⁴ As opposed to eliminating only one firm from the set of available firms.

strengthening of stability notion to strong stability. A concluding discussion follows. Some proofs are relegated to the “Appendix”.

1.2 Relation to literature

It is well known that, in a bilateral matching problem, when every agent has a strict (but arbitrary) preference over agents on the other side, the size of all stable matchings are identical, a consequence of the *Lone Wolf Theorem*⁵ (McVitie and Wilson 1970; Gale and Sotomayor 1985). This result breaks down if we allow for general preferences (Roth and Sotomayor 1990). However, in our domain with indifferences, the equivalence of stable matchings with respect to size remains true.

This paper contributes to the literature on matching with indifferences. In particular, it is closely related to the Stable Marriage problem with Ties and Incomplete Lists with a Master List (Irving et al. 2008). The term “Ties” simply means indifferences in our context; “Incomplete Lists” refer to the fact that each agent’s preference list may consist of only a subset of the members of the other side (the *acceptable* partners of this agent); and, an agent’s preference list contains her acceptable partners ranked precisely according to the “Master List”. In the above problem, Irving et al. note that even when preferences of both sides are derived from a master list,⁶ weakly stable matchings need not have the same size, and subsequently discusses the algorithmics of finding weakly stable matchings that maximize size. In our model, where preferences of only one side are derived from a master list, while preferences of the other side are not incomplete;⁷ weakly stable matchings necessarily have the same size. This result is therefore of practical significance since, in the presence of incomplete lists and ties (indifferences), finding a weakly stable matching that maximizes size is a hard problem; even under restrictions on the number and length of ties (Manlove et al. 2002), or preferences of both sides being limited to a master list (Irving et al. 2008).⁸ On structural results, Manlove (2002) proves the lattice structure of the set of strongly stable matchings with ties (without allowing for incomplete lists) and notes the absence of such a structure for the set of weakly stable matchings. Our structural results, therefore, do not concern the set of weakly stable matchings. We argue instead that, every matching that is both (weakly) stable and efficient admits a hierarchical structure in a precise sense.

Erdil and Ergin (2017) study a very general two-sided many-to-one matching domain with indifferences. Apart from PI-cycles that we use in this paper, the notion of PI-chains plays an important role in the algorithm they propose. The presence of PI-chains in a stable matching indicates the possibility for welfare improvement by accommodating a previously unmatched project and reshuffling existing matches, eventually leading to an increased matching size. By the size-equivalence result, such a scenario does not arise in our domain. In our more restricted setting, we are addi-

⁵ The theorem illustrates that the set of unmatched agents does not depend on the choice of stable matching.

⁶ Note that, there are two master lists, one for each side of the market.

⁷ We assume that each *project* prefers any *firm* over remaining unmatched.

⁸ For a comprehensive survey on algorithmic results related to matching with indifferences, we refer the reader to Manlove (2013).

tionally able to pin down the structure of stable and efficient matchings as well as strongly stable matchings. In fact, we fully characterize the set of all stable and efficient matchings, and the set of all strongly stable matchings (whenever they exist).

Other matching applications with indifferences include: the school choice problem with weak priorities (Erdil and Ergin 2008; Ehlers and Erdil 2010);⁹ the housing market model where agents are initially endowed with a house (Aziz and De Keijzer 2012); kidney exchange models (Roth et al. 2005; Andersson and Kratz 2020). Unlike in our model, in these studies, welfare of only one side of the market is relevant, i.e., the efficiency notion is one-sided.

Our mechanism design problem involves matching agents to each other without the possibility of resorting to money or lotteries. As a consequence, most fairness requirements are immediately out of reach. Any matching will therefore favour some agents and leave others largely unsatisfied. In such settings, it is widely observed that resulting mechanisms tend to have a hierarchical structure. Formally proving, or even formalizing this statement is however elusive. Under strict preferences some results are obtained: see, for instance, pure assignment models without transfers and exchange models in Svensson (1999), Pápai (2000), and Pycia and Ünver (2017). Under non-strict preferences, we are only aware of Svensson (1994) and Bogomolnaia et al. (2005). Most of these papers attempt to characterize the set of incentive-compatible (in a strong sense of strategy-proofness) and efficient mechanisms and demonstrate that they have a hierarchical structure.

We study a more general setting in this paper. Our model can be considered as an extension of the “house allocation” model. Here the role of “houses” is played by “firms”. But contrary to that model, firms now have preferences, albeit rather homogeneous ones. When all projects have the same rank, our model is reduced to the “house allocation” model with general non-strict preferences. In this reduced model, Bogomolnaia et al. (2005) showed that the set of all stable and efficient assignments is equal to the set of all assignments obtained by some refined priority rule.¹⁰ In our more general case, the results have a similar flavour. A decreasing refined priority rule always outputs a stable and efficient matching. Every stable and efficient matching is a result of a decreasing (non-refined) priority rule. The set of all stable and efficient matchings is equal to the set of all matchings that can be generated by some Pareto improved decreasing priority rule. If a preference profile admits strongly stable matching(s), then the set of all strongly stable matchings is equal to the set of all matchings that can be generated by some decreasing refined priority rule.

Finally, the preference domain studied in this paper is also related to that of the refugee matching problem discussed by Andersson and Ehlers (2020). In their model, *refugee families* (projects) are also ordered according to family size giving rise to blocks containing projects of the same size. *Hosts* (firms) have limited beds and therefore their preferences have a nested structure¹¹ over these blocks similar to ours. The differences are as follows. Firstly, in our domain, hosts are necessarily indiffer-

⁹ Preferences of agents (students) in these studies are however strict.

¹⁰ Since all projects have the same rank, the notion of a decreasing priority order plays no role.

¹¹ Hosts prefer refugee families of greater size subject to the condition that the host has a sufficient number of beds.

ent between all refugee families in the same block (as a result we have indifference classes); while in their model, hosts can have general preferences over refugee families within the same block while respecting the nested structure between blocks. Secondly, contrary to our domain, their model do not elicit any preferences from refugee families whatsoever. It is important to take into account their preferences: there is growing evidence that the initial placement of refugee families greatly affects outcomes like education, job prospects, and earnings; which in turn profoundly alters their lifetime welfare, as most refugees do not move from the localities to which they are resettled for many years (Åslund and Rooth 2007; Damm 2014; Jones and Teytelboym 2017; Åslund et al. 2010; Martén et al. 2019). As described in Jones and Teytelboym (2018), ignoring preferences for refugees has even caused families seeking shelter in Finland to cancel their asylum applications.

2 Model

Throughout the rest of this article, we use the project-firm terminology for convenience. We take the liberty to personify projects and firms, and whenever convenient, we refer to them using pronouns “he” and “she”, respectively. There is a finite set $J = \{1, 2, \dots, |J|\}$ of projects, and a finite set F of firms. Let $A = J \cup F$ denote the set of all agents. Furthermore, J is partitioned into a set of indifference classes $\{J_1, J_2, \dots, J_p\}$, and there is an exogenously given (strict) order \succ over these classes. Without loss of generality, suppose it is $J_p \succ J_{p-1} \succ \dots \succ J_1$. All the above ingredients are fixed for the rest of the paper.

The preference domain

The preference relation of every firm $f \in F$ is completely identified by a threshold $t_f \in \{1, 2, \dots, p\}$ that she reports. Let R_k^* be a weak order (complete and transitive relation) over $J \cup \{f\} (f \in F)$ defined as follows:

$$J_k P_k^* J_{k-1} P_k^* \dots J_1 P_k^* \mathbf{f} P_k^* J_p I_k^* J_{p-1} \dots I_k^* J_{k+1}$$

where P_k^* and I_k^* are the antisymmetric and symmetric parts of R_k^* , respectively, and $k \in \{1, 2, \dots, p\}$. Then, the preference relation of a firm $f \in F$ with threshold t_f , denoted by R_f , is given by $R_f \equiv R_{t_f}^*$ where $t_f \in \{1, 2, \dots, p\}$ and the \mathbf{f} in the definition of $R_{t_f}^*$ is to be interpreted simply as the corresponding firm f . Preferences of firms have a nested structure in the sense that if for two firms f and f' , we have $R_f \equiv R_k^*$ and $R_{f'} \equiv R_{k'}^*$ with $k > k'$, then over the indifference classes $J_{k'}, J_{k'-1}, \dots, J_1$, both R_f and $R_{f'}$ are identical.

Preference of a project $j \in J$, denoted by R_j , is a weak order over the set $F \cup \{j\}$. Let P_j and I_j denote the antisymmetric and symmetric parts of R_j , respectively.

Let $R = (R_i)_{i \in J \cup F}$ denote a profile of preferences. Given a profile R , we will say that a firm f (resp., project j) is *acceptable* to a project j (resp., firm f) if

$f P_j j$ (resp., $j P_f f$). We will say that a project j and a firm f are *compatible* if they are mutually acceptable to each other. We assume throughout that all projects find every firm acceptable, i.e., for any project $j \in J$, we assume that $f P_j j \forall f \in F$.

Finally, let \mathcal{R} denote the set of all such preference profiles.

Notations and definitions

A **matching** is a function $\mu : J \cup F \rightarrow J \cup F$ such that:

- $\mu(j) \in F \cup \{j\} \forall j \in J$
- $\mu(f) \in J \cup \{f\} \forall f \in F$
- $\mu(f) = j \iff \mu(j) = f, \forall j \in J, f \in F$

Let \mathcal{M} be the set of all possible matchings.

Given a preference profile $R \in \mathcal{R}$, let us define partial orders R_A, R_J , and R_F on the set of matchings as follows. Let $\mu R_J \nu$, if $\mu(j) R_j \nu(j)$ for every $j \in J$; let $\mu R_F \nu$, if $\mu(f) R_f \nu(f)$ for every $f \in F$; and let $\mu R_A \nu$, if $\mu R_J \nu$ and $\mu R_F \nu$. Let I_A, I_J and I_F denote the symmetric parts, and let P_A, P_J and P_F denote the asymmetric parts of these relations.

A matching $\mu \in \mathcal{M}$ is **individually rational** at profile $R \in \mathcal{R}$ if every agent is *acceptable* to his or her mate, i.e., $\nexists i \in A$ such that $i P_i \mu(i)$.

A matching $\mu \in \mathcal{M}$ is *strongly blocked* at profile $R \in \mathcal{R}$ by a pair $(j, f) \in J \times F$, if $j P_f \mu(f)$ and $f P_j \mu(j)$. A matching $\mu \in \mathcal{M}$ is *weakly blocked* at profile $R \in \mathcal{R}$ by a pair $(j, f) \in J \times F$, if $j R_f \mu(f)$ and $f R_j \mu(j)$, and at least one of these two relations holds strictly. A matching $\mu \in \mathcal{M}$ is **stable** at profile $R \in \mathcal{R}$, if it is individually rational and is not strongly blocked by any pair of agents. A matching $\mu \in \mathcal{M}$ is **strongly stable** at profile $R \in \mathcal{R}$, if it is individually rational and is not even weakly blocked by any pair of agents.

A matching ν *Pareto dominates* a matching μ at profile R if $\nu P_A \mu$, i.e., all projects and firms weakly prefer ν to μ , and at least one project or a firm strictly prefers ν to μ . A matching $\mu \in \mathcal{M}$ is **efficient** at profile $R \in \mathcal{R}$ if it is not Pareto dominated by another matching at R .

The **size** of a matching $\mu \in \mathcal{M}$ is defined as the total number of pairs $(j, f) \in J \times F$ such that $\mu(j) = f$.

Recall that J is partitioned into a set of indifference classes. Consider one such class $J_k, k \in \{1, 2, \dots, p\}$. Now consider a project $j \in J_k$ and a firm f with $R_f \equiv R_s^*$ where $s < k$. Then, given the way the preference R_s^* is defined, project j is not acceptable to her. Then, although project j considers firm f as an acceptable match, individual rationality dictates that they are not matched to each other. Therefore, it will be useful to keep track of all firms that considers the class J_k as acceptable.

Given a preference profile R and some $k \in \{1, 2, \dots, p\}$, let $C(J_k, R)$ denote the set of firms that find projects in J_k acceptable, i.e., for every firm $f \in C(J_k, R)$ we have $R_f \equiv R_s^*$ where $s \geq k$ (therefore, $J_k P_f f$). Given the nested structure of preferences, for $k' < k$, every firm that considers J_k acceptable also finds $J_{k'}$ acceptable but not necessarily the other way round. As a result, we have an inclusion: $C(J_p, R) \subset C(J_{p-1}, R) \cdots \subset C(J_1, R)$. Let $J_{<k}$ be a shorthand to denote J_x with $x < k$. $J_{\leq k}$,

$J_{\geq k}$ and $J_{>k}$ are similarly defined. Finally, note that every firm $f \in C(J_k, R)$ strictly prefers J_k over $J_{<k}$, i.e., for every $f \in C(J_k, R)$ we have $J_k P_f J_{<k}$.

Given a matching μ , for every indifference class $J_k, k \in \{1, 2, \dots, p\}$, the set of projects in J_k that are matched under μ are collected in the set below:

$$J_k^\mu = \{j \in J_k \mid \mu(j) \neq j\}$$

In our preference domain, stable matchings are equivalent with respect to size.

Proposition 1 *Given any profile of preference $R \in \mathcal{R}$, any two stable matchings μ and μ' must have the same size. In particular, for every $k \in \{1, 2, \dots, p\}$,*

$$|J_k^\mu| = |J_k^{\mu'}|$$

Proof Consider first the top indifference class of projects J_p . Let μ and μ' denote two stable matchings such that $|J_p^\mu| > |J_p^{\mu'}|$, i.e., fewer projects from J_p are matched under μ' compared to μ . Since a project in J_p can only be matched to some firm in $C(J_p, R)$, this means that there are at least $|J_p^\mu|$ firms in $C(J_p, R)$. But only $|J_p^{\mu'}|$ of them are matched under μ' . This implies that there exists at least one firm $f \in C(J_p, R)$ that was matched to a project in J_p under μ but is either unmatched or is matched to a project in $J_{<p}$ under μ' . But $|J_p^\mu| > |J_p^{\mu'}|$ implies there is a project $j \in J_p$ that is matched under μ but not under μ' . Pair (j, f) would then block μ' , a contradiction to the stability of μ' .

Next, let $k < p$ be the largest integer such that there exist two stable matchings μ and μ' such that $|J_k^\mu| > |J_k^{\mu'}|$, i.e. fewer projects from J_k are matched in μ' than in μ . Therefore, both μ and μ' matches the same number of projects from classes, $J_{k+1}, J_{k+2}, \dots, J_p$, i.e. we have $|J_q^\mu| = |J_q^{\mu'}| = X_q \forall q \in \{k + 1, k + 2, \dots, p\}$. Let the total number of such projects be X . Thus,

$$X = X_{k+1} + X_{k+2} + \dots + X_p$$

This means that there are at least X firms in $C(J_{k+1}, R)$. Now, since $|J_k^\mu|$ projects are matched under μ , it must be the case that there are at least $X + |J_k^\mu|$ firms in $C(J_k, R)$. However, only $X + |J_k^{\mu'}| < X + |J_k^\mu|$ firms from $C(J_k, R)$ are matched to some project from $J_{\geq k}$ under μ' . Therefore, there is a firm $f \in C(J_k, R)$ that is either unmatched or is matched to a project in $J_{<k}$ (less preferred to J_k) under μ' . Also, $|J_k^\mu| > |J_k^{\mu'}|$ implies there is at least one project $j \in J_k$ which is unmatched under μ' . Then pair (j, f) blocks μ' , a contradiction. □

3 Assignment rules for stable and efficient matchings

We propose and discuss various candidate assignment rules to solve any instance of the given problem.

Let us recall that, the exogenously given order over the indifference classes of projects $\{J_1, J_2, \dots, J_p\}$ is given by: $J_p \succ J_{p-1} \succ \dots \succ J_1$.

The assignment rules we discuss are parameterized by a priority order over projects. A **priority order (or, queue order)** over the set of projects J is a bijection mapping $\sigma : J \rightarrow J$. To refer to positions in a queue order, we will use the phrase *queue slots*.

Since preferences of one side (firms) are aligned (see, Sect. 2), a natural starting point is the priority rule (or, the serial dictator rule), where the other side (projects) is ordered in a queue and allowed to choose a firm in turn. Next, we define this rule formally.

Recall that $C(J_k, R)$ denotes the set of firms that are compatible with projects in J_k at preference profile R . For ease of presentation, in the notation below, we suppress its dependence on R and denote it simply by $C(J_k)$.

For any given subset of firms $\bar{F} \subset F$, a preference profile R , and a project $j \in J_k$, denote by

$$B_j(\bar{F} | R) = \begin{cases} \{f \in \bar{F} \cap C(J_k) \mid f R_j f' \forall f' \in \bar{F} \cap C(J_k)\}, & \text{if } \bar{F} \cap C(J_k) \neq \emptyset \\ \{j\}, & \text{otherwise} \end{cases}$$

the subset of \bar{F} consisting of **best compatible firms** for project j among all compatible firms in \bar{F} ; if every firm in \bar{F} is incompatible with project j , it is just the set containing project j .

Definition 1 The **priority rule** is parameterized by an arbitrary priority order σ over projects. For any fixed priority order σ , the priority rule is a correspondence $\psi_\sigma^P : \mathcal{R} \rightarrow \mathcal{M}$ such that, for any preference profile R we have that $\psi_\sigma^P(R) = M$, with M denoting a set of matchings, where every matching $\mu \in M$ is constructed iteratively such that:

$$\begin{aligned} \mu(\sigma(1)) &\in B_{\sigma(1)}(F | R) \\ \mu(\sigma(2)) &\in B_{\sigma(2)}(F \setminus \{\mu(\sigma(1))\} | R) \\ \mu(\sigma(3)) &\in B_{\sigma(3)}(F \setminus \{\mu(\sigma(1)), \mu(\sigma(2))\} | R) \\ &\vdots \\ \mu(\sigma(|J|)) &\in B_{\sigma(|J|)}(F \setminus \{\mu(\sigma(1)), \mu(\sigma(2)), \dots, \mu(\sigma(|J| - 1))\} | R) \end{aligned}$$

In words, the priority rule sequentially (according to priority order σ) assigns every project to one of his best compatible firms at every iteration, from the set of remaining

compatible firms at that iteration. If no compatible firm is available, then the project remains unmatched. Since preferences exhibit indifferences, based on the choice made at each iteration, several different matchings may be reached by this rule for any fixed profile of preferences.

Given any profile R , an outcome of the priority rule, i.e., a matching $\mu \in \psi_\sigma^P(R)$ for some priority order σ , is not necessarily stable. This is because the priority order σ is not necessarily aligned with the exogenous order \succ over the indifference classes of projects. That is to say, for two projects $j \in J_k$ and $j' \in J_{<k}$,¹² it may be the case that $\sigma(j') < \sigma(j)$, i.e., project j' is assigned earlier. Therefore, if project j' is assigned to some firm $f \in C(J_k, R)$ that project j strictly prefers to his own partner, then (j, f) forms a strong blocking pair since firm f is also compatible with project j and hence prefers project j over j' .

The above observation necessitates that the priority order be aligned with the exogenous order $J_p \succ J_{p-1} \succ \dots \succ J_1$. Accordingly, we define next the notion of a decreasing priority order, and correspondingly, the decreasing priority rule. Indeed, it turns out that this new rule solves the above-mentioned issue and always outputs a stable matching (see, Proposition 2).

Definition 2 For any priority order σ , the associated decreasing priority order, denoted by σ^D , is evaluated in the following way.

For any two projects i and j , let $i \in J_x$ and $j \in J_y$. Then σ^D is derived from σ such that:

- * if $x > y$, then $\sigma^D(i) < \sigma^D(j)$
- * if $x = y$, then $\sigma^D(i) < \sigma^D(j) \iff \sigma(i) < \sigma(j)$

A priority order σ is called a **decreasing priority order** if $\sigma = \sigma^D$.

In words, for any given priority order σ , the associated decreasing priority order σ^D aligns the priority of projects according to the exogenous order \succ over the indifference classes of projects: any project $j \in J_k$ gets a better priority in σ^D over all projects $j' \in J_{<k}$; while, if two projects $j, j' \in J_k$, i.e., they belong to the same indifference class, then their relative priority in σ^D remains the same as in the original priority order σ .

Definition 3 The **decreasing priority (DP) rule** ψ_σ^{DP} with respect to any fixed priority order σ , is simply the priority rule with respect to the associated decreasing priority order σ^D . That is, for any preference profile R , we have,

$$\psi_\sigma^{DP}(R) = \psi_{\sigma^D}^P(R)$$

Proposition 2 Consider a preference profile $R \in \mathcal{R}$ and let $\mu \in \psi_\sigma^{DP}(R)$, then μ is stable but not necessarily efficient.

¹² $J_{<k}$ is a shorthand to denote J_x for some $x < k$.

The proof of stability appears in the ‘‘Appendix’’. The intuition is as follows: since the priority order over projects is aligned according to \succ , whenever a firm f strictly prefers a project j over her own partner, it must be the case that this firm f was available to be assigned to project j , which in turn implies project j either prefers his current partner over firm f or is indifferent between them. As a result, the blocking pair of the kind discussed above in the context of the priority rule never forms.

Although always stable, the output (matching) produced by the DP rule is not necessarily efficient. This inefficiency stems from the combination of two facts. At any given iteration, firstly, the rule does not take into account the preferences of projects that are to be matched in subsequent iterations; and secondly, preferences of projects exhibit indifferences. The following example illustrates this inefficiency.

Example 1 Consider a single indifference class $J = \{1, 2\}$ of projects and two firms A and B , both of which are compatible with this class.

Suppose preferences of projects are as follows below:

R_1	R_2
A, B	A B

Then, $\psi_\sigma^{DP}(R) = \left\{ \begin{pmatrix} 1 & 2 \\ A & B \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix} \right\}$ for the priority order $\sigma : \sigma(1) = 1, \sigma(2) = 2$. It is easy to check that both these matchings are stable, while only the second is also efficient.

In the above example, project 1 is indifferent between firm A and B , and project 1 has the higher priority. Just ensuring that this project gets one of his best compatible firms without considering that project 2 strictly prefers firm A may lead to inefficiency. Additional refinements or adjustments are therefore necessary to guarantee efficiency. Specifically, in any iteration, if one encounters the scenario that there are multiple best compatible firms for the project in question, one must find a way to not outright eliminate any possibility that does not hurt this concerned project. Doing so allows for utility gains to be had in future iterations.

Accordingly, unlike in the case of the priority rule, where we start from the set of all firms and remove a firm at every iteration; we instead modify the procedure and start from the set of all matchings. At any iteration, it is then possible to keep every option (matching) where the currently assigned project receives one of his best compatible firms (even if there are multiple such firms), allowing for full utility gains to be had in future iterations. We call this modified procedure the *refined priority rule*.

To formally define it, we will use the notation $B^*(\cdot | R)$, which is the counterpart of $B(\cdot | R)$ presented earlier in the context of the priority rule. The difference is that B operates on a set of firms, while B^* operates on a set of matchings. We will additionally require the following notation.

Given a set of matchings $M \subset \mathcal{M}$, a preference profile R , and a project $j \in J_k$, denote by $C_j^*(M, R) := \{\mu \in M \mid \mu(j) \in C(J_k, R)\}$, the subset of M where project

j is matched to a firm that is compatible with this project at preference profile R . For ease of presentation, we suppress the dependence of $C_j^*(M, R)$ on R , and denote it simply by $C_j^*(M)$.

For any given set of matchings $M \subset \mathcal{M}$, a preference profile R , and a project $j \in J_k$, let

$$B_j^*(M | R) = \begin{cases} \{\mu \in C_j^*(M) \mid \mu(j) R_j \mu'(j) \ \forall \mu' \in C_j^*(M)\}, & \text{if } C_j^*(M) \neq \emptyset \\ \{\mu \in M \mid \mu(j) = j\}, & \text{otherwise} \end{cases}$$

denote, the subset of M consisting of the **best compatible matchings** for project j in the sense that project j is matched to one of his best-ranked compatible firms among all matchings in M . If there are no matchings where project j is matched to a compatible firm, $B_j^*(M | R)$ is simply the subset of M consisting of all matchings where project j is unmatched.

Definition 4 A **refined priority (RP) rule** with respect to any fixed priority order σ , is a correspondence $\psi_\sigma^{RP} : \mathcal{R} \rightarrow \mathcal{M}$ such that, for any preference profile R , we have that $\psi_\sigma^{RP}(R) = M_{|J|}$, where $M_{|J|}$ is defined as follows. Let $M_0 = \mathcal{M}$ (the set of all possible matchings),

$$\begin{aligned} M_1 &= B_{\sigma(1)}^*(M_0 | R) \\ M_2 &= B_{\sigma(2)}^*(M_1 | R) \\ &\vdots \\ M_{|J|} &= B_{\sigma(|J|)}^*(M_{|J|-1} | R) \end{aligned}$$

Clearly, the above rule can be multi-valued, but note that a direct consequence of its definition is that all projects are indifferent between all elements (matchings) in $M_{|J|}$.

Definition 5 The **decreasing refined priority (DRP) rule** ψ_σ^{DRP} with respect to any fixed priority order σ , is simply the RP rule with respect to the associated decreasing priority order σ^D . That is, for any preference profile R , we have,

$$\psi_\sigma^{DRP}(R) = \psi_{\sigma^D}^{RP}(R)$$

Proposition 3 Consider a preference profile $R \in \mathcal{R}$ and let $\mu \in \psi_\sigma^{DRP}(R)$, then μ is both stable and efficient at R .

The formal proof of the proposition appears in the ‘‘Appendix’’. Here we present an informal discussion. The DRP rule always uses a priority order that is aligned with the exogenous order $J_p > J_{p-1} > \dots > J_1$. This ensures that any selection (matching)

from the set $\psi_{\sigma}^{DRP}(R)$ is stable¹³ at profile R . Since this rule is also refined,¹⁴ any selection from $\psi_{\sigma}^{DRP}(R)$ also happens to be efficient at profile R . Intuitively, since at every iteration, the rule only eliminates those matchings in which, the project assigned at that iteration does not get one of his best firms conditional on every project in all previous iterations receiving one of their best firms; it leaves no room for increasing welfare further. Any attempt at increasing an agent’s welfare must come at the expense of another.

Next, we discuss the structure of matchings that are both stable and efficient. For any given preference profile, we ask, whether or not every stable and efficient matching at that profile always results from one of the earlier defined rules.

Proposition 4 *There exists a preference profile R , and a matching μ that is stable and efficient at R such that $\mu \notin \psi_{\sigma}^{DRP}(R)$.*

Proof Consider the following example.

Suppose we have indifference classes $J_2 = \{1, 2\}$ and $J_1 = \{3\}$ of projects and three firms $F = \{A, B, C\}$ that are compatible with all projects. Let preferences be as below:

(a) Projects			(b) Firms		
R_1	R_2	R_3	R_A	R_B	R_C
A, B	A	B	1, 2	1, 2	1, 2
C	C	A, C	3	3	3
	B				

The matching $\mu = \begin{pmatrix} 1 & 2 & 3 \\ A & C & B \end{pmatrix}$ is stable and efficient at R .

The two decreasing priority orders are:

$$\sigma : \sigma(1) = 1, \sigma(2) = 2, \sigma(3) = 3$$

$$\bar{\sigma} : \bar{\sigma}(1) = 2, \bar{\sigma}(2) = 1, \bar{\sigma}(3) = 3$$

It is easy to verify that $\mu \notin \psi_{\sigma}^{DRP}(R)$ and $\mu \notin \psi_{\bar{\sigma}}^{DRP}(R)$. □

Proposition 4 implies that stability and efficiency do not characterize the decreasing refined priority rule.

Proposition 5 *There exists a preference profile R , and a matching μ that is stable and efficient at R such that $\mu \notin \psi_{\sigma}^{RP}(R)$.*

Proof Consider the following example with two projects $J = \{1, 2\}$ and two firms $F = \{A, B\}$. Preferences are as under:

¹³ See discussion preceding Definition 2.

¹⁴ Refer to the section following Example 1 for an elaborate discussion.

(a) Projects		(b) Firms	
R_1	R_2	R_A	R_B
A, B	A	1	1
	B	2	2

The matching $\mu = \begin{pmatrix} 1 & 2 \\ A & B \end{pmatrix}$ is stable and efficient at this profile.

But the refined priority rule with both possible priority orders produces the matching $\mu' = \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix}$. □

For the example in the preceding proof, μ and μ' are the two stable and efficient at profile R . For priority order σ with $\sigma(1) = 1$ and $\sigma(2) = 2$, we have, $\psi_\sigma^{DP}(R) = \{\mu, \mu'\}$, while $\psi_\sigma^{RP}(R) = \{\mu'\}$. This difference is driven by the fact that the latter operates on the set of matchings while the former on the set of firms. Project 2 strictly prefers μ' to μ while project 1 is indifferent between them. Using the RP rule, both μ and μ' survives after the first iteration and μ is thereafter eliminated by project 2. But if project 1 is assigned to $A = \mu'(2)$ in the first iteration, it is no longer available for project 2. Therefore, μ can still be generated by using a DP rule.

In Proposition 6 below, we show that every stable and efficient matching can be generated by a DP rule. This demonstrates that matchings that are both stable and efficient are hierarchical by nature: they can be thought of as a result of projects being arranged in a queue order, with each project being matched to one of his best firms in turn, from the set of remaining firms. Efficiency is necessary for this result to hold. Stable but not efficient matchings do not necessarily admit this structure, as demonstrated by the following example.

Example 2 Consider the following. One indifference class $J = \{1, 2\}$ of projects. Two firms A and B , both of which are compatible with this class. Preferences of projects are as under:

R_1	R_2
A	B
B	A

The matching $\mu = \begin{pmatrix} 1 & 2 \\ B & A \end{pmatrix}$ is stable but not efficient. It is easy to check that there is no priority order σ such that $\mu \in \psi_\sigma^P(R)$.

Proposition 6 Consider a preference profile $R \in \mathcal{R}$ and let μ be a stable and efficient matching at R . Then, there is a priority order σ over projects such that $\mu \in \psi_\sigma^{DP}(R)$.

The formal proof of this proposition appears in the ‘‘Appendix’’. We briefly outline the proof steps next. Fix any indifference class of projects J_r . Since μ is stable at R , a firm $f \in C(J_r, R)$ is matched to some project in $J_{<r}$ under μ only in the event that all projects in J_r are already matched to some firm in $C(J_r, R)$ that they weakly prefer over this firm f . Thus, in a stable matching, the demand of projects in J_r is catered to before projects in $J_{<r}$; and accordingly projects in J_r receive an earlier queue slot compared to projects in $J_{<r}$. This means the priority order is decreasing. We still need the individual queue slots for projects within every indifference class. Fix the top class J_p . By efficiency, we must have that at least one project in J_p is matched to his best firm under μ . This fact identifies precisely the project(s)¹⁵ which receive the earliest queue slot within projects in J_p . Removing the corresponding matched firms from the set of firms and repeatedly applying the same arguments until we exhaust J_p gives the individual queue slots for all projects within J_p . Finally, repeating the process for $J_{p-1}, J_{p-2}, \dots, J_1$ in turn, completes the proof.

It is important to note here that, even though one can always find a priority order σ such that $\mu \in \psi_\sigma^{DP}(R)$ for any matching μ that is stable and efficient at R ; every selection (matching) from the set $\psi_\sigma^{DP}(R)$, in spite of always being stable, is not guaranteed to be efficient (see, Proposition 2 or Remark 1).

Let Σ denote the set of all possible decreasing priority orders over the set of projects. For any profile R , denote by $\mathcal{M}_{SE}(R)$ the set of all stable and efficient matchings at R . Next denote by $\mathcal{M}_{DP}(R) := \cup_{\sigma \in \Sigma} \psi_\sigma^{DP}(R)$, the set of all matchings that can be obtained using some DP rule. Similarly, let $\mathcal{M}_{DRP}(R)$ denote the set of all matchings that can be obtained using some DRP rule.

The preceding results can be summarised as follows:

Remark 1 For any preference profile $R \in \mathcal{R}$,

$$\mathcal{M}_{DRP}(R) \subseteq \mathcal{M}_{SE}(R) \subseteq \mathcal{M}_{DP}(R)$$

with, there existing profiles where one or both of these inclusions above are strict.

Next, we define another class of rules that Pareto improves upon the DP rule. We make use of the notion of Pareto improvement cycles introduced by Erdil and Ergin (2017) (henceforth, EE’17). Since the DP rule always outputs a stable matching, in the discussion below we implicitly assume that the matching is stable.

We start by noting that Pareto improving upon a stable matching preserves stability, an implication of the following lemma.

Lemma 1 *Let μ be a stable matching. Let ν be another matching that every agent weakly prefers to μ , i.e., $\nu R_A \mu$. Then, ν must also be stable.*

Proof Since μ is stable, it is individually rational, and therefore ν is also individually rational. Next, assume that ν is not stable. Suppose $(j, f) \in J \times F$ strongly blocks

¹⁵ If there are multiple such projects, any order between them works.

v . Then, $f P_j v(j)$ and $j P_f v(f)$. But then $f P_j v(j) R_j \mu(j)$ and $j P_f v(f) R_f \mu(f)$, implying that (j, f) strongly blocks μ , a contradiction. \square

Definition 6 Fix a profile R and a matching μ . A **Pareto improvement (PI) cycle** consists of projects $j_1, j_2, \dots, j_n \equiv j_0$ ($n \geq 2$) such that:

- (a) Every project j_t is matched to a firm.
- (b) $\mu(j_{t+1}) R_{j_t} \mu(j_t)$ and $j_t R_{\mu(j_{t+1})} j_{t+1} \quad \forall t \in \{0, 1, \dots, (n - 1)\}$.
- (c) At least one of the preference relations in (b) is strict, for some $t \in \{0, 1, \dots, (n - 1)\}$.

Every project j_t in a PI-cycle weakly prefers the firm of the following project j_{t+1} , and the firm $\mu(j_{t+1})$ matched to the latter would not mind swapping project j_{t+1} with j_t . Additionally, at least one project strictly *envies* the following project, or at least one firm $\mu(j_{t+1})$ strictly prefers j_t to j_{t+1} . If there is a PI-cycle, then a Pareto dominating matching v can be obtained by matching each project in the cycle to the firm of the next project:

$$v(j) = \begin{cases} \mu(j_{t+1}) & \text{if } j = j_t \text{ for some } t \in \{0, 1, \dots, n - 1\} \\ \mu(j) & \text{otherwise.} \end{cases}$$

By **implementing a PI-cycle**, we mean constructing the new matching v that Pareto dominates μ as above. The existence of at least one PI-cycle therefore imply that μ is not efficient. Proposition 7 below shows that the converse of this is also true: if μ is stable and admits no PI-cycle, then μ must be efficient. This result is an adaptation of Theorem 1 in EE'17 to our domain. In their more general setting, to ascertain if a stable matching is efficient, one needs to additionally test for the existence of *Pareto improvement (PI) chains*. PI-chains are defined in a similar fashion, except that PI-chains accommodate a previously unmatched project (thereby increasing the matching size) by readjusting assignments in the manner described above. PI-chains do not play any role in our model: by the size-equivalence result in Proposition 1, it is impossible to accommodate a previously unmatched project without hurting another.

A directed graph $G = (V, E)$ consists of a set V of vertices and a set E of directed edges, where a directed edge (henceforth, *edge*) is an ordered pair of vertices, i.e., an element of $V \times V$. Let us denote an edge (x, y) as $x \rightarrow y$. A cycle in G consists of distinct vertices x_0, \dots, x_{n-1} ($n \geq 2$) such that $x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_{n-1} \rightarrow x_n \equiv x_0$. Note that, given a directed graph, if each vertex of this graph has exactly one incoming and one outgoing edge, then each edge of the graph is part of a cycle.¹⁶

Proposition 7 *A stable matching is efficient if and only if it does not admit any PI-cycles.*

Proof (if part) Assume that μ is stable but not efficient. Suppose v Pareto dominates μ . Since μ is stable and $v P_A \mu$, Lemma 1 and Proposition 1 imply that the set of projects

¹⁶ There may be multiple cycles, but every edge is necessarily part of one such cycle.

and firms that are matched in ν is identical to that in μ . Among these matched projects, collect in $\hat{J} := \{j \in J \mid \nu(j) \neq \mu(j)\}$ all those that are matched to different firms in the two matchings. Note that $|\hat{J}| \geq 2$. Next, construct a directed graph G with \hat{J} as the set of vertices in the following manner. For any project $j \in \hat{J}$, take the unique firm f such that $\nu(f) = j$. Find the project j' such that $\mu(f) = j'$. Clearly $j' \in \hat{J}$. Let there be an edge $j \rightarrow j'$. Since $\nu P_A \mu$, it is easy to check that if $j \rightarrow j'$ then $\mu(j')R_j\mu(j)$ and $jR_{\mu(j')}j'$. If one of these preferences is strict, denote the corresponding edge by $j \rightarrow j'$, and call it a *strict* edge.

Given the way edges are defined, every vertex in G must have exactly one outgoing and one incoming edge. Therefore, each edge of G is part of a cycle. Since ν Pareto dominates μ , G must have at least one strict edge. As a result, there is at least one cycle containing a strict edge, that leads to a PI-cycle.

The other direction is easy to see. □

Definition 7 The **Pareto improved decreasing priority (PIDP) rule** is parameterized by an arbitrary priority order σ over projects. For any fixed priority order σ , the PIDP rule is a correspondence $\psi_\sigma^{PIDP} : \mathcal{R} \rightarrow \mathcal{M}$ such that for any preference profile R , we have that $\psi_\sigma^{PIDP}(R) = M$, where M denotes a set of matchings such that every $\mu \in M$ satisfies the following:

- (a) $\mu \in \psi_\sigma^{DP}(R)$.
- (b) μ does not admit a PI-cycle.

For any given priority order, at every preference profile, the PIDP rule outputs a subset of the set of matchings produced by the DP rule, by eliminating all those that admit at least one PI-cycle.

Fix a priority order σ and a profile R . Consider a matching $\mu \in \psi_\sigma^{DP}(R)$. Suppose it admits a PI-cycle. Let μ' be obtained by implementing a PI-cycle. Then, $\mu' \in \psi_\sigma^{DP}(R)$. Therefore, at any profile R , the following procedure leads to the same set of matchings (M) as the PIDP rule:

Select (iteratively) a matching from $\psi_\sigma^{DP}(R)$. It is always stable. As long as it is not efficient, by Proposition 7 there will always be a PI-cycle. If so, find a new matching by implementing a PI-cycle. This matching will also be stable by Lemma 1. Repeat this process as long as the new matching does not have a PI-cycle. Since the model is finite, an efficient matching will be found in finitely many steps. Collect this matching in the set M if it is not already present. EE'17 shows that the search for PI-cycles is computationally efficient, thereby making the process of finding a stable and efficient matching fast.

It is then clear that the PIDP rule always gives a stable and efficient matching. In fact, stability and efficiency completely characterize the class of PIDP rules (Proposition 8).

Denote by $\mathcal{M}_{PIDP}(R) := \cup_{\sigma \in \Sigma} \psi_\sigma^{PIDP}(R)$, the set of all matchings that can be obtained using some Pareto improved decreasing priority rule.

Proposition 8 For any preference profile $R \in \mathcal{R}$,

$$\mathcal{M}_{SE}(R) = \mathcal{M}_{PIDP}(R)$$

Proof Follows from Propositions 2, 6 and 7. \square

4 Strongly stable matchings

Strongly stable matchings are not guaranteed to exist in our preference domain (see, for instance, the example used in Proposition 5 which does not admit any strongly stable matching). In fact, the existence of a strongly stable matching is a rare event, and implies a very specific structure of both preferences and the strongly stable matchings themselves. We outline this phenomenon using the lemmas presented below.

In what follows we assume that the preference profile admits at least one strongly stable matching (henceforth, SSM).

Lemma 2 Every SSM is efficient.

Proof Let μ be a SSM that can be Pareto improved by matching μ' . There is an agent x (firm or project) who strictly prefers μ' to μ . Hence, x is matched at μ' , say to some agent y . Since μ' is Pareto superior to μ , y is at least as well at μ' as at μ . Hence, the pair (x, y) weakly blocks μ , a contradiction to μ being a SSM. \square

Lemma 3 In any SSM, for every $k \in \{1, 2, \dots, p\}$, the set of projects J_k is either fully matched or fully unmatched.

Proof Fix a SSM μ . Suppose there are two projects $i, j \in J_k$, such that i is matched with firm f at μ while j is unmatched. Since firm f is indifferent between projects i and j , the pair (j, f) weakly blocks μ , a contradiction. \square

For any matching μ and a set of projects \hat{J} , let $\mu(\hat{J})$ denote the set of firms matched with some project in \hat{J} .

Lemma 4 Fix a preference profile and a SSM μ . For any $k \in \{1, 2, \dots, p\}$, let F_k denote the set of firms that are not matched with some project in $\bigcup_{x=k+1}^p J_x$ at μ . Then,

- For any matched project $j \in J_k$, firm $\mu(j)$ belongs to the top indifference class of this project j among all compatible firms of project j in the set F_k .
- A firm $f \in F_k \setminus \mu(J_k)$ does not belong to the top indifference class (within compatible firms in F_k) of any project $j \in J_k$.
- $\mu(J_k) = T_k$, where $T_k \subset F_k$ is the set of firms that are in the top indifference class (within compatible firms in F_k) for at least one project $j \in J_k$.

Proof Part (a): If project j would strictly prefer some firm $f \in F_k$ to $\mu(j)$, then (j, f) would weakly block μ (firm f cannot strictly prefer its current match to j , since it is not matched to a project of better rank).

Part (b): Any firm $f \in F_k \setminus \mu(J_k)$ compatible with projects in J_k , prefers any project $j \in J_k$ to its match in μ . If this firm f is in the top (within compatible firms in F_k) indifference class of any such project $j \in J_k$, then project j is indifferent between firms f and $\mu(j)$. Hence (j, f) weakly blocks μ , a contradiction.

Part (c): Take any firm $f \in \mu(J_k)$. Then by Part (a), we have $f \in T_k$ and so, $\mu(J_k) \subset T_k$. Next, assume for contradiction that $f \in T_k$ but $f \notin \mu(J_k)$. This means $f \in F_k \setminus \mu(J_k)$. Then by Part (b) we have, $f \notin T_k$, a contradiction. Thus, $f \in \mu(J_k)$ and so, $T_k \subset \mu(J_k)$. Hence $\mu(J_k) = T_k$. \square

Lemma 5 Let μ be a SSM at preference profile R . Suppose $k^* \in \{1, 2, \dots, p\}$ is the largest integer such that there is a project $j \in J_{k^*}$ which is matched at μ . Then,

- (a) There does not exist a firm f that is compatible with projects in $J_{>k^*}$ at preference profile R .
- (b) J_{k^*} is always fully matched at every SSM at R .

Proof (a): If such a firm would exist, then that firm would form a weak blocking pair with a project of rank better than k^* .

(b): Since there is a project $j \in J_{k^*}$ which is matched at μ , the firm $\mu(j)$ finds projects in J_{k^*} compatible. By Lemma 4(a), this firm $\mu(j)$ belongs to the top indifference class of project j among all his compatible firms. Therefore, this firm $\mu(j)$ must be matched to a project in J_{k^*} (possibly different from j) in every other SSM. This implies at least one project in J_{k^*} must be matched at every SSM at R . By Lemma 3, J_{k^*} must then be fully matched in every SSM at R . \square

Lemma 6 Suppose μ and μ' are two SSMs at a given preference profile R . Then the set of firms matched to every indifference class of projects J_k remains the same across these two SSMs, i.e.,

$$\mu(J_k) = \mu'(J_k) \quad \forall k \in \{1, 2, \dots, p\}.$$

Proof Let $k^* \in \{1, 2, \dots, p\}$ be the largest threshold such that projects in J_{k^*} are compatible for some firm in F at preference profile R . Then by individual rationality, all projects in $J_{>k^*}$ are unmatched in both μ' and μ . Thus, $\mu(J_k) = \mu'(J_k) \quad \forall k \in \{k^* + 1, k^* + 2, \dots, p\}$. Moreover, Lemma 5 implies that both μ' and μ must fully match all projects in J_{k^*} . By Lemma 4(c) we have that, $\mu(J_{k^*}) = T_{k^*} = \mu'(J_{k^*})$, where T_{k^*} is the set of all firms which are in the top indifference class for at least one project in J_{k^*} .

Next let $k^{**} \in \{1, 2, \dots, k^* - 1\}$ be the largest threshold such that projects in $J_{k^{**}}$ are compatible for some firm in $F \setminus T_{k^*}$ at preference profile R . If $k^{**} < k^* - 1$, then projects in $J_{k^{**}+1}, J_{k^{**}+2}, \dots, J_{k^*-1}$ are unmatched in both μ and μ' , since there

are no compatible firms for these projects at preference profile R that are not already matched to some better-ranked project in both μ and μ' . Furthermore, by Lemma 5 and Lemma 4(c) we have $\mu(J_{k^{**}}) = T_{k^{**}} = \mu'(J_{k^{**}})$.

Proceeding in this fashion completes the proof. \square

Lemma 7 *If a preference profile admits more than one SSM, all projects are indifferent between them.*

Proof Follows from Lemmas 4(a) and 6. \square

The implications from the preceding lemmas can be summarized as follows. If a preference profile admits at least one SSM, then in every such SSM, any set of projects J_k of a given rank is either fully matched or fully unmatched. Moreover, in every such SSM, projects in J_k are matched to the same set of firms (and those firms are the best for these projects among all compatible firms not matched with a better-ranked project). These observations drive the equivalence result in Proposition 9: if a preference profile admits at least one strongly stable matching, then the set of all strongly stable matchings at that profile is exactly equal to the set of all matchings that can be generated by a decreasing refined priority rule at that profile. The proof appears in the ‘‘Appendix’’.

For any preference profile R , let $\mathcal{M}_{SS}(R)$ denote the set of all matchings that are strongly stable at profile R . Also recall that $\mathcal{M}_{DRP}(R)$ denotes the set of all matchings which can be obtained using some decreasing refined priority rule, i.e., $\mathcal{M}_{DRP}(R) = \cup_{\sigma \in \Sigma} \psi_{\sigma}^{DRP}(R)$, where Σ denotes the set of all possible decreasing priority order. The following proposition establishes the relationship between these two sets.

Proposition 9 *Assume that preference profile $R \in \mathcal{R}$ admits a strongly stable matching. Then,*

$$\mathcal{M}_{SS}(R) = \mathcal{M}_{DRP}(R)$$

5 Concluding remarks

This article proposes a two-sided matching domain where preferences of one side of the market are derived from a master preference list, while agents on the other side have general preferences. The domain permits indifferences. It has been shown that stable matchings have the same size. We have discussed why this is a positive result due to its practical significance. Our discussion on the assignment rules demonstrates the hierarchical nature of stable and efficient matchings for the problem. In particular, we have shown that the set of stable and efficient matchings in this domain is completely characterized by the Pareto improved decreasing priority rule. We have also discussed the more demanding notion of strong stability and shown that its existence is not

guaranteed. If they do exist, then the set of strongly stable matchings is completely characterized by the decreasing refined priority rule. Next, we make some additional remarks about our results.

Firstly, we have not analyzed the strategic aspects of our assignment rules. It is easy to check that the decreasing refined (and, non-refined) priority rule is not manipulable by an individual project. This is not surprising given that the serial dictatorship rule is well known to be strategy-proof. One can also easily check that they are not manipulable by firms. Recall that the set of projects J is partitioned into $\{J_1, J_2, \dots, J_p\}$. We have implicitly assumed that the block to which any particular project belongs, is publicly observable. If this information is privately held, then projects can manipulate the decreasing priority order by their reports, and thereby the assignment rules.

Secondly, note that the assignment rules we have discussed are expensive in terms of the information that the designer must elicit about every project's preference over the entire set of firms. This may become a major concern as the number of agents gets very large. It is in regard to this observation that Proposition 6 has an important implication.

We know from Proposition 6 that, for any matching μ that is both stable and efficient at a preference profile R , there is a (decreasing) priority order σ over projects and a (weakly) decreasing sequence $\{\Gamma_j\}_{j=1}^{|J|}$ of choice sets consisting of firms, such that $\mu(\sigma(j)) \in B_{\sigma(j)}(\Gamma_j, |R)$ for all $j \in J$. We conjecture the existence of a sophisticated procedure whereby a stable and efficient matching may be obtained by a recursively defined procedure, where at each step and given the data from previous steps, a choice set for every "project" is defined. The "project" then only needs to report the maximal elements from the choice set. The advantage of such a procedure would be that the projects will no longer need to report the entire preference over the complete list of firms. This can reduce the information elicitation burden greatly and hence can be of particular relevance from a practical standpoint when the number of agents is significantly large. We leave the study of the existence of such a rule open for future investigation.

Acknowledgements I am grateful to Anna Bogomolnaia for her guidance and support. Comments from Hervé Moulin on an earlier draft were very helpful. I wish to thank Onur Kesten and two anonymous referees for their insightful and detailed comments that have greatly improved the paper. I thank Swarnendu Chatterjee, Damiano Turchet, Ashraf Mahfuz, Foivos Savva and Abhinaba Lahiri for helpful discussions. This work was supported by the College of Social Science Scholarship, University of Glasgow.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix

Proof Proposition 2 Fix a preference profile $R \in \mathcal{R}$, and let $\mu \in \psi_{\sigma}^{DP}(R)$ for some decreasing priority order σ . Suppose μ is not stable: some project j and some firm f block. This implies project j and firm f form a compatible pair. Let project $j \in J_k$: then firm $f \in C(J_k, R)$. Note that, it cannot be the case that both j and f are unmatched at μ as the rule requires that a project is always assigned a firm if a compatible firm is available. There are three possible remaining cases:

1. $\mu(j) = j, \mu(f) = j'$: only firm f is matched.
2. $\mu(j) = f', \mu(f) = f$: only project j is matched.
3. $\mu(j) = f', \mu(f) = j'$: both project j and firm f are matched.

Case: 1—Since (j, f) block we have: $j P_f j'$ which implies $j' \in J_{<k}$. Since ordering is decreasing, this means $\sigma(j) < \sigma(j')$. Since project j comes ahead of project j' in the ordering and remains unmatched, it must be the case that all firms compatible with j were already matched to some project which was ahead in the queue compared to j . This means, every firm in $C(J_k, R)$ which is matched under μ , is matched to some project in $J_{\geq k}$. In particular also firm f . But then $\mu(f) = j' \in J_{<k}$, a contradiction.

Case: 2—Since a firm once matched never gets unmatched, firm f was available to project j when it was assigned f' . This implies $\mu(j) = f' R_j f$. But since (j, f) block, we have $f P_j f'$, which is not possible.

Case: 3—Since (j, f) block we have: $j P_f j'$ which implies $j' \in J_{<k}$, which implies $\sigma(j) < \sigma(j')$. Now, since project j comes ahead in the queue but $\mu(f) = j'$, it means that firm f was available when project j was assigned firm f' . This means that $f' R_j f$. But since (j, f) block, we have $f P_j f'$, which is not possible. \square

Proof of Proposition 3 Fix a preference profile $R \in \mathcal{R}$ and let $\mu \in \psi_{\sigma}^{DRP}(R)$. Stability of μ follows from arguments in the same vein as in the proof of Proposition 2 with very minor modifications. We omit it here for the sake of brevity. We proceed to show that μ is efficient. The only way to increase the utility of a project j is to expand the set $M_{\sigma(j)-1}$ of feasible matchings to choose from. This is not possible without decreasing the utility for some project j' such that $\sigma(j') < \sigma(j)$. This is because, if all projects ahead of j in the queue were utility invariant when this welfare improving matching (after carrying out the reshuffling steps) was added, they would not have removed it in the first place. In particular, all (un)matched projects in μ must remain (un)matched after any reshuffling of agents to improve the welfare of any firm. Consequently, such a firm (whose welfare is possibly improved) must have been matched at μ . Also, any such reshuffling attempt requires that the matched projects remain indifferent between partners, before and after the swaps take place.

Suppose next that firm f can be made better without hurting any other agent. Let k^* be the largest integer such that $f \in C(J_{k^*}, R)$, i.e., firm f 's best-ranked acceptable projects are J_{k^*} . Since firm f gets a utility kick, it must mean that f was matched to a project $j \in J_{<k^*}$. Let $\mu(f) = j \in J_{k_j}$. Suppose in the new matching, f gets project $j' \in J_{k_{j'}}$. Then, $k^* \geq k_{j'} > k_j$: j' has a better exogenous rank compared to j . Then firm $\mu(j')$ that becomes partner-less as a result must be assigned a project in $J_{\geq k_{j'}}$. But this project was also matched in μ . Therefore, the firm that then becomes partner-less

must also be assigned a project in $J_{\geq k_{j'}}$, and so on. Thus, the Pareto improvement swaps to begin with, must sequentially involve projects in $J_{\geq k_{j'}}$ and firms in $C(J_{k_{j'}}, R)$. Now, let X be the total number of projects in $J_{\geq k_{j'}}$ that were matched in μ . This implies that X firms in $C(J_{k_{j'}}, R)$ were matched to projects with a rank strictly better than project j . Firm f was not one such firm. But since no more than X projects in $J_{\geq k_{j'}}$ can be matched after the reshuffling, there must be a firm in $C(J_{k_{j'}}, R)$ that now is either unmatched or matched to some project in $J_{< k_{j'}}$, a contradiction. \square

Proof of Proposition 6 Fix a preference profile $R \in \mathcal{R}$ and let μ be a stable and efficient matching. We have to show that there is a (decreasing) priority order σ over projects and a (weakly) decreasing sequence $\{I_j\}_{j=1}^{|J|}$ of choice sets consisting of firms, such that $\mu(\sigma(j)) \in B_{\sigma(j)}(I_j)$ for all $j \in J$.

Consider first the top set of projects J_p . If one such project is not matched, but some firm compatible with it (i.e. firm in $C(J_p, R)$) is matched with a project in $J_{< p}$, it violates stability. Thus, either all these projects are matched or all firms compatible with them are matched to a project in J_p , depending on whether $|J_p| \leq |C(J_p, R)|$. Now, since μ is efficient, there must exist at least one project in J_p who gets his best firm (one of his best firms if there are many) in $C(J_p, R)$. To see why suppose not. Next, let an arrow go from each project in J_p to all his top choice of firms. Let another set of arrows go from all firms in $C(J_p, R)$ who are matched to some project in J_p , to their corresponding matches. Since the set of agents is finite, there always exists a cycle following arrows. Exchanging along this cycle leads to a Pareto improvement, a contradiction. Collect all such projects that get one of their best firms under μ in the set X_p^1 . Next choose a bijection $\sigma_p^1 : X_p^1 \rightarrow \{1, 2, \dots, |X_p^1|\}$. For any $Y \subset J$, let $\mu(Y)$ denote the set (possibly empty) of firms matched to some project in Y . Next, consider the set $J_p \setminus X_p^1$ of projects and the set $C(J_p, R) \setminus \mu(X_p^1)$ of firms. Since μ is efficient we again get a set X_p^2 of projects who get their top choice in $C(J_p, R) \setminus \mu(X_p^1)$. Again choose a bijection $\sigma_p^2 : X_p^2 \rightarrow \{|X_p^1| + 1, |X_p^1| + 2, \dots, |X_p^1| + |X_p^2|\}$. Remove the set of matched agents and proceed in the same fashion until we exhaust J_p . We will have a collection of n_p sets $X_p^1, X_p^2, \dots, X_p^{n_p}$ and corresponding orderings $\sigma_p^1, \sigma_p^2, \dots, \sigma_p^{n_p}$. If $|J_p| > |C(J_p, R)|$, i.e. there are not enough firms (compatible with J_p), then $X_p^{n_p}$ contains all projects in J_p which are unmatched (in the absence of compatible firms, we slightly abuse the notion of top choice to also mean unmatched). If $|J_p| < |C(J_p, R)|$, all projects are matched, and by stability we have, every project in J_p weakly prefers their matched firm over all compatible firms that are unmatched or matched to projects in $J_{< p}$. This fact, together with the way X_p^i 's are defined, we see that every project in J_p is matched to one of their best available compatible firms when they allowed to choose (following the order).

Next, consider the second best set of projects and carry out the same procedure, followed by the third best and so on. To elaborate, proceeding in decreasing order, for each $k \in \{p - 1, p - 2, \dots, 1\}$, let us consider projects in J_k and firms in $C(J_k, R) \setminus \bigcup_{i=k+1}^p \mu(J_i)$. Then, by carrying out the same steps outlined earlier we will get a collection of n_k sets $\{X_k^i\}_{i=1}^{n_k}$, and corresponding orderings $\{\sigma_k^i\}_{i=1}^{n_k}$. Once we do this for every k , we finally have a collection $\{X_k^i\}_{\substack{k=1(1)p \\ i=1(1)n_k}}$ and $\{\sigma_k^i\}_{\substack{k=1(1)p \\ i=1(1)n_k}}$.

Next, define the priority order σ such that $\sigma(j) = \sigma_k^i(j)$, if $j \in X_k^i$. Note that σ is, by construction, aligned with the exogenous order over projects: $J_p \succ J_{p-1} \succ \dots \succ J_1$.

Finally, defining the collection $\{\Gamma_j\}_{j=1}^{|J|}$ of choice sets consisting of firms such that; $\Gamma_1 = F$, and for $j = \{2, 3, \dots, |J|\}$, $\Gamma_j = F \setminus \bigcup_{i=1}^{j-1} \mu(\sigma(i))$, we have the desired proof. \square

Proof of Proposition 9 We use the lemmas in Sect. 4 including the notations therein. Suppose R is such a preference profile that admits a SSM.

Let $k^* \in \{1, 2, \dots, p\}$ be the largest threshold such that projects in J_{k^*} are compatible for some firm in F at preference profile R . Then by individual rationality, all projects in $J_{k^*+1} \cup J_{k^*+2} \cup \dots \cup J_p$ remain unmatched at every SSM at preference profile R . Moreover, all projects in $J_{k^*+1} \cup J_{k^*+2} \cup \dots \cup J_p$ are also unmatched at every matching that is output by any decreasing refined priority rule by definition.

Since there is a compatible firm for projects J_{k^*} at profile R , at least one such project, can be, and must be, matched. Moreover, since preference profile R admits a SSM, by Lemma 3,

- All projects in J_{k^*} can be matched.
- All projects in J_{k^*} can be matched to T_{k^*} such that each project is matched to a best compatible firm for that project from within the set F .
- Firms in the set $F \setminus T_{k^*}$ are not in the top indifference class (within compatible firms in F) for any project in J_{k^*} .

This implies that, no matter what ordering of projects within J_{k^*} is used by the decreasing refined priority rule, the rule always outputs a set of matchings that matches J_{k^*} to T_{k^*} (and never to a firm in $F \setminus T_{k^*}$), with every project in J_{k^*} getting one of their best compatible firm in F . Since every SSM at profile R satisfies the above bullet points, the result is true for projects in J_{k^*} .

Note that, the set of matchings from which projects of rank lower than k^* gets to pick from in the decreasing refined priority rule, never contains a matching where this lower-ranked project is matched to a firm in T_{k^*} . Moreover, firms in T_{k^*} never prefer these lower-ranked projects to their SSM match. Next, let $k^{**} \in \{1, 2, \dots, k^* - 1\}$ be the largest threshold such that projects in $J_{k^{**}}$ are compatible for some firm in $F \setminus T_{k^*}$ at preference profile R . If $k^{**} < k^* - 1$, then all projects in $J_{k^{**}+1} \cup J_{k^{**}+2} \cup \dots \cup J_{k^*-1}$ remain unmatched at every SSM at preference profile R ; as well as at every matching that is output by any decreasing refined priority rule (by definition) at profile R . By the same arguments as above, every SSM and every decreasing refined priority rule at preference profile R , matches projects in $J_{k^{**}}$ to $T_{k^{**}}$ (and never to a firm in $(F \setminus T_{k^*}) \setminus T_{k^{**}}$), with every project in $J_{k^{**}}$ getting one of their best compatible firms in $F \setminus T_{k^*}$. Continuing in this manner we have the desired result. \square

References

- Andersson T, Ehlers L (2020) Assigning refugees to landlords in Sweden: Efficient, stable, and maximum matchings. *Scand J Econ* 122(3):937–965
- Andersson T, Kratz J (2020) Pairwise kidney exchange over the blood group barrier. *Rev Econ Stud* 87(3):1091–1133

- Åslund O, Rooth DO (2007) Do when and where matter? Initial labour market conditions and immigrant earnings. *Econ J* 117(518):422–448
- Åslund O, Östh J, Zenou Y (2010) How important is access to jobs? Old question-improved answer. *J Econ Geogr* 10(3):389–422
- Aziz H, De Keijzer B (2012) Housing markets with indifferences: a tale of two mechanisms. In: Proceedings of the AAAI conference on artificial intelligence, vol 26
- Bogomolnaia A, Deb R, Ehlers L (2005) Strategy-proof assignment on the full preference domain. *J Econ Theory* 123(2):161–186
- Damm AP (2014) Neighborhood quality and labor market outcomes: evidence from quasi-random neighborhood assignment of immigrants. *J Urban Econ* 79:139–166
- Ehlers L, Erdil A (2010) Efficient assignment respecting priorities. *J Econ Theory* 145(3):1269–1282
- Erdil A, Ergin H (2008) What's the matter with tie-breaking. Improving efficiency in school choice. *Am Econ Rev* 98:669–689
- Erdil A, Ergin H (2017) Two-sided matching with indifferences. *J Econ Theory* 171:268–292
- Gale D, Sotomayor M (1985) Ms. Machiavelli and the stable matching problem. *Am Math Mon* 92(4):261–268
- Irving RW (1994) Stable marriage and indifference. *Discrete Appl Math* 48(3):261–272
- Irving RW, Manlove DF, Scott S (2008) The stable marriage problem with master preference lists. *Discrete Appl Math* 156(15):2959–2977
- Jones W, Teytelboym A (2017) The international refugee match: a system that respects refugee's preferences and the priorities of states. *Refug Surv Q* 36(2):84–109
- Jones W, Teytelboym A (2018) The local refugee match: aligning refugees' preferences with the capacities and priorities of localities. *J Refug Stud* 31(2):152–178
- Manlove DF (2002) The structure of stable marriage with indifference. *Discrete Appl Math* 122(1–3):167–181
- Manlove D (2013) *Algorithmics of matching under preferences*, vol 2. World Scientific, Singapore
- Manlove DF, Irving RW, Iwama K, Miyazaki S, Morita Y (2002) Hard variants of stable marriage. *Theor Comput Sci* 276(1–2):261–279
- Martín L, Hainmueller J, Hangartner D (2019) Ethnic networks can foster the economic integration of refugees. *Proc Natl Acad Sci* 116(33):16280–16285
- McVitie DG, Wilson LB (1970) Stable marriage assignment for unequal sets. *BIT Numer Math* 10(3):295–309
- Pápai S (2000) Strategyproof assignment by hierarchical exchange. *Econometrica* 68(6):1403–1433
- Pycia M, Ünver MU (2017) Incentive compatible allocation and exchange of discrete resources. *Theor Econ* 12(1):287–329
- Roth AE, Sotomayor MAO (1990) Two-sided matching: a study in game-theoretic modeling and analysis. In: *Econometric society monographs*. Cambridge University Press. <https://doi.org/10.1017/CCOL052139015X>
- Roth AE, Sönmez T, Ünver MU (2005) Pairwise kidney exchange. *J Econ Theory* 125(2):151–188
- Svensson LG (1994) Queue allocation of indivisible goods. *Soc Choice Welf* 11(4):323–330
- Svensson LG (1999) Strategy-proof allocation of indivisible goods. *Social Choice Welf* 16(4):557–567

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.