Stability and Pareto optimality in Refugee Allocation Matchings

Haris Aziz$^{1,2}$, Jiayin Chen$^{3,1}$, Serge Gaspers$^{2,1}$, and Zhaohong Sun$^{2,1}$

$^1$ Data61, CSIRO, Australia
$^2$ UNSW Sydney, Australia
$^3$ Australian National University, Canberra, Australia

Abstract. We focus on the refugee matching problem—a general “two-sided matching under preferences” model with multi-dimensional feasibility constraints that was formalized by Delacretaz, Kominers, and Teytelboym (2016). We propose a taxonomy of stability concepts for the refugee matching problem; identify relations between them; and show that even for two natural weakenings of the standard stability concept, non-existence and NP-hardness results persist. We then identify several natural weaker stability concepts for which we present a polynomial-time and strategy-proof algorithm that returns stable matchings.

1 Introduction

Centralized matching markets based on the preferences of the concerned agents have been one of the success stories of algorithmic economics. These approaches have been successfully deployed in school admissions, placement of hospital residents, and centralized kidney markets [see e.g., Abdulkadiroğlu and Sönmez 2003, Roth et al. 2007, Klaus et al. 2016, Manlove 2013].

In recent years, one of the most pressing issues is the safe and timely placement of refugees in places that can host them. Often, this placement is done in an ad hoc manner where neither the preferences of the refugees nor the hosts is taken into account. For example, the host locality may prefer people who speak the same language and a refugee family may prefer a country with which they have some affinity. This calls for a centralized matching market approach to the refugee problem [see e.g., Moraga and Rapoport 2014, Teytelboym and Jones 2016].

Delacretaz et al. [2016] formalized refugee allocation as a centralized matching market design problem. The problem is more general than the traditional school choice or hospital resident setting [see e.g., Abdulkadiroğlu and Sönmez 2003] because unlike a school seat that accommodates a single student, a family can only be hosted by a locality, if it can satisfy a multi-dimensional requirement of the family that could involve services such as hospital beds, children’s day care, special medical services, etc. Thus the refugee allocation problem is a generalisation of the traditional two-sided matching problem by considering multi-dimensional feasibility constraints. Delacretaz et al. [2016] pointed out
that for the refugee allocation problem, the standard stability concept may lead to non-existence of a stable matching. Hence they focus on a weaker stability notion called quasi-stability for which they propose algorithms. Andersson and Ehlers [2016] focused on a restricted version of the refugee allocation problem with unidimensional service demand and capacity vectors but with a feature that captures language compatibility of families and hosts. For this setting, they present an algorithm that finds a stable maximum matching.

Contributions In this paper we first focus on stability in refugee allocation. We present a clear taxonomy of stability concepts for the refugee allocation problem. Two of the concepts (stability and quasi-stability) have been studied in prior work [Delacrétaz et al., 2016] whereas the others (strong stability, weak stability, stability by demand, weak stability by demand, stability by master list, weak stability by master list) are natural variants of the original two that we propose in this paper. Whereas stability is too stringent to guarantee the existence of a stable matching, quasi-stability appears to be weak since an empty matching satisfies it. We prove the logical relations between the stability concepts to unify the discussion on stability in refugee allocation (see Figure 1).

Fig. 1. Logical relations between stability concepts. An arrow from (A) to (B) denotes that stability concept (A) implies stability concept (B). The difference between ‘weak’ stable concepts and their corresponding stability definitions is that a weakly blocking pair only allows each blocking family to replace at most one family with lower preference. The stability concepts in bold guarantee the existence of a corresponding stable matching.

We start from stability and weaken it in two orthogonal directions: (1) a deviating family can replace at most one other family, and (2) if a family replaces
a set of families, then at most the same number of units of each service are used by the new family as the set of families that are replaced. For each of the weakening operations, the resulting stability notions weak stability and stability by demand still do not guarantee the existence of a stable matching. We additionally show that the problems of checking whether such matchings exist are NP-complete. Based on these negative results, we focus on the stability notion weak stability by demand which is obtained from stability by applying both weakening operations (1) and (2). This notion seems to have some merit over quasi-stability. We show that a weakly stable by demand matching is guaranteed to exist. We also propose a polynomial-time, strategy-proof algorithm for computing a weakly stable by demand matching. We also present two stability concepts that are based on the master list principle that defines a global priority over the families and could be based on factors such as the education level of the families or the urgency of their resettlement. Our algorithm (Hierarchical Family Proposing Deferred Acceptance (HFPDA)) also achieves stability based on the master list principle.

Since tailor-made algorithms for stability concepts cannot easily be extended to satisfy other feasibility constraints or objectives such as maximizing the number of refugees hosted, we take an integer/constraint programming approach to the problem. This type of approach has only recently gained traction for more restricted settings such as hospital-resident matching with couples [Biro et al., 2014]. We propose constraint programming formulations for finding stable matchings in this general setting. Our formulations provide a general framework where additional constraints can easily be placed.

We then focus on Pareto optimal allocations and show that testing weak Pareto optimality as well as Pareto optimality is coNP-complete even for single-dimensional constraints. Our results provide a formal justification to the comment by [Delacrétaz et al., 2016] that finding a Pareto improvement appears to be a challenging task.

Most of our results are summarized in Table 1.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak Stability</td>
<td>in P</td>
<td>NP-c</td>
<td>–</td>
</tr>
<tr>
<td>Stability by demand</td>
<td>in P</td>
<td>NP-c</td>
<td>–</td>
</tr>
<tr>
<td>Weak stability by demand</td>
<td>in P</td>
<td>in P</td>
<td>✓</td>
</tr>
<tr>
<td>Stability by ML</td>
<td>in P</td>
<td>in P</td>
<td>✓</td>
</tr>
<tr>
<td>Weak stability by ML</td>
<td>in P</td>
<td>in P</td>
<td>✓</td>
</tr>
<tr>
<td>Pareto optimality</td>
<td>coNP-c</td>
<td>NP-h</td>
<td>✓</td>
</tr>
<tr>
<td>Weak Pareto optimality</td>
<td>coNP-c</td>
<td>NP-h</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1. Summary of results.
2 Model

Let there be a set of refugee families $F$ and a set of localities $L$. Each family $f$ has a preference ordering $≿_f$ over the set of localities $L$ and the option of being unmatched, denoted by $\emptyset$. $l ≿_f l'$ means that $f$ prefers $l$ to $l'$ or $f$ is indifferent between $l$ and $l'$. Each locality $l$ also has a priority ordering $≿_l$ over the set of families $F \cup \{\emptyset\}$. A locality $l$ is acceptable to $f$ if $l ≿_f \emptyset$ and a family $f$ is acceptable to $l$ if $f ≿_l \emptyset$. Let $≿$ denote the preference/priority profile of all families and all localities.

Unlike classical two-sided matching problems, different types of services or multidimensional constraints need to be taken into account, e.g., each family may require several units of house rooms, school seats and job vacancies. Let $S$ denote a set of services and matrix $d$ denote the service demands of all families. Each row vector $d_f$ corresponds to the demand of family $f$ and each element $d_{sf}$ specifies the demand for service $s$ of family $f$. Let matrix $c$ denote the service capacities of all localities. Each row vector $c_l$ corresponds to the capacity of locality $l$ and each element $c_{sl}$ specifies locality $l$’s capacity of service $s$.

A refugee allocation problem consists of a tuple $\mu = (F, L, ≿, S, d, c)$. A contract $x = (f, l)$ is a family-locality pair which implies $f$ and $l$ are matched to each other. An outcome of a refugee allocation problem is a set of contracts $X \subseteq F \times (L \cup \{\emptyset\})$. Let $L_f(X)$ denote the locality matched to $f$ and $F_l(X)$ be the set of families matched to $l$ in the outcome $X$. $F(X)$ is the set of matched families and $L(X)$ is the set of matched localities under $X$. We abuse the notion slightly and consider $\emptyset$ as a null family or locality.

We assume demand or capacity vectors can be compared in this way: For any two vectors $\omega = (\omega_1, ..., \omega_k), \omega' = (\omega'_1, ..., \omega'_k), \omega \leq \omega'$ if $\forall i \in [1, k], \omega_i \leq \omega'_i$. In other words, a vector $\omega$ is smaller or equal to $\omega'$ if each element of $\omega$ is smaller or equal to corresponding element of $\omega'$.

An outcome $X$ is feasible if (i) $\forall f \in F(X), |L_f(X)| \leq 1$ and (ii) $\forall l \in L(X), \sum_{f \in F_l(X)} d_f \leq c_l$. In other words, an outcome is feasible if each family is matched with at most one locality and each locality is matched with a set of families without exceeding its capacity. A feasible outcome $X$ is individually rational if (i) $\forall f \in F, L_f(X) \geq_l \emptyset$, and (ii) $\forall l \in L, \forall f' \in F_l(X), f' \geq_l \emptyset$. That is, each family is not matched to an unacceptable locality and each locality is not matched with any unacceptable family.

A mechanism is a function $\Phi : \mu \rightarrow X$ that takes a problem $\mu$ as input and yields an outcome of contracts. A mechanism is feasible if it always produces a feasible outcome for any problem and a mechanism is strategy-proof if no family has an incentive to misreport its preference, regardless of the preferences of the other families.

3 Stability

In this section, we describe a taxonomy of several stability concepts and show which concepts can guarantee the existence of stable outcomes. Stability can
be characterized by individual rationality, elimination of justified envy and non-wastefulness in school choice [Abdulkadiroğlu and Sönmez, 2003; Roth et al., 2007]. In the context of refugee allocation, Delacrétaz et al. [2016] proposed a stability notion called quasi-stability, which is an extension of elimination of justified envy with respect to multidimensional constraints.

**Definition 1 (Quasi-stability).** A feasible outcome $X$ is quasi-stable if for each locality $l \in L$ and family $f \in F_l(X)$ with $l' \neq l$, either $l' \succ_f l$ or $f' \succ_l f$ for all $f' \in F_l(X)$.

It captures the idea that any family and locality pair cannot block an outcome if the family would have the lowest priority in the new locality, even though the new locality can provide sufficient services to accommodate it. One drawback of this concept is that it does not take feasibility into account. Consider two families $f_1, f_2$ with $d_{f_1} = (2), d_{f_2} = (1)$ and one acceptable locality $l$ with $c_l = (1)$. A feasible outcome $X = (f_2, l)$ is not quasi-stable if $f_1 \succ_l f_2$. Secondly, quasi-stability can be wasteful because even an empty outcome is quasi-stable. Wastefulness might be intolerable in practice, since it is desirable to accommodate more refugee families rather than fewer or none. In contrast to Delacrétaz et al. [2016], we consider non-wastefulness as an important part in the definition of stability.

**Definition 2 (Non-wastefulness).** A feasible matching $X$ is non-wasteful if there is no such $f, l$ that (i) $l \succ_f L_f(X)$, $f \succ_l \emptyset$ and (ii) $X \cup \{(f, l)\} \setminus \{(f, L_f(X))\}$ is feasible.

Following the idea of elimination of justified envy, we can define the notions of strong blocking pair and strong stability as follows.

**Definition 3 (Strong Stability).** Given a feasible outcome $X$, a pair $(f, l) \notin X$ is called a strong blocking pair if $l \succ_f L_f(X)$ and $\exists f' \in L_f(X), f \succ_l f'$. A feasible outcome $X$ is strongly stable if it is individually rational, non-wasteful and admits no strong blocking pair.

Although the set of strongly stable outcomes can be empty, this concept serves as the connection between quasi-stability and other stability concepts. For the rest of paper, let $X$ and $X'$ be two distinct and feasible outcomes.

**Definition 4 (Stability).** A pair $(f, l) \in X' \setminus X$ is called a blocking pair if $l \succ_f L_f(X)$ and $\forall f' \in F_l(X) \setminus F_l(X'), f \succ_l f'$. A feasible outcome $X$ is stable if it is individually rational, non-wasteful and admits no blocking pair.

In words, a pair $(f, l)$ is a blocking pair if $f$ prefers $l$ to its current assigned locality and $l$ can accommodate $f$ by removing a subset of families matched to $l$ with lower priority than $f$. Note that stability is defined in the way Delacrétaz et al. [2016] did. It also coincides with the stability concept considered in hospital-resident matching markets with couples [McDermid and Manlove, 2010].
If we impose a restriction that \( l \) can accommodate \( f \) by removing exactly one family \( f' \) matched to \( l \) with lower priority than \( f \), then we can derive the following weak stability concept.

**Definition 5 (Weak Stability).** A pair \((f,l)\) is called a weakly blocking pair if \( l \succ f \) \( L_f(X) \) and \( \forall f' \in F_l(X) \setminus F_l(X'), f \succ f' \) with \(|F_l(X) \setminus F_l(X')| \leq 1 \). A feasible outcome \( X \) is weakly stable if it is individually rational, non-wasteful and admits no weakly blocking pair.

Next, we will show that the set of weakly stable outcomes can be empty, which implies the same conclusion for stable and strongly stable outcomes.

**Proposition 1.** The set of weakly stable outcomes can be empty.

**Proof.** For the proof we reuse the following counterexample that has been considered in matching with couples [McDermid and Manlove, 2010].

\[
\begin{align*}
   d_{f_1} &= (1) & l_2 & \succ f_1, l_1 & X_0 = \{(f_1, l_2), (f_2, l_1)\} \\
   d_{f_2} &= (2) & l_1 & \succ f_2, l_2 & X_1 = \{(f_3, l_2), (f_2, l_1)\} \\
   d_{f_3} &= (1) & l_1 & \succ f_3, l_2 & X_2 = \{(f_3, l_1), (f_1, l_1)\} \\
   c_{l_1} &= (2) & f_1 & \succ l_1, f_2 \succ l_1, f_3 & X_3 = \{(f_3, l_1), (f_1, l_1)\} \\
   c_{l_2} &= (1) & f_3 & \succ l_2, f_1 \succ l_2, f_2 & X_4 = \{(f_3, l_1), (f_1, l_1)\} \\
\end{align*}
\]

There are five feasible, individually rational and non-wasteful outcomes \( X_i \) for \( i \in [1, 5] \) and each \( X_i \) is blocked via \( X_{(i+1) \mod 5} \), which completes the proof. \( \square \)

**Definition 6 (Stability by demand).** A pair \((f,l)\) is called a blocking pair by demand if \( l \succ f \) \( L_f(X) \) and \( \forall f' \in F_l(X) \setminus F_l(X'), f \succ f' \), \( d_f \leq \sum_{f'} d_{f'} \). A feasible outcome \( X \) is stable by demand if it is individually rational, non-wasteful and admits no blocking pair by demand.

\( (f,l) \) can form a blocking pair by demand if the sum of demands of each family \( f' \) matched to \( l \) with lower priority than \( f \) is greater or equal to the demand of family \( f \). This reflects the idea that \( f \) cannot replace a subset of families with lower priority if \( f \) requires more services than them.

**Proposition 2.** The set of stable matchings by demand can be empty even if preferences and priorities are strict and all families and localities are acceptable to each other.

**Proof.** Consider the following instance.

\[
\begin{align*}
   d_{f_3} &= c_{l_1} = 2 & d_{f_1} = d_{f_2} = d_{f_4} = c_{l_2} = 1 \\
   f_1 : l_1 & \succ f_1, l_2 & X_1 = \{(f_1, l_1), (f_2, l_1), (f_4, l_2)\} \\
   f_2 : l_1 & \succ f_2, l_2 & X_2 = \{(f_1, l_1), (f_4, l_1), (f_2, l_2)\} \\
   f_3 : l_1 & \succ f_3, l_2 & X_3 = \{(f_2, l_1), (f_4, l_1), (f_1, l_2)\} \\
   f_4 : l_2 & \succ f_4, l_1 & X_4 = \{(f_3, l_1), (f_4, l_2)\} \\
   l_1 : f_4 & \succ f_3, f_3 \succ f_1, f_1 \succ f_1, f_2 & X_5 = \{(f_3, l_1), (f_1, l_2)\} \\
   l_2 : f_1 & \succ f_2, f_4 \succ f_2, f_2 \succ f_2, f_3 & X_6 = \{(f_3, l_1), (f_2, l_2)\} \\
\end{align*}
\]
Definition 7 (Weak stability by demand). A pair \((f, l)\) is called a weak blocking pair by demand if \(l \succ_f L_f(X)\) and \(\forall f' \in F_l(X) \setminus F_l(X'), f \succ_l f'\), \(d_f \leq \sum_{f'} d_{f'}\) with \(|F_l(X) \setminus F_l(X')| \leq 1\). A feasible outcome \(X\) is weakly stable by demand if it is individually rational, non-wasteful and admits no weak blocking pair by demand.

The distinction from stability by demand is that a weak blocking pair by demand requires that the more preferred family \(f\) has no more demand than the less preferred family \(f'\) in all services. Unlike the non-existence result regarding stability by demand, we will prove the following result in the next section.

Proposition 3. The set of weakly stable outcomes by demand is non-empty.

Next we will define stability with a master list \(\mathcal{ML}\), which is a common preference over families imposed by the policy maker to all localities. For example, it can be decided by the amount of time the families have been in the market for medical school graduates. Similar master lists have been used in practice, for example in the Scottish entry-labour market for medical school graduates.\(^4\) Let \(f \preceq_{\mathcal{ML}} f'\) denote that \(f\) has higher \(\mathcal{ML}\) priority than \(f'\) or they have the same \(\mathcal{ML}\) priority. We will assume that \(\mathcal{ML}\) is a weak ordering.

Definition 8 (Stability by \(\mathcal{ML}\)). Given a feasible matching \(X\) and a master list \(\mathcal{ML}\), a pair \((f, l)\) is called a blocking pair by \(\mathcal{ML}\) if \(l \succ_f L_f(X)\) and \(\forall f' \in F_l(X) \setminus F_l(X'), f \succ_l f'\) and \(f \preceq_{\mathcal{ML}} f'\). A feasible outcome \(X\) is stable by \(\mathcal{ML}\) if it is individually rational, non-wasteful and admits no blocking pair by \(\mathcal{ML}\).

The pair \((f, l)\) can form a blocking pair by \(\mathcal{ML}\) if \(l\) can remove a subset of matched families with lower preference and lower master list priority than \(f\) to accommodate it. Similarly, we can define weak stability as follows.

Definition 9 (Weak stability by \(\mathcal{ML}\)). Given a feasible matching \(X\) and a master list \(\mathcal{ML}\), a pair \((f, l)\) is called a weak blocking pair by \(\mathcal{ML}\) if \(l \succ_f L_f(X)\) and \(\forall f' \in F_l(X) \setminus F_l(X'), f \succ_l f'\) and \(f \preceq_{\mathcal{ML}} f'\) with \(|F_l(X) \setminus F_l(X')| \leq 1\). A feasible outcome \(X\) is weakly stable by \(\mathcal{ML}\) if it is individually rational, non-wasteful and admits no weak blocking pair by \(\mathcal{ML}\).

Remark 1. If all families have the same priority in the master list \(\mathcal{ML}\), then (weak) stability by \(\mathcal{ML}\) is the same as (weak) stability and a (weakly) stable

by \(\mathcal{ML}\) outcome is not guaranteed to exist. Therefore, we make the following assumption on the master lists throughout the paper: every two families in the same \(\mathcal{ML}\)-equivalence class have the same demand vectors \((f \sim_{\mathcal{ML}} f' \Rightarrow d_f = d_{f'})\). A special case is when the master list gives a strict priority over the families.

**Proposition 4.** Given a strict master list \(\mathcal{ML}\), the set of stable matchings by \(\mathcal{ML}\) is non-empty. Serial dictatorship always return a stable outcome by \(\mathcal{ML}\).

When the master list \(\mathcal{ML}\) is strict, we can always find a stable outcome by \(\mathcal{ML}\) through serial dictatorship. However, that algorithm completely ignores the preference of localities and yields a Pareto optimal outcome rather than a stable outcome. We will show that the restriction on master lists given in Remark 1 above is sufficient to guarantee stable outcomes in the next section.

## 4 Algorithms

First we consider the special case where all families have the same demand vectors and show that the problem reduces to the classical Hospital-Resident (HR) problem [Roth and Sotomayor, 1992].

### 4.1 Identical Demand Vectors

When all the families have the same demands, the central stability concepts considered in the paper coincide.

**Proposition 5.** If \(\forall f, f' \in F, d_f = d_{f'}\), then weak stability, stability, strong stability, weak stability by demand, and stability by demand are all equivalent.

Furthermore, under identical demand vectors, the refugee problem is equivalent to the classical Hospital-Resident (HR) problem. A refugee allocation problem can be transformed into a HR problem in which the families correspond to the residents, localities correspond to hospitals, and the corresponding preferences and priorities are unchanged. We modify the demand vectors to a single-dimensional vector with one unit of demand and set the capacity of each locality to the maximum number of families that the locality can accommodate. Then we have a one-to-one mapping from a refugee allocation problem to a HR problem. A stable matching for the HR instance gives a matching that is stable for the refugee allocation problem. As a consequence, results for the HR setting translate to this special case of refugee allocation. In particular, a stable matching can be computed by first breaking all indifferent preferences and priorities and then running the classic DA (deferred acceptance) algorithm [Roth, 2008]. We will refer to \(\text{FPDA} (\text{Family Proposing Deferred Acceptance})\) as the algorithm that transforms the refugee allocation setting with identical demands to HR and then runs DA to obtain a stable matching. The Rural Hospitals Theorem also translates to our setting.
Corollary 1 (Rural Hospitals Theorem). When all families have the same demand vectors, and the preferences and priorities are strict, then the following statements hold:

(i) The same families are assigned in all stable matchings;
(ii) Each locality is assigned the same number of families in all stable matchings;
(iii) Any locality that is under-subscribed in one stable matching is assigned exactly the same set of families in all stable matchings.

4.2 General case

Since families with different demand vectors need to be treated differently, we design the following Hierarchical Family Proposing Deferred Acceptance (HFPDA) algorithm by partitioning families into different groups.

Algorithm 1 Hierarchical Family Proposing Deferred Acceptance (HFPDA) for a given master list $ML$

1: Use $ML$ to divide the families into groups $H_1, \ldots, H_n$ in the order of $ML$ where each group $H_i$ forms an indifference equivalence class with respect to $ML$. {In each group, the families have the same demand vector.}
2: $k = 1$ {Start with the group with highest $ML$.}
3: for $k = 1, 2, \ldots, n$ do
4: Run FPDA (Family Proposing Deferred Acceptance) on $H_k$ and the localities.
5: Remove all families in $H_k$ from the market and update the corresponding capacities of each locality
6: $k \leftarrow k + 1$
7: end for

Proposition 6. HFPDA is polynomial-time, strategy-proof and always returns a stable outcome by $ML$.

Proof. Consider a refugee problem with $n$ groups $H_1, \ldots, H_n$ such that $\forall f \in H_i, \forall g \in H_j, f \succ_M g$ if $i < j$.

Strategy-proofness For any family in $H_1$, they won’t be affected by the preferences of any family from $H_i$ with $i \geq 2$. So we can consider $H_1$ as an independent market and HFPDA is strategy-proof for families. Similarly, $H_2$ can be considered as an independent market as well after removing all families from $H_1$. By induction, we will see each $H_i$ is strategy-proof for families.

Stability Give an outcome yielded by HFPDA, assume there exists a blocking pair $(f, c)$ by $ML$ in $H_1$. Since all families in $H_1$ have the same $ML$ priority, then there exists $f'$ such that $f' \in F_c(X)$ and $f \succ_c f'$. While $f$ must propose to $c$ if he is assigned with a less preferred locality under HFPDA, thus there is no blocking pair.
After removing all agents from $H_1$ and updating the capacities for all local- 


ties, any family from $H_2$ has the highest $\mathcal{ML}$ priority among all remaining 


families and we can derive the same conclusion that there is no blocking pair in 


$H_2$. By induction, we will see there will be no blocking pair in any $H_i$.

In group $H_1$, the outcome is stable by $\mathcal{ML}$ and no family has an incentive 


to misreport its preference. We can derive the same conclusion for group $H_2$ by 


removing $H_1$ from the market. By induction, we can prove Proposition 6.

We call a master list $\succeq L$ demand-respecting if $f \succeq f'$ if and only if $d_f \leq d_{f'}$. In such an ordering families with the same demand are placed in the same equivalence class. If we determine a $\mathcal{ML}$ by ranking all groups in ascending order in terms of demand vectors and arbitrarily rank incomparable groups, then stability by $\mathcal{ML}$ is the same as weak stability by demand.

**Proposition 7.** For some demand respecting master list, HFPDA returns a weakly stable outcome by demand.

5 Complexity of stability

We presented the polynomial-time HFPDA algorithm that returns a matching 


that is stable for three of the stability notions that we proposed. In this section 


we present a complete picture of the complexity of testing as well as finding 


stable matchings.

5.1 Testing Stability

Define $\text{sum}_l(X) = \sum_{f \in F_l(X)} d_f$ as the sum of the demand vectors of all families 


matched to $l$. Define $F_{\succ l}^f(X)$ as the set of families such that $f' \in F_l(X)$ and 


$f \succ_l f'$. Define $F_{\prec l}^f(X)$ as the set of families such that $f' \in F_l(X)$ and $f' \succ_l f$.

Now we can describe the polynomial algorithm to check whether a given feasible outcome $X$ satisfies different stability requirements. The idea is to check whether the outcome is wasteful and if not, then check whether there exists a blocking pair of different forms. If no blocking pair is detected, the outcome is stable for the corresponding stability concept.

For each family $f \in F$ and each locality $l$ with $l \succ_f L_f(X)$:

1) Check whether $\text{sum}_l(X) + d_f \leq c_l$ holds. If so, then $X$ is not non-wasteful, 


which implies that $X$ does not satisfy any of our stability concepts.

2) Otherwise, check whether $X$ satisfies different fairness concepts as follows:

**Stability** Check whether the following inequality holds. $d_f + \sum_{f' \in F_{\succ l}^f(X)} d_{f'} \leq c_l$. If so, then $X$ is not stable.

**Stability by demand** Check whether the following inequality holds. $d_f \leq \sum_{f' \in F_{\prec l}^f(X)} d_{f'}$ If so, then $X$ is not stable by demand.
**Stability by ML** Check whether the following inequality holds.
\[ df + \sum_{f' \in F^{\succ f}(X)} d_{f'} + \sum_{f'' \in F^{\succ f}(X) \land f' \succ f''} f'' \leq c_l \]
If so, then \( X \) is not stable by ML.

**Weak stability** Check whether there exists any family \( f' \in F^{\succ f}(X) \) that satisfies \( \sum_l(X) - df' + df \leq c_l \). If so, then \( X \) is not weakly stable.

**Weak stability by demand** Check whether there exists any family \( f' \in F^{\succ f}(X) \) that satisfies \( df \leq df' \). If so, then \( X \) is not weakly stable by demand.

**Weak stability by ML** Check whether there exists any family \( f' \in F^{\succ f}(X) \land f' \succ f \) that satisfies \( \sum_l(X) - df' + df \leq c_l \) and \( . \) If so, then \( X \) is not weakly stable by ML.

### 5.2 Decide if an instance admits a stable matching

Since checking whether a given outcome is stable in all definitions is in P, deciding if an instance admits a stable matching is in NP. [McDermid and Manlove 2010] proved that, even with one-dimensional demands and capacities consisting of 1’s and 2’s and preference lists of length at most 3, it is NP-complete to decide whether a stable matching exists (Theorem 3.7). We present complexity results concerning weak stability and stability by demand by reductions from the following strongly NP-complete problem.

**3-Partition**

*Input:* A finite set \( E = \{e_1, \ldots, e_{3n}\} \) of 3n elements, a bound \( W \) and integer weight \( w(e_j) \) for each \( e_j \in E \) such that \( \frac{W}{3} < w(e_j) < \frac{W}{2} \) and \( w(E) = \sum_{j=1}^{3n} w(e_j) = nW \).

*Question:* Can \( E \) be partitioned into \( n \) disjoint sets \( E_1, \ldots, E_n \) with weight \( w(E_i) = W \) for all \( i \in [n] \)?

**Proposition 8.** Checking whether a weakly stable matching exists is NP-complete even if there are single-dimensional demands and capacities.

**Proof.** We reduce from 3-Partition.

We can produce 3n gadgets where the \( i \)-th gadget looks as follows:

\[
\begin{align*}
  f_1 &: (1w_i) : l_2 \succ f_1 l_1 \\
  f_2 &: (1w_i) : l_1 \succ f_2 l_2 \\
  f_3 &: (2w_i) : l_1 \\
\end{align*}
\]

\[
\begin{align*}
  f_1 &: (2w_i) : f_1 \succ_l f_1 f_3 \succ_l f_1 f_2 \\
  f_2 &: (1w_i) : f_2 \succ_l f_2 f_1 \\
  f_3 &: (2w_i) : f_3 \\
\end{align*}
\]

Note that if \( f_1 \) is not present, then the following is a stable matching:
\[ \{(f_3, l_1), (f_4, l_2)\} \]

From each gadget, we can take the corresponding family \( f_1 \) that has highest preference for \( n \) other localities \( d_1, \ldots, d_n \), each of capacity \( W \). The \( f_1 \) families have a total demand of \( nW \).
The other families in the gadget are not interested or have least preference for the new localities. The new localities most prefer the \( f_1 \) type of families and then any other families. The new localities are completely indifferent among the \( f_1 \) families.

Then there exists a stable matching if and only if the 3-Partition instance is a yes-instance.

\[ \square \]

The reduction above also gives us a simpler proof for the statement that checking whether there exists a stable matching is NP-complete.

**Proposition 9.** Checking whether a stable by demand matching exists is NP-complete even if there are single-dimensional demands and capacities.

**Proof.** We can produce \( 3n \) gadgets as follows as follows. The \( i \)-th gadget looks as follows:

\[
\begin{align*}
  f_1 : (1w_i) : l_1 \succ f_1 l_2 \\
  f_2 : (1w_i) : l_1 \succ f_2 l_2 \\
  f_3 : (2w_i) : l_1 \succ f_3 l_2 \\
  f_4 : (1w_i) : l_2 \succ f_4 l_1
\end{align*}
\]

\[
\begin{align*}
  l_1 : (2w_i) : f_4 \succ l_1, f_3 \succ l_1, f_1 \succ l_1, f_2 \\
  l_2 : (1w_i) : f_1 \succ l_2, f_4 \succ l_2, f_2 \succ l_2, f_3
\end{align*}
\]

Note that if \( f_1 \) is not present, then the following is a stable matching: \( \{(f_3, l_1), (f_4, l_2)\} \)

From each gadget, we can take the corresponding family \( f_1 \) which has highest priority for \( n \) other localities \( d_1, \ldots, d_n \) each of capacity \( W \). The \( f_1 \) families have a total demand of \( nW \). The other families in the gadget are not interested or have least preference for the new localities. The new localities most prefer the \( f_1 \) type of families and then any other families. The new localities are completely indifferent among the \( f_1 \) families.

Then there exists a stable matching if and only if the 3-Partition instance is a yes instance.

\[ \square \]

## 6 Constraint programming formulation

[Delacrétaz et al. 2016] presented an Integer Programming (IP) formulation to accommodate the maximum number of refugees. On the other hand, [Delacrétaz et al. 2016] also presented tailor-made algorithms to find stable matchings for the two concepts stability and quasi-stability. In this section, we combine the two orthogonal approaches and make a case that it is desirable to capture stability constraints and incorporate them into an integer or constraint program. By doing so, one can maximize other objectives such as accommodating a maximum number of refugees while maintaining some form of stability, especially if the stability constraint does not affect or significantly affect the number of refugees accommodated. If a stable matching exists and does not lead to a significant enough decrease in the number of people hosted, then we can select that matching. Otherwise, we can gradually replace the constraints for stronger stability constraints and incorporate them into an integer or constraint program.
notion by constraints for a weaker stability notion until we are satisfied with the number of people who are matched. Such an approach also makes it possible to additionally impose other feasibility constraints not considered in the paper.

6.1 Matching with a maximum number of refugees

First we reproduce the integer programming formulation for matching maximum number of refugees [Delacrétaz et al., 2016]. Let \( n_f \) denote the number of family members of \( f \) and \( x(f, \ell) \) denote a function that \( x(f, \ell) = 1 \) if \( f \) is matched to \( \ell \) and otherwise \( x(f, \ell) = 0 \). Finding a matching with a maximum number of refugees is equivalent to solving the following optimization problem.

\[
\max \sum_{f \in F} \sum_{\ell \in L} n_f \times x(f, \ell)
\]

subject to

\[
\sum_{f \in F} d_s f \times x(f, \ell) \leq c_s \quad \forall s, \forall \ell \in L
\]

\[
\sum_{f \in F} x(f, \ell) \leq 1
\]

\[
x(f, \ell) = \{0, 1\}
\]

6.2 Capturing Stability

Below we formulate inequalities capturing each of the stability concepts.

**Stability** For each \((f, \ell)\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succeq f, \ell} x(f, \ell') \times c_s + \sum_{f' \succ f} x(f', \ell) \times d^s_{f'} + d^s_f > c_s.
\]

**Weak stability** For each \((f, \ell)\) and any \(f''\) such that \(f \succ f''\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succeq f, \ell} x(f, \ell') \times c_s + \sum_{f' \in F_l(X)} x(f', \ell) \times d^s_{f'} - x(f'', \ell) \times d^s_{f''} + d^s_f > c_s.
\]

**Stability by demand** For each \((f, \ell)\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succeq f, \ell} x(f, \ell') \times c_s + d^s_f - \sum_{f' \succ f} x(f', \ell) \times d^s_{f'} > 0.
\]
Weak stability by demand For each \((f, \ell)\) and any \(f'\) such that \(f \succ \ell f'\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succ_{\ell} \ell} x(f, \ell') \times c_s^f + d_s^f - d_s^{f'} > 0. 
\tag{5}
\]

Stability by \(\mathcal{ML}\) For each \((f, \ell)\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succ_{\ell} \ell} x(f, \ell') \times c_s^f + d_s^f - \sum_{f' \succ \ell, f' \succ_{\mathcal{ML}} f} x(f', \ell) \times d_s^{f'} > 0. 
\tag{6}
\]

Weak stability by \(\mathcal{ML}\) For each \((f, \ell)\) and \(f''\) such that \(f \succ \ell f''\) and \(f'' \succ_{\mathcal{ML}} f\), the following constraint is satisfied for at least one service \(s\):

\[
\sum_{\ell' \succ_{\ell} \ell} x(f, \ell') \times c_s^f + \sum_{f' \in F_\ell(X)} x(f', \ell) \times d_s^{f'} - x(f'', \ell) \times d_s^{f''} + d_s^f > c_s^\ell. 
\tag{7}
\]

If there is exactly one service, the stability constraints can be achieved by a polynomial number of IP constraints. If there are more than one service, one can use a constraint program that requires at least one constraint from each family of constraints to be satisfied. If we want to write an integer program for any number of services, then each disjunction of constraints can be modelled by a logarithmic number of binary variables [Vielma and Nemhauser 2008].

7 Pareto optimality

Under strict preferences, computing a Pareto optimal allocation is easy via serial dictatorship: a strict priority ordering over the families is set and then each family takes a place in the best possible locality that can accommodate it. However, finding Pareto improvements over an existing allocation seems challenging even under strict preferences. Delacrétaz et al. [2016] mention that “Finding all Pareto-improving exchanges among sets of families in general—even between two localities—would mean potentially looking at all subsets of the families in these localities and is therefore computationally intractable.” However, just because a naive algorithm for solving a problem requires exponential time does not mean that the problem itself is intractable. In what follows, we give formal argument that finding a Pareto improvement is intractable.

**Proposition 10.** Testing weak Pareto optimality is strongly coNP-complete even if families have strict preferences over localities, each family is acceptable, and there is only type of service.
Proof. To show that the complement of testing weakly Pareto optimality (TWPO)
is in NP, for a given instance \(X\), we can guess another outcome \(X'\) as a certificate
and check whether all families strictly prefer \(X'\) to \(X\) in polynomial time.

We now prove that 3-Partition \(\leq_p\) TWPO. The reduction algorithm begins
with an instance of 3-Partition. Let \(E = \{e_1, \ldots, e_{3n}\}\) be a set of \(3n\) elements
and each \(e_j \in E\) has a integer weight \(w(e_j)\) such that \(\frac{W}{3} < w(e_j) < \frac{W}{2}\) and
\(w(E) = \sum_{j=1}^{3n} w(e_j) = nW\).

Assume all families have strict preferences over localities and each family
is acceptable and there is only one type of service. We construct the refugee
allocation problem as follows. Let \(F = \{f^*, f_1, \ldots, f_{3n}\}\) be the set of families
and \(L = \{\ell^*, \ell', l_1, \ldots, c_n\}\) be the set of localities. \(f^*\) demands \(nW\) units while
each \(f_i\) requires \(w(e_i)\) units. The capacity of both \(\ell^*\) and \(\ell'\) is \(nW\) and \(W\) for each
\(c_i\). \(f^*\) has \(\ell^*\) as the second most preferred locality and \(\ell'\) as the most preferred
locality. Each \(f_i\) has \(\ell'\) as the second least preferred locality and \(\ell^*\) as the least
preferred locality. \(X\) is the allocation in which \(f^*\) is matched to \(\ell^*\) and all the
other families are matched to \(\ell'\). The construction can be done in polynomial
time.

We now show that \(E\) can be partitioned into \(n\) disjoint sets \(E_1, \ldots, E_n\) and
weight \(w(E_i) = W\) for all \(i \in [n]\) if and only if \(X\) is not weakly Pareto optimal.
First, suppose that \(E\) has a 3-Partition and let it be \(E_1 = \{e_1, e_2, e_3\}, E_2 = \{e_4, e_5, e_6\}, \ldots, E_n = \{e_{3n-2}, e_{3n-1}, e_{3n}\}\). Then consider another outcome \(X'\) in
which \(f_1, f_2, f_3\) are matched with \(l_1, \ldots, f_{3n-2}, f_{3n-1}, f_{3n}\) are matched with \(c_n\)
and \(f^*\) is matched with \(\ell^*\). Then \(X'\) weakly Pareto dominates \(X\) because every
family is strictly better off.

Now suppose that \(X\) is not weakly Pareto optimal. Then there will be an-
other outcome \(X'\) that weakly Pareto dominates \(X\). Assume in \(X', f_1, f_2, f_3\) are
matched with \(l_1, \ldots, f_{3n-2}, f_{3n-1}, f_{3n}\) are matched with \(c_n\) and \(f^*\) is matched with \(\ell^*\). Then we could find a 3-Partition for \(E\) in which \(E_1 = \{e_1, e_2, e_3\}, E_2 = \{e_4, e_5, e_6\}, \ldots, E_n = \{e_{3n-2}, e_{3n-1}, e_{3n}\}\), which completes the proof. \(\square\)

Proposition 11. Testing Pareto optimality is strongly coNP-complete even if
each family is acceptable, and there is only one type of service.

Proof. The proof is similar as the one for Proposition[10] with the change that \(f^*\)
has \(\ell'\) as the second most preferred locality and \(\ell^*\) as the most preferred locality.
Each \(f_i\) is indifferent among all localities except \(\ell^*\) which it prefers the least. \(\square\)

Acknowledgments

Haris Aziz thanks Alex Teytelboym for introducing him to issues related to
the refugee allocation problem. Aziz is supported by a Julius Career Award.
Serge Gaspers is the recipient of an Australian Research Council (ARC) Future
Fellowship (project number FT140100048) and he also acknowledges support
under the ARC’s Discovery Projects funding scheme (DP150101134).
Bibliography


