

Enabling Trade-offs in Machine Learning-based Matching for Refugee Resettlement*

Nils Olberg
University of Zurich
olberg@ifi.uzh.ch

Sven Seuken
University of Zurich
seuken@ifi.uzh.ch

First version: May 5, 2019
This version: November 10, 2019

Abstract

The Swiss State Secretariat for Migration recently announced a pilot project for a machine learning-based assignment process for refugee resettlement. This approach has the potential to substantially increase the overall employment rate of refugees in Switzerland. However, the currently proposed method ignores families' preferences. In this paper, we build on this prior work and propose two matching mechanisms that additionally take families' preferences over locations into account. The first mechanism is strategyproof while the second is not but achieves higher family welfare. Importantly, we parameterize both mechanisms, giving placement officers precise control how to trade off family welfare against overall employment success. Preliminary simulations on synthetic data show that both mechanisms can significantly increase family welfare even with only a small loss on the overall employment rate of refugees.

Keywords: Refugee Resettlement, Matching Markets, Machine Learning, Linear Programming

1 Introduction

Refugee families seeking shelter in Switzerland are currently assigned randomly to one of the 26 Swiss cantons according to a distribution key unless their asylum claim is rejected within three months after arrival. This practice ignores potential synergies between refugees and cantons, suggesting suboptimal integration outcomes. Bansak et al. (2018) proposed a machine learning-based algorithm for family placement that aims

*A large part of this paper overlaps with Acharya et al. (2019). We were initially not aware of their work, and our results were developed independently from them. After we were pointed towards their working paper, we decided to stop working on this project.

to optimize the overall employment rate of refugees. The algorithm, which could increase the employment rate of refugees in Switzerland by about 73%, works in three stages. In the first stage, a machine learning model predicts, for each refugee-location pair, the probability that a refugee will find employment at the corresponding location. In the second stage, these individual refugee probabilities are transformed to a family-level metric. In the last stage, solving an optimization problem provides the final assignment. The Swiss State Secretariat for Migration (SEM), which is responsible for assigning refugees to cantons, has recently announced to test this machine learning-based assignment process in a pilot project.

Although finding employment is important for successful integration, there are many reasons why refugee resettlement procedures should also enable families to express their individual preferences over resettlement locations, even if these preferences are potentially in conflict with employment success. Delacrétaz et al. (2016) argue that refugee families themselves know best where they are likely to thrive. Jones and Teytelboym (2017a) point out that resettlement systems which ignore families’ preferences are disempowering for refugees, and suggest that giving them a say in the assignment process could increase their well-being. As described by Jones and Teytelboym (2017b), ignoring families’ individual preferences has even caused families seeking shelter in Finland to cancel their asylum applications. This is tragic, considering the reasons why families abandon their home countries.

To address this issue, we propose two mechanisms that build on the machine learning-based approach by Bansak et al. (2018), but now also take family welfare into account (in terms of the refugees’ reported preferences). We show that our first mechanism, the *constrained random serial dictatorship mechanism (CRSD)* is family-strategyproof, i.e., families cannot benefit from misreporting their true preferences. The second mechanism, the *constrained rank value mechanism (CRV)*, is not strategyproof, but is always weakly better in terms of family welfare. While prior work has already considered incorporating refugee preferences into the resettlement process (e.g., Delacrétaz et al. (2016)) the primary innovation of our paper is that the mechanisms we design are parameterized, giving placement officers precise control over the impact of families’ preferences on the final matching. To the best of our knowledge, prior mechanisms for refugee resettlement were either optimization-based or preference-based. Our mechanisms combine both concepts. In our simulations (using synthetic ”proof-of-concept” data), we show that, for both mechanisms, the trade-off parameter can be chosen in such a way that family welfare is significantly improved (compared to the one-sided assignment mechanism) with only a minimal loss on overall employment success.

2 Preliminaries

We consider a set of refugee families F and a set of locations L . A distribution key (which typically depends on the population of each location) determines a quota $q_j \in \mathbb{N}$ for each location $j \in L$, which is the number of families that location j is obligated to host. We assume that $q = (q_j)_{j \in L}$ is chosen in such a way that

each family can be assigned to exactly one location, i.e., it holds that $\sum_{j \in L} q_j = |F|$. Location j derives its preference for family i from a predicted integration success $\pi_{ij} \in [0, 1]$. We assume that the predicted values for integration success are provided to us by machine learning models trained on historic resettlement data. As is done in Bansak et al. (2018) and Trapp et al. (2018), we assume that π_{ij} corresponds to the predicted probability that family i will find employment at location j . For now, we assume that family $i \in F$ has a weak preference order \succeq_i over locations in L . In the appendix we present an extension for the case of incomplete preferences. Let \succeq_{ij} denote the position of location $j \in L$ in preference order \succeq_i . A matching is a mapping $\mu : F \rightarrow L \cup \{\emptyset\}$, and $F(\mu) = \{i \in F \mid \mu(i) \neq \emptyset\} \subseteq F$ is the set of assigned families under matching μ . Further, let $F_j(\mu) = \{i \in F \mid \mu(i) = j\}$ denote the set of all families matched to location j under μ . A matching is feasible if $|F_j(\mu)| = q_j$ for all $j \in L$.

We call the expected number of successfully integrated families, i.e., $z(\mu) = \sum_{i \in F} \pi_{i\mu(i)}$, the *government objective* of μ . A feasible matching μ^* is *government-optimal* if $\mu^* \in \arg \max_{\mu} \{z(\mu)\}$. Our goal is to find a feasible matching μ that (1) maximizes family welfare in terms of reported preferences $\succeq = (\succeq_i)_{i \in F}$ and at the same time (2) ensures that $z(\mu)$ is within a factor of α of $z(\mu^*)$ for a previously chosen $\alpha \in [0, 1]$.

Definition 1. Let $\alpha \in [0, 1]$. A feasible matching μ is an α -approximation of the government-optimal matching if $z(\mu) \geq \alpha z^*$, where $z^* = \max_{\mu} \{z(\mu)\}$.

We measure family welfare using the average rank $\rho(\mu) = \frac{1}{|F|} \sum_{i \in F} \succeq_{i\mu(i)}$ of matching μ . Additionally, we use the *cumulative rank distribution* $\Delta(\mu)$, which provides more detailed information about the goodness of matching μ in terms of family welfare than $\rho(\mu)$.

Definition 2. Let $\delta_k(\mu)$ denote the number of families that are assigned to their k -th choice under matching μ . The cumulative rank distribution of μ is a vector $\Delta(\mu) \in \mathbb{N}^{|L|}$, where the k -th entry $\Delta_k(\mu)$ denotes the number of families that are matched to their k -th or better choice under μ , i.e., $\Delta_k(\mu) = \sum_{k'=1}^k \delta_{k'}(\mu)$. It holds that $\Delta_{|L|}(\mu) = |F_{\mu}^*|$.

Remark 3. Requiring refugees to report a preference order over locations (or even assuming that they have preformed preferences) may be problematic in some countries. For example, there can be hundreds of potential resettlement locations for a family seeking shelter in the United Kingdom (Jones and Teytelboym (2017a)). It is hard to imagine that refugees can come up with a complete preference order over that many options. In Switzerland, however, refugees can only be matched to one out of 26 cantons. Further, note that we are not requiring families to have complete preference orders over locations (see appendix). In practice, families could be provided with information on cantons after arrival, which could help them to form at least incomplete preference orders.

Remark 4. We agree with Jones and Teytelboym (2017a) that efficiency and strategyproofness are more

Algorithm 1 Constrained Random Serial Dictatorship (CRSD)

Input: $F, L, \succeq, \pi, b, \alpha$

```
1:  $\mu(i) := \emptyset$  for all  $i \in F$ .
2: Compute the objective value  $z^*$  of an optimal solution to  $\text{IP}_\mu(3.1)$ .
3:  $Q := F$ 
4: while  $Q \neq \emptyset$  do
5:   Remove randomly chosen  $i \in Q$  from  $Q$ .
6:   while  $\succeq_i \neq \emptyset$  do
7:     Let  $j$  denote  $i$ 's current top choice in  $\succeq_i$  and remove  $j$  from  $\succeq_i$ .
8:     if  $|F_j(\mu)| < q_j$  then
9:        $\mu' := \mu; \mu'(i) := j$ 
10:      Solve  $\text{IP}_{\mu'}(3.1)$  and let  $z'$  denote the objective value of the solution.
11:      if  $z' \geq \alpha z^*$  then
12:         $\mu(i) := j$ 
13:        break
14:      end if
15:    end if
16:  end while
17: end while
Output:  $\mu$ 
```

important than stability in the context of refugee resettlement mechanisms. Therefore, we will not analyze our mechanisms in terms of stability in this paper.

3 Mechanisms

In the following subsections we describe two mechanisms that consider families' preferences. The first mechanism, the *constrained random serial dictatorship mechanism* (CRSD), is family-strategyproof while the second, the *constrained rank value mechanism* (CRV), is not. However, CRV will usually achieve higher family welfare than CRSD. Both mechanisms allow placement officers to choose a parameter $\alpha \in [0, 1]$, which ensures that the computed matching is an α -approximation of the government-optimal matching.

3.1 Constrained Random Serial Dictatorship Mechanism

The constrained random serial dictatorship mechanism (CRSD) is a constrained version of the well-known random serial dictatorship mechanism. The general idea of CRSD is to let a family only choose their match from the set of remaining locations if it can be guaranteed that an α -approximation of the government-optimal matching is still achievable. Algorithm 1 provides a detailed description of the mechanism.

Initially, the algorithm computes the objective value $z(\mu^*)$ of a government-optimal solution μ^* . Afterwards, it initializes an (infeasible) empty matching μ . Families are then sorted in a random order and

processed sequentially. When it is family i 's turn to choose among the remaining locations, i is only assigned to j under μ if it can be guaranteed that an α -approximation of μ^* is still achievable.

The integer program IP(3.1) has to be solved in Line 2 and Line 10 of the mechanism. IP(3.1) ensures that the intermediate matching μ is preserved. If the objective value of an optimal solution to IP(3.1) is smaller than $\alpha z(\mu^*)$, then i is not allowed to be matched to j .

$$\begin{aligned}
\text{IP}_\mu(3.1): \quad & \text{maximize} \quad \sum_{i \in F} \sum_{j \in L} \pi_{ij} x_{ij} & (1) \\
& \text{subject to} \quad \sum_{i \in F} x_{ij} = q_j & \forall j \in L \quad (2) \\
& \quad \quad \quad \sum_{j \in L} x_{ij} = 1 & \forall i \in F \quad (3) \\
& \quad \quad \quad x_{i\mu(i)} = 1 & \forall i \in F(\mu) \quad (4) \\
& \quad \quad \quad x_{ij} \in \{0, 1\} & \forall i \in F, \forall j \in L \quad (5)
\end{aligned}$$

An optimal solution of IP(3.1) induces a matching that maximizes the overall predicted employment rate. Variable x_{ij} indicates whether family i will be assigned to location j . Constraints (2) ensure that every location hosts as many families as required for a feasible matching. Constraints (3) guarantee that every family is assigned to exactly one location. Constraints (4) preserve the intermediate matching μ .

Proposition 5. *CRSD is family-strategyproof, and the matching computed by CRSD is an α -approximation of the government-optimal matching.*

Proof. Because of the feasibility check in Line 10, we know that at each step of the algorithm there exists a feasible α -approximation μ' that preserves the intermediate matching μ . Thus, the final matching is an α -approximation of the government-optimal matching. Further, CRSD is family-strategyproof because family i cannot influence which locations will remain available to it once it is i 's turn to choose, and by stating its true preferences it is guaranteed that the best among the remaining locations is chosen. \square

Remark 6 (Computational Complexity). *In its original formulation, IP(3.1) boils down to a maximum-weight matching problem. Finding a solution to this problem can be done in polynomial time, e.g., using the Hungarian method (Kuhn (1955)). As we will see in Section 3.3, additional constraints (e.g., service constraints, capacity constraints, etc.) could easily be integrated in the CRSD mechanism. However, this transforms the maximum-weight matching problem into a NP-hard problem, which can significantly increase the overall runtime of the algorithm.*

3.2 Constrained Rank Value Mechanism

Before introducing the constrained rank value mechanism (CRV), we need to establish the concept of a rank value function. Along the lines of Featherstone (2014), we use rank value functions to assign values between 0 and 1 to positions in preference orders.

Definition 7. A rank value function is a mapping $v : \{1, \dots, |L|\} \rightarrow [0, 1]$ that is monotonically decreasing.

Definition 8. Given a rank value function v , a set of families F , a set of locations L , quotas q , preference orders \succeq , predicted employment probabilities π , and a lower bound γ , the constrained maximum rank value problem (CMRV) is to find a feasible matching μ that maximizes $\sum_{i \in F} \sum_{j \in L} v(\succeq_{ij}) x_{ij}$, such that $z(\mu) \geq \gamma$.

By solving an instance of CMRV with $\gamma = \alpha z(\mu^*)$, we can find an α -approximation of the government-optimal matching that maximizes family welfare in terms of v .

Proposition 9. The Constrained Maximum Rank Value problem is NP-hard.

Proof. Suppose that we are given an instance $I = (N, w, a, b)$ of the Knapsack problem, where $N = \{1, \dots, n\}$ is the set of items, $w_i \in \mathbb{R}_{\geq 0}$ is the value for item i , $a_i \in \mathbb{R}_{\geq 0}$ is the size of item i , and $b \in \mathbb{R}_{\geq 0}$ is the capacity of the knapsack. Construct a CMRV instance $\hat{I} = (F, L, q, \pi, \succeq, \gamma)$ as follows. Let $F = \{f_1, \dots, f_n\} \cup \{\bar{f}_1, \dots, \bar{f}_n\}$ and $L = \{\ell_1, \dots, \ell_n\} \cup \{\bar{\ell}_1, \dots, \bar{\ell}_n\}$. Without loss of generality assume that $w_1 \geq \dots \geq w_n$ and $\sum_{i=1}^n a_i = \frac{1}{4n}$. Further, assume that $b < \frac{1}{4n}$. Otherwise we would have a trivial instance where the optimal solution is to put all items in the knapsack. Set the rank value function to be

$$v(i) = \begin{cases} w_i/2 & 1 \leq i \leq n \\ 0 & n+1 \leq i \leq 2n \end{cases}.$$

All families f_i have the same preference order

$$\ell_1 \succeq_{f_i} \dots \succeq_{f_i} \ell_n \succeq_{f_i} \bar{\ell}_1 \succeq_{f_i} \dots \succeq_{f_i} \bar{\ell}_n,$$

and all families \bar{f}_i have the preference order

$$\bar{\ell}_1 \succeq_{\bar{f}_i} \dots \succeq_{\bar{f}_i} \bar{\ell}_n \succeq_{\bar{f}_i} \ell_1 \succeq_{\bar{f}_i} \dots \succeq_{\bar{f}_i} \ell_n.$$

Set $\pi_{f_i \bar{\ell}_i} = a_i + \frac{1}{4n}$ and $\pi_{f_i \ell_i} = \pi_{\bar{f}_i \ell_i} = \pi_{\bar{f}_i \bar{\ell}_i} = \frac{1}{4n}$ for all $i \in N$. For all other family-location pairs (i, j) set $\pi_{ij} = 0$. Finally, let each location have a capacity of 1 and choose $\gamma = \frac{2n+1}{4n} - b$.

Note that, due to the choice of π , f_i can only be matched to either ℓ_i or $\bar{\ell}_i$ in any feasible solution for \hat{I} . Otherwise the γ -constraint would be violated. The same holds for \bar{f}_i . Further, because of the capacity constraints, for any feasible matching μ it holds that $\mu(f_i) = \ell_i \Leftrightarrow \mu(\bar{f}_i) = \bar{\ell}_i$.

Let μ^* denote an optimal solution for \hat{I} . Using μ^* , we can obtain an optimal solution $x[\mu^*] \in \{0, 1\}^n$ for I , where $x[\mu^*]_i = 1$ corresponds to item i being placed in the knapsack, by setting $x[\mu^*]_i = 1$ if and only if $\mu^*(f_i) = \ell_i$ for all i . We call $x[\mu^*]$ the solution induced by μ^* . We now show that $x[\mu^*]$ is an optimal solution for I .

First, observe that a matching μ is a feasible solution for \hat{I} if and only if the induced solution $x[\mu]$ is feasible for I :

$$\sum_{i=1}^n \pi_{i\mu(i)} \geq \gamma \quad (6)$$

$$\Leftrightarrow \sum_{i=1}^n \frac{1}{4n} + \frac{1}{4n} + a_i(1 - x[\mu]_i) \geq \frac{2n+1}{4n} - b \quad (7)$$

$$\Leftrightarrow \sum_{i=1}^n -a_i x[\mu]_i \geq -b \quad (8)$$

$$\Leftrightarrow \sum_{i=1}^n a_i x[\mu]_i \leq b. \quad (9)$$

Second, for any feasible matching μ it holds that

$$z(\mu) = \sum_{i=1}^n \frac{w_i}{2} \mathbb{1}[\mu(f_i) = \ell_i] + \frac{w_i}{2} \mathbb{1}[\mu(\bar{f}_i) = \bar{\ell}_i] \quad (10)$$

$$= \sum_{i=1}^n w_i \mathbb{1}[\mu(f_i) = \ell_i] \quad (11)$$

$$= \sum_{i=1}^n w_i x[\mu]_i, \quad (12)$$

where the second equality comes from the fact that μ is feasible and thus $\mu(f_i) = \ell_i \Leftrightarrow \mu(\bar{f}_i) = \bar{\ell}_i$. It follows that that $x[\mu^*]$ is an optimal solution for I .

□

CMRV can be formulated as an integer program.

$$\begin{aligned}
\text{IP(3.2) :} \quad & \text{maximize } \sum_{i \in F} \sum_{j \in L} v(\succeq_{ij}) x_{ij} & (13) \\
& \text{subject to } \sum_{i \in F} x_{ij} = q_j & \forall j \in L \quad (14) \\
& \sum_{j \in L} x_{ij} = 1 & \forall i \in F \quad (15) \\
& \sum_{i \in F} \sum_{j \in L} \pi_{ij} x_{ij} \geq \gamma & (16) \\
& x_{ij} \in \{0, 1\} & \forall i \in F, \forall j \in L \quad (17)
\end{aligned}$$

The objective function of IP(3.2) maximizes family welfare in terms of the rank value function v . Analogous to IP(3.1), Constraints (14) ensure that every location hosts as many families as required for a feasible matching, and Constraints (15) ensure that every family is assigned to exactly one location. Constraint (16) is required to guarantee that $z(\mu) \geq \gamma$.

Suppose that we have a predefined rank value function v , e.g., $v(k) = \frac{1}{k}$. The constrained rank value mechanism, described in Algorithm 2, computes a family-optimal matching μ according to v , such that $z(\mu) \geq \alpha z(\mu^*)$.

Algorithm 2 Constrained Rank Value Mechanism (CRV)

Input: $F, L, \succeq, \pi, b, \alpha$

- 1: Compute the objective value $z(\mu^*)$ of an optimal solution to IP(3.1).
- 2: Let μ denote the matching induced by the solution of IP(3.2) with $\gamma = \alpha z(\mu^*)$.

Output: μ

In contrast to CRSD, it is possible to construct instances where families can benefit from misreporting their true preference orders under CRV. However, a manipulation strategy is not straightforward since a refugee family would need to have at least some knowledge about the predictions π_{ij} of the machine learning models or the government-optimal matching μ^* and the preference orders of other families. It has to be further investigated whether families could in practice exploit this weakness of CRV.

Proposition 10. *CRV is not family-strategyproof, and CRV always produces an α -approximation of the government-optimal matching.*

Proof. Constraint (16) guarantees that the matching μ computed by CRV is an α -approximation of the government-optimal matching. To see that CRV is not family-strategyproof, we refer the reader to Featherstone (2014). □

3.3 Possible Extensions

The model introduced in Section 2 is rather simple and does not necessarily capture all constraints imposed on feasible matchings in the real world. As described by Delacrétaz et al. (2016), an agency responsible for refugee resettlement might have to incorporate family sizes, i.e., q_j denotes the number of refugees instead of families a location is obligated to host. These constraints would introduce additional combinatorial complexity to the problem. Another potential modification is replacing Constraints (2) and Constraints (14) respectively by capacity constraints, i.e., interpreting q_j as an upper bound. Similarly, one could introduce additional service constraints, e.g., constraints concerning housing or medical conditions of refugee families, as described by Delacrétaz et al. (2016) and Trapp et al. (2018). All these restrictions (and others) can easily be incorporated in CRSD and CRV by adding appropriate constraints to the IP formulations IP(3.1) and IP(2).

4 Simulations

In order to compare the performance of CRSD and CRV, we run simulations on randomly generated instances. For our simulations, we assume that both mechanisms have access to the true preferences of families. Notice, however, that this assumption might be unreasonable when these mechanisms would be used in real-world applications, especially in the case of CRV since CRV is not family-strategyproof.

All simulations were run on a laptop computer with an Intel(R) Core(TM) i7-8550U CPU 1.80GHz processor and 16GB RAM running Ubuntu 18.04.

4.1 Instance Generation

We use the following approach to generate instances. Each instance consists of a total of 100 families and 26 locations. There are four types of refugee families f_1, f_2, f_3, f_4 and four types of locations $\ell_1, \ell_2, \ell_3, \ell_4$. The predicted employment probabilities of family types at location types are uniformly distributed according to Table 1.

$\omega_{f\ell}^\pi$	ℓ_1 (1)	ℓ_2 (9)	ℓ_3 (6)	ℓ_4 (10)
f_1 (15)	0.6	0.5	0.5	0.3
f_2 (25)	0.3	0.4	0.2	0.1
f_3 (20)	0.3	0.2	0.4	0.1
f_4 (40)	0.1	0.1	0.1	0.1

Table 1: Values for generating predicted employment probabilities. For family i of type f and location j of type ℓ , $\pi_{ij} \sim \mathcal{U}(0, \omega_{f\ell}^\pi)$.

The numbers in brackets indicate for each family type (location type) how many families (locations) of that type are present in an instance. Families of type f_1 have high predicted employment probability for each of the four location types except for locations of type ℓ_4 . Type f_2 families on the other hand are less likely to be employed in locations of type ℓ_1 , even less likely in locations of type ℓ_3 and ℓ_4 , and have highest probability of employment for locations of types ℓ_2 . Type f_3 is similar to f_2 , except that for those families the predicted employment probabilities for type ℓ_2 locations and type ℓ_3 locations are swapped. Families of type f_4 have a low predicted employment probability at all locations.

Locations of type ℓ_1 (ℓ_2 and ℓ_3) are obligated to host 4 (2) times more families than locations of type ℓ_4 . The preference orders of families are derived from randomly generated valuation functions $(u_i)_{i \in F}$ according to Table 2.

$\omega_{f\ell}^u$	ℓ_1 (1)	ℓ_2 (9)	ℓ_3 (6)	ℓ_4 (10)
$f_1(15)$	1.0	0.6	0.6	0.3
$f_2(25)$	0.8	1.0	0.6	0.3
$f_3(20)$	0.8	0.6	1.0	0.3
$f_4(40)$	1.0	0.6	0.6	0.3

Table 2: Values for generating preference orders. For family i of type f and location j of type ℓ , $u_{ij} \sim \mathcal{U}(0, \omega_{f\ell}^u)$.

4.2 Mechanism Performances

We include Top Trading Cycles (TTC) and Deferred Acceptance (DA) as benchmarks in our simulations. Priorities of locations over families are derived by sorting families according to π_{ij} in decreasing order.

Figure 1 illustrates the performance of CRSD, CRV, TTC and DA on 20 randomly generated instances.

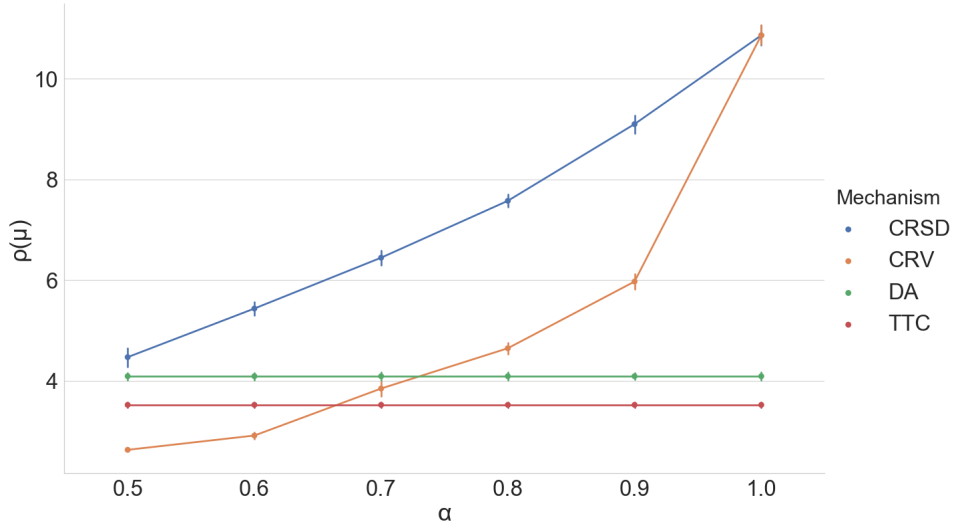


Figure 1: Performance of CRSD, CRV, TTC and DA in terms of family welfare.

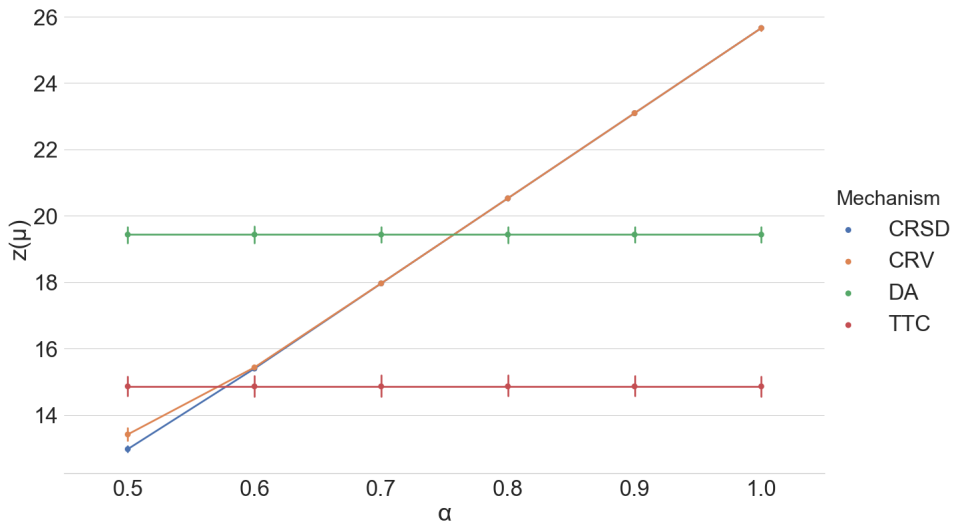


Figure 2: Performance of CRSD, CRV, TTC and DA in terms of employment probability.

For CRSD and CRV the average rank $\rho(\mu)$ strictly decreases (and thus family welfare strictly increases) with α going to 0. Keep in mind that for $\alpha = 1$, both CRSD and CRV produce a government-optimal matching. The average rank of CRV matchings strictly dominates the average rank of CRSD matchings for fixed α , which is what we would expect. However, CRSD easily outperforms the government-optimal matching, even for values of α close to 1. In this concrete setting, $\rho(\mu)$ can be decreased by almost 2 for CRSD and almost 5 for CRV, even if placement officers are only willing to sacrifice 10% of the predicted overall employment rate. Although both TTC and DA achieve high family welfare, remember that these

mechanisms cannot give any guarantees in terms of the government objective, as is shown by Figure 2.

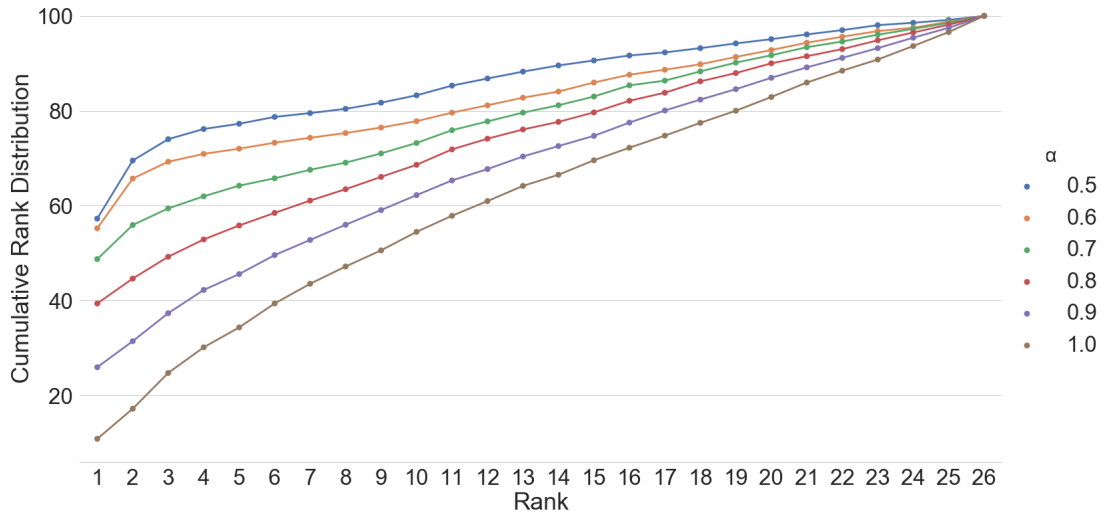


Figure 3: Cumulative rank distribution of CRSD.

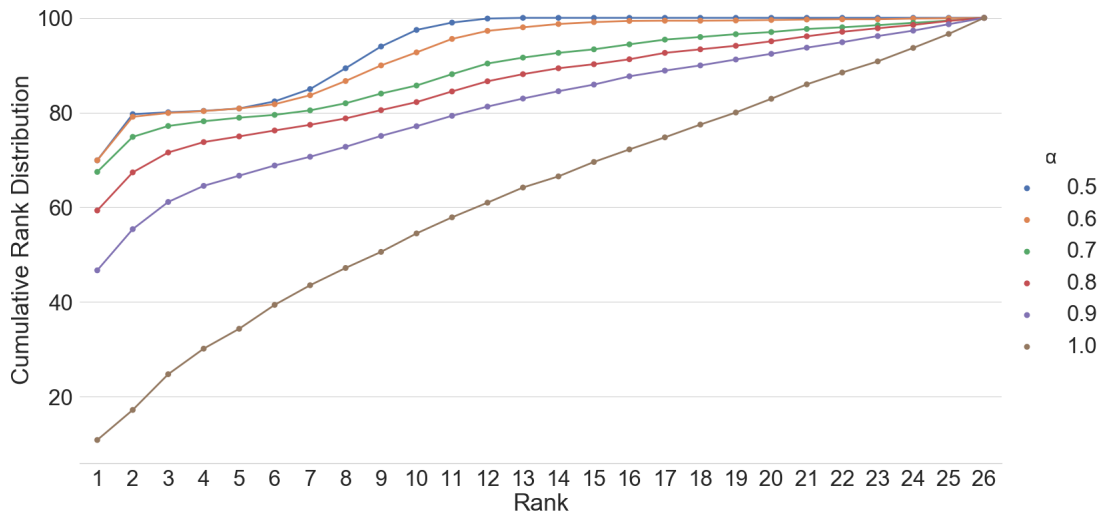


Figure 4: Cumulative rank distribution of CRV.

The cumulative rank distributions of CRSD and CRV provide a more detailed description on how family welfare improves with smaller values of α . As Figure 3 and Figure 4 show, the number of families that are assigned to their first choice can be increased by a factor of roughly 2.4 using CRSD and by a factor of roughly 4.3 using CRV by giving up just 10% of the government-optimal solution value.

5 Conclusion

In this paper, we have proposed two mechanisms, CRSD and CRV, capable of considering families' preferences while simultaneously respecting tight lower bounds on the overall predicted employment rate. While CRSD is family-strategyproof, our simulations show that CRV is in general superior in terms of family welfare when families have complete preference orders over locations. In the case of incomplete preference orders, these results are qualitatively the same and even more pronounced (see Section 6.1 in the appendix for details).

Both mechanisms require refugee families to have previously formed preferences over resettlement locations. Helping families form beliefs over where they are most likely to thrive should have a positive impact on resettlement outcomes. Therefore, future work should explore how families can extend incomplete preference orders. For example, a system which asks refugees to rank properties of cantons could derive preference orders for them (see Delacrétaz et al. (2016)). Additionally, both mechanisms should be evaluated on real-world data.

References

- Acharya, A., Bansak, K., and Hainmueller, J. (2019). Matching refugees to host country locations based on preferences and outcomes. *Working Paper*.
- Bansak, K., Ferwerda, J., Hainmueller, J., Dillon, A., Hangartner, D., Lawrence, D., and Weinstein, J. (2018). Improving refugee integration through data-driven algorithmic assignment. *Science*, 359(6373):325–329.
- Delacrétaz, D., Kominers, S. D., and Teytelboym, A. (2016). Refugee resettlement. *University of Oxford Department of Economics Working Paper*.
- Featherstone, C. R. (2014). Rank efficiency: Investigating a widespread ordinal welfare criterion. Technical report, Technical report, Working Paper.
- Jones, W. and Teytelboym, A. (2017a). The local refugee match: Aligning refugees' preferences with the capacities and priorities of localities. *Journal of Refugee Studies*, 31(2):152–178.
- Jones, W. and Teytelboym, A. (2017b). Matching systems for refugees. *Journal on Migration and Human Security*, 5(3):667–681.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.
- Trapp, A. C., Teytelboym, A., Martinello, A., Andersson, T., Ahani, N., et al. (2018). Placement optimization in refugee resettlement. Technical report.

6 Appendix

6.1 Incomplete Preference Orders

We also compare the performance of CRSD and CRV in a setting where families only have incomplete preference orders, which is closer to a real-world setting. Here, we use a slightly different way to measure family welfare. Let F_μ^* denote the set of families that were matched to one of their ranked locations under μ , i.e., $F_\mu^* = \{i \in F \mid \mu(i) \in L_i\}$, where $L_i \subseteq L$ denotes the set of locations ranked by family i . When we compute the average rank $\rho(\mu)$ of a matching, we only consider families in F_μ^* . Because $\rho(\mu)$ then only captures family welfare for families in F_μ^* , we also look at $\tau(\mu) = |F \setminus F_\mu^*|$, which is the number of families that were not matched to any location in their preference order.

When families only have incomplete preferences orders, CRSD and CRV have to be slightly modified. When CRSD is run on instances with incomplete preference orders, it can happen that a family remains unmatched after the while loop in Line 6. In our simulations, all these unmatched families are simply assigned government-optimally. In the case of CRV, we simply have to exclude a family-location pair (i, j) from the objective function if j was not ranked by i .

A families' preference order is generated just as in the complete preferences setting, but is cut off after position κ_i , where κ_i is sampled from a $\Gamma(2, 1.5)$ -distribution. Because TTC and DA – at least in their original design – do not necessarily produce feasible matchings when families only have incomplete preferences, they are excluded from our analysis in this setting.

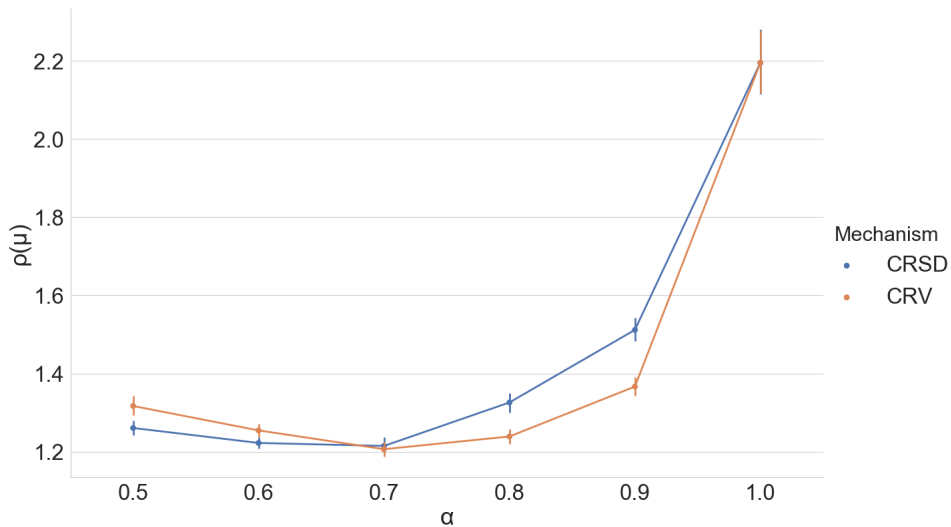


Figure 5: $\rho(\mu)$ of CRSD and CRV for different values of α .

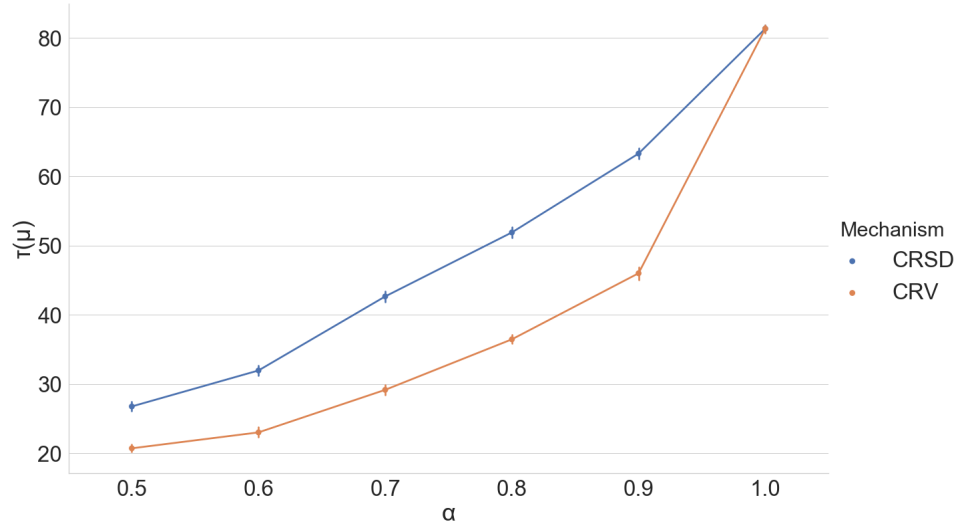


Figure 6: $\tau(\mu)$ of CRSD and CRV for different values of α .

Again, we observe that the average rank strictly decreases for values of α close to 1. Decreasing the value of α for $\alpha \in \{0.5, 0.6, 0.7\}$ slightly increases the average rank again, which can be explained by a further reduction of $\tau(\mu)$, i.e., the number of families that are not assigned to a location in their preference order continues to go down.

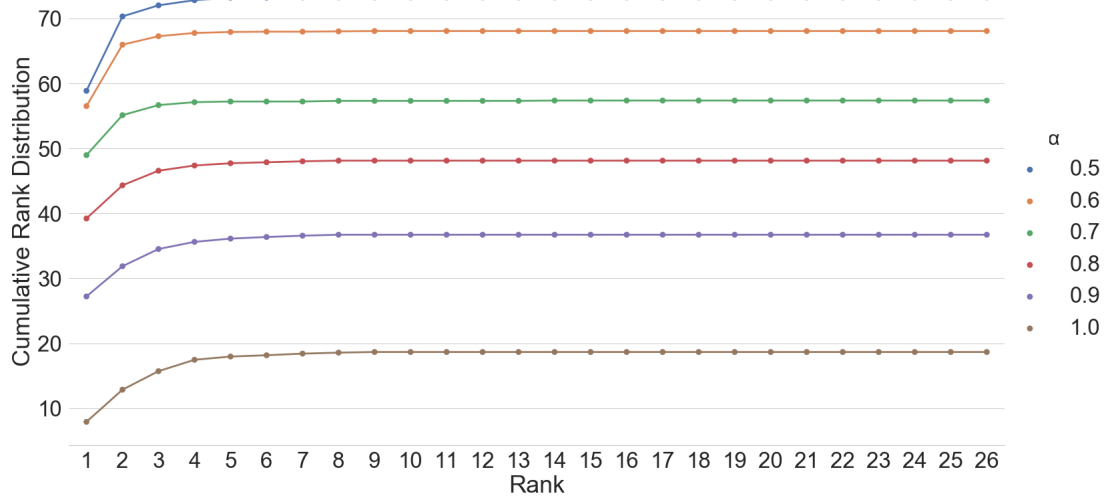


Figure 7: Cumulative rank distribution of CRSD.

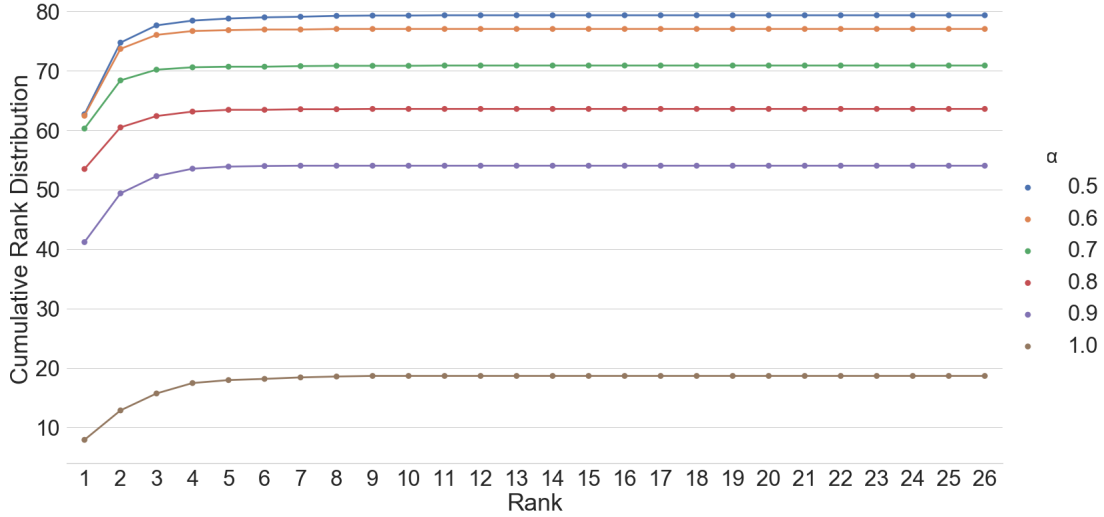


Figure 8: Cumulative rank distribution of CRV.

Choosing $\alpha = 0.9$ instead of $\alpha = 1$ increases $\Delta_1(\mu)$ from an average of 7.9 to 27.2, which corresponds to a factor of roughly 3.4. Unsurprisingly, the effect is even stronger for CRV, where $\Delta_1(\mu)$ increases to 41.1 (a factor of roughly 5.2). Those numbers continue to grow with smaller values of α .

6.2 Simulations With Negative Correlation between π and u

In Section 4.1, π and u are mostly positively correlated, which is a reasonable assumption considering that families also care about finding a job. However, our simulations suggest that also in the case where π and u are mostly negatively correlated, CRSD and CRV can significantly improve family welfare.

$\omega_{f\ell}^\pi$	ℓ_1 (1)	ℓ_2 (9)	ℓ_3 (6)	ℓ_4 (10)
f_1 (15)	0.3	0.5	0.5	0.6
f_2 (25)	0.2	0.1	0.3	0.4
f_3 (20)	0.2	0.3	0.1	0.4
f_4 (40)	0.2	0.2	0.2	0.2

Table 3: Values for generating predicted employment probabilities. For family i of type f and location j of type ℓ , $\pi_{ij} \sim \mathcal{U}(0, \omega_{f\ell}^\pi)$.

$\omega_{f\ell}^u$	ℓ_1 (1)	ℓ_2 (9)	ℓ_3 (6)	ℓ_4 (10)
$f_1(15)$	1.0	0.6	0.6	0.3
$f_2(25)$	0.8	1.0	0.6	0.3
$f_3(20)$	0.8	0.6	1.0	0.3
$f_4(40)$	1.0	0.6	0.6	0.3

Table 4: Values for generating preference orders. For family i of type f and location j of type ℓ , $u_{ij} \sim \mathcal{U}(0, \omega_{f\ell}^u)$.

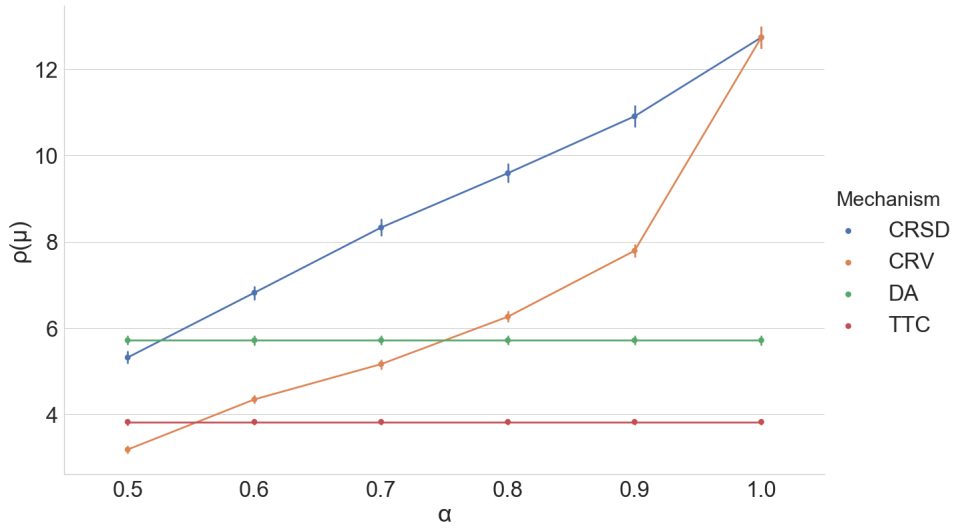


Figure 9: Performance of CRSD, CRV, TTC and DA in terms of family welfare.

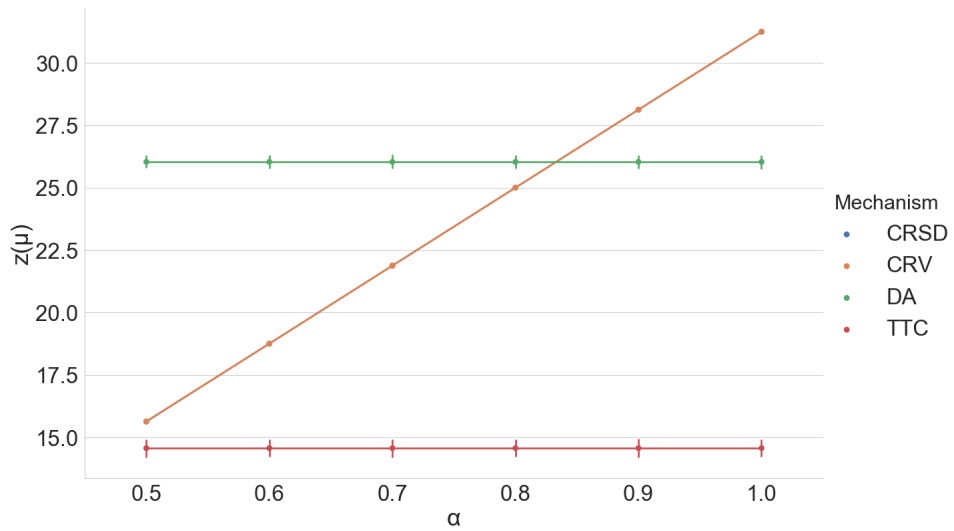


Figure 10: Performance of CRSD, CRV, TTC and DA in terms of employment probability.

This does not only hold for the setting with complete preference orders, but also for the setting with incomplete preference orders.

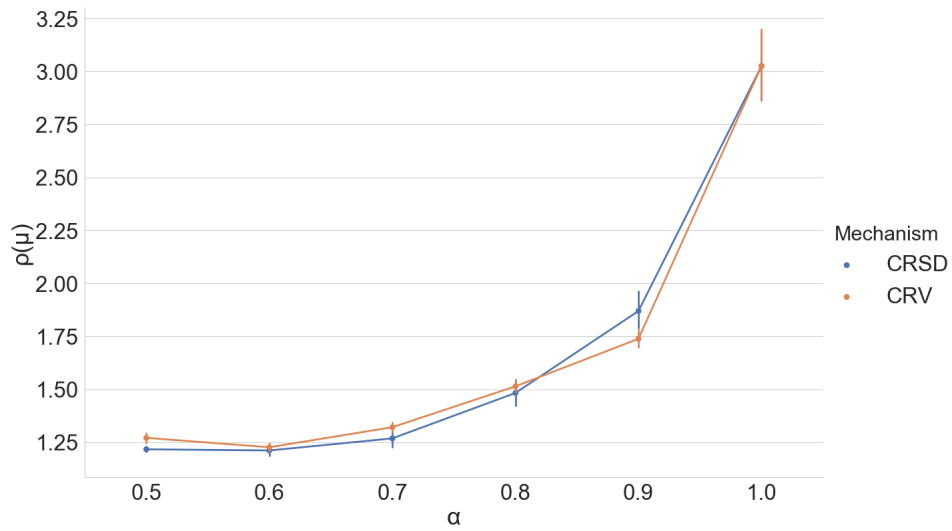


Figure 11: $\rho(\mu)$ of CRSD and CRV for different values of α .

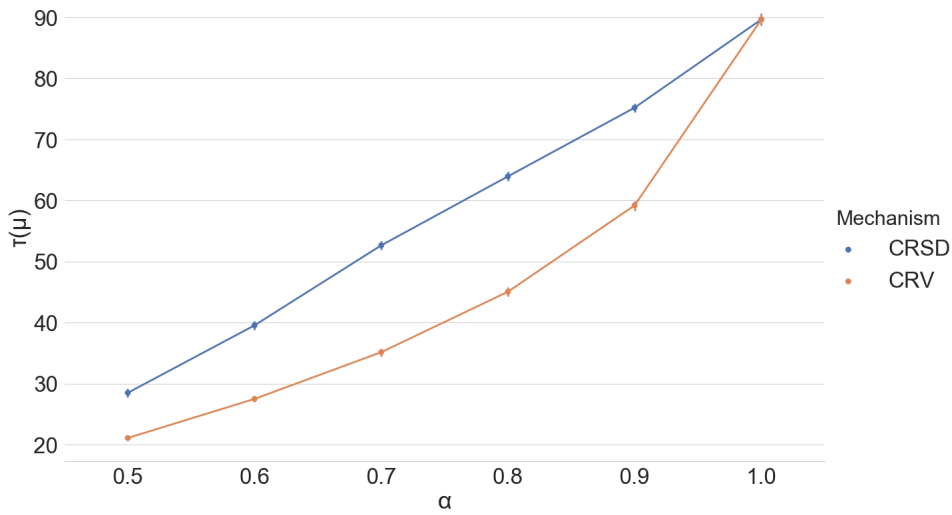


Figure 12: $\tau(\mu)$ of CRSD and CRV for different values of α .