Refugee Resettlement*

David Delacrétaz
Department of Economics
University of Melbourne

Scott Duke Kominers
Society of Fellows
Harvard University

Alexander Teytelboym
Institute for New Economic Thinking
University of Oxford

August 1, 2016

Preliminary and Incomplete:
Please do not cite or circulate without permission.

Abstract

Roughly 100,000 refugees are permanently resettled from refugee camps to hosting countries every year. Nevertheless, refugee resettlement processes in these countries are typically *ad hoc*, accounting for neither the needs and capacities of hosting communities, nor the preferences of refugees themselves. In this paper, we propose three refugee resettlement systems based on two-sided matching theory that can be used by hosting countries in different circumstances. The mechanisms we propose can improve match efficiency and reduce internal movement of refugees across localities—and might thereby encourage localities to accept more refugees in total. Additionally, the *combinatorially-constrained matching* framework we develop for our model is of independent theoretical interest and has applications beyond the refugee resettlement context.

*The authors appreciate the helpful comments of Haris Aziz, Alexander Betts, Elizabeth Collett, Carthryn Costello, Cameron Hepburn, Will Jones, Katy Long, Gregory Maniatis, Alvin Roth, Tayfun Sönmez, and seminar participants at the Melbourne Centre for Market Design and Oxford. All three authors gratefully acknowledge the support of the Oxford Martin School. Additionally, Delacrétaz thanks the Faculty of Business and Economics and the Centre for Market Design at the University of Melbourne. Kominers is grateful for the support of National Science Foundation grants CCF-1216095 and SES-1459912, the Harvard Milton Fund, the Ng Fund of the Harvard Center of Mathematical Sciences and Applications, and the Human Capital and Economic Opportunity Working Group (HCEO) sponsored by the Institute for New Economic Thinking (INET). Teytelboym is grateful for the support of the Skoll Centre for Social Entrepreneurship at the Saïd Business School, as well as the for generous fellowship and hospitality of the EU Centre for Shared Complex Challenges at the University of Melbourne.

1 Introduction

There are now 65.3 million people displaced by conflict around the world—the highest level ever recorded (UNHCR, 2016). Over 16 million of these forcibly displaced people are deemed to be *refugees* by the United Nations High Commission for Refugees (UNHCR), and half of the refugees outside Palestine come from just three countries: Syria, Afghanistan and Somalia.[1]

Refugee status is envisioned to be temporary, but the majority of refugees spend decades in camps and temporary settlements in developing countries without residence rights or work permits. Nevertheless, around 1% of refugees are referred by the UNHCR for *resettlement* to states that agree to give them permanent residence.[2] Most resettlement places are provided by the United States, Canada, Australia and the Nordic countries.

In the past, the international community has often managed to react quickly to humanitarian crises by instituting rapid and comprehensive resettlement programs.[3] In recent years, however, following escalating conflict in the Middle East, the influx of refugees into Europe has drastically increased. In 2015, a million people arrived in Greece and Italy, seeking asylum in Europe (International Organization for Migration, 2015). Germany, for example, experienced one of the highest influxes of migrants in its post-war history—and for the first time since records began the number of its non-European migrants exceeded the number of European migrants (Statistisches Bundesamt, 2016).

Given the unprecedented current scale of refugee arrival, existing policies designed to manage refugee flows have effectively collapsed. The Dublin III regulation—requiring that "irregular" refugees must be processed in the first EU country in which they arrive—has been effectively abandoned.[4] Consequently, numerous countries within the EU and elsewhere have begun to reconsider the systems they use to register, process, and allocate refugees to localities.[5]

[1] In international law, the term *refugee* was defined by the 1951 UN Convention Relating to the Status of Refugees to mean anyone who has left his or her home owing "to a well-founded fear of persecution because of his/her race, religion, nationality, membership in a particular social group, or political opinion." More recently, the definition has been expanded to include those fleeing natural and man-made disasters. An *asylum seeker*, on the other hand, is someone whose claim for refugee protection has not been evaluated by any country or by the UNHCR. Often, asylum seekers who are not registered because they arrive during a mass movement of people due to conflict are referred to as "prima facie refugees." In this paper, we will use the term "refugee" to refer to any person who is explicitly or implicitly granted legal sanctuary in another country for any reason.

[2] In 2015, 134,000 files were submitted by the UNHCR for resettlement and 81,000 refugees were formally resettled.

[3] Examples of rapid resettlement include 2.5 million Indochinese refugees to a number of Western states between 1975-1977 and 60,000 Bhutanese refugees who left Nepal for the US in 2006.

[4] Article 13(1) of the EU Parliament and Council Regulation 604/2013 of 26 June 2013 states that "Where it is established, on the basis of proof or circumstantial evidence [...], that an applicant has irregularly crossed the border into a Member State by land, sea or air having come from a third country, the Member State thus entered shall be responsible for examining the application for international protection. That responsibility shall cease 12 months after the date on which the irregular border crossing took place." But on 21 August 2015 it was reported that "Germany's Federal Office for Migration and Refugees (BAMF) suspended the otherwise obligatory examination which tests whether asylum seekers first entered the EU in another member state and whether they should be returned to that country" (Dernbach, 2015).

[5] Most asylum seekers are held in detention centers until their refugee status applications have been

Economists, refugee specialists and policymakers have developed a number of promising solutions for sharing the burden of refugee resettlement across countries, including tradeable quota systems (Schuck, 1997, Moraga and Rapoport, 2014). However, the process of resettling refugees to localities has received little attention, despite ample evidence that the locality to which a refugee is assigned matters a great deal for that refugee's lifetime outcomes (Åslund and Rooth, 2007, Åslund and Fredriksson, 2009, Åslund et al., 2010, 2011, Damm, 2014). Most countries have historically treated refugee resettlement as a purely administrative issue, and as such have not developed systematic resettlement policies—much less, transparent ones. In this paper, we take the process of the allocation of refugees to localities seriously, introducing and analyzing several market design approaches that balance competing welfare, incentive and stability objectives.

One simple policy goal is to try and maximize welfare based on observables. Meeting that objective alone requires solving a complex *knapsack* problem, with respect to quality scores that reflect the estimated quality of matches between refugees and localities. This set-up can allow us, for example, to maximize the total number of refugees or the total number of families resettled. Pure knapsack maximization, however, does not account in any way for the refugees' personal preferences over hosting communities. Refugees' preferences matter for two reasons. First, refugees have a lot of information about their own aspirations, which can affect the match quality—and which cannot be directly observed by government authorities. Second, government agencies spend significant upfront resources on integrating refugees into local communities; those resources are effectively wasted if refugees leave their assigned local areas soon after arrival. Consequently, at least in the initial stages of the resettlement process, policymakers want to minimize internal migration of refugees away from their initially assigned localities. But the priorities and hosting capacities of localities matter too; this is particularly salient when the participation of localities into resettlement is voluntary, in which case giving localities a say might serve as an additional incentive to participate.

All three of the aforementioned policy scenarios have played out in the British program to resettle Syrian refugees. Britain is similar to other popular resettlement destinations, such as Australia, Canada and the USA, in that is geographically remote and thus able to control refugee flows. Hence, the priority for Britain is the careful placement of the relatively small number of refugees that it accepts. As soon as Britain dramatically increased its target for resettling Syrian refugees, the Home Office's focus was on developing and unifying technological solutions to organize and process assignment, without any specific attention to preferences and priorities. After early numerical targets were hit, the Home Office has begun to turn its attention to more systematic matching, in order to both enhance welfare and reduce difficult-to-track internal migration.[6] Initially, localities around the UK showed

approved. If a refugee's asylum claim is approved, that refugee is granted formal (i.e., legal) refugee status and released. (If a refugee's asylum claim is not approved, that refugee is usually deported.) Many countries provide refugees with accommodation and welfare support, via a process known as "dispersal." Most of the insights in this paper can be applied directly to refugee dispersal schemes.

[6]The scheme was originally envisioned to have two stages: "Phase one has the task of immediately scaling up the existing resettlement program and phase two will work towards transforming our resettlement and protection offer, including developing ideas for community sponsorship as per the Home Secretary's commitment" (Home Office, 2015, p. 2).

a lot of willingness to accept refugees and dozens volunteered to accept them. However, anti-immigration tensions in the UK have been increasing—especially after the British referendum to leave the EU—and the Home Office might have to focus on locality priorities as well, in order to entice localities into participating. In our work, we follow the evolution of British refugee policy, describing the appropriate market design solution at each step.

Our work draws upon classical mechanisms from contexts, such as school choice, as well as on operations research techniques. However, each solution requires integrating a particular feature of the refugee resettlement problem that has not been present in previous matching market design applications: a refugee family not only takes up a house in a particular locality, but also a certain number of units of different public services, such as school places, hospital beds, slots in language classes and employment training programmes. Thus, there are explicit *combinatorial budget constraints* that limit the central authority's ability to allocate refugees to localities simply on the basis of housing needs; these additional constraints render standard market design systems for house or object allocation insufficient.

In the case in which neither preferences nor priorities are taken into account, our setting can be analyzed as an integer program called the *multiple multi-dimensional knapsack problem*. This problem is NP-hard, but the optimal solutions can be found reasonably quickly for small instances and approximate solution can be found using familiar methods from other knapsack problems (Song et al., 2008).

In the case in which the market designer fully incorporates the preferences of refugee families, the natural choice for an efficient and strategy-proof mechanism would be the top trading cycles algorithm (Shapley and Scarf, 1974, Roth and Postlewaite, 1977, Abdulkadiroğlu and Sönmez, 1999, 2003). However, the combinatorial constraints on service provision means that we cannot simply use the standard top trading cycles algorithm over houses. Instead we modify top trading cycles to run over multidimensional capacities of localities, in order to obtain an efficient mechanism in which refugee families do not have any incentive to misreport their preferences.

When the priorities of localities also need to be taken into account as part of the design, new trade-offs arise. In particular, *stable* outcomes—i.e., outcomes that fully respect the priorities of localities and the preferences of refugees—may not exist, because of a certain complementarity induced by the combinatorial resource constraints.[7] In this case, however, building on insights by Delacrétaz (2016), we develop an algorithm that finds a stable outcome whenever such an outcome exists, and is undominated by any other stable outcome.

Moraga and Rapoport (2014) first proposed matching markets for refugee resettlement as a part of international refugee quota trading. In the international context, however, refugee matching can be reasonably modeled as a standard school choice problem, and thus does not require much new theory. In a policy piece, Jones and Teytelboym (2016) introduced the idea of refugee resettlement matching in the national context, and pointed out the combinatorial constraints that arise on the local level. This paper builds theory to support Jones and Teytelboym's (2016) ideas.

Our work draws on and contributes to the applied literature on the design and implementation of matching mechanisms. The most closely-related literature has focused on design

[7]In one-sided object allocation settings such as school choice, stability is often referred to as "elimination of justified envy" (Abdulkadiroğlu and Sönmez, 2003), or just "fairness" (Balinski and Sönmez, 1999).

of many-to-one markets in which agents on the "many" side take up individual slots made available by the "one" side: residents take individual jobs at hospitals (Roth, 1984a); children take up individual seats at schools (Balinski and Sönmez, 1999, Abdulkadiroğlu and Sönmez, 2003); families occupy individual houses (Abdulkadiroğlu and Sönmez, 1999); and cadets are matched to individual branches of service (Sönmez and Switzer, 2013). Even when agents take up slots a multiple levels—as in the Japanese residency match, in which agents occupy slots at hospitals, and also count against regional quotas (Kamada and Kojima, 2012, 2015)—the slots are assumed to have a hierarchical structure.[8] With appropriate conditions on the preferences of two sides (such as "responsiveness" (Roth, 1985, Roth and Sotomayor, 1989, 1990) or "substitutability" (Kelso and Crawford, 1982, Roth, 1984b, Hatfield and Milgrom, 2005), classical many-to-one matching settings have stable outcomes, as well as elegant structure. In our setting, because refugee families may take up *several* units of *different* services, we deal with a new *combinatorially-constrained matching problem*. We cannot ensure the existence of stable outcomes—and thus, we are unable to directly apply classical matching technologies and mechanisms.[9]

Our work thus contributes to a growing literature that proposes matching mechanisms for settings in which stable outcomes may not exist. The most famous example is *matching with couples* in the National Resident Matching Program, in which residents may view pairs of hospitals as complementary (Roth and Peranson, 1999, Kojima and Pathak, 2009, Ashlagi et al., 2014, Hatfield and Kominers, 2016). There are a number of algorithms that can find stable couples matchings whenever they exist (Echenique and Yenmez, 2007, Kojima, 2015). However, the structure of our problem is different to the matching with couples problem as the barriers to stability arise from the "one" side of the market, rather than from the "many" side (as in the couples problem). Another "difficult" case for market design has been matching with minimum quotas. In this case, stability is incompatible with non-wastefulness (Goto et al., 2014), but second-best stability and non-wastefulness properties can still be satisfied alongside strategy-proofness (Fragiadakis et al., 2016). Milgrom and Segal (2014) study an auction setting in which mobile network operators have arbitrary preferences over sets of TV stations that supply spectrum bandwidth, and the allocation must satisfy complex interference constraints. The main difference between our setting and Milgrom and Segal's (2014) is that we limit localities (which correspond to mobile network operators, in the Milgrom and Segal (2014) context) to have responsive preferences, while allowing refugees (TV stations, in the Milgrom and Segal (2014) context) to have heterogeneous preferences over localities (rather than only caring about compensation, as in the Milgrom and Segal (2014) context).

Even when stable matchings exist in our setting, respecting preference and priority rules through stability may create significant welfare losses (Abdulkadiroğlu et al., 2009, Kesten, 2010). Like Erdil and Ergin (2008) and Kesten (2010), we seek to strike a balance between stability and efficiency goals: Our third algorithm ensures that any stable outcome it finds is (Pareto-)undominated by any other stable outcome.

In many market design settings (especially school choice) in which efficiency is prioritized

[8]So whenever a resident take up a place in a hospital, she would also take up a place in the region in which the hospital is located.

[9]From the point of view of localities, refugee families are neither weak substitutes (Hatfield and Kojima, 2008), nor even substitutes and symmetric complements (Alva, 2015).

over stability by the social planner, top trading cycles algorithms or their modifications (e.g., Dur and Ünver (2015)) are used instead (Abdulkadiroğlu and Sönmez, 2003). Pycia and Ünver (2016) show that in settings in which agents have single-unit demands over objects, all group strategy-proof and Pareto efficient mechanisms may be represented in terms of a general class of "trading cycles" mechanisms; following that intuition, we show that an extension of the top trading cycles mechanism achieve a Pareto efficient and strategy-proof outcome for refugees even in a setting with multi-unit, multi-dimensional demands.

Finally, our paper is related to market design applications in which the size or the aggregate quality of the overall matching matters. One such example is kidney exchange, in which maximizing the total number of kidneys exchanged for donation is of first-order importance (Roth et al., 2004, 2005a,b, 2007). Maximizing the number of high-quality matches is also a priority in adoption exchanges (Slaugh et al., 2016). More recently, and sharing some motivation with our work, Andersson and Ehlers (2016) examine a market for allocating private housing to refugees in which landlords have preferences over refugee family size and native language. Andersson and Ehlers (2016) show that as long as landlords prefer larger families compatible with their home sizes, stable and maximal matchings can be found. Krysta and Zhang (2016) have examined a similar housing market, but with knapsack constraints.[10]

The remainder of the paper is organized as follows. In Section 2, we describe the institutional setting of refugee resettlement, paying particular attention to the response of the British government to the current refugee crisis. We introduce the formal model in Section 3. In Section 4, we show how the case in which neither preferences nor priorities are taken into account can be solved as a multiple multi-dimensional knapsack problem. In Section 5, we show how the modified top trading cycle algorithm solves the case in which refugee preferences are fully accounted for. In Section 6, we present a solution to the case where localities priorities need to be respected as well. We conclude and offer suggestions for future work in Section 7. All proofs and technical descriptions of the algorithms are presented in the Appendix.

2 Institutional Context

In January 2014, the British government launched the Syrian Vulnerable Persons Resettlement (VPR) Programme alongside the UNHCR resettlement programme in order to aid efforts to resettle refugees fleeing the civil war in Syria. On September 7, 2015, the former British Prime Minister David Cameron announced that the United Kingdom would extend this programme and resettle 20,000 refugees in Britain by 2020. By April 2016, 1,854 Syrians had been resettled in Britain (House of Commons Library, 2016).

In the UK, the powers of refugee resettlement (like many other powers, including healthcare and education) are devolved from Westminster (central government) to Wales, Scotland and Northern Ireland. In England, where the bulk of the resettlement is expected to happen, local administration takes place at the level of 353 *Local Authorities* (LAs).[11] LAs are

[10] In fact, a number of papers use optimization techniques, such as linear and integer programming, to find welfare-maximizing outcomes of many-to-one matching markets (Roth et al., 1993, Baïou and Balinski, 2000, Sethuraman et al., 2006, Featherstone, 2014, Ashlagi and Shi, 2015, Bodoh-Creed, 2016).

[11] These administrative local government units (that we refer to as localities following this section) go by

responsible for the disbursement of social benefits, provision of social housing, primary and secondary education, waste collection and local amenities, such as parks and libraries.

Throughout the UK, the participation of LAs in the Syrian VPR is voluntary. At the outset, dozens of LAs declared their intention to host Syrian refugees.[12] To meet David Cameron's target, however, many more English LAs would have to become active in hosting refugees.

Syrian refugees who come to Britain are granted five-year leave to remain (after which they would be eligible to apply for permanent residence) and are given full resource to public funds. They are free to move to any part of the UK. Most refugees are housed in private accommodation and their rent is supported by the centrally administered housing benefit. State-run schools are also free, and school places are allocated by sibling and catchment area priority via popular school choice mechanisms (Pathak and Sönmez, 2013). Healthcare in Britain is publicly funded and free at the point of use to any UK resident. Thus, most of the longer-run costs of refugee resettlement—housing, unemployment and disability benefits, healthcare and education—are borne by the state. However, short-run costs—including language support, welfare, community support, and help with finding local employment—fall on LAs.[13] It is not surprising therefore that LAs prefer to accept only the refugees that they can support well. Likewise, it is natural that each LA wants the refugees it supports to remain local, instead of moving away—otherwise the LA's fixed-costs are sunk, with no future return.

As with other resettlement schemes, refugee families who have expressed a wish to be resettled are referred to British authorities by the UNHCR.[14] The British government then

different names, such as "counties", "districts" and "boroughs", depending on their location and history. For unitary authorities, all local governments duties are located under one roof whereas counties and districts/boroughs usually split their duties: for example, counties are in charge of police, while housing would be under districts/boroughs. For the purposes of this paper, these distinctions are not particularly relevant as ultimately we are only concerned with the provision of a service in a particular area. From the perspective from residents, all levels of government cooperate fairly seamlessly to deliver a full range of services. Scotland, Wales and Northern Ireland have similar systems of local government.

[12]Thus far, 71 LAs in England have accepted refugees, as have half of Scottish LAs. By April 2016, one third of all refugees on the Syrian VPR scheme came to Scotland.

[13]According to leaked government documents, the first-year costs of resettlement to LAs themselves are estimated to be £8,520 per person. The central government also expects to cover £12,700 in social benefits per adult, £5,500 for a child's schooling and £2,200 in medical expenses (Dedman, 2015) per person. LAs are compensated in full for their resettlement costs in the first year (House of Commons Library, 2016), but are expected to shoulder costs in subsequent years. With sunk costs of resettlement behind them and some refugees at work, costs after the first year may be significantly lower, but there is virtually no evidence yet for what they are likely to be. The recent government initiative for community sponsorship suggests that organization plan for annual funding of £4,500 per adult which is supposed to cover "the initial provision of cash on arrival and to fund English language tuition and interpretation costs" (Home Office, 2016b).

[14]"The UNHCR identifies people in need of resettlement based on the following criteria: women and girls at risk; survivors of violence and/or torture; refugees with legal and/or physical protection needs; refugees with medical needs or disabilities; children and adolescents at risk; persons at risk due to their sexual orientation or gender identity; and refugees with family links in resettlement countries. Individuals are not specifically identified for resettlement based on their membership of Yazidi, Druze, Christian or other communities but members of those communities may well meet one of the other vulnerability criteria set out by UNHCR. The UNHCR identifies and proposes Syrian refugees for the Vulnerable Persons Scheme scheme from among the whole of the registered refugee population in the region, over 4 million people. This includes people in formal refugee camps, informal settlements and host communities." (House of Commons Library, 2016)

reviews the files, including medical records, and carries out its own security and background checks as well as interviews. The Home Office and the UNHCR try to keep families together via linked independent applications if necessary.[15] The file is then sent to the Home Office who match the suitable accommodation supplied by LAs. Refugees have no say as to where in Britain they go, and LAs often refuse to take certain applicants. What counts are "suitable" accommodation might vary from one LA to another.[16] Once accommodation has been found and a LA is ready to receive a given refugee family, the family is informed about where it is going and the Home Office arranges transport. To reduce costs, the Home Office usually charters flights that carry several dozen families. The referral and application process typically takes between six weeks and three months.[17]

The need for an efficient matching system comes at a time when the government is facing increasing scrutiny over the use of its resources and public hostility towards immigration.[18] There are some technical obstacles, too: databases on refugees characteristics and preferences, as well as local housing and services availability, need to be merged, while some logistics of the resettlement process need to be reworked (e.g., when refugees are interviewed, and how housing availability is projected). For example, in order to facilitate information sharing on housing across LAs and ensure best possible matches, English LAs already work alongside nine regional coordinators. However, given the recent evidence that the initial placement of refugees in less desirable areas has negative lifetime impacts on their labor (Åslund and Rooth, 2007, Damm, 2014), welfare dependence (Åslund and Fredriksson, 2009) and educational achievement (Åslund et al., 2011), it is becoming apparent that improving the matching of refugees to LAs can deliver a lot of value.

3 Model

There is a set of *refugee families* F, in which a typical element f has size $|f|$, as well as a set of *houses* H. Houses are partitioned into a set of *localities* L; we denote the set of houses in locality $\ell \in L$ by H_ℓ, and write $\boldsymbol{\ell}(h)$ for the locality of house $h \in H$.

A *(housing) contract* is a family-house pair (f, h). For any set of contracts $Z \subseteq F \times H$, we define Z_f, Z_h, and Z_ℓ to be the sets of contracts in Z associated to family f, house h, and locality ℓ, respectively. A contract is *feasible* if its house meets the legal requirements to accommodate the associated family. We denote by $X \subseteq F \times H$ the set of feasible contracts, which we assume to be nonempty. We also introduce the notations $\boldsymbol{f}(x)$, $\boldsymbol{h}(x)$ and $\boldsymbol{\ell}(x) \equiv \boldsymbol{\ell}(\boldsymbol{h}(x))$ for the family, house, and locality associated with contract $x \in X$. We also use the convention that $\boldsymbol{f}(\emptyset) \equiv \boldsymbol{h}(\emptyset) \equiv \boldsymbol{\ell}(\emptyset) \equiv \emptyset$.

[15] The typical family size is six. Around half of the applicants are under 18; about half are women (Home Office, 2016a).

[16] For example, some LAs allow teenage siblings of different sexes to share a room, others do not.

[17] The Home Office tries to match refugees immediately as they come through the application pipeline. Hence, refugees who have been waiting longest typically receive the highest priority. However, as the housing market is fast-moving, matches to available houses often need to be made immediately.

[18] Surveys indicated that the proportion of British public that thought that Britain should take fewer Syrian and Libyan refugees increased from 31 percent in September 2015 to 41 percent in January 2016 (BBC, 2016). In July 2016, Theresa May, the incoming British Prime Minister, eliminated the Home Office "Minister for Syrian Refugees" position.

On top of the matching set-up just described, the refugee matching problem has *combinatorial constraints* on the set of feasible allocations: In addition to housing, refugee families require (non-housing) *services* (e.g., hospital beds, school places, language support) from a set S. We denote by ν the matrix of family *service requirements*, with typical element $\nu_s^f \in \mathbb{Z}_{\geq 0}$ denoting the number of units of service s required by family f. A refugee family f can only live in locality ℓ if ℓ can provide (non-housing) services to meet f's needs. We denote by κ the matrix of locality *service provision*, with typical element $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ denoting the number of units of service s locality ℓ can provide.

An *outcome* $Y \subseteq F \times H$ is a set of contracts. We denote by $\tau_s^\ell(Y) \equiv \sum_{x \in Y \cap X_\ell} \nu_s^{\boldsymbol{f}(x)}$ the number of units of service s demanded at locality ℓ under outcome Y. Let $\tau(Y) \equiv (\tau_s^\ell(Y))_{\ell \in L, s \in S}$ be the matrix of *service demands* at outcome Y. An outcome Y is feasible if $Y \subseteq X$, $|Y \cap X_f| \leq 1$ for all f, $|Y \cap X_h| \leq 1$ for all h and $\tau(Y) \leq \kappa$. We call $\tau(Y) \leq \kappa$ the *feasibility constraint* on outcome Y.

We say that a locality ℓ can *accommodate* a family f in outcome Y if there exists a house $h \in H_\ell$ such that $(f, h) \cup Y$ is feasible.

Example. Throughout this paper we will use an example with five families, seven houses, four localities (three of which have two houses and one of which has one house), and two services.

- Families: $F = \{f_1, f_2, f_3, f_4, f_5\}$
- Localities: $L = \{\ell_1, \ell_2, \ell_3, \ell_4\}$
- Houses: $H_{\ell_1} = \{h_{11}, h_{12}\}$, $H_{\ell_2} = \{h_{21}\}$, $H_{\ell_3} = \{h_{31}, h_{32}\}$, $H_{\ell_4} = \{h_{41}, h_{42}\}$
- Service requirements:

$$\kappa = \begin{array}{c} \\ \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{pmatrix} 4 & 2 \\ 3 & 2 \\ 2 & 2 \\ 2 & 2 \end{pmatrix} \end{array}$$

- Service provision:

$$\nu = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 2 \\ 1 & 1 \\ 3 & 0 \end{pmatrix} \end{array}$$

Clearly, for example, ℓ_4 cannot accommodate f_5 in any outcome since $\nu_{s_1}^{f_5} > \kappa_{s_1}^{\ell_4}$. Now, we consider an outcome Y, in which f_1, f_2, f_3 and f_4 are matched to h_{11}, h_{12}, h_{21} and h_{41}

respectively. The corresponding matrix of service demands for services at localities would be as follows:

$$\tau(Y) = \begin{array}{c} \\ \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{array} \begin{array}{c} \begin{array}{cc} s_1 & s_2 \end{array} \\ \begin{pmatrix} 3 & 1 \\ 0 & 2 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \end{array}$$

Clearly, as $\tau(Y) \leq \kappa$, the outcome Y is feasible.

---

4 Maximizing the number of resettled refugees

In this section, we consider the case in which the preferences of refugees and the priorities of localities are not elicited, as is often the case in rapid responses to unfolding humanitarian catastrophes or indeed in many refugee resettlement schemes around the world. For example, during the Kosovo airlift in 1999 many Kosovar-Albanian refugees boarded rescue planes without knowing where they were going (Wells, 1999). More generally, eliciting preferences requires a sophisticated IT infrastructure for sharing data across localities as well as time and resources in order to conduct interviews with the refugees.[19] Instead, the policymaker could estimate the quality of matches based on observable data and on past experience.

We summarize estimated quality of each refugee-house match as a single number called the *quality score*, $q : X \to \mathbb{R}$. Two special cases of q are worth emphasizing: Setting $q = 1$ (or any other constant) would maximise the total number of families that are resettled, while setting $q((f,h)) = |f|$, i.e. the total number of refugees in each family, would maximize the total number of refugees that are resettled. However, the quality score can be quite general. For example, the policymaker could use survey data on satisfaction with housing and locality service as well as on lifetime outcomes in order to determine which families tend to fare best in what housing and locality. In order to maximize the overall observed efficiency of the match within the feasibility constraints, the policymaker solves the following *outcome-quality maximization problem*:

$$\max_{Y \subseteq X} \sum_{y \in Y} q(y) \quad \text{subject to:} \quad Y \text{ is feasible.} \tag{1}$$

Let us now state the problem of maximizing the overall observed efficiency of the match as an integer program. Let us introduce a binary variable $\iota(f,h)$ which is equal to 1 if a contract (f,h) has been selected (family f has been assigned house h in locality $\boldsymbol{\ell}(h)$) and 0 otherwise.

[19] Even in developed countries that are accustomed to refugee resettlement, this infrastructure is often lacking.

$$\max \sum_{f \in F} \sum_{h \in H} q((f,h))\iota(f,h) \quad \text{subject to:}$$

$$\begin{aligned}
\sum_{f \in F} \sum_{h \in H_\ell} \nu_s^f \iota(f,h) &\leq \kappa_s^\ell && \forall \ell, s \\
\sum_{f \in F} \iota(f,h) &\leq 1 && \forall h \qquad (2) \\
\sum_{h \in H} \iota(f,h) &\leq 1 && \forall f \\
\iota(f,h) &\in \{0,1\} && \forall f, h;
\end{aligned}$$

Problem (3) is an example of a *0-1 multiple multi-dimensional knapsack problem* (Song et al., 2008).

Proposition 1. *The outcome-quality maximization problem* (1) *is equivalent to a 0-1 multiple multi-dimensional knapsack problem* (3)*.*

In the 0-1 multiple multidimensional knapsack problem, there are a number of multi-dimensional objects that we want to pack into one of many knapsacks which have multi-dimensional capacities.[20] Different objects yield different profits in each knapsack and the objective is to maximize the total profit, subject to not exceeding the capacity constraints and placing any one object in at most one knapsack (hence, "0-1"). Song et al. (2008) analyzed this problem (also with two-dimensional objects as in our Example) in the case of spectrum allocation in radio networks; they provide both an algorithm for an exact solution and offer a heuristic algorithm for an approximation of the solution.

Corollary 1. *Using the algorithms developed by Song et al. (2008), the outcome quality maximization problem can solved:*

- *exactly—using a "branch and bound" algorithm—by determining an upper bound by decomposing 0-1 multiple multidimensional knapsack problem into several simpler, standard knapsack problems, and*
- *approximately—using a heuristic algorithm with a tight performance ratio of* $\frac{1}{2}$*.*

Example. We now extend our earlier example to illustrate a solution to outcome-quality maximization problem. First let us assume that (f_1, h_{12}), (f_1, h_{21}), (f_1, h_{41}), (f_2, h_{12}), (f_3, h_{11}), (f_4, h_{11}), (f_4, h_{41}) and (f_5, h_{11}) are the only infeasible contracts. Second, let us introduce the

[20] This is different from a multidimensional knapsack problem in which there is only one knapsack (Fréville, 2004); from the multiple knapsack problem in which objects vary in size along only one dimension (Martello and Toth, 1980); and from the multiple-choice knapsack problems in which exactly one object from each of the many mutually exclusive classes must be used (Sinha and Zoltners, 1979, Kellerer et al., 2004, Hifi et al., 2004).

following quality score function q over the set of feasible contracts X:

$$q((f,h))$$

	h_{11}	h_{12}	h_{21}	h_{31}	h_{32}	h_{41}	h_{42}
f_1	71	$\emptyset$	$\emptyset$	8	23	$\emptyset$	38
f_2	46	$\emptyset$	49	22	30	21	91
f_3	$\emptyset$	52	68	31	43	7	20
f_4	$\emptyset$	4	75	40	96	$\emptyset$	36
f_5	$\emptyset$	92	41	48	34	6	9

The outcome of outcome-quality maximization problem is:

$$Y^{OQMP} = \{(f_1, h_{11}), (f_2, h_{42}), (f_3, h_{21}), (f_4, h_{32}), (f_5, h_{12})\}.$$

This can be verified in two simple steps. First, notice that Y^{OQMP} is feasible since all five contracts are feasible, ℓ_1 can accommodate both f_1 and f_5, ℓ_2 can accommodate f_3, ℓ_3 can accommodate f_2 and ℓ_4 can accommodate f_4. Second, observe that Y^{OQMP} generates a larger aggregate quality score than any one-to-one match between families and houses. This can be verified by inspection of the match quality matrix, by using the Hungarian algorithm or linear programming techniques. Since any feasible outcome is one-to-one, Y^{OQMP} then generates a higher aggregate quality score than any feasible outcome and is therefore optimal.

5 Accounting for Refugees' Preferences

Once a refugee crisis has been brought to the world's attention by the media, there is generally an outpouring of goodwill from many countries that decide to take on refugees.[21] While initial refugee resettlement responses are organized in a rush and on a shoestring, as countries allocate more resources to refugee resettlement, authorities gain the bandwidth necessary to elicit refugees' preferences, typically through an interview process. Incorporating refugees' preferences into the matching process is valuable because it can help prevent *internal migration*—in which refugees move away from their assigned localities soon after arriving—which localities want to avoid because they make substantial upfront investments in the refugees they host.

When locality participation is (relatively) secured, refugee resettlement takes the form of a one-sided matching problem among the refugees; in such a setting, we can maximally account for refugees' preferences by seeking a Pareto efficient outcome. Pareto efficiency also limits internal migration across participating localities by ensuring that families will not want to swap localities.

To achieve Pareto efficiency, we build upon the classical top trading cycles mechanism, which is Pareto efficient and strategy-proof (see Shapley and Scarf (1974), Abdulkadiroğlu

[21] In Britain, liberal-leaning newspapers celebrated the arrival of fifteen Syrian refugee families to the remote Scottish Isle of Bute (McKenna, 2015).

and Sönmez (2003), Pycia and Ünver (2016)). Because both refugee preferences and feasibility constraints act at the locality level, our mechanism must identify and eliminate potential Pareto-improving locality trades. However, as refugees are assigned to localities at the house level, our mechanism must clear trades at the house level, rather than the locality level.

We retain the quality score function defined in the prior section, and moreover assume that no two contracts have the exact same quality score (for any $x, x' \in X$, $q(x) = q(x')$ if and only if $x = x'$).[22] Additionally, we introduce notation for families' preferences over localities: We denote by $\succ_f$ the strict ordinal preference list of family f over $L \cup \{\emptyset\}$, and let $\succ \equiv (\succ_f)_{f \in F}$ be the preference profile of families.[23] Given the set of contracts X, we say that a locality ℓ is family f's *top-choice* if it is f's favorite locality among those that have at least one house with which f has a feasible contract. We write $\overline{\ell}_f$ for f's top-choice and $\overline{F}_\ell$ for the set of families that have ℓ as their top-choice. Formally, $\ell = \overline{\ell}_f$ if there exists $h \in H_\ell$ such that $(f, h) \in X$ and $\ell \succ_f \ell'$ for any ℓ' such that $(f, h') \in X$ for some $h \in H_{\ell'}$. $\overline{F}_\ell \equiv \{f \in F \mid \ell = \overline{\ell}_f\}$.

An outcome Y is *Pareto efficient* if it is feasible if there is no feasible outcome Y' such that for all f, either $\boldsymbol{\ell}(Y'_f) = \boldsymbol{\ell}(Y_f)$ or $\boldsymbol{\ell}(Y'_f) \succ_f \boldsymbol{\ell}(Y_f)$. In other words, if one refugee family gets a more preferred outcome than in a Pareto efficient outcome then another family must be worse off. A *mechanism* is a function $\varphi : (\succ, \pi) \to 2^X$. A mechanism φ is *strategy-proof* if there does not exist a report $\succ'_f$ such that $\boldsymbol{\ell}(\varphi((\succ_{-f}, \succ'_f), \pi)_f) \succ_f \boldsymbol{\ell}(\varphi(\succ, \pi)_f)$. Strategy-proofness requires that refugee families cannot make themselves better off by misreporting their preferences over localities.

Multidimensional Top Trading Cycles (MTTC) Algorithm

Round 1: We start with an empty current outcome. Each family f points at its top-choice locality $\overline{\ell}_f$ that can accommodate it given all other families are unmatched. A family does not point at a locality if no locality can accommodate it. Once f points $\overline{\ell}_f$, $\overline{\ell}_f$ tentatively allocates f to one of its houses with the highest quality score for f. Then looking at all the houses that have been tentatively allocated to the families, the locality points at a family with the highest quality score among the tentatively allocated houses. At least one cycle appears. Any locality can be in at most one cycle. Families in the cycle are matched to their top-choice localities that they pointed at and the corresponding contracts are added to the current outcome. The service capacities of the localities in each cycle are reduced by the service requirements of the families that have been matched to them.

Round $j > 1$: Each family points at its top-choice locality that can accommodate it given the current outcome. A family does not point at a locality if no locality can accommodate it given the current outcome. Once a family points a locality, the locality tentatively allocates the family to one of its houses with the highest quality score for that family. Then looking at all

[22] Here, the quality score may be used to encode localities' preferences over the refugees; the "no ties" condition then corresponds to a social planner's assessment of which localities' preferences take priority in different contexts.

[23] We assume without loss of generality that every family least preferred option is being unmatched.

the houses that have been tentatively allocated to the families, the locality points at one of the remaining unmatched families with the highest quality score among the tentatively allocated houses. At least one cycle appears. Any locality can be in at most one cycle. Families in the cycle are matched to the localities that they pointed at. The service capacities of the localities in each cycle are reduced by the service requirements of the families that have been matched to them.

MTTC algorithm terminates when none of families and localities have any feasible contracts to point at. The remaining families are left unmatched.

Proposition 2. *MTTC is strategy-proof and yields a Pareto-efficient outcome.*

MTTC works in much the same way as the classical top trading cycles mechanism. The main difference with the classical TTC is that even though the family is pointing at a locality it needs to be allocated a house when it is part of the cycle. Therefore, once a family points at a locality that can feasibly accommodate it, the locality first determines the best house for that family and then points at (possibly another) family that has the highest quality score for that house. As in the classical TTC, in each round there must be a cycle which is eliminated by assigning families to the localities they pointed at and adjusting the capacities of each localities by the demands of the family that has just been assigned to it.

Example. In order to illustrate MTTC, we will need to know the preferences of refugee families in addition to the quality score function q:

$$f_1 : \ell_3, \ell_4, \ell_1, \ell_2$$
$$f_2 : \ell_3, \ell_1, \ell_2, \ell_4$$
$$f_3 : \ell_4, \ell_3, \ell_1, \ell_2$$
$$f_4 : \ell_4, \ell_1, \ell_2, \ell_3$$
$$f_5 : \ell_1, \ell_2, \ell_4, \ell_3$$

In this example, the MTTC algorithm lasts two rounds, which are summarized in Table 1.

Round 1: All families and localities are active in the first round.

- In the first round, all families point at their favorite locality. f_1 and f_2 point at ℓ_3, f_3 and f_4 at ℓ_4 and f_5 at ℓ_1. ℓ_1 in turns points at f_5 since its highest quality score is for (f_5, h_{12}). Similarly, ℓ_2 and ℓ_3 point at f_4 due to (f_4, h_{21}) and (f_4, h_{32}) being their respective highest quality scores and ℓ_4 points at f_2 as (f_2, h_{42}) has the highest quality score.

- A cycle appears between f_5 and ℓ_1, resulting in f_5 being matched to h_{12} since (f_5, h_{11}) is not feasible. Another cycle appears between f_2, ℓ_3, f_4 and ℓ_4, consequentlyf_2 is matched to h_{32} and f_4 is matched to h_{42}.

Round 2: Only two families and four houses, one per locality, remain active in the second round. The relevant quality scores are:

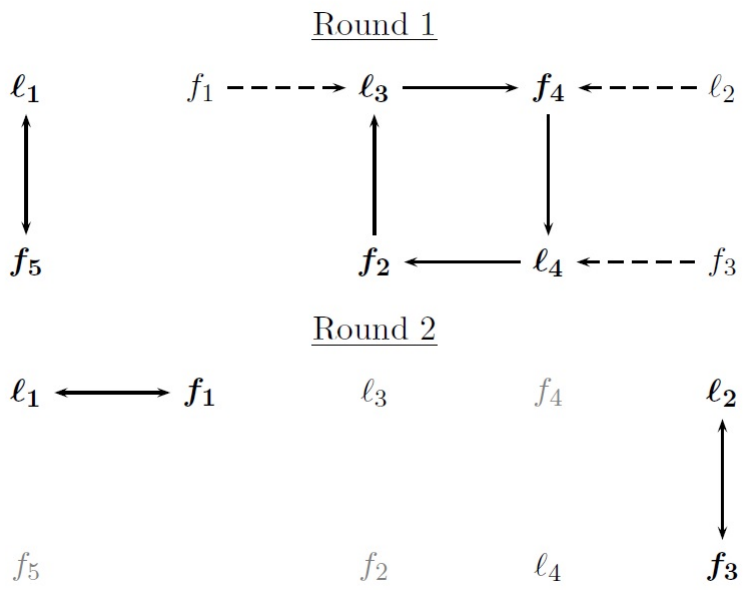


Table 1: Applying the MTTC Algorithm to the Example.

	$q((f,h))$			
	h_{11}	h_{21}	h_{31}	h_{41}
f_1	71	$\emptyset$	8	$\emptyset$
f_3	$\emptyset$	68	31	7

- f_1 cannot point at ℓ_3 because it cannot be accommodated alongside f_2. It cannot point at ℓ_4 either since $(f_1, h_{4,1})$ is not a feasible contract. f_1 then points at its third choice, ℓ_1, this is possible since (f_1, h_{11}) is feasible and f_1 and f_5 can be accommodated together at ℓ_1. f_3 can point at neither ℓ_4 nor ℓ_3 nor ℓ_1 because it cannot be accommodated alongside f_4, f_2 and f_5, respectively. Consequently f_3 points at ℓ_2.

- ℓ_1 already accommodates f_5, consequently it can still accommodate f_1 but not f_1 and points at the former. (f_3, h_{21}) is feasible but (f_1, h_{21}) is not, consequently ℓ_2 points at f_3. ℓ_3 and ℓ_4 do not points at any family since they can no longer accommodate neither f_1 nor f_3 alongside, respectively, f_2 and f_4. Two cycles appear since f_1 and ℓ_1 as well as f_3 and ℓ_2 points at each other. f_1 is assigned h_{11}, f_3 is assigned h_{21} and the algorithm terminates.

The resulting outcome is

$$Y^{MTTC} = \{(f_1, h_{11}), (f_2, h_{32}), (f_3, h_{21}), (f_4, h_{42}), (f_5, h_{12})\}$$

In this example, taking refugee preferences into account and running MTTC makes families f_2 and f_4 better-off than in the solution to outcome-quality maximization problem by allowing them to swap their houses.

6 Accounting for Refugee's Preferences and Local Areas' Priorities

As a refugee crisis goes on, it typically becomes more difficult to obtain goodwill from localities.[24] Beyond direct cash transfers and coercion, the policymaker has a market design tool that can increase localities' willingness to participate: explicitly incorporating localities' priorities over refugees. Eliciting which refugees localities want to prioritize not only gives an important sense of control to local communities, but also helps ensure that refugees are located in places where they are most welcome and where their needs can be best addressed.[25]

Now, we let π_ℓ be the strict ordinal priority list of locality ℓ over families in feasible contracts X_ℓ and let π be ordinal priority profile of the localities. Denote by $\hat{F}_{f,\ell}$ the set of families with a priority for ℓ higher than f.

We say that for an outcome $Y \subseteq X$, f and ℓ form a *blocking pair* if $\ell \succ_f \boldsymbol{\ell}(Y_f)$ and there exists a feasible outcome Y' such that $\boldsymbol{\ell}(Y'_f) = \ell$ and $\boldsymbol{\ell}(Y_{f'}) = \ell \Rightarrow \boldsymbol{\ell}(Y'_{f'}) = \ell$ for all $f' \in \hat{F}_{f,\ell}$. Y is *stable* if it is feasible and does not allow any blocking pair.

In other words, family f and locality ℓ form a blocking pair if f prefers ℓ to the current locality in which it is resettled and it is possible to accommodate f in ℓ while ensuring that families which have a priority in ℓ than f can remain in the same locality. Therefore, in a stable outcome, every refugee family prefer the locality it has been resettled in to not being resettled at all, and there is no family that prefers to be placed in another locality in which it has higher priority.

A stable outcome $Y \subseteq X$ is *(Pareto-)undominated* if there does not exist any stable outcome $Y' \subseteq X$ such that for all f, either $\boldsymbol{\ell}(Y'_f) = \boldsymbol{\ell}(Y_f)$ or $\boldsymbol{\ell}(Y'_f) \succ_f \boldsymbol{\ell}(Y_f)$. A stable outcome $Y \subseteq X$ is *family-optimal* if there does not exist another stable outcome $Y' \subseteq X$ such that $\boldsymbol{\ell}(Y'_f) \succ \boldsymbol{\ell}(Y_f)$ for any f.

Note how houses themselves play no direct role in the definitions stability, efficiency or strategy-proofness once preferences over localities and priorities over refugees are taken into account. With this in mind, it is useful to define a *family-locality assignment* (a partition on the set of families) $A \equiv \{A_\ell\}_{\ell \in L \cup \{\emptyset\}}$, where A_ℓ for any $\ell \in L$ is the set of families resettled in locality ℓ and $A_\emptyset$ is the set of families that do not get resettled. A is feasible if, for all $\ell \in L$, A_ℓ is a feasible set of families for locality ℓ. Any pair $(f, \ell) \in F \times L$ forms a blocking pair of A if $f \in A_{\ell'}$ with $\ell \succ_f \ell'$ and there exists $\tilde{A}_\ell \subseteq A_\ell$ such that ℓ can accommodate $\tilde{A}_\ell \cup \{f\}$ and $f \,\pi_\ell\, f'$ for any $f' \in A_\ell \setminus \tilde{A}_\ell$. If A does not contain a blocking pair then we say that A is unblocked. A is stable if it is feasible and does not contain a blocking pair. A is a stable undominated assignment if it is stable and for any other stable assignment A', there exists $f \in F$ such that $f \in A_\ell$, $f \in A'_{\ell'}$ and $\ell \succ_f \ell'$. Observe that an outcome Y created by assigning each family a house within the locality where it is resettled under A is stable if and only A is. Additionally, Y is a stable undominated outcome if and only if A is a stable

[24] Some of the British media, for example, soon reacted against the Isle of Bute refugees (Reid, 2015).

[25] Alternatively, we might take priorities of localities into account by treating the refugee resettlement process as a one-sided matching market among the localities. In this case, we would allow localities to report their priorities as a strategic decision, and replace refugees' preferences with the quality score function q. In the context just described, we can modify MTTC by running it from the point of view of localities and reversing the direction of matching; this would yield a strategy-proof and efficient mechanism from the point of view of localities, but treat refugees as objects.

undominated assignment. We denote $A(Y)$, the assignment associated with outcome Y.

Summary of the Top-Choice Algorithm

Even without combinatorial constraints, simultaneously respecting refugee preferences and locality priorities via stability can come at a high cost to efficiency (Kesten, 2010). The Top-Choice Algorithm, described in sections 6.1 – 6.3 below, finds a stable outcome, if one exists, that is Pareto-undominated from the refugees' perspective. We also explain how it reduces refugees' incentives to misrepresent their preferences in section 6.4.

The Top-Choice Algorithm runs in three phases and we illustrate each phase by applying it to the Example running throughout the paper. The technical descriptions of each phase of the algorithm can be found in the Appendix.

In Phase 1, described in section 6.1, we provide an algorithm to find an undominated unblocked family-locality assignment. The first phase involves running the Top-Down-Bottom-Up (TDBU) algorithm on the set of feasible contracts by systematically identifying contracts that cannot be part of any stable outcome. This leaves us with contracts $\phi(X) \subseteq X$ that could form stable outcome. A family-locality assignment can be constructed by resettling each family in their top-choice locality they prefer among those such that a contract between the family and a house in that locality exists in $\phi(X)$. We call that assignment $A^*(\phi(X))$. If $A^*(\phi(X))$ is feasible, then we have found the unique refugee-optimal stable assignment. However, $A^*(\phi(X))$ will often not be feasible.

In Phase 2, described in section 6.2, we address the potential infeasibility of $A^*(\phi(X))$ by using a search process. The aim of this Phase is to identify enough contracts that cannot be part of a stable outcome in order to find a stable outcome. We first note that $A^*(\phi(X))$ is an extreme point of a lattice of unblocked assignments partially ordered by Pareto dominance. We restart only with the contracts in $\phi(X)$ and fix an arbitrary order of families. We start by checking whether we can give the first family in the order its top-choice locality. Suppose we can. Then, we look at the top-choice of the second family. It could be the case that no stable outcome exists when first and second families have their first choice. Then, we check whether the first family could have its second choice satisfied, and so on. By keeping the order of the families fixed, prioritising contracts over families above in the ranking to the families below, we can remove more contracts at each step (called Augmented TDBU) of the Stable Undominated Assignment (SUA) algorithm. Eventually, SUA finds a stable undominated assignment or proves that it doesn't exist.

If a stable undominated assignment between families and localities has been found in Phase 2, we run the final Phase 3, described in section 6.3, which allocates houses to families. Since all the combinatorial constraints are now satisfied and blocking pair have been eliminated, this reduces to a straightforward maximum bipartite matching problem.

6.1 Phase 1: Top-Down Bottom-Up Algorithm

The TDBU algorithm is run exactly once at the beginning of the Top-Choice Algorithm. It cycles through localities and identifies family-locality pairs that cannot be part of *any* stable assignment. It does so from the top-down by identifying *guarantees* i.e. families whose priority at a given locality is high enough to ensure the two will form a blocking pair if the family were not resettled in that locality. This allows us to eliminate all contracts between

that family and houses located in any locality the family likes less. The TDBU algorithm also eliminates contracts from the bottom-up by identifying *rejections* i.e. families whose priority at a given locality is too low for any contract involving this family and a house at this locality to be part of a stable outcome. These contracts can also be eliminated. Once the TDBU algorithm stops, we end up with a set of contracts $\phi(X)$ which contains *all* stable outcomes (although they may not exist). The TDBU algorithm works as follows:

TOP-DOWN BOTTOM-UP (TDBU) ALGORITHM

Initialisation: Start with the set of all feasible contracts X as a current outcome and fix an arbitrary order of localities.

Round $i \geq 1$ *(across localities)*:
Consider the current outcome X_i. If the algorithm has cycled through every locality and the outcome has not changed, that algorithm terminates and yields $\phi(X) := X_i$.

Initialisation within locality:
Otherwise, consider locality ℓ_i (subscripts modulo $|L|$) and the set of families (size $|F_{\ell_i}|$) which have ℓ_i as their top-choice in the current outcome X_i. Order the contracts involving ℓ_i according to π_{ℓ_i}.

Step $j = 1$: Consider a family f with the highest priority in ℓ_i.
If f can be accommodated in ℓ_i, then f receives a *guarantee* for ℓ_i. Remove all the contracts involving f and a house in any locality that f prefers less than ℓ_i from the current outcome.

Otherwise f's contracts are unchanged.

Step $1 < j \leq |F_{\ell_i}|$: Consider a family f with the j^{th} highest priority in ℓ_i.

If for every set of accommodated families with higher priority that f in ℓ_i, f can be accommodated in ℓ_i, then f receives a *guarantee* for ℓ_i. Remove all the contracts involving f and a house in any locality that f prefers less than ℓ_i from the current outcome.

If for every set of accommodated families with higher priority that f in ℓ_i, f cannot be accommodated in ℓ_i, then f receives a *rejection* for ℓ_i. Remove the contracts involving f and every house in ℓ_i from the current outcome.

Otherwise f's contracts are unchanged.

Update X_i by dropping (i) all the contracts ranked below the guaranteed contracts for every family, (ii) all rejected contracts.

Continue to the next locality ℓ_{i+1} in the order.

The TDBU algorithm terminates when no further rejections can be identified reduces the set of contracts that need to be considered from X to $\phi(X) \subseteq X$. Let $A^*(\phi(X))$ be the family-locality assignment constructed by assigning each family to their top-choice given $\phi(X)$. That is, $A^*_\ell(\phi(X)) := \overline{F}_\ell$ for all $\ell \in L$.

Lemma 1. *$A^*(\phi(X))$ does not contain any blocking pair and dominates any stable assignment. It is the unique optimal stable assignment if and only if it is feasible.*

Unfortunately, $A^*(\phi(X))$ is usually not feasible.

Example. We now introduce priorities for the localities[26] into our example and apply the TDBU algorithm to take care of the first phase:

$$\ell_1 : f_2, f_1, f_4, f_5, f_3$$
$$\ell_2 : f_5, f_1, f_3, f_4, f_2$$
$$\ell_3 : f_5, f_3, f_2, f_1, f_4$$
$$\ell_4 : f_1, f_5, f_2, f_4, f_3$$

The Top-Down Bottom-Up algorithm reduces the size of the problem by eliminating contracts that are cannot be part of any stable outcome. This process is summarized in Table 2. Families are initially listed in order of priority for each locality, as can be seen in the top panel of Table 2. The second column indicates how the family ranks the locality. Of primary importance is whether the family considers this locality to be its top-choice, which is denoted by "T".

Round 1: The TDBU algorithm starts in locality ℓ_1. (This is arbitrary, the order in which localities are considered does not have any impact.) f_2 gets a guarantee thanks to its first priority and the fact that it can be accommodated, which is denoted by a "G" in the third column. This allows eliminating all contracts involving f_2's third and fourth preferences: (f_2, h_{21}), (f_2, h_{41}) and (f_2, h_{42}) are eliminated and f_2 stops contesting ℓ_2 and ℓ_4. This is denoted by a "✗" in the fourth columns of the rows devoted to f_2 and ℓ_2, respectively ℓ_4. ℓ_1's capacities for both services are sufficient to accommodate both f_1 and f_2, however both families can only be assigned h_{11}. As a result, it is impossible to resettle both f_1 and f_2 in locality ℓ_1 and f_1 does not get a guarantee. f_1 does not get rejected either since ℓ_1 is not f_2's top-choice. In contrast, f_4 does get a guarantee since it is possible to give h_{11} to either f_2 or f_1 and h_{12} to f_4 without violating ℓ_1's capacity constraint. f_4 stops contesting ℓ_2 and ℓ_3, its third and fourth choices, as a result. f_5 does not get a guarantee since both houses will be used in the case both f_2 and f_4 are resettled in ℓ_1. It does not get a rejection either as it may be that none of f_2, f_1 and f_4 end up resettled in ℓ_1, in which case f_5 would. f_3 does not get a guarantee since it cannot be resettled in ℓ_1 alongside f_5. In fact, f_3 gets rejected from ℓ_1 due to f_5 having ℓ_1 as its top-choice. To see this, observe that f_5 will be resettled in ℓ_1 unless at least one of f_2 or f_4 is. As f_3 cannot be resettled in ℓ_1 alongside either of these families, there is no situation where f_3 ends up in ℓ_1 without violating stability. f_3 stops contesting ℓ_1 and ℓ_2 becomes its third choice.

[26]With these priorities, the outcome of MTTC would be $\{(f_1, h_{42}), (f_2, h_{32}), (f_3, h_{41}), (f_4, h_{21}), (f_5, h_{12})\}$

Round 2: Families are removed from the list of localities they no longer contest and the algorithm continues in Round 2, which focuses on ℓ_2. f_5 gets a guarantee since it can be accommodated, consequently it will no longer contest ℓ_3 and ℓ_4. f_1 is rejected because it cannot be assigned the unique house of ℓ_2. f_3 receives neither a guarantee nor a rejection since h_{21} may or may not be assigned to f_5. f_5 is removed from ℓ_3 and ℓ_4's list and f_1 is removed from ℓ_2's list, the algorithm continues in Round 3, which focuses on ℓ_3.

Round 3: f_3 gets a guarantee at ℓ_3 but f_2 does not since accommodating both families requires three units of Service 2 and ℓ_3 only has two units available. f_2 is however not rejected since f_3 may give up its priority for ℓ_3 if it is resettled in ℓ_4. f_1 can be accommodated alongside f_3 but not f_2, as a result f_1 will be resettled at ℓ_3 if and only if f_3 is. f_3 is removed from ℓ_2's list and the algorithm continues in Round 4, which focuses on ℓ_4.

Round 4: This is similar to the previous round, f_1 gets a guarantee while f_4 and f_3 neither get a guarantee nor a rejection because f_1 can be accommodated alongside f_3 but not alongside f_4. As a result, f_1 gets a guarantee while f_3 and f_4 neither get a guarantee nor a rejection. This leads to f_1 being removed from ℓ_1's list. Going through all localities again in Rounds 5-8, no additional contract can be eliminated. The TDBU algorithm ends with the following set of contracts:

$$\begin{aligned} Y^{TDBU} = \{&(f_1,h_{31}),(f_1,h_{32}),(f_1,h_{42}),(f_2,h_{11}),(f_2,h_{31}),(f_2,h_{32}),(f_3,h_{31}),\\ &(f_3,h_{32}),(f_3,h_{41}),(f_3,h_{42}),(f_4,h_{12}),(f_4,h_{42}),(f_5,h_{12}),(f_5,h_{21})\} \end{aligned}$$

Since f_1 and f_2 cannot be accommodated together at ℓ_3 and the same is true for f_3 and f_4 at ℓ_4, resettling all families in their top-choice locality is infeasible. The search for a stable outcome requires eliminating additional contracts. We do so by using a pre-determined general priority over families in Phase 2.

6.2 Phase 2: Searching For a Stable Undominated Outcome

The TDBU algorithm reduces the set of contracts by eliminating some that are incompatible with stability. Given that new set of contracts, if all top-choices can be accommodated then the resulting assignment is the optimal stable one. Unfortunately, it is not always possible to accommodate all top-choices, as a result the search for a stable assignment requires finding additional rejections. Our goal is to find additional rejections until a stable undominated assignment is found whenever the set of stable assignments is nonempty.

We begin with a descriptions of the first steps of the SUA algorithm and then explain how its intermediate steps – ATDBU algorithm – work. We order the families, labelling them $f^1, f^2, ..., f^{|F|}$. This can be done arbitrarily or reflect a *general priority*, for example due to their level of vulnerability or the waiting time in the resettlement pipeline. The order does not matter for whether the stable outcome is found or not. As we shall see, it is advantageous for families to be as high on general priority list as possible and our Example shows that different orders may give different running times for the algorithm.

We start SUA with $\phi(X)$ and in the first round give f^1 an *artificial guarantee* for its top-choice $\bar{\ell}_{f^1}$ meaning that we tentatively eliminate all contracts involving f^1 and houses

Round 1												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}	G		f_5	2^{nd}			f_5	4^{th}			f_1	2^{nd}		
f_1	3^{rd}			f_1	4^{th}			f_3	2^{nd}			f_5	3^{rd}		
f_4	2^{nd}	G		f_3	4^{th}			f_2	T			f_2	4^{th}		✗
f_5	T			f_4	3^{rd}		✗	f_1	T			f_4	T		
f_3	3^{rd}	R	✗	f_2	3^{rd}		✗	f_4	4^{th}		✗	f_3	T		

Round 2												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}			f_5	2^{nd}	G		f_5	4^{th}		✗	f_1	2^{nd}		
f_1	3^{rd}			f_1	4^{th}	R	✗	f_3	2^{nd}			f_5	3^{rd}		✗
f_4	2^{nd}			f_3	3^{rd}			f_2	T			f_4	T		
f_5	T							f_1	T			f_3	T		

Round 3												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}			f_5	2^{nd}			f_3	2^{nd}	G		f_1	2^{nd}		
f_1	3^{rd}			f_3	3^{rd}		✗	f_2	T			f_4	T		
f_4	2^{nd}							f_1	T			f_3	T		
f_5	T														

Round 4												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}			f_5	2^{nd}			f_3	2^{nd}			f_1	2^{nd}	G	
f_1	3^{rd}		✗					f_2	T			f_4	T		
f_4	2^{nd}							f_1	T			f_3	T		
f_5	T														

Rounds 5-8												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}	G		f_5	2^{nd}	G		f_3	2^{nd}	G		f_1	2^{nd}	G	
f_4	2^{nd}							f_2	T			f_4	T		
f_5	T							f_1	T			f_3	T		

Table 2: Top-Down Bottom-Up Algorithm: Example.

within localities f^1 likes less than $\bar{\ell}_{f^1}$. This hypothetical elimination allows us to commence our search along the lattice of stable assignments, looking for a feasible one. Once f is artificially guaranteed a house in locality $\bar{\ell}_{f^1}$, we reduce the set of contracts and check for a feasible matching by running a version of TDBU, which we call Augmented TDBU.[27]

Let us suppose that f^1 can be feasibly assigned to $\bar{\ell}_{f^1}$ given the artificial guarantee. In that case, let us give f^2, its top-choice $\bar{\ell}_{f^2}$ as an artificial guarantee and removing all

[27]ATDBU is equivalent to TDBU whenever the sets of artificial guarantees and rejections are empty.

its contracts involving localities that are less preferred to $\bar{\ell}_{f^2}$. Suppose, we find that it is infeasible to assign both f^1 and f^2 to their top-choice localities in any stable outcome. In that case, we tentatively reject f^2 from $\bar{\ell}_{f^2}$ and remove its contracts. We call this hypothetical rejection an *artificial rejection* as it is only dictated by the current set of artificial guarantees.

More generally, in each step of the SUA, the ATDBU algorithm takes the sets of artificial guarantees and artificial rejections as additional inputs while checking for feasibility of stable assignments (and outputting unblocked outcomes, sets of artificial guarantees and artificial rejections). It does two other things. It stops if a contradiction occurs, which happens either when a family with an artificial guarantee is rejected or when a family with an artificial rejection receives a guarantee. If either one of those occurs, then no stable assignment exists that satisfies the artificial guarantees and rejections of the current round. Second, it finds additional artificial rejections in a special case (as in the case with family f^2 above). This occurs when, at some locality ℓ, a family, say f, has a higher priority than another one, say f', who has received an artificial guarantee. If f cannot be accommodated alongside f', then it must be rejected. It is however possible for f to be given a guarantee later on, since this only depends on the families with a higher priority (hence not on f'). Such scenario yields a contradiction since it means resettling f' in ℓ creates a blocking pair between f and ℓ. Giving f an artificial rejection allows identifying this contradiction.

Let us now state the ATDBU algorithm:

AUGMENTED TOP-DOWN BOTTOM-UP (ATDBU) ALGORITHM

Initialisation: Start with any set of feasible contracts as a current outcome, set of artificial guarantees G_A and a set of artificial rejections R_A and fix an arbitrary order of localities.

Round $i \geq 1$ (across localities):
Consider the current outcome X_i. If the algorithm has cycled through every locality and the outcome, the set of artificial guarantees and the set of artificial rejections have not changed, that algorithm terminates.

Initialisation within locality:
Otherwise, consider locality ℓ_i (subscripts modulo $|L|$) and the set of families (size $|F_{\ell_i}|$) which have ℓ_i as their top-choice in the current outcome X_i. Order the contracts involving ℓ_i according to π_{ℓ_i}.

Step $1 < j \leq |F_{\ell_i}|$: Consider a family f with the j^{th} highest priority in ℓ_i.

If for every set of accommodated families with higher priority that f in ℓ_i, f can be accommodated in ℓ_i, then f receives a *guarantee* for ℓ_i. Remove all the contracts involving f and a house in any locality that f prefers less than ℓ_i from the current outcome. Remove (f, ℓ_i) from the set of artificial guarantees.

If for every set of accommodated families with higher priority that f in ℓ_i, f cannot be accommodated in ℓ_i, then f receives a *rejection* for ℓ_i.

If f had already received an *artificial guarantee*, the algorithm stops with no stable matching.
Otherwise, remove the contracts involving f and every house in ℓ_i from the current outcome.

If f had not received a rejection in ℓ_i when the set of artificial guarantees was empty, but, f cannot be accommodated in ℓ_i for every set of accommodated families with higher priority given the current set of artificial guarantees, then f receives an *artificial rejection* for ℓ_i.

If f had already received a *guarantee*, the algorithm stops with no stable matching.
Otherwise, add (f, ℓ_i) to the set of artificial rejections.

If f has received neither a guarantee, nor a rejection nor an artificial rejection, set of rejections, artificial rejections and artificial guarantees are unchanged.

If f had already received an *artificial guarantee*, then add (f, ℓ_i) to the set of guarantees.
Otherwise f's contracts are unchanged.

Update X_i by dropping (i) all the contracts ranked below the guaranteed contracts for every family, (ii) all artificially rejected contracts, (iii) all rejected contracts. Continue to the next locality ℓ_{i+1} in the order.

The ATDBU algorithm run at each step of the SUA can end in three different ways:

1. ATDBU finds enough new rejections to identify a stable undominated assignment.
2. ATDBU produces a situation where a stable assignment may exist but more rejections are needed to find it.
3. ATDBU no stable assignment exists given the current set of artificial guarantees.

In the first case, the SUA also stops and we are done. Our final algorithm deals with the other two cases when there are either not enough rejections or not enough guarantees.

In the second case, we continue by giving an artificial guarantee to next family in the general priority and running the ATDBU algorithm again. We do this along the general priority until either a stable assignment is found or we find that such an assignment is infeasible.

In the third case, the SUA retraces its steps by removing the top-choice of the latest family with an artificial guarantee, rejecting it and giving the next preferred option for this family as a top-choice. Once all locality options for a particular family are exhausted, SUA rejects the artificial guarantee of the next family up the general priority list runs ATDBU and so on.

SUA starts out with TDBU and then proceeds to iterate ATDBU until it ends.

Stable Undominated Assignment (SUA) Algorithm

Initialisation: Run the TDBU algorithm to obtain $\phi(X)$. Let the current outcome $X_i = \phi(X)$. Arbitrarily order the remaining families. Let the sets of artificial guarantees and artificial rejections be empty.

Round 1:
Give the top family f in the order an *artificial guarantee* of its top-choice locality $\bar{\ell}_f$ by removing its less preferred contracts. Run ATDBU on the resulting input.

If the output of ATDBU is empty, then remove f's artificial guarantee. Report $\star\star$.
If the output of ATDBU is non-empty, then add $(f, \bar{\ell}_f)$ to the set of artificial guarantees. Report $\star$.

Round $i > 1$:

If the previous round reported $\star\star$:

Give the next family f' in the order an artificial guarantee of its top-choice locality $\bar{\ell}_{f'}$ from the current outcome by removing its less preferred contracts. Run ATDBU on the resulting input.

If the output of ATDBU is empty, then remove f''s artificial guarantee. Report $\star\star$.
If the output of ATDBU is non-empty, then add $(f', \bar{\ell}_{f'})$ to the set of artificial guarantees.

If this was the last family in the order, report the current outcome as stable outdominated assignment.
Otherwise, report $\star$.

If the previous round reported $\star$:

Remove previous family's top-choice from the current outcome and add it to the set of artificial rejections. Run ATDBU on the resulting input.

If the output of ATDBU is empty:

Go to the last family f'' for which the current outcome, the set of artificial rejections and guarantees did not change to the subsequent round. Report $\star\star$.
If this was the first family, then report an empty stable outcome.

If the output of ATDBU is not empty, remove the top-choice of f'' from the current outcome and add it to the set of artificial rejection. Run ATDBU on the resulting input. Report $\star$.

SUA continues until either it finds a stable undominated assignment, which always happens if one exists, or by identifying that the set of stable assignments is empty. The latter happens if an impossibility is found when after f^1's top-choice is artificially guaranteed and then eventually rejected because the set of possible stable outcomes has narrowed.

Example. We now illustrate how the SUA algorithm finds a stable undominated outcome in our example using the ATDBU algorithm at each step. We only use contracts remaining from running TDBU in order to initialize the process. We will show that the outcome depends on the general priority order of families.

First Stable Undominated Assignment

Suppose first that f_1 is at the top of the general priority (for example, the general priority could be f_1, f_2, f_3, f_4, f_5). We give f_1 an artificial guarantee for his top choice, that is we look for a stable assignment where f_1 is resettled in ℓ_3. We run the Augmented Top-Down Bottom-Up algorithm to identify additional contracts that can be eliminated as a result. The algorithm is summarized in Table 3. The artificial guarantee is denoted by "AG" in the fourth column of the relevant row. Nothing of interest occurs in the first two rounds, f_2 and f_5 receive a guarantee for ℓ_1 and ℓ_2, respectively, but this does not allow eliminating any contract. The action begins in Round 3, which focuses in ℓ_3. As a consequence of f_2's artificial guarantee, f_2 receives an artificial rejection.

The difference between a rejection and an artificial rejection is a subtle yet important one. A family f receives a rejection if the families with a higher priority will necessarily take enough resources from the locality to prevent f from being resettled there. A family that receives a rejection could never receive a guarantee later on and stops contesting the locality that rejected it. Observe that the rejection decision only depends on families with a higher priority. An artificial rejection occurs when a family is does not receive a normal rejection but cannot be resettled in a locality because families with a *lower* priority have received an artificial guarantee. In the second panel of Table 3, f_2 is not rejected since it is possible that f_3 will be resettled ℓ_4, however f_2 can only be resettled in ℓ_3 if f_1 is not. Resettling f_2 in ℓ_3 contradicts the artificial guarantee given to f_1. Everything works as if f_2 had been rejected, its top choice will move to ℓ_1 in the next round and it will no longer be taken into account when assessing f_1 for a guarantee or a rejection, however f_2 remains in ℓ_3's list. The reason for this is that it could still get a guarantee if f_3 was to be resettled in ℓ_4. It is important to keep track of this to identify cases where a stable matching may not exist.

Because f_2 receives an artificial rejection, only f_3 is relevant to determine whether f_1 receives a guarantee. Since f_1 and f_3 can be accommodated together, f_1 does receive a guarantee, which means it will no longer requires the artificial guarantee in the following rounds. f_1 is removed from ℓ_4's list due to its (artificial) guarantee at ℓ_3 and ℓ_1 is no f_2's top-choice after the latter was artificially rejected from ℓ_3.

In Round 4, f_4 receives a guarantee for ℓ_4 and is removed from ℓ_1's list. f_3 receives a rejection since it cannot be accommodated at ℓ_4 alongside f_4 and the latter will no be resettled in ℓ_4 in any stable matching. f_3 is removed from ℓ_4's list and ℓ_3 becomes its top-choice. The algorithm continues in Round 5, which focuses on ℓ_1. f_2 receives a guarantee and, because ℓ_1 cannot accommodate both families, f_5 receives a rejection. ℓ_2 becomes f_5's top-choice as a result. The algorithm continues in Round 6, which does not yield any additional elimination: f_5 receives a guarantee for ℓ_2 but does not contest any other locality. In Round 7, f_3 receives a guarantee for ℓ_3, which is its top-choice. As a result it will be matched with ℓ_3 in any stable matching. Since f_3 and f_2 cannot be both resettled at ℓ_3, f_2 receives a rejection. We can now be certain that the artificial rejection is received earlier does not lead to any contradiction. In contrast, f_1 receives a guarantee since it can be accommodated in ℓ_3 alongside f_3. In Rounds 8-11, only top-choices remain and now additional rejections can be found, the algorithm ends. Since these top-choices can all be accommodated, a stable undominated assignment has been found: f_1 and f_3 are resettled in ℓ_3, f_2 in ℓ_1, f_4 in ℓ_4 and f_5 in ℓ_2. Hence:

$$A^{SUA} = \{A^{SUA}_{\ell_i}\}_{i=1,2,3,4} = \{\{f_2\}, \{f_5\}, \{f_1, f_3\}, \{f_4\}\}$$

Second Stable Undominated Assignment

If f_4 is at the top of the general priority, the ATDBU algorithm operates in an almost analogous way. f_4's artificial guarantee for ℓ_4 means that f_3 is rejected and ℓ_3 becomes its new top-choice. f_2 is rejected from ℓ_3 as a result and its top choice moves to ℓ_1, leading to the rejection of f_5. As before, f_1 and f_3 are resettled in ℓ_3, f_2 in ℓ_1, f_4 in ℓ_4 and f_5 in ℓ_2.

If f_2 is at the top of the general priority, its artificial guarantee implies the rejection of f_1 from ℓ_3. In turn, f_4 is rejected from ℓ_4 and f_5 from ℓ_1, yielding an assignment where f_1 and f_3 are resettled in ℓ_4, f_2 in ℓ_3, f_4 in ℓ_1 and f_5 in ℓ_2. Compared to the other assignment, f_2 and f_3 are better-off while f_1 and f_4 are worse-off. Unsurprisingly, the same outcome is found if f_3 is at the top of the general priority.

$$A^{\dagger SUA} = \{A^{\dagger SUA}_{\ell_i}\}_{i=1,2,3,4} = \{\{f_4\}, \{f_5\}, \{f_2\}, \{f_1, f_3\}\}$$

Non-existence of other stable assignments and irrelevance of general priority

The two assignments found above are in fact the only two stable ones in this matching problem. This can be shown in two steps. First, observe that when one of f_1, f_2, f_3 or f_4 is at the top of the priority list, only top choices remain when the ATDBU algorithm terminates, hence only one stable matching exists with one of these families receives its top-choice. Second, if none of them obtains its top-choice, f_1 forms a blocking pair with ℓ_3 and f_3 forms a blocking pair with ℓ_4. Finally, we need to follow what happens when f_5 is at the top of the list, as we do now. The algorithm is summarized in Table 4.

Rounds 1-2												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}	G		f_5	2^{nd}	G		f_3	2^{nd}			f_1	2^{nd}		
f_4	2^{nd}							f_2	T			f_4	T		
f_5	T							f_1	T		AG	f_3	T		

Round 3															
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}			f_5	2^{nd}			f_3	2^{nd}			f_1	2^{nd}		✗
f_4	2^{nd}							f_2	T		AR	f_4	T		
f_5	T							f_1	T	G	AG	f_3	T		

Round 4															
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	T			f_5	2^{nd}			f_3	2^{nd}			f_4	T	G	
f_4	2^{nd}		✗					f_2			AR	f_3	T	R	✗
f_5	T							f_1	T						

Round 5															
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	T	G		f_5	2^{nd}			f_3	T			f_4	T		
f_5	T	R	✗					f_2			AR				
								f_1	T						

Rounds 6-7															
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	T			f_5	T	G		f_3	T	G		f_4	T		
								f_2		R	✗				
								f_1	T	G					

Rounds 8-11															
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	T	G		f_5	T	G		f_3	T	G		f_4	T	G	
								f_1	T	G					

Table 3: SUA Algorithm with f_1 on Top of the General Priority.

In Round 1, f_5's artificial guarantee yields an artificial rejection for f_2 and f_4 since neither of them can be resettled in ℓ_1 alongside f_5. Recall that f_2 and f_4 receive an artificial rejection rather than a regular one because they are rejected due to a family with a lower priority. They could still receive a guarantee, in fact this happens to f_2 since it is at the top ℓ_1's priority list. The fact that f_2 receives both a guarantee and an artificial rejection means that f_5's artificial guarantee violates f_2's priority for ℓ_1. f_2 and ℓ_1 will form a blocking pair unless f_2 is resettled in a locality it prefers to ℓ_1. In this example, ℓ_3 is the only option. This

Round 1												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2	2^{nd}	G	AR	f_5	2^{nd}	✗		f_3	2^{nd}			f_1	2^{nd}		
f_4	2^{nd}		AR					f_2	T		AG	f_4	T		
f_5	T		AG					f_1	T			f_3	T		

Rounds 2-3												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_2			✗					f_3	2^{nd}	G	AR	f_1	2^{nd}		
f_4			AR					f_2	T		AG	f_4	T		
f_5	T		AG					f_1	T	R	✗	f_3	T		AG

Round 4												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_4			AR					f_3			AR	f_1	T	G	
f_5	T		AG					f_2	T		AG	f_4	T	R	✗
												f_3	T	G	AG

Round 5												
ℓ_1				ℓ_2				ℓ_3				ℓ_4			
f_4		G	AR					f_3			AR	f_1	T		
f_5	T	G	AG					f_2	T		AG	f_3	T		

Table 4: SUA Algorithm with f_5 on Top of the General Priority.

means that f_2 must be matched with ℓ_3 in any stable matching that respects f_5's artificial guarantee. The ATDBU algorithm accounts for this by giving f_2 an artificial guarantee for ℓ_3. This guarantee is artificial because f_2 could still be rejected by ℓ_3 later on. Notice finally that f_5's artificial guarantee allows removing f_5 from ℓ_2's list.

Nothing occurs in Round 2 since ℓ_2 does not have any family listed. In Round 3, f_2's artificial guarantee yields an artificial rejection for f_3 since the pair cannot be accommodated together in ℓ_3 and f_3 has a higher priority than f_2. Since f_3 also gets a guarantee, it receives an artificial guarantee for ℓ_4. f_1 receives a rejection from ℓ_3 since it has a lower priority than f_2 and the two families cannot both be resettled in ℓ_3. ℓ_4 consequently becomes f_1's top-choice. Finally, f_2's artificial guarantee for ℓ_3 allows removing f_2 from ℓ_1's list.

In Round 4, f_1 receives a guarantee and, since ℓ_4 is its top-choice, the fact that f_1 and f_4 cannot both be resettled in ℓ_4 means that f_4 is rejected. f_3 then gets a guarantee since it can be accommodated alongside f_1.

In Round 5, f_4 gets a garantee for ℓ_1 while it already has an artificial rejection. This means that f_5's artificial guarantee for ℓ_1 violates f_4's priority for ℓ_1 unless it can be resettled in a location it prefers. Since f_4 is no longer contesting any other locality, this is impossible. We can then conclude that f_5's artificial guarantee contradicts stability, in other words there does not exist any stable matching where f_5 is resettled in ℓ_1.

The search continues by going back to the outcome of the TDBU algorithm and removing

f_5 from ℓ_1's list. f_5 is then guaranteed its second choice, ℓ_2. As this fails to remove any of f_1, f_2, f_3 or f_4 from their top-choice, the second family of the general priority must be given an artificial guarantee. By an argument analogous to the one developed above, the first stable undominated assignment is found if either f_1 or f_4 is second on the general priority list and the second stable assignment is found if it is either f_2 or f_3.

6.3 Phase 3: Assignment of Houses

The SUA algorithm finds an undominated family-locality assignment whenever one exists. Finding an outcome requires determining, for all localities, the matching between families and houses. As the family-locality assignment is feasible, it is always possible to give each family a house with which it has a feasible contract. When there is more than one possibility, we choose the one that maximizes total match quality. This problem is equivalent to finding a maximum bipartite matching in each locality, which can be solved using linear programming techniques or the Hungarian algorithm.

Given $(X, (H_\ell)_{\ell \in L}, q, \kappa, \nu, \succ, \pi)$, the Top-Choice Algorithm runs as follows:

1. Run the TDBU algorithm to obtain a smaller set of contracts, $\phi(X)$.
2. Run the SUA algorithm to obtain a stable undominated assignment A^*, if one exists.
3. For each ℓ, solve

$$\max_{Y \subseteq X} \sum_{(f,h) \in Y} q((f,h)) \quad \text{subject to:} \qquad \begin{array}{l} f \in A^*_\ell \\ h \in H_\ell \\ Y \text{ is feasible} \end{array} \tag{3}$$

   in order to obtain a stable undominated outcome.

Example. Let's illustrate the final phase of the Top-Choice Algorithm that assigns houses. First, let's deal with A^{SUA}. At ℓ_1, f_2 receives h_{11} since (f_2, h_{12}) is incompatible. At ℓ_2, f_5 receives the only house, h_{21}. At ℓ_3, $q(f_1, h_{31}) + q(f_3, h_{32}) = 51 < 54 = q(f_1, h_{32}) + q(f_3, h_{31})$, hence f_1 receives h_{32} and f_3 receives h_{31}. At ℓ_4, feasibility dictates that f_4 be given h_{42}. This yields the following stable undominated outcome:

$$Y^{TCA} = \{(f_1, h_{32}), (f_2, h_{11}), (f_3, h_{31}), (f_4, h_{42}), (f_5, h_{21})\}$$

$A^{\dagger SUA}$ can also be quickly turned into a stable undominated outcome by assigning houses within each locality. At ℓ_1, f_4 is assigned h_{12} since f_4, h_{11} is not feasible. At ℓ_2, f_5 is again assigned the only house h_{21}. At ℓ_3, f_2 is assigned h_{32} to maximize match quality. At ℓ_4,

feasibility dictates that f_1 be assigned h_{42} and, consequently, f_3 h_{41}. This yields another stable undominated outcome:

$$Y^{\dagger TCA} = \{(f_1, h_{42}), (f_2, h_{32}), (f_3, h_{41}), (f_4, h_{12}), (f_5, h_{21})\}$$

We can now state our main result in this section.

Proposition 3. *Top-Choice Algorithm finds a stable undominated outcome if and only if the set of stable outcomes is nonempty.*

In the event the set of stable outcomes is empty, it is possible to run a version of the SUA algorithm where families are given artificial rejections until a feasible family-locality assignment is obtained. This assignment is not stable, but wasteful i.e. all blocking pairs it contains involve a family and a locality such that the locality has enough capacity to accommodate the family. The assignment can be made stable if the service provision of some localities were appropriately reduced. Capacity that is not used in the current round can be transferred to the matching batch, ensuring that it is not wasted entirely.

6.4 Discussion of strategic issues

Unlike MTTC, the Top-Choice Algorithm is a not a strategy-proof mechanism. As Kesten (2010) shows "No Pareto-efficient mechanism that can Pareto improve upon [family]-optimal stable mechanism is fully immune to strategic action". But just because the mechanism is manipulable is does not mean it is easy to manipulate (Roth and Rothblum, 1999, Ehlers, 2008). Refugees submit preferences in an environment of great uncertainty about what the effect of other refugees' strategies as the priorities of localities shift unpredictably over time. Hence, it is hard to learn from past matches how to manipulate the system as parents have done over time in place where the Boston mechanism was used for school choice. First, we should point to one case in which the Top-Choice Algorithm *is* strategy-proof: when A^* is feasible. Second, manipulability increases as the SUA algorithms searches deeper into the lattice of unblocked assignments. The "further" away from the TDBU outcome the SUA algorithm stops, the worse is the outcome for the refugees and the more manipulable it is. The social planner might well have a lot of discretion over mild feasibility violations and can easily put refugees into the next matching period if that ensures an outcome closer to the refugee-optimal one for the current batch.

7 Conclusion

This paper has described the matching problems that arise at different stages of refugee resettlement and shown three ways in which systems for refugee resettlement can be enhanced using insights from operations research and market design. First, we showed how to maximize the overall efficiency of the match based entirely on a quality score estimated without eliciting the preferences of refugees or the priorities of the localities. Then, we adapted the

classical top trading cycles mechanism to achieve Pareto efficient, strategy-proof matching of refugees to localities. And finally, we developed a mechanism that finds stable outcomes that are as efficient as possible from the perspective of refugees. Our hope is that if the matching of refugees to localities works well, then government and local communities will be less resistant to accepting refugees. Although we followed the case of Britain, our insights could well apply to other countries. The Canadian system operates much like the British system, but it also includes private and mixed sponsorship (via the "group of five" scheme); our framework could be adapted to the Canadian system by allowing private sponsors to operate alongside the centralized matching system. In the US, however, the Office of Refugee Resettlement delegates one of nine voluntary agencies (e.g., the Lutheran Immigration and Refugee Service) the bulk of the responsibility for allocating each refugee family; a matching system could operate within (or preferably, across) these voluntary agencies.

Beyond the context of refugee resettlement, our work is relevant wherever matching problems incorporate combinatorial feasibility/resource constraints. Combinatorial constraints might arise, for example, in a setting of client–firm matching, in which each client takes up different types of resources available at the firm. While combinatorially-constrained matching markets are structurally messier than more classical many-to-one matching frameworks, they offer an exciting area of theoretical and applied research.

8 Appendix: to be completed

Proof of Proposition 1. To be completed. Introduce an indicator ι. Write out the matrix form with the constraint. □

Proof of Lemma 1. Suppose that a family f receives its first preferences, i.e. $f \in A_\ell(\phi(X))$ such that $\ell \succ_f \ell'$ for any $\ell \neq \ell'$. Then, by definition, f cannot be part of a blocking pair. Suppose now that is not assigned its first preferences, i.e. $f \in A_e ll$ and there exists $\ell' \succ_f \ell$. In this case, (f, ℓ') could potentially form a blocking pair. However observes that either f and ℓ do not have any feasible contract ($(f,h) \notin X$ for any $h \in H_{\ell'}$) or ℓ' has rejected f in the TDBU algorithm. In the first case, f and ℓ' do not form a blocking pair since ℓ' does not have the resources to accommodate f.In the second case, ℓ only rejects f if, whatever happens in other localities, there will always be a set of families with ℓ' as their top-choice and a higher priority than f that cannot be accommodated alongside f. It follows that (f, ℓ') does not constitute a blocking pair. □

By the above argument, if $f \in A_\ell(\phi(X))$ and ℓ is not f's first preference, then any locality it prefers either does not have any feasible contract with it or has rejected it during the TDBU algorithm. By definition, an infeasible contract cannot be part of a stable outcome. Additionally, if ℓ has rejected f, then any contract involving f and a house located within ℓ is incompatible with stability. It follows that in any stable outcome, f is resettled in either ℓ or a locality it likes less. □

By definition, this implies that $P(\phi(X))$ is stable and dominates any other stable assignment if and only if it is feasible. ■ □

Proof of Proposition 2. To be completed. □

Proof of Proposition 3. To be completed. □

9 Technical descriptions of the algorithms

9.1 MTTC

Round $n \geq 1$: Each family $f \in F_{n-1}$ points at their favorite locality that can accommodate them. f does not point at any locality if no locality can accommodate it. That is, locality ℓ can accommodate family f if there exists $h \in H_\ell$ such that $(f,h) \in X_{n-1}$ and $\nu_s^f \leq \kappa_{s,n-1}^\ell$ for all s. Family f points at locality ℓ such that ℓ can accommodate it and, for any other locality ℓ' that can accommodate f, $\ell \succ_f \ell'$.

Each locality points at the family with the highest quality score among its feasible contracts. If there is no feasible contract ($Z_\ell \cap X_{n-1} = \emptyset$) then it does not point at any family. That is, locality ℓ points at family f such that, for some $h \in H_\ell$, $(f,h) \in X_{n-1}$ and, for any $(f', h') \in X_{n-1}$ with $f' \neq f$ and $h' \in H_\ell$, $q(f,h) > q(f',h')$.

Since $X_{n-1} \neq \emptyset$, at least one cycle appears. Each family in a cycle is matched with the house, among those in the locality at which it was pointing for it has the highest quality score. That is, if f points at ℓ and is in a cycle, f is matched with $h \in H_\ell \cap X_{n-1}$ such that for all $h' \in H_\ell \setminus \{h\}$, either $(f,h') \notin X_{n-1}$ or $q(f,h) > q(f,h')$. All pairs matched together are added to the outcome, yielding $Y_n \supset Y_{n-1}$. All matched families are removed, yielding

$F_n \subset F_{n-1}$. All service provisions used by matched families are removed from the localities' capacities, yielding $\kappa_n < \kappa_{n-1}$. That is, for any locality ℓ matched with family f and any service s, $\kappa^{\ell}_{s,n} \equiv \kappa^{\ell}_{s,n-1} - \nu^{f}_{s}$. All contracts involving a matched family or a pair such that the locality can no longer provide the services required by the family are removed from the set of feasible contracts, yielding $X_n \subset X_{n-1}$. That is, $X_n \equiv \{(f,h) \in X_{n-1} \mid f \in F_n \text{ and } \nu^{f}_{s} \leq \kappa^{\ell}_{s,n} \text{ for all } s\}$.

Termination The algorithm terminates and yields the outcome $Y \equiv Y_n$ if $X_n = \emptyset$.

9.2 TDBU

First, we describe how to find guarantees algorithmically. For any $n = 1, ...m$, let $\hat{F}_{f_n,\ell}$ be the set of families with a priority for ℓ higher than f_n's. Then $\hat{F}_{f_n,\ell} = \{f_1, f_2, ..., f_{n-1}\}$. We also define $G_{f_n,\ell} \subseteq \hat{F}_{f_n,\ell}$ as the set of families with a higher priority than f_n who have received a guarantee. As argued above, families in $\hat{F}_{f_n,\ell} \cap \overline{F}_{\ell} \cap \hat{G}_{\ell}$ will be matched with a house in H_ℓ in any stable outcome. Families in $\hat{F}_{f_n,\ell} \setminus (\overline{F}_{\ell} \cap \hat{G}_{f_n,\ell})$ may or may not be matched with a house in H_ℓ. Family f_n therefore receives a guarantee if it can be accommodated alongside any feasible subset of $\hat{F}_{f_n,\ell}$ that contains $\overline{F}_{\ell} \cap \hat{G}_{\ell}$. Formally, f_n receives a guarantee if $T \cup \{f_n\}$ is feasible for all feasible T such that $\overline{F}_{\ell} \cap \hat{G}_{f_n,\ell} \subseteq T \subseteq \hat{F}_{f_n,\ell}$. Intuitively, a family f_n receives a guarantee from its top-choice locality ℓ if the ℓ has enough capacity to accommodate it with any subset of the set of families for which ℓ is top-choice and which have higher priority in ℓ (taking into account that some of these families also have guarantees).

Second, we describe how to find rejections algorithmically. Let $\hat{R}_{f_n,\ell} \subseteq \hat{F}_{f_n,\ell}$ be the set of families with a priority higher than f_n for ℓ who have been rejected. f_n is then rejected if $T \cup \{f_n\}$ is infeasible for any T such that $\overline{F}_{f_n,\ell_i} \cap \hat{G}_{f_n,\ell} \subseteq T \subseteq \hat{F}_{f_n,\ell} \setminus \hat{R}_{f_n,\ell}$ and $f \in (\overline{F}_{\ell_i} \cap \hat{F}_{(f_n,\ell_i)}) \setminus (T \cup R_{(f_n,\ell_i)})$ implies $T \cup \{f\}$ is infeasible. The intuition behind this rejection rule is that stability dictates that a set T of the families with a higher priority of ℓ than f_n is resettled in ℓ. If f_n cannot be accommodated along any of them, it cannot be resettled in ℓ under any stable outcome.

Top-Down Bottom-up Algorithm

Round 0:

Let $X_1 := X$ and arbitrarily label the localities such that $L \equiv \{\ell_1, \ell_2, ..., \ell_{|L|}\}$. Additionally, for any $n \in \mathbb{N}$ and $j = 1, 2, ..., |L|$, let $\ell_{n \cdot |L| + j} := \ell_j$.

Round $i \geq 1$:

Step 0:

If $i > |L|$ and $X_i = X_{i-|L|}$, the algorithm terminates and yields $\phi(X) := X_i$.

Otherwise, consider locality ℓ_i and, given X_i, let F_{ℓ_i} be the set of families who have at least one contract involving a house located within ℓ. Let the elements of that set be labeled in order of priority: $F_{\ell_i} \equiv \{f_{(i,1)}, f_{(i,2)}, ..., f_{(i,|F_{\ell_i}|)}\}$ with $f_{(i,1)} \, \pi_{\ell_i} \, f_{(i,2)} \, \pi_{\ell_i} \, ... \, \pi_{\ell_i} \, f_{(i,|F_{\ell_i}|)}$. Let $G_{(i,0)} := \emptyset$ and $R_{(i,0)} := \emptyset$.

Step $j \geq 1$:

Consider family $f_{(i,j)}$. Given X_i, let $F_{(i,j)} := \hat{F}_{f_{(i,j)},\ell_i} = \{f_{(i,1)}, f_{(i,2)}, ..., f_{(i,j-1)}\}$. Compute $G_{(i,j+1)}$ and $R_{(i,j+1)}$ as follows:

If, for any feasible subset of families T such that $\overline{F}_{\ell_i} \cap G_{(i,j)} \subseteq T \subseteq F_{(i,j)}$, $T \cap \{f_{(i,j)}\}$ is feasible, then family $f_{(i,j)}$ receives a guarantee for locality ℓ_i. Let $G_{(i,j+1)} := G_{(i,j)} \cup \{f_{(i,j)}\}$ and $R_{(i,j+1)} := R_{(i,j)}$.

If $T \cup \{f_{(i,j)}\}$ is infeasible for any T such that $\overline{F}_{\ell_i} \cap G_{(i,j)} \subseteq T \subseteq F_{(i,j)} \setminus R_{(i,j)}$ and $f \in (\overline{F}_{\ell_i} \cap F_{(i,j)}) \setminus (T \cup R_{(i,j)})$ implies $T \cup \{f\}$ is infeasible, then locality ℓ_i rejects family $f_{(i,j)}$. Let $G_{(i,j+1)} := G_{(i,j)}$ and $R_{(i,j+1)} := R_{(i,j)} \cup \{f_{(i,j)}\}$.

If neither of the above conditions is met, let $G_{(i,j+1)} := G_{(i,j)}$ and $R_{(i,j+1)} := R_{(i,j)}$.

If $j < |F_{\ell_i}|$, the algorithm continues to Step $j+1$.

If $j = |F_{\ell_i}|$, update the set of feasible matchings as follows:

$$X_{i+1} := X_i \setminus \big(\{(f,h) \in X_i \mid f \in G_{(i,j+1)},\ h \in H_{\ell'} \text{ for some } \ell' \text{ such that } \ell_i \succ_f \ell'\} \cup R_{(i,j)}\big)$$

The algorithm continues to Round $i+1$.

9.3 ATDBU

Augmented Top-Dow Bottom-Up (ATDBU) Algorithm

Round 0:

Given the set of artificial guarantees G_A, artificial rejections R_A and contracts $\tilde{X}$, let $G_{A,(1,1)} := G_A$, $R_{A,(1,1)} := R_A$ and $X_1 := \tilde{X}$. Arbitrarily label the localities such that $L \equiv \{\ell_1, \ell_2, ..., \ell_{|L|}\}$. Additionally, for any $n \in \mathbb{N}$ and $j = 1, 2, ..., |L|$, let $\ell_{n \cdot |L|+j} := \ell_j$.

Round $i \geq 1$:

Step 0:

If $i > |L|$, $G_{A,(i,1)} = R_{A,(i-|L|,1)}$, $R_{A,(i,1)} = R_{A,(i-|L|,1)}$ and $X_i = X_{i-|L|}$, the algorithm terminates and yields $\tilde{\phi}(G_A, R_A, \tilde{X}) := (G_{A,(i,1)}, R_{A,(i,1)}, X_i)$.

Otherwise, consider locality ℓ_i and, given X_i, let F_{ℓ_i} be the set of families who have at least one contract involving a house located within ℓ. Let the elements of that set be labeled in order of priority: $F_{\ell_i} \equiv \{f_{(i,1)}, f_{(i,2)}, ..., f_{(i,|F_{\ell_i}|)}\}$ with $f_{(i,1)}\, \pi_{\ell_i}\, f_{(i,2)}\, \pi_{\ell_i} \ldots \pi_{\ell_i}\, f_{(i,|F_{\ell_i}|)}$. Let $G_{(i,0)} := \emptyset$ and $R_{(i,0)} := \emptyset$.

Step $j \geq 1$:

Consider family $f_{(i,j)}$. Given X_i, let $F_{(i,j)} := \hat{F}_{f_{(i,j)},\ell_i} = \{f_{(i,1)}, f_{(i,2)}, ..., f_{(i,j-1)}\}$. Compute $G_{(i,j+1)}$, $R_{(i,j+1)}$, $G_{A,(i,j+1)}$ and $R_{A,(i,j+1)}$ as follows:

If, for any feasible subset of families T such that $\overline{F}_{\ell_i} \cap G_{(i,j)} \subseteq T \subseteq F_{(i,j)}$, $T \cap \{f_{(i,j)}\}$ is feasible, then family $f_{(i,j)}$ receives a guarantee for locality ℓ_i. Let $G_{(i,j+1)} := G_{(i,j)} \cup \{f_{(i,j)}\}$, $R_{(i,j+1)} := R_{(i,j)}$, $G_{A,(i,j+1)} := G_{A,(i,j)} \setminus (f_{i,j}, \ell_i)$ and $R_{A,(i,j+1)} := R_{A,(i,j)}$.

If $T \cup \{f_{(i,j)}\}$ is infeasible for any T such that $\overline{F}_{\ell_i} \cap G_{(i,j)} \subseteq T \subseteq F_{(i,j)} \setminus R_{(i,j)}$ and $f \in (\overline{F}_{\ell_i} \cap F_{(i,j)}) \setminus (T \cup R_{(i,j)})$ implies $T \cup \{f\}$ is infeasible, then locality ℓ_i rejects family $f_{(i,j)}$.

If $(f_{(i,j)}, \ell_i) \in G_{A,(i,j)}$, the algorithm terminates and yields $\tilde{\phi}(G_A, R_A, \tilde{X}) := \{\emptyset\}$.

Otherwise, let $G_{(i,j+1)} := G_{(i,j)}$, $R_{(i,j+1)} := R_{(i,j)} \cup \{f_{(i,j)}\}$, $G_{A,(i,j+1)} := G_{A,(i,j)}$ and $R_{A,(i,j+1)} := R_{A,(i,j)}$.

If $f_{(i,j)}$ did not receive a rejection and $T \cup \{f_{(i,j)}\}$ is infeasible for any T such that $\overline{F}_{\ell_i} \cap (G_{(i,j)} \cup G_{A,(i,j)}) \subseteq T \subseteq (F_{(i,j)} \setminus R_{(i,j)}) \cup (\overline{F}_{\ell_i} \cap G_{A,(i,j)})$ and $f \in (\overline{F}_{\ell_i} \cap F_{(i,j)}) \setminus (T \cup R_{(i,j)})$ implies $T \cup \{f\}$ is infeasible, then family $f_{(i,j)}$ receives an artificial rejection from locality ℓ_i.

If $(f_{(i,j)})$ received a guarantee ($f_{(i,j)} \in G_{(i,j+1)}$), the algorithm terminates and yields $\tilde{\phi}(G_A, R_A, \tilde{X}) := \{\emptyset\}$.

Otherwise, let let $G_{(i,j+1)} := G_{(i,j)}$, $R_{(i,j+1)} := R_{(i,j)}$, $G_{A,(i,j+1)} := G_{A,(i,j)}$ and $R_{A,(i,j+1)} := R_{A,(i,j)} \cup \{f_{(i,j)}\}$.

If $f_{(i,j)}$ has received neither a guarantee, a rejection nor an artificial rejection, let $R_{(i,j+1)} := R_{(i,j)}$, $G_{A,(i,j+1)} := G_{A,(i,j)}$ and $R_{A,(i,j+1)} := R_{A,(i,j)}$

If $(f_{(i,j)}, \ell_i) \in G_{A,(i,j)}$, $G_{(i,j+1)} := G_{(i,j)} \cup \{(f_{(i,j)}, \ell_i)\}$.

Otherwise $G_{(i,j+1)} := G_{(i,j)}$.

If $j < |F_{\ell_i}|$, the algorithm continues to Step $j+1$.

If $j = |F_{\ell_i}|$, update the set of feasible matchings as follows:

$$X_{i+1} := X_i \setminus \big\{\{(f,h) \in X_i \mid f \in G_{(i,j+1)},\, h \in H_{\ell'} \text{ for some } \ell' \text{ such that } \ell_i \succ_f \ell'\} \cup R_{(i,j)} \cup R_{A,(i,j)}$$

The algorithm continues to Round $i+1$.

9.4 SUA

Stable and Undominated Assignment Algorithm (SUA)

Round 0:

Run the TDBU algorithm to obtain $\phi(X)$. Let the families be ordered from 1 to $|F|$ so that $F \equiv \{f^1, f^2, ..., f^{|F|}\}$. Let $X^1 := \phi(X)$, $G_A^1 := \emptyset$ and $R_A^1 := \emptyset$. Additionally, initialize the family index $k^1 := 1$ and the case $c^1 :=$ "guarantee".

Round $i \geq 1$:

If $c^i =$ "guarantee":

Run the ATDBU algorithm to obtain $\tilde{\phi}(G_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, R_A^i, X^i)$.

If $\tilde{\phi}(G_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, R_A^i, X^i) = \{\emptyset\}$, then let $G_A^{i+1} := G_A^i$, $R_A^{i+1} := R_A^i$, $X^{i+1} := X^i$ and $c^{i+1} :=$ "rejection". The algorithm continues to Round $i+1$.

If $\tilde{\phi}(G_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, R_A^i, X^i) \in \mathcal{P}(F \times L) \times P(F \times L) \times P(F \times L)$, then let $(G_A^{i+1}, R_A^{i+1}, X^{i+1}) := \tilde{\phi}(G_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, R_A^i, G^i)$.

If $k^i = |F|$, the algorithm terminates and yields $\psi(X) := A^*(X^i)$.

Otherwise, let $k^{i+1} := k^i + 1$ and $c^i :=$ "guarantee". The algorithm continues to Round $i+1$.

If $c^i =$ "rejection":

Run the ATDBU algorithm to obtain $\tilde{\phi}(G_A^i, R_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, X^i \setminus \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\})$.

If $\tilde{\phi}(G_A^i, R_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, X^i \setminus \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}) = \{\emptyset\}$, then

If $k^i > 1$, let $(G_A^{i+1}, R_A^{i+1}, X^{i+1}) := (G_A^j, R_A^j, X^j)$, where j is the last round such that $k^j = k^i - 1$ ($j \geq j'$ for any j' such that $k^{j'} = k^i - 1$). Let $k^{i+1} := k^i$ and $c^{i+1} =$ "rejection". The algorithm continues to Round $i+1$.

If $k^i = 1$, the set of stable assignments is empty. The algorithm terminates and yields $\psi(X) := \{\emptyset\}$.

If $\tilde{\phi}(G_A^i, R_A^i \cup \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}, X^i \setminus \{(f^{k^i}, \overline{\ell}_{f^{k^i}})\}) \in \mathcal{P}(F \times L) \times \mathcal{P}(F \times L) \times \mathcal{P}(F \times L)$, then let $(G_A^{i+1}, R_A^{i+1}, X^{i+1}) := \tilde{\phi}(G_A^i, R_A^i \cup \{(f^{k^i-1}, \overline{\ell}_{f^{k^i-1}})\}, X^i \setminus \{(f^{k^i-1}, \overline{\ell}_{f^{k^i-1}})\})$, $k^{i+1} := k^i$ and $c^{i+1} :=$ "guarantee". The algorithm continues to Round $i+1$.

References

Abdulkadiroğlu, A., P. A. Pathak, and A. E. Roth (2009). Strategyproofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *American Economic Review 99*, 1954–1978.

Abdulkadiroğlu, A. and T. Sönmez (1999). House allocation with existing tenants. *Journal of Economic Theory 88*, 233–260.

Abdulkadiroğlu, A. and T. Sönmez (2003). School choice: A mechanism design approach. *American Economic Review 93*, 729–747.

Alva, S. (2015). Pairwise stability and complementarity in many-to-one matching. Mimeo.

Andersson, T. and L. Ehlers (2016). Assigning refugees to landlords in sweden: Stable maximum matchings. Working Paper 18, Lund University.

Ashlagi, I., M. Braverman, and A. Hassidim (2014). Stability in large matching markets with complementarities. *Operations Research 62*(4), 713–732.

Ashlagi, I. and P. Shi (2015). Optimal allocation without money: An engineering approach. *Management Science 62*(4), 1078–1097.

Åslund, O., P.-A. Edin, P. Fredriksson, and H. Grönqvist (2011). Peers, neighborhoods, and immigrant student achievement: Evidence from a placement policy. *American Economic Journal: Applied Economics 3*(2), 67–95.

Åslund, O. and P. Fredriksson (2009). Peer effects in welfare dependence quasi-experimental evidence. *Journal of human resources 44*(3), 798–825.

Åslund, O., J. Östh, and Y. Zenou (2010). How important is access to jobs? old question, improved answer. *Journal of Economic Geography 10*(3), 389–422.

Åslund, O. and D.-O. Rooth (2007). Do when and where matter? initial labour market conditions and immigrant earnings. *The Economic Journal 117*(518), 422–448.

Baïou, M. and M. Balinski (2000). The stable admissions polytope. *Mathematical programming 87*(3), 427–439.

Balinski, M. and T. Sönmez (1999). A tale of two mechanisms: Student placement. *Journal of Economic Theory 84*, 73–94.

BBC (2016). Attitudes harden towards refugees from Syria and Libya, BBC poll suggests. *BBC 4 February*.

Bodoh-Creed, A. L. (2016). Programming Approaches to School Choice Problems. Mimeo.

Damm, A. P. (2014). Neighborhood quality and labor market outcomes: Evidence from quasi-random neighborhood assignment of immigrants. *Journal of Urban Economics 79*, 139–166.

Dedman, S. (2015, October). Syria refugees to cost 'up to £23k each' in first year in UK. *British Broadcasting Corporation*.

Delacrétaz, D. (2016, February). Matching with quantity. Mimeo.

Dernbach, A. (2015). Germany suspends Dublin agreement for Syrian refugees (translated by Samuel Morgan). *Der Tagesspiegel*.

Dur, U. and M. U. Ünver (2015, March). Two-sided matching with balanced exchange: Tuition and worker exchanges. Mimeo.

Echenique, F. and M. B. Yenmez (2007). A solution to matching with preferences over colleagues. *Games and Economic Behavior 59*, 46–71.

Ehlers, L. (2008). Truncation strategies in matching markets. *Mathematics of Operations Research 33*(2), 327–335.

Erdil, A. and H. Ergin (2008). What's the matter with tie-breaking? improving efficiency in school choice. *American Economic Review 98*(3), 669–689.

Featherstone, C. R. (2014). Rank efficiency: Investigating a widespread ordinal welfare criterion. Technical report, Working Paper.

Fragiadakis, D., A. Iwasaki, P. Troyan, S. Ueda, and M. Yokoo (2016). Strategyproof matching with minimum quotas. *ACM Transactions on Economics and Computation 4*(1), 6.

Fréville, A. (2004). The multidimensional 0–1 knapsack problem: An overview. *European Journal of Operational Research 155*(1), 1–21.

Goto, M., N. Hashimoto, A. Iwasaki, Y. Kawasaki, S. Ueda, Y. Yasuda, and M. Yokoo (2014). Strategy-proof matching with regional minimum quotas. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1225–1232. International Foundation for Autonomous Agents and Multiagent Systems.

Hatfield, J. W. and F. Kojima (2008). Matching with contracts: Comment. *American Economic Review 98*, 1189–1194.

Hatfield, J. W. and S. D. Kominers (2016). Contract design and stability in many-to-many matching. *Games and Economic Behavior*. forthcoming.

Hatfield, J. W. and P. Milgrom (2005). Matching with contracts. *American Economic Review 95*, 913–935.

Hifi, M., M. Michrafy, and A. Sbihi (2004). Heuristic algorithms for the multiple-choice multidimensional knapsack problem. *Journal of the Operational Research Society 55*(12), 1323–1332.

Home Office (2015, October). Syrian Vulnerable Person Resettlement (VPR) Programme: Guidance for local authorities and partners. Technical report, Home Office.

Home Office (2016a, 26 May). Asylum: Statistics for January to March 2016. Technical report.

Home Office (2016b). Full Community Sponsorship: Guidance for prospective sponsors. Technical report, Home Office, Department for Communities and Local Government, Department for International Development.

House of Commons Library (2016, June). Syrian refugees and the UK response. Briefing paper 06805, House of Commons Library.

International Organization for Migration (2015, 22 December 2015). Irregular Migrant, Refugee Arrivals in Europe Top One Million in 2015. Press release, IOM.

Jones, W. and A. Teytelboym (2016, March). The local refugee match: Aligning refugees' preferences with the capacities and priorities of localities. Mimeo.

Kamada, Y. and F. Kojima (2012). Stability and strategy-proofness for matching with constraints: A problem in the Japanese medical match and its solution. *American Economic Review 102*(3), 366–370.

Kamada, Y. and F. Kojima (2015). Efficient matching under distributional constraints: Theory and applications. *American Economic Review 105*, 67–99.

Kellerer, H., U. Pferschy, and D. Pisinger (2004). The multiple-choice knapsack problem. In *Knapsack Problems*, pp. 317–347. Springer.

Kelso, A. S. and V. P. Crawford (1982). Job matching, coalition formation, and gross substitutes. *Econometrica 50*, 1483–1504.

Kesten, O. (2010). School choice with consent. *Quarterly Journal of Economics 125*, 1297–1348.

Kojima, F. (2015). Finding all stable matchings with couples. *Journal of Dynamics and Games 2*(3/4), 321–330.

Kojima, F. and P. A. Pathak (2009). Incentives and stability in large two-sided matching markets. *American Economic Review 99*, 608–627.

Krysta, P. and J. Zhang (2016). House markets with matroid and knapsack constraints. In *ICALP*.

Martello, S. and P. Toth (1980). Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research 4*(4), 276–283.

McKenna, K. (2015, November). It's a wonderful life: Scotland's holiday isle prepares to greet its first Syrian refugees. *The Observer*.

Milgrom, P. and I. Segal (2014). Deferred-acceptance auctions and radio spectrum reallocation. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pp. 185–186. ACM.

Moraga, J. F.-H. and H. Rapoport (2014). Tradable immigration quotas. *Journal of Public Economics 115*, 94–108.

Pathak, P. A. and T. Sönmez (2013). School admissions reform in Chicago and England: Comparing mechanisms by their vulnerability to manipulation. *American Economic Review 103*, 80–106.

Pycia, M. and M. U. Ünver (2016). Incentive compatible allocation and exchange of discrete resource. *Theoretical Economics*. Forthcoming.

Reid, S. (2015, December). Senseless! Why are Syrian refugees being foisted on a remote Scottish island with high unemployment and poverty - then given perks some locals don't enjoy? *Daily Mail*.

Roth, A. E. (1984a). The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy 92*, 991–1016.

Roth, A. E. (1984b). Stability and polarization of interests in job matching. *Econometrica 52*, 47–57.

Roth, A. E. (1985). The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory 36*, 277–288.

Roth, A. E. and E. Peranson (1999). The redesign of the matching market for American physicians: Some engineering aspects of economic design. *American Economic Review 89*, 748–780.

Roth, A. E. and A. Postlewaite (1977). Weak versus strong domination in a market with indivisible goods. *Journal of Mathematical Economics 4*(2), 131–137.

Roth, A. E. and U. G. Rothblum (1999). Truncation strategies in matching markets—in search of advice for participants. *Econometrica 67*(1), 21–43.

Roth, A. E., U. G. Rothblum, and J. H. Vande Vate (1993). Stable matchings, optimal assignments, and linear programming. *Mathematics of operations research 18*(4), 803–828.

Roth, A. E., T. Sönmez, and M. Ünver (2007). Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *The American economic review 97*(3), 828–851.

Roth, A. E., T. Sönmez, and M. U. Ünver (2004). Kidney exchange. *The Quarterly Journal of Economics 119*(2), 457–488.

Roth, A. E., T. Sönmez, and M. U. Ünver (2005a). A kidney exchange clearinghouse in new england. *American Economic Review Papers & Proceedings 95*(2), 376–380.

Roth, A. E., T. Sönmez, and M. U. Ünver (2005b). Pairwise kidney exchange. *Journal of Economic theory 125*(2), 151–188.

Roth, A. E. and M. Sotomayor (1989). The college admissions problem revisited. *Econometrica 57*, 559–570.

Roth, A. E. and M. Sotomayor (1990). *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press.

Schuck, P. H. (1997). Refugee burden-sharing: A modest proposal. *Yale Journal of International Law 22*, 243.

Sethuraman, J., C.-P. Teo, and L. Qian (2006). Many-to-one stable matching: geometry and fairness. *Mathematics of Operations Research 31*(3), 581–596.

Shapley, L. and H. Scarf (1974). On cores and indivisibility. *Journal of Mathematical Economics 1*(1), 23–37.

Sinha, P. and A. A. Zoltners (1979). The multiple-choice knapsack problem. *Operations Research 27*(3), 503–515.

Slaugh, V. W., M. Akan, O. Kesten, and M. U. Ünver (2016). The pennsylvania adoption exchange improves its matching process. *Interfaces 46*(2), 133–153.

Song, Y., C. Zhang, and Y. Fang (2008). Multiple multidimensional knapsack problem and its applications in cognitive radio networks. In *MILCOM 2008-2008 IEEE Military Communications Conference*, pp. 1–7. IEEE.

Sönmez, T. and T. B. Switzer (2013). Matching with (branch-of-choice) contracts at United States Military Academy. *Econometrica 81*, 451–488.

Statistisches Bundesamt (2016). Nettozuwanderung von Ausländerinnen und Ausländern im Jahr 2015 bei 1,1 Millionen. Pressemitteilung 105, Statistisches Bundesamt.

UNHCR (2016). Global trends. Report, UNHCR, Geneva.

Wells, R. (1999, April). UK Kosovo refugees settle in . *BBC Monday, April 26*.