ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240892

Parameterized Algorithms for Optimal Refugee Resettlement

Jiehua Chen[a,1,*], **Ildikó Schlotter**[b,c,1] **and Sofia Simola**[a,1]

[a]TU Wien, Austria
[b]HUN-REN Centre for Economic and Regional Studies, Hungary
[c]Budapest University of Technology and Economics, Hungary
ORCID (Jiehua Chen): https://orcid.org/0000-0002-8163-1327, ORCID (Ildikó Schlotter): https://orcid.org/0000-0002-0114-8280, ORCID (Sofia Simola): https://orcid.org/0000-0001-7941-0018

Abstract. We study variants of the Optimal Refugee Resettlement problem where a set F of refugee families need to be allocated to a set P of possible places of resettlement in a feasible and optimal way. Feasibility issues emerge from the assumption that each family requires certain services (such as accommodation, school seats, or medical assistance), while there is an upper and, possibly, a lower quota on the number of service units provided at a given place. Besides studying the problem of finding a feasible assignment, we also investigate two natural optimization variants. In the first one, we allow families to express preferences over P, and we aim for a Pareto-optimal assignment. In a more general setting, families can attribute utilities to each place in P, and the task is to find a feasible assignment with maximum total utilities. We study the computational complexity of all three variants in a multivariate fashion using the framework of parameterized complexity. We provide fixed-parameter algorithms for a handful of natural parameterizations, and complement these tractable cases with tight intractability results.

1 Introduction

At the 2023 Global Refugee Forum, the UN High Commissioner for Refugees reported that *114 million* people are currently displaced due to persecution, human rights violations, violence, and wars, and made a direct appeal to everyone to join forces to help refugees find protection.[2] This immense number highlights the critical need for effective resettlement strategies that cater to diverse populations.

Refugee resettlement involves not just relocating individuals but also families, each with distinct needs and service requirements ranging from accommodation to education and medical assistance. Delacrétaz et al. [15] and Ahani et al. [2] propose a *multi-dimensional* and *multiple knapsack* model to address these challenges. Their model takes into account the specific needs of *refugee families* who require a range of services, as well as the capacity constraints of potential hosting places that have specific upper and lower quotas on the services they can offer. The goal is to determine a *feasible* assignment from the families to the places which satisfies the specific needs of the families while ensuring that no place is over- or under-subscribed according to its capacity constraints. Additionally, the model may include a *utility score* for each family–place pair which estimates the "profit" that a family may contribute to a place; such profit could be for example the employment outcome. Optimizing the assignment means finding a feasible assignment that yields a maximum total utility.

If we care about the welfare and choices of the refugee families, we may allow them to express *preferences* over places which they find acceptable [15]. A standard optimality criterion in such a case is Pareto-optimality, which means that we aim for a feasible assignment for which no other feasible assignment can make one family better off without making another worse off.

Unfortunately, it is computationally intractable (i.e., NP-hard) to determine whether a feasible assignment exists [8]. Similarly, it is NP-hard to find a feasible assignment with maximum total utility or one that is Pareto-optimal, even if there are no lower quotas [7, 18, 2]. To tackle these complexities, we examine the parameterized complexity of the three computational problems for refugee resettlement that we study, FEASIBLE-RR, MAXUTIL-RR and PARETO-RR, and provide parameterized algorithms for them. We focus on canonical parameters such as the number of places (m), the number of refugee families (n), the number of services (t), and the desired utility (u^*). We also consider additional parameters that are motivated from real-life scenarios, including the maximum number $r_{\max}$ of units required by a family per service and the maximum utility $u_{\max}$ a family can contribute. The service units can reasonably be assumed to be small integers in practical situations when a family's requirements describe their need for housing (e.g., number of beds or bedrooms) or education (e.g., the number of school seats or kindergarten places). Our study provides new insights into the parameterized complexities of these problems, presenting fixed-parameter (FPT) algorithms for several natural parameterizations, and contrasting these with strong intractability results. See Table 1 for an overview. We summarize our main contributions as follows.

Single service. We develop an FPT algorithm w.r.t. $r_{\max}$ for FEASIBLE-RR; the algorithm also applies to MAXUTIL-RR and PARETO-RR when all families have the same utilities for all places (*equal utilities*) or are indifferent between all of them (*equal preferences*), respectively; see Theorem 1. The main idea is to group all families together that have the same requirements, and group all places together

* Corresponding Author. Email: jiehua.chen@ac.tuwien.ac.at.
1 Equal contribution.
2 https://www.unhcr.org/global-refugee-forum-2023

with the same lower and upper quotas. Then, we observe that either the upper quotas are small (i.e., bounded by a function in $r_{\max}$) so we can brute-force search all possible partitions of the families into different places, or there is a so-called *homogeneous ρ-block* (see Section 3 for the formal definition) that can be exchanged across the places, which enables us to replace the upper quota of each place with a value bounded by a function of $r_{\max}$. In this way, we bound the number of groups of families and places, and can use Integer Linear Programming (ILP) to obtain an FPT algorithm for $r_{\max}$.

We also propose an FPT algorithm for the combined parameter $m + r_{\max}$ for the general case when families may have different utilities or preferences (Theorem 3). The generalized algorithm additionally uses the idea that *homogeneous ρ-blocks* can be exchanged across places, and combines dynamic programming with color-coding [4] to find an optimal solution in FPT time.

Multiple services. In Theorem 6, we extend the FPT algorithm of Theorem 1 for the setting of equal preferences or utilities to multiple service types by combining the parameters $r_{\max}$ and t, the number of services; we use the technique of N-fold integer programming [20]. We present a more general FPT algorithm for PARETO-RR with parameter $r_{\max}+t+m$ which also solves MAXUTIL-RR if the number of different utility values is bounded (Theorem 7); this result relies on Lenstra's result on solving ILPs with bounded dimension [25]. Contrasting our algorithmic results, we prove that PARETO- and MAXUTIL-RR are both NP-hard already for three places, even if there are no lower quotas, all upper quotas are 1, and families have equal preferences or utilities, respectively; see Theorem 5.

Related work. The model we study is the same as that of Ahani et al. [2]. They formulate MAXUTIL-RR via ILP and study its performance. The same model without lower quotas has attracted previous study: It was introduced in a working paper by Delacrétaz et al. [14] (see also [15]). The paper provides an algorithm for finding a Pareto-optimal matching when the preferences are strict, and also studies other stability concepts. Aziz et al. [7] show that *finding* a Pareto-optimal assignment is NP-hard even when the families are indifferent between places, and study a few other stability concepts. Nguyen et al. [28] use fractional matchings to find group-stable assignments which violate the quotas only a little. None of the works above focuses on parameterized complexity analysis.

As already mentioned by Ahani et al. [2], the MAXUTIL-RR problem is a generalization of the MULTIPLE/MULTIDIMENSIONAL KNAPSACK problem [2]. The parameterized complexity of the latter has been studied by Gurski et al. [18], and several of our hardness-results are obtained either directly from them or from modifications of their reductions. MULTIPLE/MULTIDIMENSIONAL KNAPSACK however has neither lower quotas nor different profits for items depending on which knapsack they are placed in. They also assume the sizes and profits are encoded in binary, whereas we assume they are encoded in unary. Hence, their parameterized algorithms are not directly applicable to our problems. FEASIBLE-RR generalizes BIN PACKING [21] and hence SIMPLE MULTIDIMENSIONAL PARTITIONED SUBSET SUM [16]; note that the latter two problems are equivalent. Since BIN PACKING is W[1]-hard w.r.t. the number of bins and the bins correspond to the places in our setting, W[1]-hardness for FEASIBLE-RR follows; see Proposition 1.

The problem can be seen as an extension of different classical matching problems. We can model MATCHING WITH DIVERSITY CONSTRAINTS [11, 19, 12, 8, 1, 24] by using services as types. In the case where we have a single service, the problem can be seen as a variant of MATCHING WITH SIZES [10, 27], where the service requirements correspond to the sizes.

Refugee resettlement has also been studied in the literature under other types of models: Online setting [6, 3, 9], one-to-one housing [5], preferences based on weighted vectors [30], hedonic games [23], and placing refugees on a graph [22, 26, 29].

Paper structure. In Section 2, we formally define refugee resettlement. We investigate the case when there is only one service and when there are multiple services in Section 3 and Section 4, respectively. In Section 4, we first look at FEASIBLE-RR, followed by PARETO-RR, and finally MAXUTIL-RR. We conclude with a discussion on potential areas for future research in Section 5. Additional results and the proofs for the statements marked with ⋆ are deferred to the full version of the paper [13].

2 Preliminaries

For an integer z, we use $[z]$ to denote the set $\{1, 2, \ldots, z\}$. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of the same length, we write $\boldsymbol{x} \le \boldsymbol{y}$ if for each coordinate i it holds that $\boldsymbol{x}[i] \le \boldsymbol{y}[j]$.

An *instance* of REFUGEE RESETTLEMENT (RR) is a tuple $(F, P, S, (\boldsymbol{r}_i)_{f_i \in F}, (\underline{\boldsymbol{c}}_j, \bar{\boldsymbol{c}}_j)_{p_j \in P})$ with information:

- F denotes a set of n refugee *families* with $F = \{f_1, \ldots, f_n\}$,
- P denotes a set of m *places* with $P = \{p_1, \ldots, p_m\}$, and
- S denotes a set of t *services* $S = \{s_1, \ldots, s_t\}$, such that
- each family $f_i \in F$ has a *requirement* vector $\boldsymbol{r}_i \in \mathbb{N}^t$ where, for every service $s_k \in S$, the value $\boldsymbol{r}_i[k]$ determines how many units of service s_k the family f_i requires, and
- each place $p_j \in P$ has two vectors $\underline{\boldsymbol{c}}_j, \bar{\boldsymbol{c}}_j \in \mathbb{N}^t$, denoted as *lower quota* and *upper quota* which indicate for every service $s_k \in S$, the minimum and maximum number of units place p_j can provide. Non-zero lower quotas for places may for example follow from an obligation for a place to house at least a certain number of refugees. If the lower quota of every place is a zero-vector, then we say that the instance has *no lower quotas*.

Assignments. Given an instance of RR, an *assignment* is a function $\sigma \colon F \to P \cup \{\perp\}$; we say that $f_i \in F$ is *assigned* to a place $p_j \in P$ if $\sigma(f_i) = p_j$, and f_i is *unassigned* if $\sigma(f_i) = \perp$. We define the *load* vector of a place $p_j \in P$ under σ as $\mathrm{load}(p_j, \sigma) \coloneqq \sum_{f_i \in \sigma^{-1}(\ell_j)} \boldsymbol{r}_i[k]$; for each service $s_k \in S$, $\mathrm{load}(p_j, \sigma)[k]$ denotes the number of units that are required by the refugees that are assigned to p_j. An assignment is *complete* if it does not leave any families unassigned. An assignment is *feasible* if for every place $p_j \in P$ the load vector is within the lower and upper quota, i.e., $\underline{\boldsymbol{c}}_j \le \mathrm{load}(p_j, \sigma) \le \bar{\boldsymbol{c}}_j$. Place p_j can *accommodate* a set of families $F' \subseteq F$ if $\sum_{f_{i'} \in F'} \boldsymbol{r}_{i'}[k] \le \bar{\boldsymbol{c}}_j[k]$ for each $s_k \in S$.

Utilities. Each family may contribute a certain utility to each place. To model this, each family $f_i \in F$ expresses an integral *utility* vector $\boldsymbol{u}_i \in \mathbb{Z}^m$, where for every $p_j \in P$, the value $\boldsymbol{u}_i[j]$ indicates the utility of family f_i if assigned to p_j. Note that we also allow negative utilities, but it will be evident that all hardness results hold even if the utilities are non-negative. Given an assignment σ, we define the *(total) utility* of the assignment as the sum of all utilities contributed by the families, i.e., $\mathrm{util}(\sigma) = \sum_{p_j \in P} \sum_{f_i \in \sigma^{-1}(p_j)} \boldsymbol{u}_i[j]$. We consider two special kinds of utility vectors. We say that the families have *equal utilities* if all utility values $\boldsymbol{u}_i[j]$ are equal and positive over all families $f_i \in F$ and places $p_j \in P$, and families have *binary* utilities if each utility value is either zero or one.

Preferences and Pareto-optimal assignments. Each family $f_i \in F$ may only find a subset of places *acceptable* and may have a *preference list* $\succeq_i$ over the acceptable places, i.e., a weak order over a subset of P. For a family f_i and two places p and p' in its preference list, $p \succeq_i p'$ means that f_i *weakly prefers* p to p'. If $p \succeq_i p'$ and $p' \succeq_i p$, then we write $p \sim_i p'$ and say that f_i is *indifferent* between p and p'. We write $p \succ_i p'$ to denote that f_i *(strictly) prefers* p to p', meaning that $p \succeq_i p'$ but $p \not\succeq_i p'$. If the preference list of f_i contains p, then f_i finds p *acceptable*. We assume that each family f_i prefers being assigned to some place in his preference list over being unassigned; accordingly, we write $p \succ_i \bot$. An assignment is *acceptable* if every family is either unassigned or assigned to a place it finds acceptable.

We also define *equal* and *dichotomous* preferences: If every family finds every place acceptable and is additionally indifferent between them, then the preferences are *equal*. If every family is indifferent between every place it finds acceptable, then the preferences are *dichotomous*.

A feasible and acceptable assignment σ is *Pareto-optimal* if it admits no *Pareto-improvement*, that is, a feasible and acceptable assignment σ' such that $\sigma'(f_i) \succeq_i \sigma(f_i)$ for every $f_i \in F$ and there exists at least one family $f_{i'} \in F$ such that $\sigma'(f_{i'}) \succ_{i'} \sigma(f_{i'})$.

We present an example of our model in the full version [13].

Central problems. We are now ready to define our problems.

FEASIBLE-RR
Input: An instance I of REFUGEE RESETTLEMENT.
Question: Is there a feasible assignment σ for I?

MAXUTIL-RR
Input: An instance I of refugee resettlement, a utility vector $\boldsymbol{u}_i \in \mathbb{Z}^m$ for each family $f_i \in F$, and an integer bound u^*.
Question: Is there a feasible assignment σ for I such that $\text{util}(\sigma) \geq u^*$?

PARETO-RR
Input: An instance I of refugee resettlement and a preference order $\succeq_i$ for every $f_i \in F$.
Task: *Find* a feasible and acceptable Pareto-optimal assignment σ for I or report that none exists.

We remark that there is a straightforward way to reduce PARETO-RR to the optimization variant of MAXUTIL-RR in the following sense. Suppose that an algorithm $\mathcal{A}$ finds a maximum-utility feasible assignment for each instance of MAXUTIL-RR that admits a feasible assignment. Such an algorithm can be used to solve an instance I the PARETO-RR problem as follows.

Observation 1 (⋆). *Given an instance I of* PARETO-RR, *construct an instance I' of* MAXUTIL-RR *as follows. For each family $f_i \in F$:*
- *for every place p_j that f_i finds acceptable, set $\boldsymbol{u}_i[j] = |\{p_{j'} \in P \mid p_j \succeq_i p_{j'}\}|$;*
- *for each place p_j that f_i finds unacceptable, set $\boldsymbol{u}_i[j] = -m \cdot n$.*

Let σ be a maximum-utility feasible assignment for I'. If $\text{util}(\sigma) > 0$, *then σ is a feasible, acceptable, and Pareto-optimal assignment for I; otherwise there is no feasible and acceptable assignment for I.*

Parameterization. We study the following parameters:
- number of places ($m = |P|$),
- number of refugee families ($n = |F|$),
- number of services ($t = |S|$),
- maximum number of units required for all services and by all families ($r_{\max} = \max\{\boldsymbol{r}_i[k] : f_i \in F, s_k \in S\}$).

We also study the following parameters for MAXUTIL-RR: the total utility bound u^* and the maximum utility brought by a family $u_{\max} = \max\{\boldsymbol{u}_i[j] : f_i \in F, p_j \in P\}$. Additionally, we consider the highest upper quota any place has for a service $c_{\max} = \max_{p_j \in P, s_k \in S} \bar{\boldsymbol{c}}_j[k]$, but discover that this parameter behaves very

Parameter	FEASIBLE	MAXUTIL LQ=0 / LQ≠0		PARETO LQ=0 / LQ≠0	
m	**W1h** [P1] **XP** [P7]	**W1h**°/**W1h**° **XP** / **XP**	[P1] [P7]	**W1h**$^=$/**W1h**$^=$ **XP** / **XP**	[P1] [P7]
$r_{\max}$ eq. util./pref.	**FPT** [T1] – –	**NPh** / **NPh** **FPT** / **FPT**	[T2] [T1]	**NPh** / **NPh** **FPT** / **FPT**	[T2] [T1]
$m + r_{\max}$	**FPT** [T3]	**FPT** / **FPT**	[T3]	**FPT** / **FPT**	[T3]
$u_{\max}$ u^*	– – – –	NPh°/NPh° **FPT**/**NPh**°	[18] [T4]/[P1]	– –	– –
$m + r_{\max}$	**NPh** [P2]	NPh°/NPh°	[18]	**NPh**$^=$/**NPh**$^=$	[T5]/[P2]
t	**NPh** [P1]	NPh°/NPh°	[18]	NPh$^=$/NPh$^=$	[7]
n	**FPT** [P6]	**FPT** / **FPT**	[P6]	**FPT** / **FPT**	[P6]
$m + t$	**W1h** [P1] **XP** [P7]	**W1h**°/**W1h**° **XP** / **XP**	[P1] [P7]	**W1h**$^=$/**W1h**$^=$ **XP** / **XP**	[P1] [P7]
$t + r_{\max}$ eq. util./pref.	**FPT** [T6] – –	**NPh** / **NPh** **FPT**/**FPT**	[T2] [T6]	**NPh** / **NPh** **FPT** / **FPT**	[T2] [T6]
$m + t + r_{\max}$ binary util.	**FPT** [T6] – –	**XP** / **XP**, ? **FPT**	[P7] [T7]	**FPT** / **FPT** –	[T7] –
u^*	– – – –	W1h°/**NPh**° **XP**/**NPh**°	[18]/[P2] [P8]/[P2]	– –	– –

Table 1. All three problems are NP-hard in general; see [8], [18, T32],[7, P7.1]. Above: Results for the single-service case ($t = 1$). We skip the parameterization by n since for this case it is FPT for the more general case. Below: Results for the general case. We skip the parameterization by $u_{\max}$ since it is already NP-hard for the single-service case. Bold faced results are obtained in this paper. LQ=0 (resp. LQ≠0) refers to the case when lower quotas are zero (resp. may be positive). NPh means that the problem remains NP-hard even if the corresponding parameter is a constant. All hardness results hold for dichotomous preferences or binary utilities. Additionally, ° (resp. $^=$) means hardness results hold even for equal utilities (resp. preferences). The results for the remaining parameter combinations are deferred to the full version [13].

similarly to the smaller and better-motivated parameter $r_{\max}$. Note that we may assume that for each family there is at least one place that can accommodate it, otherwise we can remove the family from our instance; this implies that we can assume $c_{\max} \geq r_{\max}$.

We obtain FPT results w.r.t. the sum of the capacities of the places, that is, $c_\Sigma = \sum_{p_j \in P, s_k \in S} \bar{\boldsymbol{c}}_j[k]$, and the sum of the requirements of the families $r_\Sigma = \sum_{f_i \in F, s_k \in S} \boldsymbol{r}_i[k]$. If there are no lower quotas, we also have an FPT result w.r.t. the sum of utilities $u_\Sigma = \sum_{f_i \in F, p_j \in S} \boldsymbol{u}_i[j]$. If the instance has non-zero lower quotas, then the problem is hard even when all utilities are zero, and this parameter is not helpful. We also study the complexity w.r.t. the number $n_\sim$ of agents who have ties in their preference lists. Finally, we observe that the parameter "the maximum length of the ties in preference lists" does not help in designing parameterized algorithms since it is upper-bounded by m, and most problems are already hard w.r.t. m. Discussion on parameters c_Σ, r_Σ, u_Σ, and $n_\sim$ are deferred to the full version [13].

3 Single service

Let us assume that there is only a single service in our input instance. Thus, we will simply refer to $\boldsymbol{r}_i[1]$ as the *requirement* of a family $f_i \in F$, and we will write $\boldsymbol{r}_i = \boldsymbol{r}_i[1]$ accordingly. Observe that we may assume w.l.o.g. that each family has a positive requirement. Similarly, we will refer to $\bar{\boldsymbol{c}}_j[1]$ and $\underline{\boldsymbol{c}}_j[1]$ as the *upper* and the *lower quota* of a place $p_j \in P$, writing also $\bar{\boldsymbol{c}}_j = \bar{\boldsymbol{c}}_j[1]$ and $\underline{\boldsymbol{c}}_j = \underline{\boldsymbol{c}}_j[1]$.

The reader may observe that when our sole concern is feasibility, then the problem can be seen as a multidimensional variant of the classic BIN PACKING or KNAPSACK problems. On the one hand, it is not hard to show that the parameterized hardness of BIN PACKING w.r.t. the number of bins as parameter translates to parameterized hardness of FEASIBLE-RR w.r.t. the number of places; see Proposition 1. On the other hand, the textbook dynamic programming technique for KNAPSACK was used by Gurski et al. [18, Proposition 34] to solve the so-called MAX MULTIPLE KNAPSACK problem which in our model coincides with the MAXUTIL-RR problem without lower quotas. This approach can be adapted in a straightforward way to solve the MAXUTIL-RR problem even for the case when there are multiple services and lower quotas; in Proposition 7 we present an algorithm running in $O((c_{\max})^{mt}nm)$ time.

Proposition 1 (⋆)**.** *The following problems are W[1]-hard w.r.t. m for $t = 1$:*

- FEASIBLE-RR*;*
- PARETO-RR *with no lower quotas and equal preferences;*
- PARETO-RR *when all families have strict preferences;*
- MAXUTIL-RR *with no lower quotas and equal utilities;*
- MAXUTIL-RR *with $u^* = 0$.*

In spite of the strong connection between FEASIBLE-RR and BIN PACKING (or between MAXUTIL-RR and KNAPSACK), the context of REFUGEE RESETTLEMENT motivates parameterizations that have not been studied for these two classical problems. One such parameter is $r_{\max}$, the maximum units of a service that any refugee family may require. Theorem 1 presents an efficient algorithm for FEASIBLE-RR for the case when $r_{\max}$ is small; the proposed algorithm can be used to solve PARETO- and MAXUTIL-RR as well, assuming equal preferences or utilities, when the task is to assign as many refugee families as possible.

Let us introduce an important notion used in our algorithms. Let ρ denote the least common multiple of all integers in the set $\{1, \dots, r_{\max}\}$; then $\rho \le (r_{\max})!$ is clear. We say that a set $F' \subseteq F$ of families is a *homogeneous ρ-block*, if all families in F' have the same requirement, and their total requirement is exactly ρ.

Observation 2 (⋆)**.** *Suppose that the number of services is $t = 1$. If $F' \subseteq F$ is a set of families such that $\sum_{f_i \in F'} \boldsymbol{r}_i > r_{\max}(\rho - 1)$, then F' contains a homogeneous ρ-block.*

In the case where $r_{\max}$ is a constant and the families are indifferent between the places, we can use Observation 2 to bound the number of different possible places. The idea is to observe that while the location capacities are unbounded, we can bound the "relevant part" of them by a function of $r_{\max}$. If the total requirement of families assigned to a place is more than $r_{\max}(\rho - 1)$, there must be a homogeneous ρ-block among them. We can treat these homogeneous ρ-blocks separately from the places they originate from, and thus bound the upper quotas by a function of $r_{\max}$. The family requirements are also trivially bounded by $r_{\max}$.

Since we have now bounded both the maximum requirements of the families and the capacities of the locations, we can enumerate all the different ways families may be matched to places. We can create an ILP that has a variable for each such way and additionally variables for the homogeneous ρ-blocks. As the number of variables and constraints is bounded above by a function of $r_{\max}$, we can solve this ILP in FPT time w.r.t. $r_{\max}$ [25].

Theorem 1. PARETO- *and* MAXUTIL-RR *are FPT w.r.t. $r_{\max}$ when $t = 1$ and families have equal preferences or utilities, respectively.*

Proof. We start by showing that we can find an assignment that matches the maximum number of families in FPT time w.r.t. $r_{\max}$.

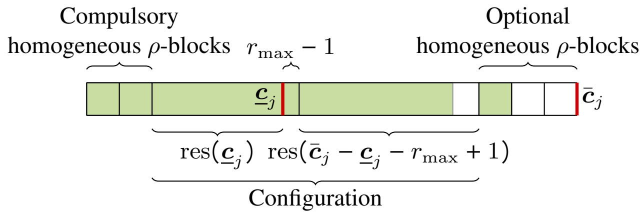

Figure 1. Illustration for the proof of Theorem 1. The bar shows the upper and lower quotas of a place, and the green area represents the requirements of the families matched to it.

Under equal preferences (resp. utilities) this assignment must maximize utility (resp. be Pareto-optimal).

We start by defining two functions which will be useful for typing places by their service quotas: the function $\arg\text{res}\colon \mathbb{N} \to \mathbb{N}$ and the function $\text{res}\colon \mathbb{N} \to \{0, \dots, r_{\max}(\rho - 1)\}$:

$$\text{res}(x) = \max_{\alpha \in \mathbb{N}}\{x - \alpha \cdot \rho : 0 \le x - \alpha \cdot \rho \le r_{\max}(\rho - 1)\}$$

$$\arg\text{res}(x) = \arg\max_{\alpha \in \mathbb{N}}\{x - \alpha \cdot \rho : 0 \le x - \alpha \cdot \rho \le r_{\max}(\rho - 1)\}.$$

Observe that $\quad \forall x \in \mathbb{N}, x = \rho \cdot \arg\text{res}(x) + \text{res}(x). \qquad (1)$

Using the res function we can associate each place with a type τ^P. Let $\tau^P(p_j) = (\text{res}(\underline{\boldsymbol{c}}_j), x)$, where

$$x = \begin{cases} \text{res}(\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j - r_{\max} + 1) + r_{\max} - 1, & \text{if } \bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j \ge r_{\max} - 1, \\ \bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j, & \text{otherwise.} \end{cases}$$

The first element of the type tells us the lower quota of the place after we have discounted all the homogeneous ρ-blocks that are used to satisfy the lower quota. The second element tells us the size of the "optional" quota $\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j$ when we have again discounted all the homogeneous ρ-blocks that this part may contain. We however only compute the residue on the part of $\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j$ that is larger than $r_{\max} - 1$. This is because the total requirements of the families that are used to satisfy $\underline{\boldsymbol{c}}_j$ may be slightly greater than $\underline{\boldsymbol{c}}_j$, and thus there may be a family whose requirement is partially counted for $\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j$, however it may have been used for a homogeneous ρ-block. See Figure 1 for intuition of how the lower and upper quotas of a place are divided into homogeneous ρ-blocks and a reconfiguration; see below for the definition.

Let $\mathcal{T}^P = \{0, \dots, r_{\max}(\rho - 1)\} \times \{0, \dots, r_{\max}(\rho - 1) + r_{\max} - 1\}$ be the set of possible place types. It is clear that their number is bounded above by $r_{\max}^2 \rho^2$, which is a function of $r_{\max}$. We additionally know that every place must have $\arg\text{res}(\underline{\boldsymbol{c}}_j)$ many homogeneous ρ-blocks assigned to it. We call these *compulsory* homogeneous ρ-blocks. Similarly, we may assign at most $\arg\text{res}(\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j - r_{\max} + 1)$ additional homogeneous ρ-blocks to p_j, which we call *optional* homogeneous ρ-blocks. Because the families are indifferent between the places, we do not need to keep track of the place to which the compulsory and optional homogeneous ρ-blocks belong, and we only enforce that the families assigned to compulsory homogeneous ρ-blocks create exactly the number of compulsory homogeneous ρ-blocks needed, and the families assigned to optional homogeneous ρ-blocks create at most the number of optional homogeneous ρ-blocks.

Now with a bound on the upper and lower quotas of the places discounting the homogeneous ρ-blocks, we can enumerate all possible ways to satisfy these quotas. Let $\hat{\rho} \coloneqq 2r_{\max}(\rho - 1) + r_{\max} - 1$. This is the maximum sum of requirements that may be assigned to any place and that are not part of a homogeneous ρ-block. We create configurations $c^F \in \{0, \dots, \hat{\rho}\}^{r_{\max}}$ that tell us the number of families of each requirement type. Let us denote the set of possible configurations $C^F := \{0, \ldots, \hat{\rho}\}^{r_{\max}}$. It is clear that the number of possible configurations is bounded above by $(\hat{\rho}+1)^{r_{\max}}$, which is a function of $r_{\max}$.

We say that a place type τ^P is *suitable* for a configuration c^F if $\tau^P[1] \le \sum_{r \in [r_{\max}]} c^F[r] \cdot r \le \tau^P[1] + \tau^P[2]$. This means that if the families are assigned to a place according to the configuration, its upper and lower quotas are satisfied.

We create an ILP with the following variables and constants:

- non-negative integer variables $\underline{b}_r$ and $\bar{b}_r$ for each $r \in [r_{\max}]$, representing the number of compulsory or optional, respectively, homogeneous ρ-blocks filled with families of requirement r.
- non-negative integer variable $x(\tau^P, c^F)$ for every $\tau^P \in \mathcal{T}^P$, $c^F \in C^F$ such that τ^P is suitable for c^F, counting the number of places of type τ^P that are assigned families according to configuration c^F.
- m_{τ^P} is the number of places of type τ^P for each $\tau^P \in \mathcal{T}^P$;
- $\underline{B} = \sum_{p_j \in P} \arg\operatorname{res}(\underline{\boldsymbol{c}}_j)$ (resp. $\overline{B} = \sum_{p_j \in P} \arg\operatorname{res}(\max(\bar{\boldsymbol{c}}_j - \underline{\boldsymbol{c}}_j - r_{\max} + 1), 0))$ is the number of compulsory (resp. optional) homogeneous ρ-blocks;
- n_r is the number of families $f_i \in F$ such that $\boldsymbol{r}_i = r$, for each $r \in [r_{\max}]$.

We create the following ILP:

(ILP$_1$)

$$\max \sum_{r \in [r_{\max}]} \left(\frac{\rho}{r}(\underline{b}_r + \bar{b}_r) + \sum_{\substack{\tau^P \in \mathcal{T}^P \\ c^F \in C^F}} c^F[r] x(\tau^P, c^F) \right) \quad \text{s.t.}$$

$$\forall \tau^P \in \mathcal{T}^P\colon \sum_{\substack{c^F \in C^F \\ c^F \text{ is suitable for } \tau^P}} x(\tau^P, c^F) = m_{\tau^P} \tag{2}$$

$$\sum_{r \in [r_{\max}]} \underline{b}_r = \underline{B} \quad \text{and} \quad \sum_{r \in [r_{\max}]} \bar{b}_r \le \overline{B} \tag{3}$$

$$\forall r \in [r_{\max}]\colon \frac{\rho}{r}(\underline{b}_r + \bar{b}_r) + \sum_{\substack{\tau^P \in \mathcal{T}^P \\ c^F \in C^F}} c^F[r] x(\tau^P, c^F) \le n_r \tag{4}$$

Constraint (2) enforces that every place has families matched to it according to some suitable configuration. Constraint (3) enforces that every compulsory homogeneous ρ-block is filled with refugee families and that no non-existing optional homogeneous ρ-blocks are filled with refugee families. Constraint (4) enforces that for each service-requirement, only available number of refugees are used. The objective function formulates the total number of families assigned.

It is clear that the number of variables is bounded above by $2r_{\max} + (\hat{\rho}+1)^{r_{\max}} r_{\max}^2 \rho^2$, and the number of constraints by $3r_{\max}^2 \rho^2 + r_{\max}$, which are functions of $r_{\max}$. Thus the problem can be solved in FPT time w.r.t. $r_{\max}$ [25]. The correctness of this approach follows from Claim 1.

Claim 1 ($\star$). *ILP$_1$ admits a solution with value u^* if and only if there is an assignment of families with utility u^*.* □

When preferences or utilities are not equal, parameterization by $r_{\max}$ alone does not yield fixed-parameter tractability: as established by Theorem 2, the case $r_{\max} = c_{\max} = 2$ is NP-hard even in a very restricted case, when there are no lower quotas, families have dichotomous preferences (or binary utilities), and each family finds at most two places acceptable (or of positive utility).

Let us remark that a slightly weaker result (the statement without the condition that each family finds exactly two places acceptable, or has positive utility for exactly two paces) follows via a fairly straightforward reduction from the MATCHING WITH COUPLES problem [10, 17].

Theorem 2 ($\star$)**.** *PARETO-RR and MAXUTIL-RR for $t = 1$ are NP-hard even when $r_{\max} = c_{\max} = 2$ and there are no lower quotas. The result holds for PARETO-RR even if all families have dichotomous preferences and find exactly two places acceptable, and for MAXUTIL-RR even if utilities are binary and each family has positive utilities for exactly two places.*

To tackle the computational intractability of Theorem 2, we focus on the parameter $m + r_{\max}$ and propose an FPT algorithm with this parameterization in Theorem 3.

To prove Theorem 3, we are going to present an algorithm for MAXUTIL-RR that constructs a feasible assignment with maximum utility, or concludes that no feasible assignment exists. By Observation 1, such an algorithm can be used to solve the PARETO-RR problem as well. Let I denote our input instance of MAXUTIL-RR.

We use a two-phase dynamic programming approach based on the following key idea: once we have obtained an optimal assignment σ for a partial instance $\mathcal{J}$, then a small modification to this partial instance results in an instance $\mathcal{J}'$ that admits an optimal assignment σ' that is "close" to σ. By guessing how σ' differs from σ, we can compute σ' efficiently. Let us give a high-level view of our algorithm.

In the first phase, we disregard lower quotas, and starting from an instance with only a single family, we add families one by one. For each $i \in [n]$, let $F_i = \{f_1, \ldots, f_i\}$ denote the set of the first i families, and I_i the instance obtained by restricting I to F_i and setting all lower quotas to zero. Starting from a maximum-utility feasible assignment σ_1 for I_1, we construct a maximum-utility feasible assignment σ_i for $i = 2, \ldots, n$ by slightly modifying the assignment σ_{i-1}.

In the second phase, starting from the instance $\hat{I}_0 = I_n$ without lower quotas, we define a sequence $\hat{I}_1, \ldots, \hat{I}_{\underline{c}_\Sigma}$ of instances where each instance is obtained from the previous one by raising the lower quota of a single place by one in an arbitrary way so long as the lower quotas for I are not exceeded; notice that this implies $I = \hat{I}_{\underline{c}_\Sigma}$ where $\underline{c}_\Sigma = \sum_{p_j \in P} \underline{\boldsymbol{c}}_j$. Then starting from $\hat{\sigma}_0 := \sigma_n$ we compute a maximum-utility feasible assignment $\hat{\sigma}_q$ for $\hat{I}_q$, $q = 1, \ldots, \underline{c}_\Sigma$, from the assignment $\hat{\sigma}_{q-1}$ by applying small modifications.

Theorem 3. *MAXUTIL- and PARETO-RR for $t = 1$ are FPT w.r.t. $m + r_{\max}$.*

Proof. We may assume w.l.o.g. that for all $i \in [n]$, there exists a feasible assignment for I_i with maximum utility that is complete. Indeed, to ensure completeness, we can simply create a dummy place whose upper quota is $\sum_{f_h \in F} \boldsymbol{r}_h$ and towards which all families have zero utility; this also shows that we can assume $m \ge 2$. For brevity's sake, we say that an assignment is *optimal* if it is feasible and complete, and has maximum utility among all feasible assignments.

Let $\rho_m^\star = m^m \cdot \rho \cdot r_{\max}$. Notice that since ρ is a function of $r_{\max}$, we know that $\rho_m^\star$ is a function of m and $r_{\max}$ only.

The first phase of our algorithm relies on Claim 2, which proves that given an optimal assignment σ_i for I_i for some $i \in [n-1]$, we can obtain an optimal assignment for I_{i+1} whose distance from σ_i is bounded by a function of m and $r_{\max}$. We measure the distance of two assignments σ and σ' as the sum of the requirements of all families that are assigned to different places by σ and σ', that is,

$$\Delta(\sigma, \sigma') = \sum \left\{ \boldsymbol{r}_h : f_h \in F_\cap, \sigma(f_h) \ne \sigma'(f_h) \right\},$$

where $F_\cap$ is the intersection of the domains of σ and σ'.

Claim 2 (⋆). *Suppose that $i \in [n-1]$, and let $\sigma_i : F_i \to P$ denote an optimal assignment for I_i. Then there exists an optimal assignment $\sigma_{i+1} : F_{i+1} \to P$ for I_{i+1} such that $\Delta(\sigma_i, \sigma_{i+1}) \leq m \cdot \rho^\star_m$.*

The second phase of our algorithm relies Claim 3 which is an analog of Claim 2 with a quite similar proof.

Claim 3 (⋆). *Suppose that $q \in [\underline{c}_\Sigma]$, and let $\hat{\sigma}_q : F \to P$ denote an optimal allocation for $\hat{I}_q$. Then there exists an optimal allocation $\hat{\sigma}_{q+1} : F \to P$ for $\hat{I}_{q+1}$ such that $\Delta(\hat{\sigma}_q, \hat{\sigma}_{q+1}) \leq m \cdot \rho^\star_m$.*

We are now ready to present our algorithm for MaxUtil-RR based on Claims 2 and 3. We use a combination of dynamic programming and color-coding.

Initially, we compute a maximum-utility feasible allocation σ_1 for I_1 by assigning family f_1 to a place that can accommodate it, and among all such places, yields the highest utility for f_1. Then, in the first phase of the algorithm, for each $i \in [n-1]$ we compute an optimal assignment for I_{i+1} by slightly modifying σ_i. In the second phase, starting from the assignment $\hat{\sigma}_0 := \sigma_n$ for $\hat{I}_0 := I_n$, we compute an optimal assignment for $\hat{I}_q$ by slightly modifying $\hat{\sigma}_{q-1}$ for each $q \in [\underline{c}_\Sigma]$. In each step of the first and second phases, we apply a procedure based on color-coding; the remainder of the proof contains the description of this procedure and its proof of correctness.

Let I_{curr} be the instance of phase 1 or 2 for which we have already computed an optimal assignment σ_{curr}, and suppose that I_{next} is the next instance for which we aim to compute an optimal assignment σ_{next}. Thus, I_{next} is either obtained from I_{curr} by adding some family $f_i \in F$, or by raising the lower quota for one of the places in P by one. Let F_{curr} and F_{next} denote the set of families in I_{curr} and in I_{next}, respectively. Due to Claims 2 and 3, we can choose σ_{next} so that $\Delta(\sigma_{\text{next}}, \sigma_{\text{curr}}) \leq m \cdot \rho^\star_m$.

Guessing step. Let $X(p_j, p_{j'}, r)$ denote the set of all families with requirement r that are assigned to p_j by σ_{curr} but are moved to $p_{j'}$ by σ_{next}. We guess the number $x(p_j, p_{j'}, r) = |X(p_j, p_{j'}, r)|$ for each $p_j, p_{j'} \in P$ and $r \in [r_{\max}]$. By our choice of σ_{next}, we have

$$\sum_{j \in [m]} \sum_{j' \in [m] \setminus \{j\}} \sum_{r \in [r_{\max}]} x(p_{j'}, p_j, r) = \Delta(\sigma_{\text{next}}, \sigma_{\text{curr}}) \leq m \cdot \rho^\star_m.$$

Since we need to guess $m \cdot (m-1) \cdot r_{\max}$ values that add up to at most $m \cdot \rho^\star_m$, there are no more than $(\rho^\star_m)^{m \cdot (m-1) \cdot r_{\max}}$ possibilities to choose all values $x(p_{j'}, p_j, r)$. Thus, the number of possibilities for all our guesses is bounded by a function of m and $r_{\max}$ only.

Color-coding step. We proceed by randomly coloring all families in I_{next} with m colors in a uniform and independent way. We say that a coloring is *suitable* for σ_{next}, if for each $p_j \in P$, all families in $\sigma^{-1}_{\text{next}}(p_j) \setminus \sigma^{-1}_{\text{curr}}(p_j)$ have color j. Thus, in a suitable coloring, each family whose assignment changes between σ_{next} and σ_{curr} must be assigned by σ_{next} to the place corresponding to its color. Considering that I_{next} may contain one more family than I_{curr}, we get

$$\sum_{p_j \in P} \left| \sigma^{-1}_{\text{next}}(p_j) \setminus \sigma^{-1}_{\text{curr}}(p_j) \right| \leq 1 + \Delta(\sigma_{\text{next}}, \sigma_{\text{curr}}) \leq m \cdot \rho^\star_m + 1.$$

Therefore, the probability that the algorithm produces a suitable coloring is at least $m^{-m\rho^\star_m + 1}$.

Modification step. Assume that our coloring χ is suitable. In the first phase, this implies that the unique family $f_i \in F_{\text{next}} \setminus F_{\text{curr}}$ must be assigned by σ_{next} to $p_{\chi(f_i)}$. Thus, we fix the assignment on f_i as $p_{\chi(f_i)}$. We proceed with the remaining families of F_{next} as follows.

For each $p_j, p_{j'} \in P$ and $r \in [r_{\max}]$, we compute the set $D(p_j, p_{j'}, r) := \{f_h \in F_{\text{curr}} : \sigma_{\text{curr}}(f_h) = p_j, \chi(f_h) = j', \boldsymbol{r}_h = r\}$; the suitability of χ means that $X(p_j, p_{j'}, r) \subseteq D(p_j, p_{j'}, r)$. With each family $f_h \in D(p_j, p_{j'}, r)$, we associate the value $\boldsymbol{u}^{j'}_h - \boldsymbol{u}^j_h$ which describes the increase in utility caused by moving a_h from p_j to $p_{j'}$. We order the families in $D(p_j, p_{j'}, r)$ in a non-increasing order of these values, and we pick the first $x(p_j, p_{j'}, r)$ families according to this ordering; denote the obtained set $\widetilde{D}(p_j, p_{j'}, r)$. We can now define σ'_{i+1} as follows for each $f_h \in F_{\text{curr}}$:

$$\sigma'_{\text{next}}(f_h) = \begin{cases} p_{\chi(f_h)} & \text{if } f_h \in F_{\text{next}} \setminus F_{\text{curr}}; \\ p_{j'} & \text{if } \exists j, r : f_h \in \widetilde{D}(p_j, p_{j'}, r); \\ \sigma_{\text{curr}}(f_h) & \text{otherwise.} \end{cases}$$

Observe that the total requirement of all families assigned to some place $p_j \in P$ is the same in σ'_{next} as in σ_{next}, due to the definition σ'_{next} and the correctness of our guesses. Therefore, σ'_{next} is feasible. Furthermore,

$$\sum_{\substack{f_h \in F_{\text{curr}},\\ p_j = \sigma'_{\text{next}}(f_h)}} \boldsymbol{u}_h[j] = \text{util}(p_j, \sigma_{\text{curr}}) + \sum_{\substack{\exists j, j', r:\\ f_h \in \widetilde{D}(p_j, p_{j'}, r)}} (\boldsymbol{u}_h[j'] - \boldsymbol{u}_h[j])$$

$$\geq \text{util}(p_j, \sigma_{\text{curr}}) + \sum_{\substack{\exists j, j', r:\\ f_h \in X(p_j, p_{j'}, r)}} (\boldsymbol{u}_h[j'] - \boldsymbol{u}_h[j]) = \sum_{\substack{f_h \in F_{\text{curr}},\\ p_j = \sigma_{\text{next}}(f_h)}} \boldsymbol{u}_h[j]$$

where the inequality follows from our choice of the sets $\widetilde{D}(p_j, p_{j'}, r)$ and the facts $|\widetilde{D}(p_j, p_{j'}, r)| = |X(p_j, p_{j'}, r)|$ and $X(p_j, p_{j'}, r) \subseteq D(p_j, p_{j'}, r)$, which in turn follow from our assumptions that our guesses are correct and that the coloring χ is suitable. Since σ'_{next} coincides with σ_{next} on $F_{\text{next}} \setminus F_{\text{curr}}$, the above inequality implies that σ'_{next} is a maximum-utility feasible assignment for I_{next}, proving the correctness of our algorithm.

The presented algorithm can be derandomized using standard techniques, based on $(n, m \cdot \rho^\star_m + 1)$-perfect families of perfect hash functions [4]. Since both the number of possible guesses and the number of families that we have to color correctly are bounded by a function of $m + r_{\max}$, the modification procedure applied in the first or second phases of the algorithm runs in FPT time when parameterized by $m + r_{\max}$. As we have to carry out this procedure $n + \underline{c}_\Sigma$ times and we can assume w.l.o.g. that $\underline{c}_\Sigma \leq n \cdot r_{\max}$, the total running time is FPT w.r.t. $m + r_{\max}$. □

We close this section by showing that if the desired total utility u^* is small and there are no lower quotas, then MaxUtil-RR for $t = 1$ can be solved efficiently. Recall that with lower quotas, even the case $u^* = 0$ is NP-hard by Proposition 1. The algorithm of Theorem 4, presented in the full version [13], starts with a greedily computed assignment, and then deletes irrelevant families to obtain an equivalent instance with at most $(u^*)^3$ families that can be solved efficiently.

Theorem 4 (⋆). MaxUtil- *and* Pareto-RR *for $t = 1$ are FPT w.r.t. u^*, the desired utility, if there are no lower quotas.*

4 Multiple services

Let us now consider the model when there are several services, i.e., $t > 1$. We start with a strong intractability result for Feasible-RR. Then we focus in Pareto-optimality, and propose several algorithms that solve Pareto-RR but not MaxUtil-RR, contrasted by tight hardness results. We close by investigating MaxUtil-RR.

Feasibility. When the number of services can be unbounded, a simple reduction from INDEPENDENT SET by Gurski et al. [18, Theorem 23] shows that MAXUTIL-RR is NP-hard even if $m = 1$, there are no lower quotas and the utilities are equal. With a slight modification of their reduction, we obtain Proposition 2 which shows the NP-hardness of FEASIBLE-RR in a very restricted setting.

Proposition 2 (⋆). *The following problems are NP-hard even if* $c_{\max} = r_{\max} = 1$ *and* $m = 1$:

- FEASIBLE-RR;
- PARETO-RR *with equal preferences;*
- MAXUTIL-RR *with equal utilities.*

Pareto-optimality. The reduction from INDEPENDENT SET used by Gurski et al. [18] and also in Proposition 2 can be adapted to show the NP-hardness of PARETO-RR in the case when there are no lower quotas, $m = 2$, and we allow $c_{\max}$ to be unbounded; see Proposition 3. Notice that in instances without lower quotas, a feasible, acceptable and Pareto-optimal assignment always exists. Hence, our hardness results for PARETO-RR rely on the following fact.

Observation 3 (⋆). *We can decide whether an instance I of* PARETO-RR *with dichotomous preferences admits a feasible, acceptable, and complete assignment by solving* PARETO-RR *on I.*

Proposition 3 (⋆). PARETO-RR *and* MAXUTIL-RR *are NP-hard even if* $m = 2, r_{\max} = 1$*, there are no lower quotas, and families have equal preferences or utilities.*

In the reduction proving Proposition 3, the value $c_{\max}$ is unbounded. Next, we show a reduction from 3-COLORING proving that even the case when $c_{\max} = 1$ is NP-hard if there are at least 3 places.

Theorem 5 (⋆). PARETO-RR *and* MAXUTIL-RR *are NP-hard even when* $m = 3, c_{\max} = 1$*, there are no lower quotas, and families have equal preferences or utilities, respectively.*

Contrasting the intractability result of Proposition 3 for $m = 2$, we show that a simple, greedy algorithm solves PARETO-RR for $m = 1$ in polynomial time assuming that there are no lower quotas.

Proposition 4 (⋆). PARETO-RR *for* $m = 1$ *is polynomial-time solvable if there are no lower quotas.*

Our next results shows that PARETO-RR can be solved efficiently if there are only a few families whose preferences contain ties, assuming that there are no lower quotas. Recall that in the presence of lower quotas, PARETO-RR is NP-hard even if $t = 1$ and $n_{\sim} = 0$, i.e., all preferences are strict, as shown in Proposition 1. The algorithm of Proposition 5 first applies serial dictatorship among families whose preferences do not contain ties, and then tries all possible assignments for the remaining families.

Proposition 5 (⋆). PARETO-RR *is FPT w.r.t. the number of families with ties* $n_{\sim}$*, if there are no lower quotas.*

Maximizing utility. Let us start with a simple fixed-parameter tractable algorithm for MAXUTIL-RR w.r.t n, the number of families. Proposition 6 presents an FPT algorithm for parameter n based on the following approach: We first guess the partitioning $\mathcal{F}$ of families arising from a maximum-weight feasible assignment, and then we map the partitions of $\mathcal{F}$ to the places by computing a maximum-weight matching in an auxiliary bipartite graph.

Proposition 6 (⋆). FEASIBLE-, PARETO-, *and* MAXUTIL-RR *are FPT w.r.t.* n.

Let us now present a generalization of Theorem 1 for MAXUTIL-RR restricted to equal preferences. The algorithm for Theorem 6 is based upon an N-fold IP formulation for this problem. By Observation 1, the obtained algorithm also implies tractability for PARETO-RR for equal utilities.

Theorem 6 (⋆). FEASIBLE-RR, PARETO-RR *for equal preferences, and* MAXUTIL-RR *for equal utilities, are FPT w.r.t.* $t + r_{\max}$.

Our next algorithm is applicable in a more general case than Theorem 6 (which works only when preferences or utilities are equal) at the cost of setting $m + t + r_{\max}$ as the parameter. Theorem 6 is based on an ILP formulation that solves PARETO-RR for arbitrary preferences as well as MAXUTIL-RR for a broad range of utilities.

Theorem 7 (⋆). *The following problems are FPT w.r.t. parameter* $m + t + r_{\max}$:

- PARETO-RR,
- MAXUTIL-RR *on instances where the number of different utility values is at most* $g(m+t+r_{\max})$ *for some computable function g.*

Proof sketch. The main idea is that there is no need to distinguish between families that have the same utilities and requirements. Since the number of possible requirement vectors and the number of possible utility vectors are both bounded by a function of the parameter, the number of family types will also be bounded. This allows us to define a variable for each place and family type describing the number of families of a given type assigned to a given place. The resulting ILP contains a bounded number of variables and constraints, and is therefore solvable by standard techniques in FPT time [25]. □

Taking an even stronger parameterization than Theorem 7, namely $m + t + c_{\max}$, yields fixed-parameter tractability: in Proposition 7 we present an algorithm running in $O((c_{\max})^{mt} nm)$ time. This algorithm is a straightforward adaptation of the textbook dynamic programming method for KNAPSACK. The same approach was also used by Gurski et al. [18, Proposition 34] to solve a simpler variant of MAXUTIL-RR without lower quotas and with a single service.

Proposition 7 (⋆). FEASIBLE-, PARETO-RR, *and* MAXUTIL-RR *are in XP w.r.t.* $m + t$ *and are FPT w.r.t.* $m + t + c_{\max}$.

We close this section by mentioning that a simple XP algorithm exists for the case when the parameter is the desired total utility u^*, and there are no lower quotas (cf. Proposition 1 stating the intractability of the case $u^* = 0$ when lower quotas are allowed).

Proposition 8 (⋆). FEASIBLE-, PARETO- *and* MAXUTIL-RR *are in XP w.r.t. the desired utility* u^* *if there are no lower quotas.*

5 Conclusion

We provided a comprehensive parameterized complexity analysis for three variants of REFUGEE RESETTLEMENT, which focus on ensuring feasibility, maximizing utility, and achieving Pareto optimality. There remain some interesting parameter combinations for which the complexity of these problems is open, e.g., is MAXUTIL-RR FPT w.r.t. parameter $m + t + r_{\max}$ for arbitrary utilities? Another exciting line of future research is to explore the possibilities of tailoring the proposed algorithms to efficiently solve practical instances, and determining which parameterizations are the most relevant in different real-world applications. We believe that our ILP algorithms could perform substantially faster then their theoretical bounds, as the current day ILP solvers are efficient. However, it could also be that a straightforward ILP formulation with a variable for each family and each place outperforms our specialized formulations.

Acknowledgements

Jiehua Chen and Sofia Simola are supported by the Vienna Science and Technology Fund (WWTF) [10.47379/VRG18012]. Ildikó Schlotter is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and its János Bolyai Research Scholarship.

References

[1] A. Abdulkadiroğlu. College admissions with affirmative action. *International Journal of Game Theory*, 33:535–549, 2005.

[2] N. Ahani, T. Andersson, A. Martinello, A. Teytelboym, and A. C. Trapp. Placement optimization in refugee resettlement. *Operations Research*, 69(5):1468–1486, 2021.

[3] N. Ahani, P. Gölz, A. D. Procaccia, A. Teytelboym, and A. C. Trapp. Dynamic placement in refugee resettlement. *Operations Research*, 72 (3):1087–1104, 2023.

[4] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995.

[5] T. Andersson and L. Ehlers. Assigning refugees to landlords in Sweden: Efficient, stable, and maximum matchings. *The Scandinavian Journal of Economics*, 122(3):937–965, 2020.

[6] T. Andersson, L. Ehlers, and A. Martinello. Dynamic refugee matching. Technical Report 2018-10, Université de Montréal, Département de sciences économiques, 2018.

[7] H. Aziz, J. Chen, S. Gaspers, and Z. Sun. Stability and Pareto optimality in refugee allocation matchings. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*, pages 964–972, 2018.

[8] H. Aziz, S. Gaspers, Z. Sun, and T. Walsh. From matching with diversity constraints to matching with regional quotas. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)*, pages 377–385, 2019.

[9] K. Bansak and E. Paulson. Outcome-driven dynamic refugee assignment with allocation balancing. *Operations Research*, 2024.

[10] P. Biró and E. McDermid. Matching with sizes (or scheduling with processing set restrictions). *Discrete Applied Mathematics*, 164:61–67, 2014.

[11] P. Biró, T. Fleiner, R. W. Irving, and D. Manlove. The College Admissions problem with lower and common quotas. *Theoretical Computer Science*, 411(34-36):3136–3153, 2010.

[12] J. Chen, R. Ganian, and T. Hamm. Stable matchings with diversity constraints: Affirmative action is beyond NP. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 146–152, 2020.

[13] J. Chen, I. Schlotter, and S. Simola. Parameterized algorithms for optimal refugee resettlement, 2024. arXiv preprint, arXiv:2408.08392.

[14] D. Delacrétaz, S. D. Kominers, and A. Teytelboym. Refugee resettlement. Technical report, 2016. Working paper. Available at http://www.t8el.com/jmp.pdf.

[15] D. Delacrétaz, S. D. Kominers, and A. Teytelboym. Matching mechanisms for refugee resettlement. *American Economic Review*, 113(10): 2689–2717, 2023.

[16] R. Ganian, S. Ordyniak, and C. S. Rahul. Group activity selection with few agent types. *Algorithmica*, 85(5):1111–1155, 2023.

[17] C. A. Glass and H. Kellerer. Parallel machine scheduling with job assignment restrictions. *Naval Research Logistics*, 54(3):250–257, 2007.

[18] F. Gurski, C. Rehs, and J. Rethmann. Knapsack problems: A parameterized point of view. *Theoretical Computer Science*, 775:93–108, 2019.

[19] K. Hamada, K. Iwama, and S. Miyazaki. The Hospitals/Residents problem with lower quotas. *Algorithmica*, 74(1):440–465, 2016.

[20] R. Hemmecke, S. Onn, and L. Romanchuk. n-fold integer programming in cubic time. *Mathematical Programming, Series A*, 137:325–341, 2013.

[21] K. Jansen, S. Kratsch, D. Marx, and I. Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79 (1):39–49, 2013.

[22] D. Knop and S. Schierreich. Host community respecting refugee housing. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*, page 966–975, 2023.

[23] B. Kuckuck, J. Rothe, and A. Weißenfeld. Refugee allocation in the setting of hedonic games. In *Proceedings of the 6th International Conference on Algorithmic Decision Theory (ADT 2019)*, pages 65–80. Springer, 2019.

[24] R. Kurata, N. Hamada, A. Iwasaki, and M. Yokoo. Controlled school choice with soft bounds and overlapping types. *Journal of Artificial Intelligence Research*, 58:153–184, 2017.

[25] H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[26] G. Lisowski and Š. Schierreich. Swap stability in refugee housing: A story about anonymous preferences. *The 10th European Starting AI Researchers' Symposium (STAIRS)*, 2023.

[27] E. J. McDermid and D. F. Manlove. Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization*, 19:279–303, 2010.

[28] H. Nguyen, T. Nguyen, and A. Teytelboym. Stability in matching markets with complex constraints. *Management Science*, 67(12):7438–7454, 2021.

[29] Š. Schierreich. Anonymous refugee housing with upper-bounds, 2023. arXiv preprint, arXiv:2308.09501.

[30] P. Xepapadeas and I. Mourtos. Refugee allocation mechanisms: Theory and applications for the European Union. *Operational Research*, 22(4): 4557–4584, 2022.