

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

(Job Market Paper)

Latest version: www.t8el.com/jmp.pdf

November 8, 2016

Abstract

Over 100,000 refugees are permanently resettled from refugee camps to hosting countries every year. Nevertheless, refugee resettlement processes in most countries are *ad hoc*, accounting for neither the priorities of hosting communities nor the preferences of refugees themselves. Building on models from two-sided matching theory, we introduce a new framework for *matching with multidimensional constraints* that models refugee families' needs for multiple units of different services, as well as the service capacities of local areas. We propose several refugee resettlement mechanisms that can be used by hosting countries under various institutional and informational constraints. Our mechanisms can improve match efficiency, incentivize refugees to report where they would like to settle, and respect priorities of local areas thereby encouraging them to accept more refugees overall. Beyond the refugee resettlement context, our model has applications ranging from the allocation of daycare slots to the incorporation of complex diversity constraints in public school assignment.

*The authors appreciate the helpful comments of Tommy Andersson, Ramnik Arora, Georgy Artermov, Haris Aziz, Ivan Balbuzanov, Péter Biró, Vincent Crawford, Sarah Glatté, Jens Gudmundsson, Guillaume Haeringer, Cameron Hepburn, Will Jones, Fuhito Kojima, Bettina Klaus, Simon Loertscher, Alex Nichifor, Alvin Roth, Yang Song, Tayfun Sönmez, Bassel Tarbush, and seminar participants at the 2016 NBER Market Design Working Group Meeting, Melbourne Centre for Market Design, Hungarian Academy of Sciences, Transatlantic Theory Workshop, University of Oxford, and Lund University. 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 for the generous fellowship and hospitality of the EU Centre for Shared Complex Challenges at the University of Melbourne.

Contents

1	Introduction	3
2	Institutional context	10
3	Model	13
3.1	Basic ingredients	13
3.1.1	Relationship to prior models	14
3.2	Housing	16
3.2.1	Housing as service constraints	16
3.2.2	General permissible housing constraints	16
4	Welfare maximization without preferences or priorities	18
5	Accounting for preferences of refugee families	21
5.1	Using only preferences of refugee families	23
5.2	Improving outcomes of refugee families from an initial allocation	26
6	Accounting for the preferences of refugee families and the priorities of localities	30
6.1	Stable outcomes	30
6.1.1	Identical priorities	31
6.1.2	Finding stable outcomes in general	32
6.2	Quasi-stable outcomes	37
6.2.1	Mechanism for a family-optimal quasi-stable outcome	38
6.2.2	Strategic properties of the PFDA algorithm	40
6.2.3	A strategy-proof mechanism that finds quasi-stable outcomes	42
7	Tradeoffs between different mechanisms	46
8	Applications to other resettlement contexts	48
8.1	United States	48
8.2	Canada	48
9	Further applications of matching with multidimensional constraints	49
9.1	Multidimensional diversity constraints	49
9.2	Daycare matching	50
10	Conclusion	50
Appendix		51
A	The running Example	51
B	Proofs	53
C	The Top Choice algorithm	65
C.1	The Top-Down Bottom-Up (TDBU) algorithm	65
C.2	The Augmented Top-Down Bottom-Up (ATDBU) algorithm	69
C.3	Depth-First Search (DFS)	70
D	Additional examples of PFDA and MRDA algorithms	75
D.1	Another example to compare PFDA and MRDA algorithms	75
D.2	Two examples of manipulability of the PFDA algorithm	78
D.3	Example of MRDA algorithm producing a family-optimal outcome	80

1 Introduction

By the end of 2015, there were 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* under the mandate of the United Nations High Commission for Refugees (UNHCR). Half of these refugees come from just three countries: Syria, Afghanistan, and Somalia.¹

In recent years, following escalating conflict in the Middle East, the influx of refugees into Europe has drastically increased. In 2015, a million people arrived by sea in Greece and Italy, seeking asylum in Europe (International Organization for Migration, 2015). Germany received almost 450,000 asylum applications in 2015 compared to 175,000 applications in 2014 (Eurostat, 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 European Union (EU) country in which they arrive—has effectively been abandoned.² Consequently, numerous countries within the EU and elsewhere have begun to reconsider the systems they use to register, process, and allocate refugees to local areas.³ Economists, refugee specialists, and policymakers are developing a number of promising solutions for sharing the burden of refugee resettlement across countries, including tradeable quota systems (Schuck, 1997; Moraga and

¹In 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. Refugee status is envisioned to be temporary, but the majority of refugees spend years in camps and temporary settlements in developing countries without residence rights or work permits. There are also 5.2 million Palestinian refugees registered by the United Nations Relief and Works Agency for Palestine Refugees in the Near East (UNRWA). 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. Those who 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.

²Article 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).

³Most asylum seekers are held in detention centers until their refugee status applications have been 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 asylum seekers who have been granted refugee status with accommodation and welfare support, via a process known as “dispersal”. The insights in this paper can also be applied directly to refugee dispersal schemes.

Rapoport, 2014).

The UNHCR estimates that around one million refugees will not be able return to their country safely in the future.⁴ These are some of the most vulnerable refugees in the world: a third of them are fleeing persecution, a quarter are survivors of torture, and a tenth are women and girls at risk of violence (UNHCR, 2016). UNHCR deems these refugees eligible for *resettlement* in states that agree to give them permanent residence.⁵ Most resettlement places are provided by the United States, Canada, Australia, and the Nordic countries. But, in 2015, only 107,000 refugees were resettled (UNHCR, 2016).⁶ In fact, in any given year over the past decade only around ten percent of refugees who are in need of resettlement were actually resettled.

There is, therefore, a huge shortage of resettlement places. Yet, despite this shortage, a recent UNHCR report notes that since 2009:

“...the total annual number of resettlement country places...were *not fully utilized*” (UNHCR, 2015, p. 23)

Not only are resettlement places being wasted but also little attention has been paid to the process of determining which refugees ought be resettled to which local areas in the hosting country. This is despite ample evidence that the local area (or *locality*, for short) to which refugees are initially matched matters a great deal for the refugees’ lifetime outcomes (Åslund and Rooth, 2007; Åslund and Fredriksson, 2009; Åslund et al., 2010, 2011; Damm, 2014; Feywerda and Gest, 2016). 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. But there is a growing consensus that the role of resettlement in refugee protection needs to be drastically expanded. In September 2016, following the first ever heads-of-state UN summit on refugees, 50 countries pledged to resettle at least 360,000 refugees in 2017 (BBC, 2016b). In this paper, we take the process of allocating of refugees to localities seriously, introducing and analyzing several matching market design approaches that balance competing welfare, incentive, and stability objectives, while being mindful of computational constraints.

⁴This number is projected to rise to 1.19 million in 2017, a fifty percent increase since 2012 (UNHCR, 2016).

⁵In the past, the international community has often managed to react quickly to humanitarian crises by instituting rapid and comprehensive resettlement programs. 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.

⁶81,000 of these refugees were resettled with UNHCR’s assistance. In fact, a total of 134,000 files were submitted by the UNHCR for resettlement in 2015 (UNHCR, 2016).

Importance of matching market design in refugee resettlement

One simple policy goal is to try and maximize refugees' welfare based on the observable characteristics of refugees and local areas. This approach allows us, for example, to maximize the total number of families resettled, but it does not account in any way for the refugees' personal preferences over hosting communities. Refugees' preferences matter for two reasons. First, refugees have information about their own aspirations, which can affect the match quality—and which cannot be directly observed by government authorities.⁷ Second, government agencies spend significant upfront resources on integrating refugees into local communities; those resources are effectively wasted if refugees leave their assigned localities soon after arrival.⁸ Consequently, at least in the early stages of the resettlement process, policymakers want to minimize internal migration of refugees away from their initially assigned localities. The priorities and hosting capacities of localities matter, too. This is particularly salient when the participation of localities in resettlement is voluntary, in which case giving localities a say might serve as an additional incentive to participate. Localities are more likely to follow through on their promises to host refugees if they have some control over which refugees they would be expected to host.

All three of the aforementioned policy concerns are playing out in the British program to resettle Syrian refugees. The United Kingdom is similar to other popular resettlement destinations such as Australia, Canada, and the United States, in that it is geographically remote and thus largely able to control overall refugee entry. When the UK dramatically increased its target for resettling Syrian refugees in 2015, the Home Office's focus was on developing and unifying technological solutions to organize and process assignment, without any specific attention to refugees' preferences and localities' priorities.⁹ After early numerical targets have been 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 which could undermine the willingness of localities to participate. Initially, localities around the UK showed a lot of willingness to host refugees and dozens volunteered to accept them. However, anti-immigration tensions in the UK have since been increasing—especially after the British referendum to leave the EU in June 2016—and the Home Office might have to focus on locality priorities in order to entice more localities into participating. In our work, we follow the evolution of the Syrian refugee resettlement policy in the UK, describing the

⁷For example, a refugee might be keen to retrain or start a business.

⁸One such investment is the provision of language classes for children and adults.

⁹The 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).

appropriate matching market design solution at each step. As we show, our insights are valuable both for designing new resettlement programs and for improving existing ones.

Contribution of this paper

This paper proposes seven mechanisms that hosting countries can use for refugee resettlement under different informational and institutional constraints. Our work draws upon classical matching mechanisms from contexts such as public school choice and housing allocation. In a standard school choice model, for example, there is a number of schools with different numbers of seats and any one student takes up exactly one school seat.¹⁰ However, refugee resettlement requires us to take into account a feature 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 seats, hospital beds, slots in language classes, and employment training programs. Thus, there are explicit multidimensional constraints that limit the central authority’s ability to allocate refugees to localities simply on the basis of housing needs. These additional constraints render most standard matching mechanisms for allocation of objects, houses, or school seats insufficient for refugee resettlement.

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 multidimensional knapsack problem”. We show how to apply the “branch-and-bound method”, familiar in operations research and combinatorial optimization, to find an exact solution to this problem (Proposition 1).

In the case where the social planner fully respects the preferences of refugee families, we show that the Multidimensional Top Trading Cycles (MTTC) algorithm—a slight modification on the classical Top Trading Cycles (TTC) algorithm of [Shapley and Scarf \(1974\)](#)—allows us to incorporate multidimensional capacities of localities and housing constraints and obtain a Pareto-efficient mechanism in which refugee families do not have any incentive to misreport their preferences (Proposition 2). However, the social planner may not want to rely entirely on the preferences of refugee families to determine the allocation. Instead the planner might start with an exogenous tentative allocation of families to localities: For example, with the maximal outcome that has been obtained from the integer program. We show that in this case the MTTC algorithm fails because some Pareto-improving trading

¹⁰Students have heterogeneous preferences over schools and schools have priorities over students (having a sibling or living in the neighborhood typically gives students a higher priority). The social planner’s objective is to elicit truthful preferences over schools from students (schools are assumed to be non-strategic and school seats are treated as objects) and to deliver an efficient matching of students to schools in which no student envies another student’s seat.

cycles might be infeasible. Moreover, obtaining a Pareto-efficient outcome without matching any refugee family to a locality it likes less than its match in the maximal outcome imposes a huge computational burden. In this case, we propose a Serial Multidimensional Top Trading Cycles (SMTTC) algorithm that reduces waste and finds simple Pareto-improving cycles in a strategy-proof way (Proposition 3).

When priorities of localities also need to be taken into account as part of the design, new trade-offs arise unless these priorities are identical (Proposition 4). In particular, *stable* outcomes—i.e. outcomes that fully respect the priorities of localities and the preferences of refugee families—may not exist.¹¹ For this case, however, building on insights by Delacrétaz (2014), we are able to develop an algorithm which finds a stable outcome—whenever such an outcome exists—that is Pareto-undominated for the families by any other stable outcome (Proposition 5).

In general, determining whether a stable outcome exists in our model is a computationally intractable problem. For that reason, we introduce an alternative solution concept called *quasi-stability*. Quasi-stable outcomes ensure that a family can only block an outcome if the family is not the lowest priority among families matched to the desired locality. We show that family-optimal quasi-stable outcomes can be found via a modification of the classical Deferred Acceptance (DA) algorithm (Gale and Shapley, 1962), which we call the Priority-Focused Deferred Acceptance (PFDA) algorithm (Proposition 6). However, unlike in contexts such as school choice, this modification of the DA algorithm is manipulable, except under low information conditions (Proposition 7). To address this, we also develop a benchmark strategy-proof and quasi-stable mechanism, called the Maximum Rank Deferred Acceptance (MRDA) algorithm (Proposition 8). When looking for quasi-stable outcomes, we show that there is a clear trade-off between truth-telling incentives and efficiency.

Relationship to prior work

Matching markets for refugee resettlement were first proposed by Moraga and Rapoport (2014) as a part of a system of international refugee quota trading (Schuck, 1997). In the international context of matching refugees to countries, however, the refugee matching market is “thick”—any country can be expected to host any family up to its capacity—and can be reasonably modeled as a standard school choice problem (Abdulkadiroğlu and Sönmez, 2003). Jones and Teytelboym (2016) introduced the idea of refugee resettlement matching in the national context and pointed out the multidimensional constraints and the thinness of matching markets that arise on the local level. The theory we develop in this

¹¹In one-sided object allocation settings such as school choice, stability is often referred to as “elimination of justified envy” (Abdulkadiroğlu and Sönmez, 2003), or simply as “fairness” (Balinski and Sönmez, 1999).

paper will allow us to realize [Jones and Teytelboym’s \(2016\)](#) ideas for local refugee matching.

Our work draws upon and contributes to the applied literature on the design and implementation of matching mechanisms. The most closely related literature has focused on design of many-to-one markets in which agents on one side take up individual slots made available by the other side: doctors take up individual residencies at hospitals ([Roth, 1984a](#)); children occupy individual seats at schools ([Balinski and Sönmez, 1999](#); [Abdulkadiroğlu and Sönmez, 2003](#)); families move between individual houses ([Abdulkadiroğlu and Sönmez, 1999](#)); and cadets serve in individual slots in their branches of military service ([Sönmez, 2013](#); [Sönmez and Switzer, 2013](#)). Even when agents take up slots at multiple levels—as in the Japanese residency match, in which doctors not only occupy slots at hospitals but also count against regional quotas ([Kamada and Kojima, 2015](#))—the slots are assumed to have a hierarchical structure.¹² With appropriate conditions on the preferences of the two sides, such as “responsiveness” ([Gale and Shapley, 1962](#); [Crawford and Knoer, 1981](#); [Roth, 1985](#); [Roth and Sotomayor, 1989, 1990](#)) or “substitutability” ([Kelso and Crawford, 1982](#); [Roth, 1984b](#); [Hatfield and Milgrom, 2005](#)), these many-to-one matching markets always have a lattice of stable outcomes. In our setting, because refugee families may take up several units of different services, we cannot ensure the existence of stable outcomes—and thus, we are unable to directly apply classical matching technologies and mechanisms. In fact, even determining whether stable outcomes exist in our model can be computationally intractable ([McDermid and Manlove, 2010](#)).

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 (NRMP), in which residents may view jobs in nearby hospitals as complementary ([Roth and Peranson, 1999](#); [Klaus and Klijn, 2005](#); [Klaus et al., 2007](#); [Haake and Klaus, 2009](#)). There are a number of algorithms that can find stable matchings in the couples model whenever they exist ([Echenique and Yenmez, 2007](#); [Kojima, 2015](#)). However, the structure of our problem is different to the matching with couples problem since the barriers to stability in our context arise from the constraints on the locality (hospital) side, rather than from the family (doctor) side (as in the couples problem). Stable outcomes also do not exist in general in the market for trainee teachers in Slovakia and Czechia, where teachers are expected to teach two out of three subjects and schools have capacities for each subject ([Ceclárová et al., 2015](#)). Another difficult case for market design has been matching with minimum quotas, in which stable outcomes also typically do not exist ([Goto et al., 2014](#); [Fragiadakis et al., 2016](#)). [Milgrom and Segal \(2014\)](#)

¹²So whenever a resident takes up a place in a hospital, she would also take up a place in the region in which the hospital is located.

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 that of [Milgrom and Segal \(2014\)](#) is that we limit localities (which correspond to mobile network operators in their context) to have responsive preferences, while allowing refugees (TV stations in their context) to have heterogeneous preferences over localities (rather than only caring about compensation as in their context).

Even when stable matchings exist in our setting, respecting preferences and priorities through stability may create significant welfare losses in strategy-proof mechanisms ([Abdulkadiroğlu et al., 2009](#); [Erdil and Ergin, 2008](#); [Kesten, 2010](#)). We seek to strike a balance between stability and efficiency goals: the Top Choice algorithm that we propose is manipulable, but it ensures that any stable outcome it finds (whenever one exists) is Pareto-undominated for families by any other stable outcome. On the other hand, when we use quasi-stability as our solution concept, insisting on strategy-proofness can come at a high cost to efficiency.

In many matching market design settings, such as school choice in New Orleans or housing allocation, in which efficiency is prioritized over stability by the social planner, the Top Trading Cycles algorithm ([Balinski and Sönmez, 1999](#); [Abdulkadiroğlu and Sönmez, 2003](#)) or its modifications ([Pápai, 2000](#); [Dur and Ünver, 2015](#)) are used instead of stable mechanisms. [Pycia and Ünver \(2016\)](#) show that in settings where agents have single-unit demands over objects, all Pareto-efficient mechanisms that cannot be manipulated by a group of agents can be represented in terms of a general class of Trading Cycles mechanisms. Interestingly, in school choice settings, these variations on the Top Trading Cycles algorithms usually have the same properties irrespective of the initial endowment. In our case, this is no longer true because not all trading cycles are feasible. Hence, when we start from an exogenous allocation of refugees to localities, achieving Pareto efficiency without violating individual rationality, strategy-proofness or imposing a huge computational burden is not always possible.

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. They 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.¹³

The remainder of the paper is organized as follows. In Section 2, we describe the institutional context of the Syrian Vulnerable Persons Resettlement Programme in the UK. We describe the formal model in Section 3 and introduce a running **Example** that we use throughout the paper. In Section 4, we show how the case in which neither preferences nor priorities are taken into account can be solved as a multiple multidimensional knapsack problem. In Section 5, we explain how two variations on the Top Trading Cycles algorithm can fully incorporate preferences of refugees. In Section 6, we present four solutions to the case where priorities of the localities need to be respected as well. We point out the tradeoffs between the mechanisms in Section 7. In Section 8, we describe how our model can be applied to other large resettlement programs and in Section 9 we illustrate the applicability of our model beyond the refugee resettlement context. We conclude and offer suggestions for future work in Section 10. In the Appendix, we recap the entire running **Example** (A), provide all proofs (B), give a technical description of the Top Choice algorithm (C), and supply additional examples of our mechanisms at work (D).

2 Institutional context

In January 2014, the British government launched the Syrian Vulnerable Persons Resettlement (VPR) Programme alongside the broader UNHCR resettlement program in order to help 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 program and resettle 20,000 refugees in Britain by 2020. By June 2016, 2,800 Syrians had been resettled in Britain (UNHCR, 2016).

In the UK, the powers of refugee resettlement (like many other powers, including health-care and education) are devolved from Westminster (the 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).¹⁴

¹³In fact, a number of papers use optimization techniques, such as linear and integer programming, to find welfare-maximizing outcomes in many-to-one matching markets (Roth et al., 1993; Baiou and Balinski, 2000; Sethuraman et al., 2006; Featherstone, 2014; Ashlagi and Shi, 2015; Bodoh-Creed, 2016).

¹⁴These administrative local government units (that we refer to as *localities* following this section) go by 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 the purview of districts/boroughs. For the purposes of this paper, these distinctions are irrelevant as ultimately we are only concerned with the provision of a service in a particular area. From the perspective of residents, all levels of government cooperate seamlessly to deliver a full range of services. Scotland, Wales, and Northern Ireland have similar systems of local government.

LAs are responsible for the disbursement of social benefits, the 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.¹⁵ Meeting the government target, however, is likely to require participation of many more LAs.

Syrian refugees who come to Britain are granted a five-year leave to remain (after which they would be eligible to apply for permanent residence) and are given full access to public services. They are free to move to any part of the UK, but this is unlikely to happen until they can fully support themselves. 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.¹⁶ 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 cannot recoup its costs of resettlement.

As with other resettlement schemes, refugee families who have expressed a wish to be resettled are referred to the British authorities by the UNHCR.¹⁷ The British authorities then

¹⁵By April 2016, 71 LAs in England had accepted refugees, as have half of Scottish LAs. One third of all refugees on the Syrian VPR scheme came to Scotland (House of Commons Library, 2016).

¹⁶According 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 the evidence of what these costs actually are is still lacking. The recent government initiative for community sponsorship suggests that sponsors should plan to provide funding of £4,500 per adult per year which is supposed to cover “the initial provision of cash on arrival and to fund English language tuition and interpretation costs” (Home Office, 2016b).

¹⁷“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

review the files, including medical records, and carry out their own security and background checks as well as interviews. In the Syrian VPR Programme, the typical refugee family size is six and the Home Office endeavours to keep all families together via linked independent applications.¹⁸ Once the application is approved, the file is sent to a team that matches the family to suitable accommodation supplied by LAs. What counts as “suitable” accommodation might vary from one LA to another.¹⁹ Refugees have no direct say as to where in Britain they go (unless they have family in a particular LA), and LAs sometimes reject applicants. Once accommodation has been found and the 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 timeframe between application approval and arrival—during which the matching of a cohort of refugees to LAs happens—is typically between six weeks and three months.²⁰

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.²¹ There are also some technical obstacles: 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.

formal refugee camps, informal settlements and host communities.” (House of Commons Library, 2016)

¹⁸Around half of the applicants are under 18; about half are women (Home Office, 2016a).

¹⁹For example, some LAs allow teenage siblings of different sexes to share a room, others do not.

²⁰The 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.

²¹Surveys 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, 2016a). In July 2016, Theresa May, the incoming British Prime Minister, eliminated the Home Office “Minister for Syrian Refugees” position.

3 Model

3.1 Basic ingredients

There is a finite set of *refugee families* F . A family $f \in F$ has size $|f|$. There is finite set of *localities* L . We use \emptyset for the null object to represent being unmatched.

A *contract* is a family-locality pair $(f, \ell) \in X \equiv F \times (L \cup \{\emptyset\})$. We denote by $\mathbf{f}(x)$ and $\ell(x)$ as the family and locality associated with contract $x \in X$.

An *outcome* $Y \subseteq X$ is a set of contracts and \mathcal{X} is the set of all outcomes. For any $Y \subseteq X$, $\mathbf{F}(Y)$ and $\mathbf{L}(Y)$ denote the families and the localities whose contracts appear in outcome Y at least once. For any $\ell \in L$ and $Y \subseteq X$, $\mathbf{F}_\ell(Y)$ is the set of families associated with locality ℓ under Y . Finally, Y_f and Y_ℓ denote the contracts that family f and locality ℓ are associated to in outcome Y .

On top of the matching set-up just described, the refugee matching problem has *multidimensional constraints* on the set of outcomes: Refugee families require *multiple units of different services* (e.g., hospital beds, school seats, language support) from a set S . We denote by ν the matrix of family *service needs*, with typical element $\nu_s^f \in \mathbb{Z}_{\geq 0}$ denoting the total number of units of service s required by family f . A refugee family f can only live in locality ℓ if ℓ can provide services to meet f 's needs. We denote by κ the matrix of locality *service capacities*, with typical element $\kappa_s^\ell \in \mathbb{Z}_{\geq 0}$ denoting the number of units of service s locality ℓ can provide.²²

We denote by $\tau_s^\ell(Y) \equiv \sum_{f \in \mathbf{F}_\ell(Y)} \nu_s^f$ 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 $Y \subseteq X$ is *feasible* if (i) $|Y_f| = 1$ for all f , i.e. each family is either matched to one locality or is unmatched, and (ii) $\tau(Y) \leq \kappa$ i.e. the *feasibility constraint* is not violated. A set of families $F' \subseteq F$ can be *accommodated* under Y if $Y \cup \{(f, \ell)\}_{f \in F'}$ is feasible. Let \mathcal{Y} denote the set of all feasible outcomes.

Multidimensional constraints introduce a particular *complementarity* into the choices of localities, which is absent in many matching models. Suppose there is a locality ℓ with two units for a single service s ($\kappa_s^\ell = 2$). Families f_1 and f_3 need one unit of s each ($\nu_s^{f_1} = \nu_s^{f_3} = 1$), but family f_2 needs two units of the service ($\nu_s^{f_2} = 2$). Suppose that in this locality f_1 has the highest priority, followed by f_2 , and then f_3 . Then, we would then match $\{f_1, f_3\}$ when all three families $\{f_1, f_2, f_3\}$ apply to the locality since f_1 would be matched first but f_2 could not be accommodated, leaving a unit of the service for f_3 . When $\{f_2, f_3\}$ apply, only

²²As is standard, the null object has infinite capacity for every service. All results in the paper go through with minimal or no modification, if we assume that $\nu_s^f, \kappa_s^\ell \in \mathbb{R}_{\geq 0}$. As an example, we might wish to capture the fact that a refugee may require $\frac{1}{7}$ of a dialysis machine because she needs to use it once a week.

f_2 is matched to the locality as it takes up both units of the service.²³ Hence, because of the multidimensional constraint, ℓ views f_1 and f_3 as complementary since f_3 is *less* likely to be matched to ℓ if f_1 does not apply. This kind of complementarity is precisely what prevents us from using classic tools in matching theory and we return to the problems this creates in Section 6.²⁴

3.1.1 Relationship to prior models

Our model generalizes a number of existing matching models, including the following:

- School choice (Abdulkadiroğlu and Sönmez, 2003): Every student takes up a single seat at any school. Let us relabel a student as a family and a school as a locality. In our model, this corresponds to having only one service ($|S| = 1$) and any family needing exactly one unit of the service ($\nu_s^f = 1$ for all $f \in F$).
- Controlled school choice or college admissions with affirmative action and m type-specific quotas (Abdulkadiroğlu and Sönmez, 2003; Abdulkadiroğlu, 2005; Westkamp, 2013): Each student is one of m types and each school has a quota for each of the m types. Let us again relabel a student as a family, a school as a locality and a type as a service. In our model, this corresponds to having m services in each locality ($|S| = m$). Each family needs exactly one unit of one of the services (ν_s^f are m -dimensional unit vectors for every $f \in F$).
- School choice with majority quotas (Kojima, 2012; Hafalir et al., 2013): Each student is either a majority or a minority student. Each school has a overall cap on the number of students, which includes a cap for majority students. Let us again relabel a majority/minority student as a majority/minority family and a school as a locality. Let us also relabel “any student seats” as service s_1 and “majority student seats” as service s_2 ($|S| = 2$). In our model, then the capacity of any locality for s_1 is greater than the capacity for s_2 ($\kappa_{s_1}^\ell > \kappa_{s_2}^\ell$ for all $\ell \in L$). A majority family f needs a unit of both services ($\nu^f = (1, 1)$) whereas a minority family f' only needs a unit of s_1 ($\nu^{f'} = (1, 0)$).
- Hungarian college admissions (Biró et al., 2010): Students take up a college seat as well as a faculty seat. Both colleges and faculties have their own capacities. Let us relabel

²³Note that the example would be analogous if there were two services s_1 and s_2 , the locality had one unit of each service available, families f_1 and f_3 required one unit of s_1 and s_2 respectively, while family f_2 required one unit of each service.

²⁴From the point of view of localities, refugee families are neither “weak substitutes” (Hatfield and Kojima, 2008), nor even “substitutes and symmetric complements” (Alva, 2015).

a student as a family and a college as a locality. Let us also relabel “college capacity” as the capacity of the locality for service s_1 ($\kappa_{s_1}^\ell$). Let us relabel the faculties as the remaining services $S \setminus \{s_1\}$. Therefore, each family has needs $\nu^f = (1, 0, 0, \dots, 1, \dots, 0, 0)$ where the second “1” is the need for a unit of one service $s \in S \setminus \{s_1\}$.

- Allocation of trainee teachers to schools in Slovakia and Czechia (Cechlárová et al., 2015): Teachers are required to teach two out of three subjects and each school has a capacity for all three subjects. Let us relabel a teacher as a family, a school as a locality, and a subject as a service. In our model, this corresponds to having three services ($|S| = 3$) and any family having needs $\nu_s^f \in \{0, 1\}$ for any two different s .
- College admission with multidimensional privileges in Brazil (Aygün and Bó, 2016): Students can claim any combination of three privileges. Colleges have quotas for each privilege, but a single student can claim more than one privilege. Let us relabel a student as a family, a college as a locality, and a privilege as a service. In our model, this corresponds to having three services ($|S| = 3$) and any family having needs $\nu_s^f \in \{0, 1\}$.
- Object allocation (Nguyen et al., 2016) or course assignment (Budish, 2011): Agents (students) demand a certain number of different objects (courses) that are supplied by a single seller (a business school). Let us relabel agents (students) as families, different objects (courses) as services, and the single seller as a single locality. In our model, this corresponds to having only one locality ($|L| = 1$).
- Resident-hospital matching with sizes (McDermid and Manlove, 2010): Doctors apply to hospitals, but the doctors can take up more than one seat at a hospital, e.g. because they arrive as couples. Let us relabel doctors as families and hospitals as localities. In our model, this corresponds to having one service ($|S| = 1$) and families having a need of arbitrary size for this service.²⁵

Most of the models described above use further assumptions and develop solution approaches that suit their particular contexts but differ substantially from ours. Nevertheless, as we note throughout the paper, several impossibility and complexity results established in these papers will apply immediately to our framework.

²⁵This model in turn generalizes the resident-hospital matching with inseparable couples (i.e. when couples have the same preference list and prefer to be unmatched to being in different hospitals) as well as resident-hospital matching with couples which have “consistent” preferences (McDermid and Manlove, 2010, Lemma 2.1). In both cases, we set $|S| = 1$ and $\nu_s^f \in \{1, 2\}$ in our model.

3.2 Housing

One of the most important concerns in refugee resettlement is finding appropriate housing for refugee families. Refugee families do not get a choice over specific housing units. Housing is extremely heterogeneous and different localities can impose different constraints on which family can be housed where. That is, some housing units may be *impermissible* for some families. In general, we say that there is set of *houses* H . Each house belongs to a locality $\ell \in L$. The set of houses in locality ℓ is denoted H_ℓ .

3.2.1 Housing as service constraints

In some cases, we can express housing as a subset of services and capture heterogeneous housing with the multidimensional service constraints. In Appendix A, we show how to incorporate different housing features and sizes directly into service constraints.

Definition 1. Housing is *reducible* if housing can be incorporated into the multidimensional service constraints given a fixed number of housing features and sizes.

For one simple case of reducible housing, suppose that houses differ only by size and a family needs a house of a certain size but can be accommodated in any house that is larger. Denote $s^{h_i} \in S^h$ the service representing housing size $i \in \{1, \dots, M\}$. Let $(\kappa_{s^{h_i}}^\ell)_{i \in \{1, \dots, M\}}$ and $(\nu_{s^{h_i}}^f)_{i \in \{1, \dots, M\}}$ denote the subvectors of $(\kappa_s^\ell)_{s \in S}$ and $(\nu_s^f)_{s \in S}$ respectively representing housing constraints. We then denote $\kappa_{s^{h_i}}^\ell$ as the number of houses of size *at least* i in locality ℓ . On the side of refugee needs, if a family requires a house of size i , then denote $\nu_{s^{h_j}}^f = 1$ for all $j \leq i$.

3.2.2 General permissible housing constraints

Some housing requirements cannot be captured by multidimensional service constraints with a fixed number of housing features and sizes. For example, in addition to minimum house size regulation there are often also maximum size regulations: In the UK, for example, refugee families may be denied housing benefit if their house has too many unused bedrooms. Consider all possible family-house pairs $F \times (H \cup \{\emptyset\})$ and

$$\mathcal{D} := \{D \in 2^{F \times (H \cup \{\emptyset\})} \mid |D \cap \{(f, h)\}| = 1 \forall f \in F \text{ and } |D \cap \{(f, h)\}| \leq 1, h \in H \cup \{\emptyset\}\}$$

which is the set of all housing assignments D i.e., sets of pairs such that there is exactly one pair per family and at most one pair per house. Denote the housing assignment function $A : \mathcal{X} \mapsto \mathcal{D}$ such that $A((f, \ell)) \mapsto (f, h)$ where $h \in H_\ell$ for all $f \in F$ and $\ell \in L$ and

$A((f, \emptyset)) = (f, \emptyset)$ for all $f \in F$. That is the housing assignment function maps each family to a house in the locality of its contract (unmatched families are not housed anywhere). We say that a family-house pair is *permissible* if house meets the legal requirements to accommodate the associated family. We denote the set of housing assignments that only contain permissible pairs as $\mathcal{D}^* \subseteq \mathcal{D}$. Hence, in the case when housing is not reducible, an outcome Y is *feasible* if $Y \subseteq X$, $|Y_f| = 1$ for all $f \in F$, $\tau(Y) \leq \kappa$, and there is $A \in \mathcal{A}$ such that $A(Y) \in \mathcal{D}^*$.

Most of our results do not rely on reducible housing and we explicitly point out when they do.

Example

To be able to compare different mechanisms in this paper, we now introduce a single running Example which we will return to throughout paper. The Example has 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 capacities: Service needs:

$$\kappa = \begin{matrix} & & s_1 & s_2 \\ \ell_1 & \begin{pmatrix} 4 & 2 \end{pmatrix} \\ \ell_2 & \begin{pmatrix} 3 & 2 \end{pmatrix} \\ \ell_3 & \begin{pmatrix} 2 & 2 \end{pmatrix} \\ \ell_4 & \begin{pmatrix} 2 & 2 \end{pmatrix} \end{matrix} \qquad \nu = \begin{matrix} & s_1 & s_2 \\ f_1 & \begin{pmatrix} 1 & 0 \end{pmatrix} \\ f_2 & \begin{pmatrix} 2 & 1 \end{pmatrix} \\ f_3 & \begin{pmatrix} 0 & 2 \end{pmatrix} \\ f_4 & \begin{pmatrix} 1 & 1 \end{pmatrix} \\ f_5 & \begin{pmatrix} 3 & 0 \end{pmatrix} \end{matrix}$$

In the Example, ℓ_4 cannot accommodate f_5 in any outcome since $\nu_{s_1}^{f_5} = 3 > 2 = \kappa_{s_1}^{\ell_4}$. On the other hand, consider an outcome Y , in which f_1 and f_2 are matched to ℓ_1 and f_3 and f_4 are matched to ℓ_2 and ℓ_4 respectively. The corresponding matrix of service demands for services at localities would be as follows:

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

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

Let us introduce some housing constraints in the Example. We will 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 impermissible family-house pairs. In Appendix A, we show that housing is reducible if there are three housing features and two sizes. However, this housing constraint cannot be represented as a multidimensional service constraint with a single house size and a single feature.

We illustrate the Example in Figure 1 below. For each family service needs are represented the number of solid blocks (first for s_1 and then for s_2). Service provision for each locality is represented with empty blocks. We color code houses according to which family-house pairs are permissible.

4 Welfare maximization without preferences or priorities

In this section, we consider the case in which the preferences of refugees and the priorities of localities are not elicited explicitly, as is often the case in rapid responses to unfolding humanitarian catastrophes or indeed in many refugee resettlement schemes around the world.²⁶ 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.²⁷ Instead, the social planner could estimate the quality of matches based on observable data and on past experience.

We summarize the estimated quality of each refugee-locality match as a single number called the *quality score*, $q : X \rightarrow \mathbb{R}_{\geq 0}$ (normalizing $q((f, \emptyset)) = 0$ for all $f \in F$). In order to maximize the overall observed efficiency of the match within the feasibility constraints, the social planner solves the following *outcome-quality maximization problem* (OQMP):

²⁶For example, during the Kosovo airlift in 1999 many Kosovar-Albanian refugees boarded rescue planes without knowing where they were going (Wells, 1999).

²⁷Even in developed countries that are accustomed to refugee resettlement, this infrastructure is often lacking.

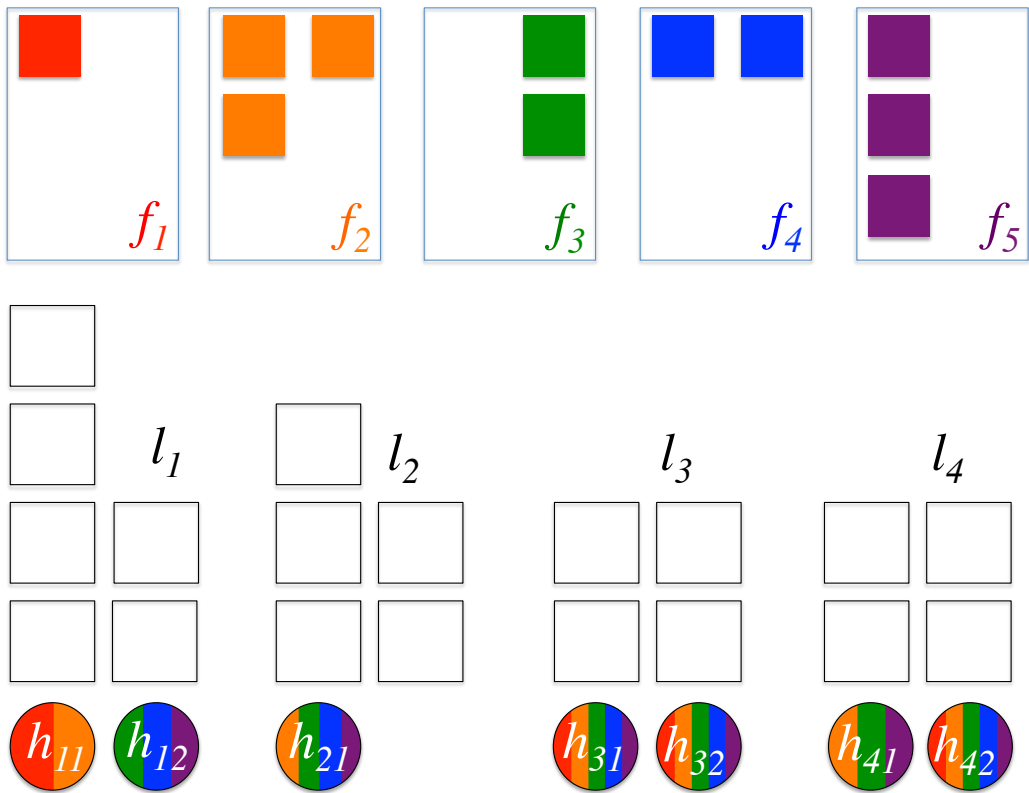


Figure 1: Set-up for the running Example

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

Two special cases of q are worth emphasizing: Setting $q = 1$ (or any other constant) would maximize the total number of families that are resettled, while setting $q((f, \ell)) = |f|$, 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 more general. For example, the social planner could use survey data on satisfaction with localities as well as on lifetime outcomes (such as the probability of employment within a given timeframe) in order to determine which families tend to fare best in which locality.

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

$$\max \sum_{f \in F} \sum_{\ell \in L} q((f, \ell)) \iota(f, \ell) \quad \text{subject to:} \quad (2)$$

$$\begin{aligned} \sum_{f \in F} \sum_{\ell \in L} \nu_s^f \iota(f, \ell) &\leq \kappa_s^\ell && \forall \ell, s \\ \sum_{\ell \in L} \iota(f, \ell) &\leq 1 && \forall f \\ \iota(f, \ell) &\in \{0, 1\} && \forall f, \ell \end{aligned}$$

Problem (2) is an example of a *0-1 multiple multidimensional knapsack problem* (Song et al., 2008).

Proposition 1. *Suppose that housing is reducible. Then the outcome-quality maximization problem (1) is equivalent to the 0-1 multiple multidimensional knapsack problem (2) and can be solved exactly using the “branch-and-bound” method.*

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.²⁸ Different objects yield different profits in each knapsack and the objective is to maximize the total profit, subject to not exceeding the capacity constraints

²⁸This 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 even from the multiple-choice knapsack problems in which exactly one object from each of the many mutually exclusive classes must be used (Sbihi, 2007).

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. In our proof of Proposition 1, we extend their approach by establishing upper bound and lower bounds of the maximand, which are the key inputs in the “branch-and-bound” method.

Example for Section 4

In addition to the set up of the [Example](#), let us introduce the following quality score q over the set of feasible contracts X :

	$q((f, \ell))$			
	ℓ_1	ℓ_2	ℓ_3	ℓ_4
f_1	71	\emptyset	23	38
f_2	46	49	30	91
f_3	52	68	43	20
f_4	4	75	96	36
f_5	92	41	\emptyset	\emptyset

Then, the solution to the outcome-quality maximization problem is:

$$Y^{OQMP} = \{(f_1, \ell_1), (f_2, \ell_4), (f_3, \ell_2), (f_4, \ell_3), (f_5, \ell_1)\}$$

since this outcome gives each family the locality with the highest quality score and is feasible. Note that since f_1 and f_5 are both allocated to ℓ_1 but family-house pair (f_1, h_{12}) is impermissible, it must be case that f_1 is assigned to h_{11} and f_5 is assigned to h_{12} .

5 Accounting for preferences of refugee families

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.²⁹ 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 only refugees themselves know in what kinds of areas they are most likely to thrive. But knowing preferences can also help

²⁹In Britain, some newspapers celebrated the arrival of fifteen Syrian refugee families to the remote Scottish Isle of Bute ([McKenna, 2015](#)).

prevent *internal migration*—the movement of refugees away from their assigned localities soon after arrival—which localities want to avoid because they make substantial upfront investments in hosting refugees.

Eliciting preferences of refugees over localities is difficult task. In school choice, for example, parents can be reasonably expected to rank their top twelve schools. In the case of refugee matching, not only are there many localities in most countries, but refugees often lack the necessary information to make decisions in their own interest. In this case, it is reasonable to ask refugees to rank the *properties* of areas that are important to them (e.g. proximity to a city, low crime, presence of a co-ethnic or a co-religious community etc.). The resettlement authority can then use locality-level data and refugees’ rankings to infer a likely preference profile over the localities. In order to ensure that this inference is as accurate as possible, the authority will need to give refugee families clear incentives to submit truthful reports over area properties.

When locality participation is (relatively) secured, the priorities of localities are not very important; 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 non-manipulable. Because both refugee preferences and feasibility constraints act at the locality level, our mechanism must identify and eliminate potential Pareto-improving locality trades while paying attention to feasibility constraints.

We retain the quality score defined in the previous 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 $x = x'$). 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 and \succ the set of all preference profiles.³⁰ Given any outcome $Y \subseteq X$, we will say that $\ell \in L \cup \{\emptyset\}$ is family f ’s *top-choice locality* (or simply f ’s *top choice*) if $(f, \ell) \in Y$ and for all $\ell' \neq \ell$ such that $\ell' \in \mathbf{L}(Y)$ we have that $\ell \succ_f \ell'$. We will denote f ’s top choice in Y as $\bar{\ell}_f(Y)$. We say that f ’s top choice is *feasible* under Y if f can be accommodated in its top choice under Y and that f is *permanently matched* under Y if $|Y_f| = 1$.

An outcome Y is *Pareto-efficient* if it is feasible and there is no feasible outcome Y' such that for any $f \in F$, either $\ell(Y'_f) = \ell(Y_f)$ or $\ell(Y'_f) \succ_f \ell(Y_f)$. In other words, if one refugee family is matched to a more preferred locality than in a Pareto-efficient outcome, then another family must be worse off. A (direct) *mechanism* is a function $\varphi : \succ \rightarrow \mathcal{X}$.

³⁰We assume without loss of generality that every family’s least preferred option is being unmatched.

A mechanism φ is *strategy-proof* if for any $f \in F$ there does not exist a report of a preference list \succ'_f such that

$$\ell(\varphi((\succ'_f, \succ_{-f}))_f) \succ_f \ell(\varphi(\succ)_f).$$

Strategy-proofness requires that refugee families cannot make themselves better off by misreporting their preferences over localities.

5.1 Using only preferences of refugee families

Our first mechanism, described in Algorithm 1, is an extension of the Top Trading Cycles mechanism to matching with multidimensional constraints.

The Multidimensional Top Trading Cycles (MTTC) algorithm works in much the same way as the classical Top Trading Cycles (TTC) algorithm: in each round, each family points at its most preferred locality that can accommodate it and each locality points at the highest-quality family that it can accommodate. There must be at least one cycle which is eliminated by matching families to the localities they pointed at and adjusting the service capacities of each localities by the service needs of the family that has just been matched to it. The main difference between the MTTC algorithm and the TTC algorithm is that even though the family is pointing at a locality it needs to be assigned to a house when it is part of the cycle. This is not trivial when housing is not reducible. Therefore, the pointing family and the pointing locality are solving independent housing assignment problems. Hence, even though families are permanently matched to localities in each round, they are only tentatively assigned to housing until the last period. Indeed, in contrast to the TTC algorithm, in the MTTC algorithm, the locality might stop pointing at a family even if it still has capacity and the family is still unmatched because the housing constraints mean that it is no longer possible to accommodate the family in the locality.

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

Example for Section 5.1

In order to illustrate the MTTC algorithm, in addition to the set-up of the [Example](#) and the quality score in the [Example for Section 4](#), we will need to know the preferences of refugee families:

$$f_1 : l_3 \succ l_4 \succ l_1 \succ l_2.$$

Algorithm 1: MULTIDIMENSIONAL TOP TRADING CYCLES (MTTC) ALGORITHM

Start with the set of all contracts X . For each locality, rank the families some order of priority (e.g. using the quality score). Remove all infeasible contracts under the empty outcome to obtain outcome Y^1 .

Round $i \geq 1$:

Under outcome Y^i , every family f that is not permanently matched points at its top choice (see Proof of Proposition 2 for the procedure).

For each locality ℓ , consider all families that have a contract with ℓ at Y but are not permanently matched. If that set is nonempty, ℓ points at the family in that set that has the highest quality score for ℓ (see Proof of Proposition 2 for the procedure). Otherwise ℓ does not point.

At least one cycle appears. Any family and any locality is in at most one cycle. For every family involved in a cycle, remove all its contracts that do not involve its top choice. These families are now permanently matched.

Given the outcome obtained, remove all contracts (f, ℓ) such that ℓ cannot accommodate f alongside all other families that are permanently matched to it. If at least one family becomes permanently matched as a result of this step, repeat this step.

Let Y^{i+1} be the outcome obtained after the previous step was repeated for the last time. If all families are permanently matched under Y^{i+1} , the algorithm ends and generates Y^{i+1} . Otherwise proceed to Round $i + 1$.

$$f_2 : \ell_3 \succ \ell_1 \succ \ell_2 \succ \ell_4.$$

$$f_3 : \ell_4 \succ \ell_3 \succ \ell_1 \succ \ell_2.$$

$$f_4 : \ell_4 \succ \ell_1 \succ \ell_2 \succ \ell_3.$$

$$f_5 : \ell_1 \succ \ell_2 \succ \ell_4 \succ \ell_3.$$

In the Example, the MTTC algorithm lasts two rounds.

Round 1: In the first round, all families point at their top choice. Families f_1 and f_2 point at ℓ_3 , f_3 and f_4 at ℓ_4 , and f_5 at ℓ_1 . Locality ℓ_1 in turns points at f_5 , ℓ_2 and ℓ_3 point at f_4 , and ℓ_4 points at f_2 . A cycle appears between f_5 and ℓ_1 (resulting in f_5 being assigned to h_{12} since h_{11} is not permissible for f_5). Another cycle appears between f_2 , ℓ_3 , f_4 and ℓ_4 .

Round 2: Only two families are not permanently matched in the second round. The relevant quality scores are:

	$q((f, \ell))$			
	ℓ_1	ℓ_2	ℓ_3	ℓ_4
f_1	71	\emptyset	8	\emptyset
f_3	\emptyset	68	31	7

Family 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_{41}) is an impermissible family-house pair. Family f_1 then points at its third choice, ℓ_1 , this is possible since family-house pair (f_1, h_{11}) is feasible and f_1 and f_5 can be accommodated together at ℓ_1 . Family f_3 can point at neither ℓ_4 , ℓ_3 nor ℓ_1 because it cannot be accommodated alongside f_4 , f_2 or f_5 , respectively. Consequently, f_3 points at ℓ_2 . Locality ℓ_1 already accommodates f_5 in house h_{12} and only h_{11} remains available. Note that ℓ_1 's current highest-quality family f_3 is not available because family-house (f_3, h_{11}) is impermissible. At this point ℓ_1 needs to work through its preference list and check the highest-quality family that would solve the (admittedly trivial) housing assignment problem in the locality. Family f_1 solves the problem, so ℓ_1 points at f_1 . Family-house pair (f_3, h_{21}) is impermissible but (f_1, h_{21}) is permissible (note the role of the housing assignment problem again), consequently ℓ_2 points at f_3 . Localities ℓ_3 and ℓ_4 do not point at any family since they can no longer accommodate either f_1 or 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 point at each other. Family f_1 is matched to ℓ_1 (and assigned house h_{11}), f_3 is matched to ℓ_2 (and assigned house h_{21}), and the algorithm terminates.

The resulting outcome is

$$Y^{MTTC} = \{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2), (f_4, \ell_4), (f_5, \ell_1)\}.$$

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

5.2 Improving outcomes of refugee families from an initial allocation

A resettlement agency might be reasonably apprehensive about relying solely on refugees' preferences in the allocation process. Resettlement agencies often have specific goals in mind, such as the likelihood of early employment or avoidance of segregation, which might conflict with the refugees' preferences over localities. Therefore, it can be important to take into account the goals of the resettlement agency alongside the preferences of refugees. We propose to find preference-based improvements from an exogenous tentative initial allocation favored by the social planner. The natural initial allocation would be the one produced by the outcome-quality maximization problem.

Consider a feasible outcome Y^E , which we will refer to as the *endowment*. We say that outcome Y is *individually rational* if for all $f \in F$ either $\ell(Y_f) = \ell(Y_f^E)$ or $\ell(Y_f) \succ_f \ell(Y_f^E)$. In the case of school choice, starting from any endowment and running the TTC algorithms (in which any non-empty school points at its most preferred student), produces an individually rational and Pareto-efficient outcome in a strategy-proof way. The reason is that any that Pareto improvement from the endowment can be broken down into simple cycles: If two students from one school want to swap with two students from another school, we can first swap one pair of students and then the other. Hence, a student only needs to point at its most preferred school, which gives an opportunity for the school's most preferred student to make a choice.

In our context, simple Pareto-improving cycles will not achieve Pareto efficiency because families, which have needs for multiple units of different services, may have to swap in groups in order to find Pareto improvements. To illustrate this with our running [Example](#), let us start with an endowment

$$Y^E = \{(f_1, \ell_1), (f_2, \ell_4), (f_3, \ell_1), (f_4, \ell_4), (f_5, \ell_2)\}$$

and assume moreover that f_4 can no longer be moved. In that case, there is a Pareto improvement if f_1 and f_3 swap with f_2 . But this Pareto improvement cannot be achieved by a pairwise swap or a cycle.

Finding all Pareto-improving exchanges among sets of families in general—even between

two localities—would mean potentially looking at all subsets of the families in these localities and is therefore computationally intractable. On the other hand, because of feasibility constraints, if all families simultaneously point at their top-choice locality as in the MTTC algorithm, a feasible cycle is not guaranteed to exist.

We now introduce the Serial Multidimensional Top Trading Cycles (SMTTC) algorithm that finds some of the possible Pareto improvements in a way that preserves individual rationality and strategy-proofness. Unlike the TTC algorithm, Hierarchical Exchange rules (Pápai, 2000), or more general Trading Cycle rules (Pycia and Ünver, 2016), the SMTTC algorithm is run sequentially—one cycle at a time. We do this in order to ensure that whenever a cycle appears (at the end of every round of the algorithm), it is feasible and any family and locality will be part of at most one cycle. First, we determine *active* and *available* localities. Any active locality participating in a possible Pareto-improving cycle has one family that can feasibly swap places with other families from all other active localities. Any available locality has enough capacity to accommodate any family from all active localities.³¹ All other localities are inactive. Every active locality points at its highest-quality family—an *active family*—participating in the possible swap. Then all active families point at their most preferred locality. Whenever cycles appear, we match families to localities they pointed at. If there is no cycle, this is because some family has pointed at an inactive locality. In that case, families are picked one at a time and their contracts involving inactive localities are removed until a cycle appears. The reason why a cycle eventually appears is that all families that have been picked point at an active locality. Once a cycle is found, families are matched to the localities they pointed at. Crucially, the remaining families and localities continue being active and pointing in the next round. Our family and locality selection rule is precisely what ensures that any cycle that appears is feasible and non-manipulable, but this rule is also restrictive and might not find all Pareto-improving swaps of sets of families. When the endowment is empty, the SMTTC algorithm reduces to the MTTC algorithm. The SMTTC algorithm is described in Algorithm 2.

Proposition 3. *The Serial MTTC algorithm is strategy-proof and yields an individually rational outcome from any endowment.*

If the Serial MTTC algorithm finds any cycles between families and localities that these families are not endowed with, its outcome Pareto-dominates the endowment.

³¹If a locality is both active and available, we label it active. Labelling it available would have no bearing on the properties of the mechanism.

Algorithm 2: SERIAL MULTIDIMENSIONAL TOP TRADING CYCLES ALGORITHM

Arbitrarily index localities such that $L \equiv \{\ell_1, \dots, \ell_{|L|}\}$. For each locality, rank the families some order of priority (e.g. using the quality score). Start with the set of all contracts X . Remove all contracts which are infeasible under the empty outcome. Consider the current outcome Y^i .

Round $i \geq 1$:

If all families endowed with a locality are permanently matched, label all families that are not yet permanently matched and all localities that can accommodate at least one of these families alongside the families that are permanently matched to it as *active*. Every active family points at its top choice and every active locality points at the active family with the highest priority among those it can accommodate. Proceed to Step $|L| + 2$.

Else label all families that were active in the previous round as *active* and proceed to Step 1.

Steps $1 \leq j \leq |L|$

For locality ℓ_j :

If there exists an active family endowed with ℓ_j , then let $f_{\ell_j}^*$ be that family. Label ℓ_j as *active*.

Else if ℓ_j can accommodate any active family alongside those with which it is permanently matched or for which it is the endowment, then label ℓ_j *available*.

Else if there exists a family whose endowment is ℓ_j , is not permanently matched and can swap with all active families, let $f_{\ell_j}^*$ be the one with the highest priority for ℓ_j . Label ℓ_j and $f_{\ell_j}^*$ as *active*.

Else label ℓ_j as *inactive*.

Step $|L| + 1$: Every active family $f_{\ell_j}^*$ points at its top choice. Every active locality ℓ_j points at $f_{\ell_j}^*$. Every available locality points at the active family with the highest priority for it. Inactive families and localities do not point.

If a cycle exists, proceed to Step $|L| + 2$.

Else arbitrarily pick one family and remove its contracts involving inactive localities. Return to the start of Step $|L| + 1$.

Step $|L| + 2$: For each family involved in a cycle, remove all contracts involving that family and a locality that is not its top choice.

Step $|L| + 3$: Remove all contracts (f, ℓ) such that f cannot be accommodated at ℓ alongside all families permanently matched to ℓ . If any family has become permanently matched as a result of this step, repeat this step. Otherwise continue to Step $|L| + 4$.

Step $|L| + 4$: Let Y^{i+1} be the updated outcome. If all families are permanently matched, the algorithm ends and generates Y^{i+1} . Otherwise proceed to Round $i + 1$.

Example for Section 5.2

We will use family preferences from [Example for Section 5.1](#). Let us start with an endowment

$$Y^E = \{(f_1, \ell_1), (f_2, \ell_4), (f_3, \ell_2), (f_4, \ell_3), (f_5, \ell_1)\}$$

which is outcome the of outcome-quality maximization problem.

We keep the same index of families as in the Example so that ℓ_1 is considered first, ℓ_2 second, ℓ_3 third, and ℓ_4 last.

Round 1: We consider ℓ_1 and make f_5 active since it has a higher quality score than f_1 . At ℓ_2 , f_3 is made active since f_5 and f_3 can feasibly exchange their localities. As f_5 can neither be accommodated at ℓ_3 nor ℓ_4 , these two localities are inactive in that round. Localities ℓ_1 and ℓ_2 point at f_5 and f_3 respectively while ℓ_3 and ℓ_4 do not point. Families f_5 and f_3 point at their respective top choices: ℓ_1 and ℓ_4 . A cycle occurs between f_5 and ℓ_1 . Family f_5 is permanently matched to ℓ_1 .

Round 2: Family f_3 is immediately made active since it was active in Round 1 but did not get permanently matched. At ℓ_1 , f_1 is the only candidate left since f_5 has been permanently matched, however f_1 cannot be accommodated at ℓ_2 as (f_1, h_{21}) is not a permissible family-house pair, and as a result ℓ_1 remains inactive in this round. In contrast, f_4 and f_2 are made active since they can both be accommodated at ℓ_2 and f_3 can be accommodated at both ℓ_3 and ℓ_4 . Locality ℓ_1 does not point, ℓ_2 points at f_3 , ℓ_3 points at f_4 and ℓ_4 points at f_2 . All three families can be accommodated at their respective first preferences, therefore f_3 and f_4 point at ℓ_4 and f_2 points at ℓ_3 . A cycle occurs between f_2 , ℓ_3 , f_4 and ℓ_4 . Families f_2 and f_4 are permanently matched to ℓ_3 and ℓ_4 respectively.

Round 3: Family f_3 is again made active immediately so that ℓ_1 remains inactive. As f_4 and f_2 have been permanently matched, ℓ_3 and ℓ_4 are also inactive. A simple cycle occurs between f_3 and ℓ_4 so f_3 is permanently matched to ℓ_4 .

Round 4: All families except f_1 have been permanently matched, as a result the latter is the only one to be active. Family f_1 and locality ℓ_1 point at one another and the algorithm ends.

The outcome of the Serial MTTC algorithm with endowment from the OQMP is the same as the outcome of the MTTC algorithm as it allows f_2 and f_4 to exchange their localities:

$$Y^{SMTTC} = \{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2), (f_4, \ell_4), (f_5, \ell_1)\}.$$

6 Accounting for the preferences of refugee families and the priorities of localities

As a refugee crisis goes on, it typically becomes more difficult to obtain goodwill from localities.³² Beyond direct cash transfers and coercion, the social planner has a market design tool that can increase localities' willingness to participate: explicitly incorporating localities' priorities over refugees. Eliciting priorities over refugee families 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. Moreover, satisfying priorities serves as an important fairness criterion in the resettlement process.

In this section, we offer mechanisms that respect priorities of localities in addition to the preferences of refugee families.³³

6.1 Stable outcomes

We let π_ℓ be the strict ordinal priority list of locality ℓ over families F and let π be the ordinal priority profile of the localities.³⁴ 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 $\ell \succ_f \ell(Y_f)$ and there exists a feasible outcome Y' such that $\ell(Y'_f) = \ell$ and if $\ell(Y_{f'}) = \ell$ then $\ell(Y'_{f'}) = \ell$ for all $f' \in \hat{F}_{(f,\ell)}$. Y is *stable* if it is feasible and does not allow any blocking pairs.

In words, family f and locality ℓ form a blocking pair if f prefers ℓ to its current match and it is possible to accommodate f in ℓ while ensuring that families that have a higher priority in ℓ than f can remain in the same locality. Therefore, in a stable outcome there is no family f that prefers another locality ℓ in which it can be accommodated alongside other families in ℓ that have a higher priority than f in ℓ . This stability concept is an extension of “elimination of justified envy” used in the school choice to our case with multidimensional constraints (Abdulkadiroğlu and Sönmez, 2003).

While in school choice models stable outcomes always exist, in a model with multidimensional constraints they do not. The reason is that the complementarity over families from the point of view of localities. Let us return briefly to the example of complementarity that

³²Some of the British media, for example, soon reacted against the Isle of Bute refugees (Reid, 2015).

³³It is straightforward to incorporate locality priorities while ignoring the preferences of the refugees by reversing the direction of matching (locality is matched with its highest-quality family) in the MTTC algorithm.

³⁴These priorities could, of course, come directly from the quality score.

we described immediately after our model in Section 3.1 and add another locality ℓ' with a single unit of the service. Assume moreover that f_3 has higher priority than f_1 in ℓ' and the following preferences

$$f_1 : \ell' \succ \ell \succ \emptyset$$

$$f_2 : \ell \succ \emptyset$$

$$f_3 : \ell \succ \ell' \succ \emptyset.$$

In this example, there is no stable outcome precisely because of the complementarity between f_1 and f_3 .³⁵

In fact, determining whether a stable outcome exists is a computationally intractable problem (McDermid and Manlove, 2010).³⁶ This means that the running time of an algorithm that guarantees to find a stable outcome or proves that none exists will increase exponentially with the problem size. This can be an impediment to practical applications in large matching markets.

A stable outcome $Y \subseteq X$ is *(Pareto-)undominated* if there does not exist any stable outcome $Y' \subseteq X$ such that either $\ell(Y'_f) = \ell(Y_f)$ or $\ell(Y'_f) \succ_f \ell(Y_f)$ for all $f \in F$. That is, a stable outcome is undominated if there is no other stable outcome in which some families stay in their existing localities but others are matched to localities they prefer. A stable outcome $Y \subseteq X$ is *family-optimal* if there does not exist another stable outcome $Y' \subseteq X$ such that $\ell(Y'_f) \succ_f \ell(Y_f)$ for any $f \in F$. A family-optimal stable outcome is unanimously preferred to any other stable outcome by all families. Note that, similarly to school choice, we only consider welfare from the point of view of refugee families and treat slots in localities as objects and locality priorities as non-strategic decisions.

6.1.1 Identical priorities

There is one case in which stable outcomes can be found straightforwardly: When the priorities of all localities are identical. This is common: For example, localities could agree that refugees families who have spent longer in the the refugee camps or those who are in urgent medical need should have a higher priority. In this case, every stable outcome coincides with an outcome from a simple mechanism—the *serial dictatorship*: first, the top-ranked family is permanently matched to its top-choice locality that can accommodate it,

³⁵This instability is a consequence of our focus on local refugee matching markets which are thin and heterogeneous. If we assumed a continuum of families and localities, our set-up would admit a stable outcome (Azevedo and Hatfield, 2015, Theorem 1).

³⁶The decision problem of the existence of stable outcomes in our setting is NP-complete, meaning there is no known efficient (e.g. with polynomial running time in the number of families or localities) method of solving it. Finding a stable outcome, like solving the multiple multidimensional knapsack problem, is therefore NP-hard i.e. as hard as the hardest computational problems. Verifying whether a particular outcome is stable only involves checking all possible blocking pairs which is a simple computational problem.

then the second-ranked family is matched to its top-choice locality given that the top-ranked family has already been permanently matched, and so on.

Proposition 4. *If priorities of localities are identical, then the outcome of the serial dictatorship is the unique stable outcome.*

Since the MTTC algorithm collapses to the serial dictatorship when priorities are identical, the serial dictatorship is also strategy-proof and the outcome it produces is Pareto-efficient.

6.1.2 Finding stable outcomes in general

We now informally describe the Top Choice algorithm, which finds a stable outcome, if one exists, that is undominated from the refugees families' perspective, and reports that the set of stable outcomes is empty otherwise. This algorithm does not run in polynomial time in general, however, many instances of the refugee resettlement problem, especially in small resettlement schemes like the UK's, comprise a few dozen families and a couple of dozen localities for one cohort, so even computationally slow algorithms can perform well in practice. After we discuss the mechanism, we comment on its strategic properties. The Top Choice algorithm runs in three phases. The technical descriptions of each phase of the algorithm and an application of this algorithm to our running [Example](#) can be found in [Appendix C](#).

Phase 1: Top-Down Bottom-Up (TDBU) algorithm

The Top-Down Bottom-Up (TDBU) algorithm is run exactly once at the beginning of the Top Choice algorithm. TDBU algorithm dramatically reduces the search space for a solution used in the subsequent phase. The TDBU algorithm cycles through localities and identifies contracts that cannot be part of *any* stable outcome. 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 matched in that locality. This allows us to eliminate all contracts between that family and any localities it prefers 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 between this family and 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 $\phi(X) \subseteq X$ which could form *all* stable outcomes (although they may not exist). That is, if a stable outcome exists then any contract in that outcome is in $\phi(X)$.

Let us attempt to construct an outcome by matching each family to their top-choice locality from the remaining contracts $\phi(X)$. If this outcome $\phi^*(X)$ is feasible, then we have found the unique family-optimal stable outcome, and the Top Choice algorithm stops.

Phase 2: Depth-First Search for a stable undominated outcome

If $\phi^*(X)$ is not feasible, we continue to this phase.

We begin with a description of the first steps of the Depth-First Search (DFS) for a stable undominated outcome and then explain how its intermediate steps—the Augmented TDBU (ATDBU) algorithm—work. We order the families, labelling them $f_1, f_2, \dots, f_{|F|}$. This can be done arbitrarily or reflecting a *general priority order*, for example due to their level of vulnerability or the waiting time in the resettlement pipeline. The general priority order does not affect whether the stable undominated outcome is found. However, in the case of multiple undominated stable outcomes, the general priority order impacts which one undominated stable outcome is found. As we shall see, it is advantageous for families to be as high on general priority order as possible because it allows them to secure an undominated stable outcome they prefer. In Appendix C, we illustrate how different general priority orders might result in different outcomes and different running times for the algorithm.

An illustration of the DFS is in Figure 2. This illustration is not related to our running Example and is only there to convey the mechanics of the algorithm. The steps correspond to number labels on the nodes in this Figure. The general version of the DFS is in Appendix C.3.

Description of the DFS as illustrated in Figure 2

Step 1: We start the DFS 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 localities which f_1 ranks below $\bar{\ell}_{f_1}$. This hypothetical elimination allows us to begin our search along the lattice of possibly stable outcomes (a solution tree), looking for a feasible one. Once f is artificially guaranteed locality $\bar{\ell}_{f_1}$, we reduce the set of contracts and check for a feasible outcome by running a version of the TDBU algorithm, which we call the Augmented TDBU (ATDBU) algorithm (see Appendix C.2). The ATDBU algorithm works in the same way as the TDBU algorithm described above except that it also keeps track of artificial guarantees and artificial rejections (defined below). Let us suppose that f_1 can be feasibly matched to $\bar{\ell}_{f_1}$ given the artificial guarantee.

Step 2: Let us now give f_2 its top choice $\bar{\ell}_{f_2}$ as an artificial guarantee and remove all its contracts involving localities that f_2 ranks below $\bar{\ell}_{f_2}$.

Step 3: Let us now give f_3 its top choice $\bar{\ell}_{f_3}$ as an artificial guarantee and remove all its

contracts involving localities that f_3 ranks below $\bar{\ell}_{f_3}$.

Step 4: Suppose that after running the ATDBU algorithm, one of the artificial guarantees is rejected. Hence, we find that it is infeasible to match both f_1 , f_2 , and f_3 to their top-choice localities in any stable outcome. In that case, we go back to the situation obtained at the end of Step 2 (before f_3 get an artificial guarantee) and instead tentatively remove contract $(f_3, \bar{\ell}_{f_3})$. We call this hypothetical rejection an *artificial rejection* as it is only dictated by the current set of artificial guarantees.³⁷

Step 5: Let us re-run the ATDBU algorithm with $(f_1, \bar{\ell}_{f_2})$ and $(f_2, \bar{\ell}_{f_2})$ as artificial guarantees and $(f_3, \bar{\ell}_{f_3})$ as an artificial rejection.

Step 6: If the ATDBU algorithm returns an empty outcome (because the artificial rejection $(f_3, \bar{\ell}_{f_3})$ has been guaranteed), this means that there is no stable outcome in which f_1 and f_2 get their top-choice localities.

Step 7: Let us return back to the case where f_1 had an artificial guarantee for its top choice locality and let us artificially reject f_2 's top-choice locality by removing $(f_2, \bar{\ell}_{f_2})$.

Step 8: This time the ATDBU algorithm might not give us an empty outcome. This means that there is a possible stable outcome where f_1 has its top choice and f_2 has something other than its top choice (which we have artificially rejected). Now, let us give f_2 an artificial guarantee for its next top-choice locality. Continue trying along f_2 's preference list until the ATDBU algorithm gives a non-empty outcome for some artificially guaranteed locality for f_2 . We can then proceed searching for a stable outcome. Let us try giving f_3 an artificial guarantee for its top choice again (as we did in Step 3).

Steps 9-12: These follow analogously to Steps 2-5.

Step 13: The ATDBU algorithm returns a feasible outcome, which is a stable undominated outcome.

More generally, the ATDBU algorithm run at each step of the DFS can end in three different ways:

1. **● Outcome of the ATDBU algorithm is feasible:** This means that the DFS has found enough new rejections to identify a stable undominated outcome (Step 13).
2. **○ Outcome of the ATDBU algorithm is non-empty:** The ATDBU algorithm shows that a stable outcome given the current set of artificial guarantees and artificial rejections may exist but more rejections are needed to find it. We continue by giving an artificial guarantee of the next top choice to the next family in the general priority and running the ATDBU algorithm again (Step 1-3 and 8-10). We give top choices to

³⁷The ATDBU algorithm is equivalent to the TDBU algorithm whenever the sets of artificial guarantees and rejections are empty.

families according to the general priority order until either a stable outcome is found or we find that such an outcome is infeasible.

3. **● Outcome of the ATDBU algorithm is empty:** The ATDBU algorithm shows that no stable outcome exists given the current set of artificial guarantees and artificial rejections. This can only happen if either an artificial guarantee has been rejected or an artificial rejection has been guaranteed, or both. The DFS retraces its steps by removing the top choice of the latest family with an artificial guarantee and re-running the ATDBU algorithm (Steps 4-5 and 11-12). If the outcome of the ATDBU algorithm is empty again, then the DFS retraces its steps to the latest step at which the set of artificial guarantees and rejections was the same and removes the top choice of the family in that step (Step 6-7). If the outcome of the ATDBU algorithm is not empty, the DFS goes back to giving top choices to families according to the general priority (Step 8).

The DFS continues until either it finds a stable undominated outcome, which always happens if one exists, or by identifying that the set of stable outcomes is empty. The latter happens if an impossibility is found after all of f_1 's choices (including being unmatched) are first artificially guaranteed but then eventually rejected (or alternatively f_1 's top choice is artificially guaranteed and rejected and then an artificial rejection of f_1 's top choice is immediately guaranteed).

Phase 3: Assigning housing

If a stable undominated outcome Y between families and localities has been found in Phase 2, we run the final phase, which assigns houses to families. We simply apply any permissible housing assignment function to Y .

Proposition 5. *The Top Choice algorithm finds a stable undominated outcome if and only if the set of stable outcomes is nonempty.*

The Top Choice algorithm is not a strategy-proof mechanism. But this does not mean it is easy to manipulate. The general priority does not need to be revealed to the families and creates randomness that can punish potential manipulation. Refugee families submit preferences in an environment of uncertainty since the priorities of localities can shift unpredictably over time. Hence, it is also hard to learn from past matches how to manipulate the system (as parents have done over time when the highly manipulable Boston mechanism was used for school choice). Intuitively, manipulability increases as the DFS searches deeper (i.e. further to the right of the solution tree in Figure 2). The longer the DFS runs, the

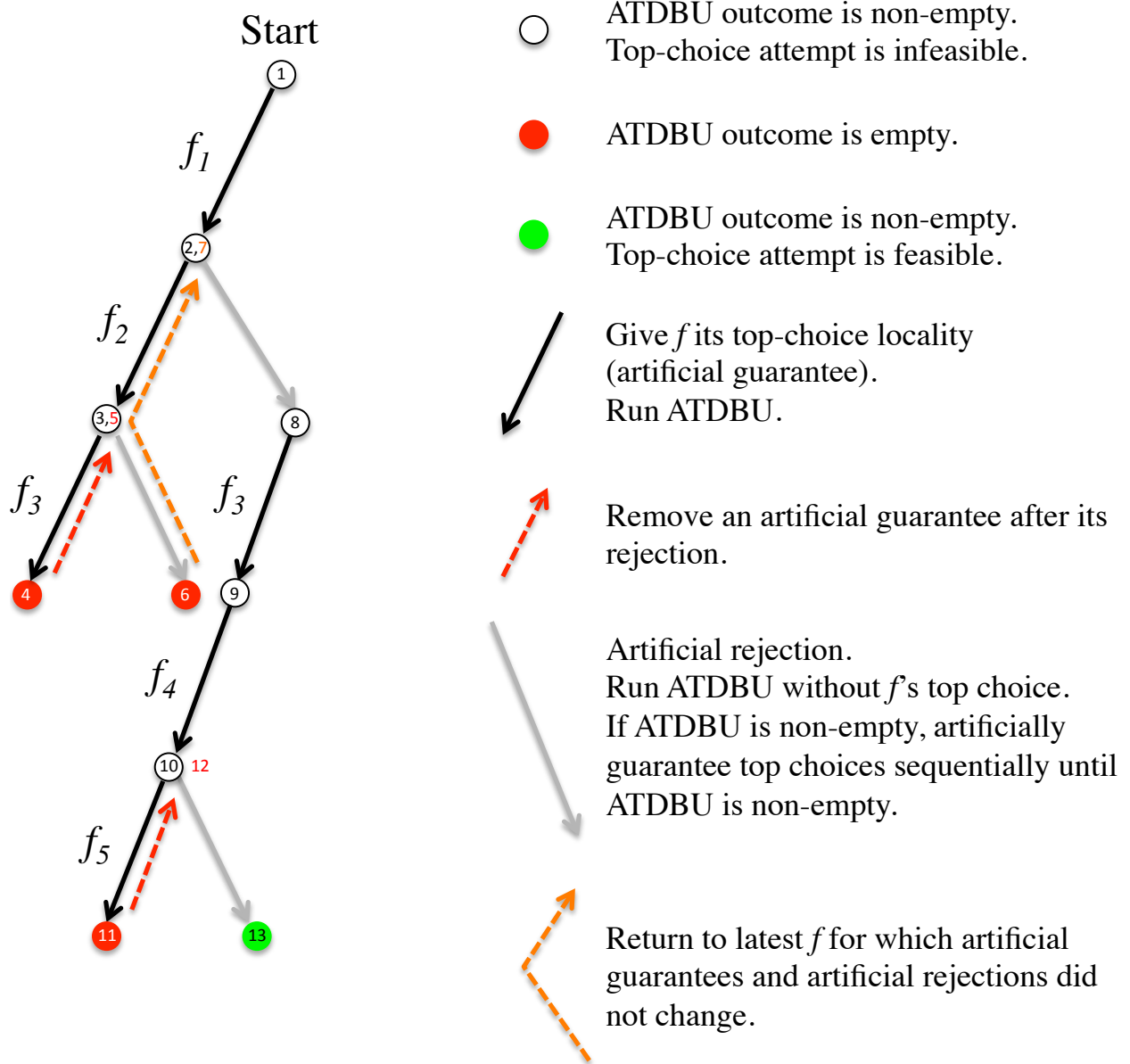


Figure 2: Depth-first search of the Top Choice algorithm

Algorithm 3: TOP CHOICE ALGORITHM

Phase 1: Run the TDBU algorithm to obtain $\phi(X)$. If $\phi^*(X)$ is feasible, stop and report the family-optimal stable outcome. If $\phi^*(X)$ is not feasible, let the current outcome be $Y = \phi(X)$. Create an arbitrary general priority order. Let the sets of artificial guarantees and artificial rejections be empty.

Phase 2: Run the DFS with the ATDBU algorithm as intermediate steps, keeping track of not only guarantees and rejections, but also of artificial guarantees and artificial rejections.

Phase 3: If an undominated stable outcome is reported at the end of Phase 2, assign housing using a permissible housing assignment function.

worse is the potential stable outcome for the refugee families and hence the stronger is their incentive to manipulate (Erdil and Ergin, 2008; Kesten, 2010).

6.2 Quasi-stable outcomes

The possible non-existence of stable outcomes and the computational challenges involved in finding them motivates us to seek an alternative stability concept. We now introduce *quasi-stability*, which respects priorities of localities, but introduces possible underuse in service capacities. In the context of refugee resettlement, this may well be tolerable. Refugees arrive to many localities regularly and many services, such as hospital beds and school places, are durable and unlikely to disappear if they are not immediately used. Any unused service capacities can simply be used for the next cohort of resettled refugee families.

Definition 2 (Quasi-Stability). A feasible outcome Y is quasi-stable if, for any locality ℓ and family $f' \in \mathbf{F}_\ell(Y)$ with $\ell' \neq \ell$, either $\ell' \succ_{f'} \ell$ or $f \pi_\ell f'$ for all $f \in \mathbf{F}_\ell(Y)$.

Quasi-stability does not allow families to block an outcome if the family has the lowest priority in the new locality. This immediately shows that quasi-stability itself is a permissive stability concept—an empty outcome is a quasi-stable outcome therefore quasi-stable outcomes always exist. Nevertheless, any quasi-stable outcome maintains complete respect for the priorities of the localities, and in the context of school choice any stable outcome is quasi-stable.³⁸

³⁸Even in the context of school choice, our definition of quasi-stability is stronger than envy-freeness (Wu and Roth, 2016), simplicity (Sotomayor, 1996), and firm-quasi-stability (Blum et al., 1997).

6.2.1 Mechanism for a family-optimal quasi-stable outcome

In Algorithm 4, we present the Priority-Focused Deferred Acceptance (PFDA) algorithm—a modified version of the classic deferred-acceptance (DA) algorithm (Gale and Shapley, 1962)—which finds the family-optimal quasi-stable outcome. In each round families apply to localities that have not rejected them yet. In order not to be rejected from a locality, the locality must be able to accommodate the family alongside families with a higher priority at that locality and must not have already rejected a family with a higher priority for that locality in this or an earlier round. The key difference between the DA and the PFDA algorithms is the condition that requires any family with a lower priority than one already rejected from a locality to be rejected from that locality immediately. This happens automatically in the DA algorithm in models without multidimensional constraints, such as school choice. In these contexts, the PFDA and the DA algorithms coincide. However, naively running the DA algorithm in our model means that a “smaller” family with a lower priority could be accepted after a “larger” family with a higher priority is rejected (because it could not be accommodated), resulting in an outcome which is neither stable not quasi-stable.

Proposition 6. *The PFDA algorithm yields a family-optimal quasi-stable outcome.*

The PFDA algorithm also makes it very clear how much “wasted capacity” the family-optimal, quasi-stable outcome can leave: when $|S| = 1$ in any locality, the maximum possible amount of capacity of any service that could be used by a low-priority family is the highest demand for this service minus 1. When there is more than one service, “large” families with high priorities for localities with small capacities could create greater waste.

Example for Section 6.2.1

In addition to the set-up of the Example and family preferences in the Example for Section 5.1, we now introduce priorities for the localities³⁹ into our Example.

$$\begin{aligned} \ell_1 &: f_2, f_1, f_4, f_5, f_3 \\ \ell_2 &: f_5, f_1, f_3, f_4, f_2 \\ \ell_3 &: f_5, f_3, f_2, f_1, f_4 \\ \ell_4 &: f_1, f_5, f_2, f_4, f_3 \end{aligned}$$

The PFDA algorithm calculates that outcome in four rounds, which are displayed below:

³⁹With these priorities, the outcome of the MTTC algorithm would be $\{(f_1, \ell_4), (f_2, \ell_3), (f_3, \ell_4), (f_4, \ell_2), (f_5, \ell_1)\}$

Algorithm 4: PRIORITY-FOCUSSED DEFERRED ACCEPTANCE ALGORITHM

Start with the set of all contracts X . Remove all contracts which are infeasible under the empty outcome. Consider the current outcome Y^1 .

Round 1:

Each family proposes to its top-choice locality under Y^1 . Each locality ℓ does not reject family f if f can be accommodated in ℓ alongside all families from set $\hat{F}_{(f,\ell)}$ that are proposing to ℓ . Remove all contracts involving the rejected families and the localities they proposed to from Y^1 . If at least one family is rejected, update current outcome to Y^2 and proceed to Round 2. Otherwise each family is permanently matched to the locality to which it last proposed and the algorithm ends.

Round $i > 1$:

Each family proposes to its top-choice locality under Y^i . Each locality ℓ does not reject f if:

Family f can be accommodated in ℓ alongside all families from set $\hat{F}_{(f,\ell)}$ that are proposing to ℓ , and

Family f does not have a lower priority than a family which was rejected from ℓ in this or an earlier round.

Any family that cannot be accommodated alongside proposing families or has a lower priority than a family which was rejected from this locality in this or an earlier round is rejected. Remove all contracts involving the rejected families and the localities they proposed to from Y .

If at least one family is rejected update current outcome to Y^{i+1} and proceed to Round $i+1$. Otherwise each family is permanently matched to the locality to which it last proposed and the algorithm ends.

Round 1	Round 2	Round 3	Round 4
$f_1 \rightarrow \ell_3 \times$	$f_1 \rightarrow \ell_4 \checkmark$	$f_1 \rightarrow \ell_4 \checkmark$	$f_1 \rightarrow \ell_4 \checkmark$
$f_2 \rightarrow \ell_3 \checkmark$	$f_2 \rightarrow \ell_3 \times$	$f_2 \rightarrow \ell_1 \checkmark$	$f_2 \rightarrow \ell_1 \checkmark$
$f_3 \rightarrow \ell_4 \times$	$f_3 \rightarrow \ell_3 \checkmark$	$f_3 \rightarrow \ell_3 \checkmark$	$f_3 \rightarrow \ell_3 \checkmark$
$f_4 \rightarrow \ell_4 \checkmark$	$f_4 \rightarrow \ell_4 \times$	$f_4 \rightarrow \ell_1 \checkmark$	$f_4 \rightarrow \ell_1 \checkmark$
$f_5 \rightarrow \ell_1 \checkmark$	$f_5 \rightarrow \ell_1 \checkmark$	$f_5 \rightarrow \ell_1 \times$	$f_5 \rightarrow \ell_2 \checkmark$

If a family is proposing to a locality and has not been rejected by the locality, we say it has been *tentatively accepted*.

Round 1: Families f_1 and f_2 are in competition for ℓ_3 . Family f_2 is tentatively accepted as it has a higher priority than f_1 and can be accommodated by itself. Family f_1 on the other hand cannot be accommodated alongside f_2 at ℓ_3 . Therefore it is rejected. Similarly, f_4 is tentatively accepted by ℓ_4 , but f_3 is rejected since it cannot be accommodated. Family f_5 is tentatively accepted by ℓ_1 as it is the only family proposing and can be accommodated by itself.

Round 2: Family f_1 proposes to its second choice and competes with f_4 for ℓ_4 . Family f_1 is tentatively accepted as it has a higher priority and can be accommodated by itself, however f_1 and f_4 cannot be accommodated alongside one another as h_{41} is not permissible for either of them. Family f_4 is rejected. Locality ℓ_3 can only accommodate one of f_2 and f_3 , so f_2 is rejected since it has a lower priority. As in Round 1, ℓ_1 tentatively accepts f_5 .

Round 3: Families f_1 and f_3 are again tentatively accepted by ℓ_4 and respectively ℓ_3 since they are the only proposers. Families f_2 , f_4 , and f_5 all propose to ℓ_1 . Families f_2 and f_4 are tentatively accepted as they both have a higher priority than f_5 and ℓ_1 can accommodate both of them. f_5 is rejected since ℓ_1 only has two houses.

Round 4: Family f_5 proposes to ℓ_2 and is tentatively accepted since it can be accommodated there. The other families propose to the same localities as in the previous round and are tentatively accepted as well. All families are now permanently matched and the algorithm ends.

The family-optimal quasi-stable outcome is:

$$Y^{PFDA} = \{(f_1, \ell_4), (f_2, \ell_1), (f_3, \ell_3), (f_4, \ell_1), (f_5, \ell_2)\}.$$

6.2.2 Strategic properties of the PFDA algorithm

The PFDA algorithm is appealing because it finds family-optimal quasi-stable outcomes quickly while respecting priorities of localities, but it is not strategy-proof. The reasoning is straightforward: When a family f applies to a locality and is rejected, it may also trigger another rejection (first example in Appendix D.2). Moreover, even when a family has proposed

and has not been rejected (but is rejected subsequently), its proposal may trigger multiple rejections (second example in Appendix D.2). In either case, the other rejected families end up competing with f for other localities. Therefore, if f does not get its top-choice locality, its chances of getting its second- or third-choice locality can also be reduced. Hence, manipulability of the PFDA algorithm is reminiscent of the manipulability of the Boston mechanism as families have an incentive to carefully consider what they report as their first preference since an early rejection may affect their eventual outcome.

Although the PFDA algorithm is not strategy-proof, it shares enough structure with the DA algorithm so that in a low-information environment truth-telling is preferred to other strategies. Following Roth and Rothblum (1999), we consider two informational environments: (i) $\{\ell, \ell'\}$ -symmetric information in which families assign the same probability to any submitted preference profile and its symmetric profile in which the ranking of ℓ and ℓ' is interchanged (ii) completely symmetric information in which families have $\{\ell, \ell'\}$ -symmetric information about any pair $\{\ell, \ell'\}$. Although under these low information conditions, families' beliefs are required to treat localities equally, it does not require families' beliefs to be independent. In fact, families' preferences can be highly correlated. We informally state the proposition and leave the technical details of the statement and the proof for the Appendix.

Proposition 7. *The PFDA algorithm is not strategy-proof, but*

- *for any family with $\{\ell, \ell'\}$ -symmetric information, truth-telling stochastically dominates reports that swap the order of ℓ and ℓ' under the PFDA algorithm.*
- *for any family with completely symmetric information, truth-telling stochastically dominates any other report under PFDA algorithm. Hence, if all families have completely symmetric information, truth-telling is an ordinal Bayesian Nash equilibrium of preference revelation game under the PFDA algorithm.*

This strategy-proofness result for the PFDA algorithm is rather weak. Indeed, even the highly-manipulable Boston mechanism (Abdulkadiroğlu and Sönmez, 2003) and the Stable Improvement Cycles (Erdil and Ergin, 2008) mechanism are strategy-proof in low information settings. In the school choice context, Kesten (2010) shows that for his “efficiency-adjusted deferred acceptance mechanism”, under a certain commonality of preferences, even partially symmetric information is sufficient for truth-telling to be an ordinal Bayesian Nash equilibrium. However, in our case, even when preferences of families are identical, manipulation of the PFDA algorithm is possible.⁴⁰

⁴⁰Specifically, our first counterexample to the strategy-proofness of the PFDA algorithm in Appendix D.2 shows that Case 2 of the proof of Proposition A.2. in Kesten (2010, p. 1343) fails when the quality classes are $\{\ell_1, \ell_2\}$ and $\{\ell_3\}$ and relabelling s as ℓ_1 , s' as ℓ_2 , and x as ℓ_3 .

6.2.3 A strategy-proof mechanism that finds quasi-stable outcomes

Our discussion of the manipulability of the PFDA algorithm reveals two important properties that a strategy-proof mechanism ought to have. First, if a family proposes to a locality and is rejected straight away, it does not start a rejection chain. Second, if a family proposes to a locality and is not rejected, the rejection chain of other families either comes back and triggers the family’s subsequent rejection or makes it more difficult to obtain its next choice, but both should not happen at the same time. If these properties are satisfied, then whenever the family is rejected by the locality to which it last proposed, it has the same competition for its next choice as it would have had without the proposal.

We now introduce the Maximum Rank Deferred Acceptance (MRDA) algorithm in which we first assign each family-locality pair a *Maximum Rank* and then use it to define a rejection rule in the strategy-proof mechanism that finds a quasi-stable outcome.

Consider a family f and a family f' which is just above f on the priority list of ℓ . The Maximum Rank for a family f in locality ℓ is either the minimum number of families that have a higher priority than f in ℓ alongside which f cannot be accommodated in ℓ , or the minimum number of families that have a higher priority than f' in ℓ alongside which f' cannot be accommodated in ℓ —whichever is smaller. Family f ’s Maximum Rank for locality ℓ is 0 if f cannot be accommodated at ℓ even on its own, i.e. it is “too big to fit”. Family f ’s Maximum Rank for locality ℓ is ∞ if f can be accommodated at ℓ alongside any set of families that have a higher priority at ℓ than f .

For example, if f' is 5th on ℓ ’s priority list and has a Maximum Rank of 3 while f is 6th on that list, f ’s Maximum Rank can be at most 3, even if it can be accommodated alongside all subsets of size greater than 3. Maximum Ranks must then be assigned recursively for each locality, starting from the family at the top of the locality’s priority list and going through that list one family at a time. This can be computed quickly if housing is reducible: Families can be sorted by sizes for each service and being able to fit with any subset of size n requires to be able to fit with the largest subset of size n for all services.

Once Maximum Ranks have been assigned to every family in every locality, we can use a variation on the DA algorithm where the rejection rule is tied to Maximum Ranks (Algorithm 5): A family is rejected if the number of families with a higher priority proposing to the same locality in a particular round is no less than the family’s Maximum Rank for that locality. The MRDA algorithm shares other important similarities with the DA algorithm that ensure its strategy-proofness: The Maximum Rank of family f for locality ℓ , which depends on priorities and constraints but not on preferences, determines the number of families with a higher priority for ℓ than f that can be matched to ℓ before f is rejected. The algorithm then aims to satisfy the preferences families as much as possible given that constraint. In the

school choice model, the MRDA algorithm collapses to the DA algorithm since the Maximum Rank of any student at any school is the school’s capacity minus one.

Proposition 8. *The MRDA algorithm is strategy-proof and yields a quasi-stable outcome. If housing is reducible, then the MRDA algorithm runs in polynomial time.*

Running the MRDA algorithm for our Example is straightforward and we show a richer example in Appendix D.1. The Example vividly highlights that the efficiency cost of strategy-proofness in the MRDA algorithm can be high compared to the outcomes of the PFDA algorithm. Unsurprisingly, for our Example, the outcome obtained by the PFDA algorithm dominates the one obtained by the MRDA algorithm: Families f_1 , f_2 and f_5 are matched to the same localities but f_3 and f_4 are unmatched under the MRDA algorithm. But this is not always the case. In Appendix D.3, we show a manipulable example of PFDA algorithm for which the MRDA algorithm still produces a family-optimal quasi-stable outcome but removes the temptation of agents to manipulate the mechanism.

Example for Section 6.2.3

In addition to the set-up of the Example and family preferences in the Example for Section 5.1, we use the priorities of localities introduced in the Example for Section 6.2.1.

Phase 1

The Maximum Ranks of all families for all localities are summarized below:

ℓ_1	ℓ_2	ℓ_3	ℓ_4
f_2 ∞	f_5 ∞	f_5 0	f_1 ∞
f_1 1	f_1 0	f_3 0	f_5 0
f_4 1	f_3 0	f_2 0	f_2 0
f_5 1	f_4 0	f_1 0	f_4 0
f_3 1	f_2 0	f_4 0	f_3 0

Family f_2 has the highest priority for ℓ_1 and can be accommodated there, hence its Maximum Rank is ∞ . Family f_1 ’s Maximum Rank for ℓ_1 is 1 as it can be accommodated by itself but not alongside f_2 . As a result, the Maximum Rank of all other families with a lower priority is at most 1. Since they can all be individually accommodated at ℓ_1 they all get a Maximum Rank of 1. Family f_5 can be accommodated at ℓ_2 and has the highest priority, hence its Maximum Rank is ∞ . Family f_1 , however, cannot be accommodated at ℓ_2 since (f_1, h_{21}) is impermissible. Its Maximum Rank for ℓ_2 is 0, which implies that the

Algorithm 5: MAXIMUM RANK DEFERRED ACCEPTANCE ALGORITHM

Phase 1:

Arbitrarily index localities $L = \{\ell_1, \ell_2, \dots, \ell_{|L|}\}$. Label families so that $f_{i,j}$ refers to the family with the j -th highest priority for locality ℓ_i . Proceed to Round 1.

Round $i \geq 1$

Step 1: Set family $f_{i,1}$'s Maximum Rank for locality ℓ_i to ∞ if $f_{i,1}$ can be accommodated at ℓ_i and to 0 otherwise. Proceed to Step 2.

Step $j > 1$:

If $f_{i,j}$ cannot be accommodated at ℓ_i , set its Maximum Rank for ℓ_i to 0.

Else if $f_{i,j}$ can be accommodated at ℓ_i alongside all other families with a higher priority, set its Maximum Rank for ℓ_i to ∞ .

Else consider the Maximum Rank of $f_{i,j-1}$ for ℓ_i :

If the Maximum Rank of $f_{i,j-1}$ for ℓ_i is 0, set $f_{i,j}$'s Maximum Rank for ℓ_i to 0.

Else if the Maximum Rank of $f_{i,j-1}$ for ℓ_i is $n \in \mathbb{N}$ and $f_{i,j}$ can be accommodated at ℓ_i alongside all subsets of $n - 1$ families with a higher priority, set $f_{i,j}$'s Maximum Rank for ℓ_i to n .

Else the Maximum Rank of $f_{i,j-1}$ for ℓ_i is ∞ or it is $n \in \mathbb{N}$ and there is a subset of $n - 1$ families with a higher priority alongside which $f_{i,j}$ cannot be accommodated. In that case set $f_{i,j}$'s Maximum Rank for ℓ_i to be $m \leq n$ such that $f_{i,j}$ can be accommodated at ℓ_i alongside all subsets of $m - 1$ families with a higher priority but there exists a subset of m families with a higher priority alongside which $f_{i,j}$ cannot be accommodated at ℓ_i .

If $j < |F|$, proceed to step $j + 1$.

Else if $j = |F|$ and $i < |L|$, proceed to Round $i + 1$.

Else $j = |F|$ and $i = |L|$, proceed to Phase 2.

Phase 2:

Start with the set of all contracts X . Remove all contracts which are infeasible under the empty outcome. Consider the current outcome Y^1 .

Round $i' \geq 1$

Every family proposes to its top-choice locality under $Y^{i'}$. A family is rejected if the number of families with a higher priority proposing to the same locality is no less than the family's Maximum Rank for that locality. Construct $Y^{i'+1}$ by removing all contracts between the rejected families and the localities they proposed to from $Y^{i'}$. If at least one rejection occurs proceed to Round $i' + 1$, otherwise each family is permanently matched to the locality to which it last proposed and the algorithm ends.

Maximum Rank of f_3 , f_4 and f_2 is also 0. Family f_5 needs three units of service s_1 and cannot be accommodated at ℓ_3 or ℓ_4 since they can only provide two units of service s_1 . Since f_5 has the highest priority for ℓ_3 and the second highest priority for ℓ_4 , all Maximum Ranks involving ℓ_3 or ℓ_4 are 0 with the exception of the Maximum Rank of f_1 for ℓ_4 . The latter is ∞ since ℓ_4 can accommodate f_1 .

Phase 2

The second phase of the MRDA algorithm lasts five rounds, which are summarized below:

Round 1	Round 2	Round 3	Round 4	Round 5
$f_1 \rightarrow \ell_3 \times$	$f_1 \rightarrow \ell_4 \checkmark$	$f_1 \rightarrow \ell_4 \checkmark$	$f_1 \rightarrow \ell_4 \checkmark$	$f_1 \rightarrow \ell_4 \checkmark$
$f_2 \rightarrow \ell_3 \times$	$f_2 \rightarrow \ell_1 \checkmark$	$f_2 \rightarrow \ell_1 \checkmark$	$f_2 \rightarrow \ell_1 \checkmark$	$f_2 \rightarrow \ell_1 \checkmark$
$f_3 \rightarrow \ell_4 \times$	$f_3 \rightarrow \ell_3 \times$	$f_3 \rightarrow \ell_1 \times$	$f_3 \rightarrow \ell_2 \times$	$f_3 \rightarrow \emptyset \checkmark$
$f_4 \rightarrow \ell_4 \times$	$f_4 \rightarrow \ell_1 \times$	$f_4 \rightarrow \ell_2 \times$	$f_4 \rightarrow \ell_3 \times$	$f_4 \rightarrow \emptyset \checkmark$
$f_5 \rightarrow \ell_1 \checkmark$	$f_5 \rightarrow \ell_1 \times$	$f_5 \rightarrow \ell_2 \checkmark$	$f_5 \rightarrow \ell_2 \checkmark$	$f_5 \rightarrow \ell_2 \checkmark$

If a family is proposing to a locality and has not been rejected by that locality, we say it has been *tentatively accepted*.

Round 1: Locality ℓ_3 rejects f_1 and f_2 and ℓ_4 rejects f_3 and f_4 because the Maximum Rank of these families is 0. Locality ℓ_1 tentatively accepts f_5 since it is the only family to propose and its Maximum Rank is 1.

Round 2: Family f_1 is tentatively accepted by ℓ_4 since its Maximum Rank is ∞ . Families f_2 , f_4 and f_5 propose to ℓ_1 . Locality ℓ_1 tentatively accepts the family with the highest priority, f_2 , and rejects the other two since their Maximum Ranks are 1. Family f_3 is rejected by ℓ_3 as its Maximum Rank is 0. Because f_1 and f_2 's Maximum Ranks for ℓ_4 and ℓ_1 respectively are ∞ , both families continue to propose to these localities for the remainder of the algorithm and are permanently matched to them in the end.

Round 3: Locality ℓ_1 rejects f_3 since it has a lower priority than f_2 and a Maximum Rank of 1. Locality ℓ_2 tentatively accepts f_5 , which has a Maximum Rank of ∞ but rejects f_2 as its Maximum Rank is 0. Family f_5 will continue to propose to ℓ_2 for the remainder of the algorithm and will be permanently matched to it in the end.

Round 4: Families f_3 and f_4 are rejected by ℓ_2 and ℓ_3 respectively since their Maximum Rank is 0.

Round 5: Families f_3 and f_4 propose to the null object in Round 5 and all families are permanently matched. The algorithm ends and yields the following outcome:

$$Y^{MRDA} = \{(f_1, \ell_4), (f_2, \ell_1), (f_3, \emptyset), (f_4, \emptyset), (f_5, \ell_2)\}.$$

	Preferences	Priorities	Manipulability	Computation
OQMP	–	–	–	NP-hard
MTTC	Pareto-efficient	–	Strategy-proof	Polynomial
Serial MTTC	Individually rational	–	Strategy-proof	Polynomial
Serial dictatorship	Pareto-efficient	Stable (identical priorities)	Strategy-proof	Polynomial
Top Choice	Family-undominated	Stable	Difficult	NP-complete
PFDA	Family-optimal	Quasi-stable	Only strategy-proof under low information	Polynomial
MRDA	–	Quasi-stable	Strategy-proof	Polynomial if housing is reducible

Table 1: Properties of different mechanisms for matching with multidimensional constraints

7 Tradeoffs between different mechanisms

This paper developed seven different mechanisms for matching with multidimensional constraints (Table 1). Some of these, such as the serial dictatorship and the Multidimensional Top Trading Cycles algorithm, are based on familiar matching market design tools. Others are completely tailored to our context. We showed how information about the preferences of refugees and the priorities of localities can be incorporated into the refugee matching system. However, the eventual choice of the refugee matching algorithm not only depends on the information available to the social planner, but also on the structure of the problem: (i) its size (the number of refugees, localities, and services), and (ii) the importance and heterogeneity of locality priorities (Table 2).

For smaller problems (e.g., 50 refugees families arriving on one plane being matched to a dozen localities), and without any information about preferences or priorities, the only option is to solve the outcome-quality maximization problem. If preference information is available, then it is possible to quickly find Pareto improvements upon a solution to the outcome-quality maximization problem using the SMTTC algorithm. If priorities also matter, then stable

	Market size	
	Small	Large
Priorities are not important	MTTC, Serial MTTC (from OQMP)	MTTC, Serial MTTC (from any endowment)
Priorities are important	Top Choice	PFDA, MRDA

Table 2: Comparing mechanisms for matching with multidimensional constraints

undominated outcomes can be found (whenever they exist) in a reasonable time using the Top Choice algorithm. In small-size problems with little capacity for some services that are in high demand (especially when a single family’s demand exceeds capacities in certain localities where they have high priority), the PFDA and MRDA algorithms may perform poorly. This points to a need for sensible *priority design*: localities that have certain services in very limited supply should avoid prioritizing families that have a great need for those services.

When priorities are sufficiently homogeneous, the Top Choice and MTTC algorithms would produce similar outcomes, and the MTTC algorithm is likely to be preferred for computational reasons. However, when priorities are heterogeneous and matter a great deal, the social planner faces a stark choice. While the Top Choice algorithm can deal with small instances, for large problems there is no way of knowing whether stable outcomes exist.

In larger matching markets, including those outside the refugee resettlement context (see Section 9), the outcome-quality maximization problem and the Top Choice algorithm might be too computationally demanding. If priorities are identical, there is no computational concern as the serial dictatorship delivers a stable, Pareto-efficient outcome in a strategy-proof way. If priorities are not identical but can be ignored, the sensible mechanism to use is the MTTC algorithm. In fact, a hybrid mechanism that first quickly approximates a reasonable solution to the outcome-quality maximization problem followed by the SMTTC algorithm is might also be appealing in this case. Finally, in large problems in which priorities must be fully respected, both PFDA and MRDA algorithms are good options: their outcomes are close to stable ones, they have little waste, and they can be implemented quickly. Selecting between two quasi-stable mechanisms—the PFDA and MRDA algorithms—creates similar trade-offs to school choice settings: efficiency comes at a cost to strategy-proofness (Erdil and Ergin, 2008; Kesten, 2010).

8 Applications to other resettlement contexts

The context of the British resettlement program for Syrian refugees is particularly interesting because it is a comparatively new large-scale program—which will still evolve. However, as we discuss below, our market design insights also apply in contexts where the resettlement programs are more mature.

8.1 United States

The United States has resettled over 70,000 refugees annually between 2013 and 2015, 85,000 in 2016, and is committed to resettling at least 110,000 in the 2017 (White, 2016). The resettlement process is similar to the UK. The US State Department conducts the security and background checks in conjunction with UNHCR. Refugees are allowed to list family members (around a half do so) who live in the United States and are almost certain to be reunited with them. Refugees can also list friends around the US. Given that it is easy to report and conceal a friend, these indicators could be treated as weak preferences. The job of matching refugees to local areas, however, is delegated to nine Voluntary Agencies (VolAgs). VolAgs establish their own links to local communities that are willing to host refugees. Often, this is done through religious institutions, such as churches and synagogues. VolAgs consult the communities about which categories of refugees they are interested in hosting and attempt to incorporate these priorities. Every week, the agencies first distribute the arriving casework among themselves (using a priority mechanism since agency preferences for the kinds of refugees they want to resettle are highly correlated). The agency is then responsible for placing refugee families in communities. The government provides support for refugees only for the first 90 days and the resettlement agencies are evaluated on their success of getting refugees employed within that period.

In recent work, Feywerda and Gest (2016) find that the matching of refugees to local areas by one large American voluntary agency is almost random, despite an explicit incentive to maximize the expected number of refugees in employment. This work suggests that systematic matching could add a lot of value in the American resettlement process.

8.2 Canada

Canada operates three resettlement schemes: private sponsorship, government sponsorship, and mixed private-government sponsorship.⁴¹ Our results could apply directly to the gov-

⁴¹Québec runs a separate and independent resettlement scheme.

ernment and mixed schemes.⁴² The institutional context of Canadian resettlement is very similar to the UK: the process is centralized, but operates in close cooperation to the federal provinces and territories. Moreover, Canada and the UK share similar welfare systems. Most importantly, unlike its American counterparts, the Canadian resettlement authority does not focus on a single metric for refugee success. Since the fall of 2015, Canada has substantially stepped up its resettlement efforts for Syrian refugees. It keeps a live website recording all the Syrian arrivals: Between November 2015 and October 2016, Canada resettled 31,919 Syrian refugees into 316 communities.⁴³ From a modeling perspective, the Canadian resettlement system is a larger-scale version of the British one.

9 Further applications of matching with multidimensional constraints

9.1 Multidimensional diversity constraints

Many public institutions prefer a diverse membership and implement diversity targets or affirmative action policies. Recently, a number of papers in matching theory has sought to integrate diversity concerns analyzing a variety of reserve, quota, and balancing schemes (Abdulkadiroğlu and Sönmez, 2003; Westkamp, 2013; Echenique and Yenmez, 2015; Kominers and Sönmez, 2016). Most of the prior work has implicitly assumed that any agent fills a quota for their specific unidimensional type. This ignores the fact that some agents can affect several quota dimensions. Our model allows public institutions to implement diversity targets directly by allowing for multidimensional type constraints. For example, a student could be represented by a vector of characteristics denoting their gender, socioeconomic status, race, and ethnicity (in the same way as a refugee family has needs for multiple services). A school could in turn implement multidimensional diversity quotas for these characteristics (same as locality capacities). This could allow the same student to fill a college quota for race and gender simultaneously.⁴⁴

⁴²There are interesting design issues to address in the private sponsorship scheme, such as the optimal length of waiting time before matching.

⁴³See <http://www.cic.gc.ca/english/refugees/welcome/milestones.asp>

⁴⁴We do not require that the student necessarily take up a full race-slot and a full gender-slot. Aygün and Bó (2016) consider a case of multidimensional privileges in Brazilian college admissions. In their case, the preferences of colleges are constrained and reporting privileges is a strategic decision.

9.2 Daycare matching

Parents of children in daycare are often in part-time work. Hence, they only require daycare on particular days of the week or, indeed, for certain parts of the day. Daycare centers have capacity constraints on staff that might also vary day-by-day. Our model allows parents to express their preferences over particular days and times without modifying the priorities of the daycare center (e.g. using neighborhood or sibling priority) or violating the center’s capacity constraints over the course of the week. For example, a parent might wish to only send one of their children to daycare for the morning of Monday and both children for the afternoon of Thursday and Friday, while the daycare center may be able to accommodate twelve children on Thursday morning and only eight children on Wednesday morning. Each “service” is then a day and a time slot and the needs of families for each “service” are the numbers of children they wish to send to daycare during that time slot.

10 Conclusion

This paper described matching problems that arise at different stages of refugee resettlement and showed three ways in which systems for refugee resettlement can be enhanced using insights from market design: by maximizing welfare based on observables, by using refugee family preferences, or by using both preferences and locality priorities. 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 algorithm (i) to achieve a Pareto-efficient outcome from an empty endowment and (ii) to find Pareto improvements from any endowment in a strategy-proof way. We showed that, unless priorities are identical, stable outcomes may not exist and can be computationally hard to find. We then introduced another stability concept—quasi-stability—that fully respects the priorities of localities. We showed that there was a tradeoff between reaching family-optimal and strategy-proof quasi-stable outcomes. In general, different mechanisms we proposed would work well depending on the size of the matching market and the importance of satisfying heterogeneous priorities. Policy work to implement our matching tools is already beginning in several countries.⁴⁵ Our framework for matching with multidimensional constraints has a number of possible applications from incorporating complex diversity constraints in school choice or college admissions to designing new systems that would match children to daycare centers. Thus matching with multidimensional constraints offers an exciting area of theoretical and applied research.

⁴⁵See www.refugees-say.com.

Appendix

A The running Example

- 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 capacities: Service needs:

$$\kappa = \begin{matrix} & s_1 & s_2 \\ \ell_1 & \begin{pmatrix} 4 & 2 \end{pmatrix} \\ \ell_2 & \begin{pmatrix} 3 & 2 \end{pmatrix} \\ \ell_3 & \begin{pmatrix} 2 & 2 \end{pmatrix} \\ \ell_4 & \begin{pmatrix} 2 & 2 \end{pmatrix} \end{matrix} \qquad \nu = \begin{matrix} & s_1 & s_2 \\ f_1 & \begin{pmatrix} 1 & 0 \end{pmatrix} \\ f_2 & \begin{pmatrix} 2 & 1 \end{pmatrix} \\ f_3 & \begin{pmatrix} 0 & 2 \end{pmatrix} \\ f_4 & \begin{pmatrix} 1 & 1 \end{pmatrix} \\ f_5 & \begin{pmatrix} 3 & 0 \end{pmatrix} \end{matrix}$$

We 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 impermissible family-house pairs.

Priorities of localities:	Preferences of families:	Quality score: $q((f, \ell))$
$\ell_1 : f_2, f_1, f_4, f_5, f_3$	$f_1 : \ell_3 \succ \ell_4 \succ \ell_1 \succ \ell_2$	$\ell_1 \quad \ell_2 \quad \ell_3 \quad \ell_4$
$\ell_2 : f_5, f_1, f_3, f_4, f_2$	$f_2 : \ell_3 \succ \ell_1 \succ \ell_2 \succ \ell_4$	$f_1 \quad 71 \quad \emptyset \quad 23 \quad 38$
$\ell_3 : f_5, f_3, f_2, f_1, f_4$	$f_3 : \ell_4 \succ \ell_3 \succ \ell_1 \succ \ell_2$	$f_2 \quad 46 \quad 49 \quad 30 \quad 91$
$\ell_4 : f_1, f_5, f_2, f_4, f_3$	$f_4 : \ell_4 \succ \ell_1 \succ \ell_2 \succ \ell_3$	$f_3 \quad 52 \quad 68 \quad 43 \quad 20$
	$f_5 : \ell_1 \succ \ell_2 \succ \ell_4 \succ \ell_3$	$f_4 \quad 4 \quad 75 \quad 96 \quad 36$
		$f_5 \quad 92 \quad 41 \quad \emptyset \quad \emptyset$

Outcomes of mechanisms proposed in this paper for the running Example:

- Outcome-quality maximization problem (OQMP): [Example for Section 4](#).
- Multidimensional Top Trading Cycles (MTTC): [Example for Section 5.1](#).
- Serial Multidimensional Top Trading Cycles: [Example for Section 5.2](#).
- Top Choice: Example in Appendix C.
- Priority-Focused Deferred Acceptance (PFDA): [Example for Section 6.2.1](#).
- Maximum Rank Deferred Acceptance (MRDA): [Example for Section 6.2.3](#).

Reducible housing and housing in the Example

In general, suppose that houses differ by size $\{1, \dots, M\}$ and provide a subset of features from a set Θ . The *type* of house h is then $\Theta^h \subseteq \Theta$. Each family requires a subset of the features in their house $\Theta^f \subseteq \Theta$. Family f can be accommodated in house h as long as the house size is greater than $|f|$ and $\Theta^f \subseteq \Theta^h$. Consider a subset of service constraints that represents housing $S^h \subseteq S$. Let $s^{hi, \Theta}$ be the service representing a house of size $i \in \{1, \dots, M\}$ and type $\Theta \subseteq \Theta$. Then $\left(\kappa_{s^{hi, \Theta}}^\ell\right)_{i \in \{1, \dots, M\}, \Theta \subseteq \Theta}$ and $\left(\nu_{s^{hi, \Theta}}^f\right)_{i \in \{1, \dots, M\}, \Theta \subseteq \Theta}$ denote the subvectors of $\left(\kappa_s^\ell\right)_{s \in S}$ and $\left(\nu_s^f\right)_{s \in S}$ respectively representing housing constraints both of dimension $|M \times 2^\Theta|$. For a family of size i that requires a set of features Θ^f , the demand for housing services is $\nu_{s^{hj, \Theta}}^f = 1$ if $j \leq i$ and $\Theta \subseteq \Theta^f$ and zero otherwise. $\kappa_{s^{hj, \Theta}}^\ell$ is then the number of houses of size at least j that contain features $\Theta \supseteq \Theta^h$ in locality ℓ .

In the running Example, housing is reducible with three housing features and two sizes. Let us suppose that there are two house and family sizes: {small, large} and three house features: $\Theta = \{\theta_1, \theta_2, \theta_3\}$. A small family can fit into a large house, but not vice versa.

Family needs			Housing		
Family	size	type	House	size	type
f_1	small	$\theta_1, \theta_2, \theta_3$	h_{11}	small	$\theta_1, \theta_2, \theta_3$
f_2	large	θ_2	h_{12}	large	θ_1
f_3	large	\emptyset	h_{21}	large	θ_1, θ_2
f_4	large	θ_1	h_{31}	large	$\theta_1, \theta_2, \theta_3$
f_5	large	\emptyset	h_{32}	large	$\theta_1, \theta_2, \theta_3$
			h_{41}	large	θ_2
			h_{42}	large	$\theta_1, \theta_2, \theta_3$

B Proofs

Proof of Proposition 1. First note that:

$$\max_{Y \subseteq X} \sum_{y \in Y} q(y) \quad \text{corresponds to:} \quad \max \sum_{\ell \in L} \sum_{f \in F} q(f, \ell) \iota(f, \ell) \quad \iota(f, \ell) \in \{0, 1\}.$$

While for the two feasibility constraints on Y , we have

$$\begin{aligned} |Y \cap X_f| = 1 \text{ for all } f & \quad \text{corresponds to:} \quad \sum_{\ell \in L} \iota(f, \ell) \leq 1 \quad \forall f \quad \iota(f, \ell) \in \{0, 1\}, \\ \tau(Y) \leq \kappa & \quad \text{corresponds to:} \quad \sum_{f \in F} \sum_{\ell \in L} \nu_s^f \iota(f, \ell) \leq \kappa_s^\ell \quad \forall \ell, s \quad \iota(f, \ell) \in \{0, 1\}. \end{aligned}$$

Since the any subset of families can be feasibly accommodated in the null locality, it does not impose a constraint on the problem hence the outcome-quality maximization problem is precisely the multiple multidimensional knapsack problem.

In order to apply the branch-and-bound algorithm, we first need to a determine an upper bound on the value of the maximand. Here we cannot use standard greedy solutions to a linear relaxation of the knapsack problem. We extend the approach of Song et al. (2008) to our multidimensional case and decompose our problem into several standard knapsack problems, which can all be independently linearly relaxed to obtain a tight upper bound for the whole problem. The lower bound is, of course, zero, since all families can be unmatched.

First, let us identify each service constraint:

$$\begin{aligned} \max \sum_{\ell \in L} \sum_{f \in F} q(f, \ell) \iota(f, \ell) & \tag{3} \\ \sum_{f \in F} \nu_{s_1}^f \iota(f, \ell) & \leq \kappa_{s_1}^\ell \quad \forall \ell \\ & \vdots \\ \sum_{f \in F} \nu_{|S|}^f \iota(f, \ell) & \leq \kappa_{|S|}^\ell \quad \forall \ell \\ \sum_{\ell \in L} \iota(f, \ell) & \leq 1 \quad \forall f \\ \iota(f, \ell) & \in \{0, 1\}. \end{aligned}$$

Let us introduce dual variables λ_f for all $f \in F$ and γ_ℓ for all $\ell \in L$. Convert Problem (3) into a problem with a one-dimensional constraint:

$$\begin{aligned}
& \max_{\boldsymbol{\nu}} L(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) \tag{4} \\
& \sum_{f \in F} \nu_{s_1}^f \iota(f, \ell) \leq \kappa_{s_1}^\ell \quad \forall \ell \\
& \iota(f, \ell) \in \{0, 1\} \\
& \text{where: } L(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = \sum_{\ell \in L} \sum_{f \in F} q(f, \ell) \iota(f, \ell) \\
& \quad - \sum_{f \in F} \lambda_f \left(\sum_{\ell \in L} \iota(f, \ell) - 1 \right) \\
& \quad - \sum_{s \in S \setminus \{s_1\}} \sum_{\ell \in L} \gamma_s^\ell \left(\sum_{f \in F} \nu_s^f \iota(f, \ell) - \kappa_s^\ell \right).
\end{aligned}$$

Denote \tilde{O} as the optimum value of Problem (4), which is also an upper bound for the outcome-quality maximization problem for any $\lambda_f, \gamma_\ell \geq 0$. For a tight upper bound, let us minimize

$$\min_{\boldsymbol{\lambda}, \boldsymbol{\gamma}} L(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$$

and rewrite Problem (4) as:

$$\begin{aligned}
L(\boldsymbol{\nu}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) &= \sum_{\ell \in L} \sum_{f \in F} q(f, \ell) \iota(f, \ell) - \sum_{\ell \in L} \sum_{f \in F} \lambda_f \iota(f, \ell) - \sum_{s \in S \setminus \{s_1\}} \sum_{\ell \in L} \sum_{f \in F} \gamma_s^\ell \nu_s^f \iota(f, \ell) \tag{5} \\
&+ \sum_{f \in F} \lambda_f + \sum_{s \in S \setminus \{s_1\}} \sum_{j=1}^m \gamma_s^j \kappa_s^j \\
&= \sum_{\ell \in L} \left\{ \sum_{f \in F} \left(\underbrace{q(f, \ell) - \lambda_f - \sum_{s \in S \setminus \{s_1\}} \gamma_s^\ell \nu_s^f}_{\tilde{q}_\ell} \right) \iota(f, \ell) \right\} \\
&+ \sum_{f \in F} \lambda_f + \sum_{s \in S \setminus \{s_1\}} \sum_{\ell \in L} \gamma_s^\ell \kappa_s^\ell.
\end{aligned}$$

Let us define $\tilde{q}_\ell = q(f, \ell) - \lambda_f - \sum_{s \in S \setminus \{s_1\}} \gamma_s^\ell \nu_s^f$ as ℓ^{th} decomposed knapsack problems,

namely

$$\begin{aligned}
\max \sum_{f \in F} \tilde{q}_\ell \iota(f, \ell) & \tag{6} \\
\sum_{f \in F} \nu_{s_1}^f \iota(f, \ell) & \leq \kappa_{s_1}^\ell \\
\iota(f, \ell) & \in \{0, 1\}.
\end{aligned}$$

Now, let us find an upper bound for all $|L|$ problems \tilde{q}_ℓ by the following linear relaxation and a greedy algorithm for the upper bound (see Kellerer et al. (2004)):

$$\begin{aligned}
\max \sum_{f \in F} \tilde{q}_\ell \iota(f, \ell) & \tag{7} \\
\sum_{f \in F} \nu_{s_1}^f \iota(f, \ell) & \leq \kappa_{s_1}^\ell \\
\iota(f, \ell) & \in [0, 1].
\end{aligned}$$

We denote the upper bound of the ℓ^{th} subproblem (6) as o_ℓ . The upper bound of the outcome quality maximization problem is then:

$$\sum_{\ell \in L} o_\ell + \sum_{f \in F}^n \lambda_i + \sum_{s \in S \setminus \{s_1\}} \sum_{\ell \in L} \gamma_s^\ell \kappa_s^\ell$$

Once the upper bound has been determined, a standard branch-and-bound method can be applied. Finding a heuristic that gives an approximation to this problem that is independent of the number of services is an open problem and beyond the scope of this paper. \square

Proof of Proposition 2. First, we show how to determine the top-choice locality and the highest-quality family that can be accommodated at each round. Given a current outcome Y , consider a family f and a locality ℓ . Consider the set of families that are permanently matched to ℓ , $\mathbf{F}_\ell(Y)$. We are interested in whether $f \notin F$ can be accommodated in ℓ alongside $\mathbf{F}_\ell(Y)$. If housing is reducible, this is trivial: we simply check whether feasibility constraints at ℓ are violated when $Y \cup \{(f, \ell)\}$ is an outcome.

Without reducible housing, consider H_ℓ , the set of houses in locality ℓ . The problem is non-trivial if $|H_\ell| \geq |\mathbf{F}_\ell(Y) \cup \{f\}|$. We have an instance of a maximum bipartite matching problem. Construct a graph as follows: The nodes are partitioned into two subsets H_ℓ and $\mathbf{F}_\ell(Y) \cup \{f\}$ and any edge represents a permissible family-house pair (f, h) such that $f \in \mathbf{F}_\ell(Y)$ and $h \in H_\ell$. All edges are given a weight of one except edges adjacent to f , which have weight $1 + \epsilon$ (s.t. $\epsilon < \frac{1}{|H_\ell|}$) in order to break ties. We solve this instance of the maximum bipartite matching problem (e.g. using the Ford-Fulkerson algorithm) and see whether f is included in the solution. If every family, including f , has an adjacent edge in

the optimal solution, then f can point at ℓ (or ℓ can point at f).

Once any locality and any family is able to determine the next family it wants to point at, it is straightforward to see that the proofs of efficiency and strategy-proofness in our case follow proofs Proposition 3 and 4 in [Abdulkadiroğlu and Sönmez \(2003\)](#) with trivial modifications (e.g. adjusting capacity of every service rather than adjusting the capacity of the total school quota). \square

Proof of Proposition 3. We prove each property in turn.

Individual Rationality

At any stage in the algorithm when f has been made active but not yet permanently matched, f points at top-choice locality while the locality it is endowed with, say ℓ , points at f . As f can always be accommodated in ℓ , f first points either at ℓ or a locality it prefers to ℓ . Therefore either f is permanently matched to a locality it prefers to ℓ or it ends up pointing at ℓ . In that case f is always matched to ℓ . Consequently, f is always permanently matched to ℓ or a locality it prefers to ℓ .

Strategy-proofness

A family cannot impact its outcome before it is made active or after it is permanently matched. Therefore an incentive to manipulate preferences exists if and only if at some point between the round in which a family is made active and the round in which it is permanently matched, the family can improve its outcome by pointing at a locality other than the one it prefers among those where it can be accommodated. We complete the proof by showing that no such deviation exists.

Consider any round of the algorithm where f has already been made active but not yet permanently matched. Let (ℓ_1, \dots, ℓ_n) be the localities where f can be accommodated, ordered by f 's preference. By construction, f can only point at one of these localities. It remains to show that pointing at ℓ_1 is a (weakly) dominant strategy. Let ℓ_j (for some $j = 1, \dots, n$) be f 's endowment. (By definition, ℓ_j is always accessible.) Locality ℓ_j points at f so, f is permanently matched to ℓ_j if it points back at it. A chain may exist such that a locality points at a family, which points at another locality, that locality points at a family and so on until a family points at ℓ_j . In that case, a cycle forms if f points at the first locality in that chain. Denote by Z the the set of localities in this chain. Let ℓ_k ($k = 1, \dots, j$) be the locality f prefers among localities in Z . Then f is permanently matched to ℓ_k if it points at ℓ_k . If $k = 1$, pointing at ℓ_k is clearly a dominant strategy since it ensures f is matched to the best locality it can still possibly obtain. We consider below the case where $k > 1$.

Family f has three possible strategies. First, it can point at ℓ_k and be permanently matched to ℓ_k . Second, there may exist one or more locality in Z to which f would be matched if it pointed at any of them. As f prefers ℓ_k to all these localities in Z , the first strategy dominates the second. Third, f can point at any other locality. No matter which of these localities f points at, the same cycles appear and f is not part of any of them. Additionally, ℓ_k as well as the families and localities involved in the chain from ℓ_k to f do not appear in a cycle either since f does not point at ℓ_k . After the cycles are carried out, f faces a new (smaller) set of localities where it can be feasibly matched and is still matched to ℓ_k if it points at it. The third strategy dominates the first since it ensures f can be matched

to ℓ_k or a locality it prefers to ℓ_k in later rounds of the algorithm. It follows that pointing at any locality such that no cycle is formed is a (weakly) dominant strategy for f . Since ℓ_1 is among these localities, pointing at ℓ_1 is a (weakly) dominant strategy. \square

Proof of Proposition 4. (\Rightarrow) Suppose, towards a contradiction, that for some outcome of the serial dictatorship there is blocking pair (f, ℓ) . This means that (i) ℓ can accommodate f (perhaps after removing another lower priority family or families from that locality) (ii) f could have been permanently matched to that locality when its priority order was called. Hence, family f must have preferred not to be at ℓ . Hence, there is no blocking pair. A contradiction.

(\Leftarrow) Suppose, towards a contradiction, that there is a stable mechanism that produces an outcome that is different from an outcome of a serial dictatorship. Consider a family f that is highest in the priority order whose locality is different from the locality that it is matched to by the serial dictatorship. But the serial dictatorship gives f its most preferred locality given that all the families with a higher priority than it have the same allocation under both mechanisms. Hence, the family prefers the serial dictatorship outcome to its current outcome. Moreover, it has the highest priority in its top-choice locality, following families before it whose outcome was identical. Hence, there is a blocking pair and the other outcome is not stable. A contradiction. \square

Proof of Proposition 5. First, let us define the terms we use.

Definition 3. For any X , π , and \succ , we say that f receives a *guarantee* for ℓ if in every stable outcome Y , we have $(f, \ell') \in Y$ for some $\ell' \succ_f \ell$ or $\ell' = \ell$.

We now describe how to find guarantees algorithmically. Let $\overline{F}_\ell(Y) \equiv \{f \in F \mid \ell = \bar{\ell}_f(Y)\}$ be the set of families that have ℓ as their top choice.⁴⁶ For any $f \in F$, let $\hat{F}_{(f, \ell)}$ be the set of families that have a higher priority for ℓ than f . We also define $\hat{G}_{(f, \ell)} \subseteq \hat{F}_{(f, \ell)}$ as the set of families with a higher priority at ℓ than f who have received a guarantee at ℓ . According to Definition 3, families in $\overline{F}_\ell \cap \hat{G}_{(f, \ell)}$ will be matched to ℓ in any stable outcome (for any f). Families in $\hat{F}_{(f, \ell)} \setminus (\overline{F}_\ell \cap \hat{G}_{(f, \ell)})$ may or may not be matched to ℓ in some stable outcome. Family f therefore receives a guarantee if it can be accommodated alongside every feasible subset of $\hat{F}_{(f, \ell)}$ that contains $\overline{F}_\ell \cap \hat{G}_{(f, \ell)}$. Formally, f receives a guarantee if for all feasible F' such that $\overline{F}_\ell \cap \hat{G}_{(f, \ell)} \subseteq F' \subseteq \hat{F}_{(f, \ell)}$, $F' \cup \{f\}$ is feasible. Intuitively, a family f receives a guarantee from its top-choice locality ℓ if 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).

Definition 4. For any X , π , and \succ , we say that f receives a *rejection* for ℓ if in every stable outcome Y , we have $(f, \ell) \notin Y$.

We now describe how to find rejections algorithmically. Family f is then rejected if there is no subset F' such that $\overline{F}_\ell \cap \hat{G}_{(f, \ell)} \subseteq F' \subseteq \hat{F}_{(f, \ell)}$ and

- $F' \cup \{f\}$ is feasible, and
- For every $f' \in (\overline{F}_\ell \cap \hat{F}_{(f, \ell)}) \setminus F'$ we have that $(F_{(f, \ell)} \cap F') \cup \{f'\}$ is infeasible.

⁴⁶We drop argument Y when it is clear what the relevant outcome is.

The intuition behind this rejection rule is the following: f is rejected from ℓ if there is no subset of families with a higher priority than f that includes all families with ℓ as their top choice and with a guarantee for ℓ which (i) can accommodate f and (ii) is not blocked by any family with a higher priority than f which has ℓ as its top choice (but no guarantee for ℓ).

Artificial guarantees and rejections

Let us define $G^A \subseteq F \times L$ and $R^A \subseteq F \times L$ as the sets of artificial guarantees and rejections. Artificial guarantees and artificial rejections induce the *same actions* within the algorithm as guarantees and rejections. Given an artificial guarantee of family f for locality ℓ , we remove all the contracts for the localities that f ranks below ℓ . Given an artificial rejection of family f for locality ℓ , we remove all contracts between f and ℓ . Because artificial guarantees and rejections are tentative i.e. not final, a family can have: an artificial guarantee and a rejection for a given locality, or an guarantee and an artificial rejection, or an artificial guarantee and an artificial rejection.

Lemma 1. *The TDBU and the ATDBU algorithms do not remove any contracts that could be part of a stable outcome.*

Proof. See technical descriptions of the algorithms in Section C.1. First, if a family f can be accommodated in a locality ℓ alongside any subset of families that have a higher priority in that locality, then if it is matched to a locality ℓ' that is preferred less, then in any stable outcome, it would be able to block with ℓ since (i) it prefers ℓ (ii) it will not affect the match of any family with a higher priority in ℓ . Therefore, any contract (f, ℓ') for any $\ell \succ_f \ell'$ can be removed without affecting the stable outcome. Second, any set of families that has a guarantee for ℓ ($G_{(f,\ell)}$) and which has ℓ as its top-choice locality (\overline{F}_ℓ) must be matched to ℓ in any stable outcome. Third, if a family f cannot be accommodated in ℓ alongside $\overline{F}_\ell \cap \hat{G}_{(f,\ell)}$ and any subset of families without a guarantee that has a higher priority than f in ℓ , then f cannot be part of a stable outcome because (f, ℓ) can never be a blocking pair. Hence, (f, ℓ) can be removed without affecting any stable outcome. Since none of the possible stable outcomes have been affected by the TDBU algorithm, if all the families receive their top-choice locality among the remaining contracts and this outcome is feasible, then the TDBU algorithm produces a unique (because each family's top choice locality is unique given any remaining set of contracts), stable (because every family gets its top choice and does not wish to block), and family-optimal (because among the remaining contracts that can be part of a stable outcome, this one gives each family its top choice).⁴⁷ Finally, note that the ATDBU algorithm runs in exactly the same way as the TDBU algorithm except that it checks for consistency between guarantees and rejections as well as artificial guarantees and artificial rejections. \square

Define a *solution tree* as a directed graph (see example of a solution tree in Figure 2). Each node represents a set of artificial guarantees. The root of the tree has no artificial guarantees. Nodes in the first level contain artificial guarantees for all localities for f_1 ,

⁴⁷Note that it may not be Pareto-efficient because contracts were removed according to the stability criterion and some of these could have led to a Pareto improvement.

nodes in the second level contain artificial guarantees for all localities for f_2 given artificial guarantee for f_1 , and so on. The bottom of the solution tree has artificial guarantees for all localities for family $f_{|F|}$ given the artificial guarantees of all the other families. The DFS therefore searches through this solution tree.

Lemma 2. *The DFS reaches the bottom of the solution tree with the ATDBU algorithm at the final step reporting a non-empty outcome if and only if there is stable outcome.*

Proof. (\Rightarrow): If the DFS reaches the bottom of the solution tree, then every family has an artificial guarantee. Moreover, the ATDBU algorithm has not rejected any artificial guarantees. The outcome must be feasible otherwise infeasible contracts would necessarily be rejected. The outcome is stable because at each step of the algorithm every family's artificial guarantee was its top choice conditional on the remaining contracts. Since contracts are only removed as the DFS moves towards the bottom of the solution tree, no new blocking opportunities can arise.

(\Leftarrow): Suppose there is a stable outcome. Let us make a trivial observation. Any stable outcome can be constructed in the following way: For every family f matched to locality ℓ , remove all contracts that include localities that f prefers to ℓ and give each family its top-choice locality (e.g. as artificial guarantee). Therefore, if an attempt to give every family its top choice at any step of the DFS results in a feasible outcome, it must be stable because the ATDBU algorithm never removes contracts that could be part of a stable outcome. Therefore, any stable outcome is a sequence of artificial guarantees for every family that are not rejected. This sequence only occurs at the bottom of the solution tree. \square

Lemma 3. *There is no stable outcome if and only if for any family an artificial guarantee for every $L \cup \{\emptyset\}$ is rejected.*

Proof. (\Leftarrow): If family f 's artificial guarantee for locality ℓ has been rejected, then, by definition, f cannot be placed in ℓ in any stable outcome. Let us artificially reject f from ℓ i.e. remove (f, ℓ) . If the ATDBU algorithm returns an empty outcome, that means in any stable outcome f cannot be placed in any outcome other than ℓ (including being unmatched). This means that f cannot be placed in any locality or be unmatched in any stable outcome.

(\Rightarrow): Suppose, towards a contradiction, there is no stable outcome, but there is a family f_1 for which an artificial guarantee has not been rejected for some $\ell \in L \cup \{\emptyset\}$ in the DFS. Start the DFS by giving f_1 an artificial guarantee of ℓ and making it the first family in the general priority order. But note that the DFS either stops at the bottom of the solution tree or returns to f_1 . We have assumed that this artificial guarantee cannot be rejected therefore the DFS can only stop at the bottom of the solution tree. Hence, since the DFS only stops at the bottom of the solution tree if it finds a stable outcome, we have a contradiction. \square

Lemma 4. *The DFS finds a stable undominated outcome if a stable outcome exists.*

Proof. Since a stable outcome exists, the DFS must stop at the bottom of the solution tree. Since the DFS can access any outcome in the solution tree, the general priority order is not relevant for whether a stable outcome is found. Index families in general priority with: f_1, f_2, \dots . Note that the DFS artificially guarantees f_1 's top choice locality for as long as possible and only rejects it when it shows that f_2 cannot be matched to *any* locality given

f_1 's top choice. Then the DFS gives every f_i its best possible choice as artificial guarantee conditional on all subsequent f_j ($j > i$) not having their best possible artificial guarantee rejected. Hence, for any family any stable outcome found by the DFS must be at least as another outcome found when this family is lower in the general priority order. Therefore, any outcome found by the DFS cannot be Pareto-dominated by another stable outcome found by the DFS. \square

Housing assignment does not affect the stability or undominatedness of the outcome. The (\Rightarrow) part of the Proposition follows from Lemma 4. The (\Leftarrow) part of the Proposition follows from Lemma 3 and Lemma 2 and the fact that DFS either stops at the bottom of the solution tree or rejects the artificial guarantee of the least preferred locality of f_1 , the first family in the general priority order. \square

Proof of Proposition 6. By construction, in each round of the PFDA algorithm, a family is rejected if another family with a higher priority is rejected or has been rejected before. It follows that by the time the algorithm ends, any family permanently matched to a given locality has a higher priority for that locality than any family that was previously rejected. The outcome Y^* reached by the time the algorithm ends is consequently quasi-stable. It remains to show that Y^* dominates all other quasi-stable outcomes.

Suppose that there exists a quasi-stable outcome Y' such that, for some f , we have $f \in \mathbf{F}_{\ell'}(Y')$ and $f \in \mathbf{F}_{\ell}(Y^*)$ with $\ell' \succ_f \ell$. By construction, f was rejected by ℓ' in some Round r of the PFDA algorithm.

Suppose now the existence of a family f_i and a locality ℓ_i such that $f_i \in \mathbf{F}_{\ell_i}(Y^*)$ and ℓ_i rejected f_i in Round $1 \leq r_i \leq r$ of the PFDA algorithm. Consider all families that proposed to ℓ_i in at least one round between 1 and r_i and have a higher priority for ℓ_i than f_i . The fact that f_i was rejected in Round r_i implies that it cannot be accommodated at ℓ_i alongside all these families. Feasibility then dictates that at least one of them, say f_{i+1} , must be matched to another (possibly null) locality: $f_{i+1} \in \mathbf{F}_{\ell_{i+1}}(Y')$ with $\ell_{i+1} \neq \ell_i$. Because $f_{i+1} \pi_{\ell_i} f_i$, quasi-stability in turn dictates that $\ell_{i+1} \succ_{f_{i+1}} \ell_i$, which implies that f_{i+1} proposed to ℓ_{i+1} in the PFDA algorithm before he proposed to ℓ_i . It follows that ℓ_{i+1} rejected f_{i+1} in Round $r_{i+1} < r_i$.

By induction, there exists a family f_j and a locality ℓ_j such that $f_j \in \mathbf{F}_{\ell_j}(Y')$ and ℓ_j rejected f_j in Round 1. Family f_j cannot be accommodated at ℓ_j alongside all families proposing to ℓ_j in Round 1 who have a higher priority. Quasi-stability then dictates that at least one of these families be matched to a locality they prefer to ℓ_j , a contradiction since families propose to their top choice in Round 1. \square

Proof of Proposition 7. Let us set up the technical machinery following Roth and Rothblum (1999) and (Ehlers, 2008). Recall that \succ_f is the set of all preference profiles of family f and $\succ \equiv \times_{f \in F} \succ_f$ is the preference domain. For a $f \in F$, denote $\succ_{-f} \equiv \times_{f \in F \setminus \{f\}} \succ_f$. A

random preference profile is a probability distribution $\tilde{\succ}_{-f}$ over \succ_{-f} . A random outcome \tilde{Y} is a probability distribution over the set of all feasible outcomes \mathcal{Y} . Let $\tilde{Y}(f)$ denote the distribution which \tilde{Y} induces over the set of all f 's feasible contracts $\{Y_f | Y \in \mathcal{Y}\}$. Given a mechanism ϕ and $\succ_f \in \succ_f$, each randomized preference profile $\tilde{\succ}_f$ induces a random outcome $\phi(\succ_f, \tilde{\succ}_{-f})$ in the following way: for all $Y \in \mathcal{Y}$

$$\Pr\{\phi(\succ_f, \tilde{\succ}_{-f}) = Y\} = \sum_{\succ_{-f} \in \mathcal{Y}_{-f}: \phi(\succ_f, \succ_{-f}) = Y} \Pr\{\tilde{\succ}_{-f} = \succ_{-f}\}$$

Given $f \in F$, $\succ_f, \succ'_f, \succ''_f \in \mathcal{Y}_f$, and a random preference profile $\tilde{\succ}_{-f}$, we say that a strategy \succ'_f *stochastically \succ_f -dominates* the strategy \succ''_f if for all $\ell \in L \cup \{\emptyset\}$, denoted

$$\mathbf{l}(\phi(\succ'_f, \tilde{\succ}_{-f})_f) \succ_{\succ_f} \mathbf{l}(\phi(\succ''_f, \tilde{\succ}_{-f})_f)$$

we have

$$\Pr\{\mathbf{l}(\phi(\succ'_f, \tilde{\succ}_{-f})_f) \succeq_{\succ_f} \ell\} \geq \Pr\{\mathbf{l}(\phi(\succ''_f, \tilde{\succ}_{-f})_f) \succeq_{\succ_f} \ell\}.$$

Given $f \in F$, $\succ_f \in \mathcal{Y}_f$ and $\ell, \ell' \in L$, let $\succ_f^{\ell \leftrightarrow \ell'}$ denote f 's preference list that changes the positions of ℓ and ℓ' and leaves the other positions of other localities in \succ_f unchanged. Also, for any $\succ \in \mathcal{Y}$, let $\succ^{\ell \leftrightarrow \ell'} = (\succ_f^{\ell \leftrightarrow \ell'}, \succ_{-f}^{\ell \leftrightarrow \ell'})$. Given an outcome $Y \in \mathcal{Y}$, let $Y^{\ell \leftrightarrow \ell'}$ be the outcome in which ℓ and ℓ' exchange the families matched to it under Y .

Definition 5 (Anonymity). For all $\succ \in \mathcal{Y}$, all $Y \in \mathcal{Y}$, and all $\ell, \ell' \in L$, if $\phi(\succ) = Y$, then $\phi(\succ^{\ell \leftrightarrow \ell'}) = Y^{\ell \leftrightarrow \ell'}$.

Definition 6 (Positive association). For all $\succ \in \mathcal{Y}$, $f \in F$, and all $\ell, \ell' \in L$, if $\mathbf{l}(\phi(\succ)_f) = \ell$, and $\ell' \succ_{\succ_f} \ell$, then $\mathbf{l}(\phi(\succ_f^{\ell \leftrightarrow \ell'}, \succ_{-f})_f) = \ell$.

We then say that family f 's information $\tilde{\succ}_{-f}$ is $\{\ell, \ell'\}$ -*symmetric* if for every profile \succ_{-f} , both \succ_{-f} and $\succ_{-f}^{\ell \leftrightarrow \ell'}$ are equally probable. Family f 's information is *completely symmetric* if $\tilde{\succ}_{-f}$ is symmetric for any two localities.

Lemma 5. *The PFDA algorithm satisfies positive association.*

Proof of Lemma 5. Consider a family f that is permanently matched to locality ℓ (not its top choice, otherwise the result is trivial) under a report of \succ_f in the PFDA algorithm. This means that it can be accommodated alongside other families that proposed to ℓ before it and it had a high enough priority to ensure it was not rejected because a higher priority family than f was rejected. Now for some ℓ' such that $\ell' \succ_f \ell$, consider submitting list $\succ_f^{\ell \leftrightarrow \ell'}$. All proposals before f applies to ℓ are the same. Consider what happens when f proposes to ℓ . Clearly, it can be accommodated in ℓ since in this round fewer families had proposed to ℓ . Moreover, since fewer families had proposed, no family with a higher priority who has been rejected from ℓ could have proposed. Therefore, f will not be rejected. But in later rounds, there will be weakly fewer higher priority families that apply to ℓ . Therefore, f cannot be rejected from ℓ before the end of the algorithm. \square

Since in addition to positive association PFDA algorithm also satisfies anonymity, the Proposition 7 holds as a corollary of Theorem 3.1 in Ehlers (2008). \square

Proof of Proposition 8. First, we prove a lemma that establishes important properties of the algorithm.

Lemma 6. Consider a locality ℓ and a subset of families \tilde{F} that are tentatively accepted by ℓ after some round of the MRDA algorithm and a family f that is the only family other than those in \tilde{F} to propose to ℓ in the same round.

- (i) If f is rejected, then it has a lower priority than all families in \tilde{F} and all families in \tilde{F} continue proposing.
- (ii) If f is not rejected, at most one family in \tilde{F} is rejected. That family has a lower priority than f and than any other family in \tilde{F} .
- (iii) If f is not rejected without triggering a rejection, then all families in \tilde{F} would continue proposing if any other family were to propose to ℓ instead of f .

Let us define the *Current Rank* of family f for locality ℓ in Round k , denoted $CR_{f,\ell}^k$, to be the number of families with a higher priority than f proposing to ℓ in Round k plus 1. The Current Rank then corresponds to the family's relative priority among those proposing to ℓ in Round k . Let us abbreviate Maximum Rank to MR .

Proof of Lemma 6. (i) If f is rejected in some Round k , then $CR_{f,\ell}^k > MR_{f,\ell}$. Any other proposing family f' with a lower priority such that $CR_{f',\ell}^k > CR_{f,\ell}^k$ and $MR_{f',\ell} < MR_{f,\ell}$ and is also rejected.

(ii) The result is trivial if $|\tilde{F}| \leq 1$. Otherwise, again let f' and f'' be the families with respectively the lowest and n^{th} ($2 \leq n \leq |\tilde{F}|$) lowest priority among all families in \tilde{F} . Before f proposes, $CR_{f',\ell} \leq MR_{f',\ell}$ since f' is not rejected. As f'' has $n - 1$ fewer proposing families with a higher priority than f' does, $CR_{f'',\ell} = CR_{f',\ell} - (n - 1) \leq MR_{f',\ell} - (n - 1) \leq MR_{f'',\ell} - (n - 1)$. When f proposes, f'' 's Current Rank goes up by one if $f \pi_\ell f''$ and remains the same otherwise, therefore it does not exceed $MR_{f'',\ell} - (n - 2) \leq MR_{f'',\ell}$ and f'' is tentatively accepted. It follows that no family other than f' is rejected. As $CR_{f',\ell} \leq MR_{f',\ell}$, f' may only be rejected if $f \pi_\ell f'$.

(iii) The result is trivial if $|\tilde{F}| = 0$. Otherwise again let f' be the family with the lowest priority among all families in \tilde{F} . Before f proposes, $CR_{f',\ell} \leq MR_{f',\ell}$ since f' is accepted. If $CR_{f',\ell} = MR_{f',\ell}$, f is rejected unless $f \pi_\ell f'$ but in that case f' 's current rank goes up by one so f' is rejected. Therefore if f is tentatively accepted without triggering a rejection, $CR_{f',\ell} \leq MR_{f',\ell} - 1$. Then if any family proposes to ℓ , f' 's current rank reaches at most $MR_{f',\ell}$. Family f' is not rejected and, by Part (ii), neither is any other family in \tilde{F} . \square

We now prove each property in turn.

Quasi-stability

Suppose, towards a contradiction, that the outcome is infeasible. Then, in the last round, there is a locality that tentatively accepts a subset of n families that it cannot accommodate. Consider the family with the last priority among that subset. That family's Current Rank is n but its Maximum Rank is at most $n - 1$ since there exists a subset of $n - 1$ families with which it cannot be accommodated. That family could not have been tentatively accepted in the last round of the algorithm, a contradiction.

Suppose now that the outcome Y allows a (quasi-)blocking pair, that is there exists $f \in \mathbf{F}_{\ell'}(Y)$ and $f' \in \mathbf{F}_\ell(Y)$ such that $\ell \succ_f \ell'$ and $f \pi_\ell f'$. By construction, $MR_{f,\ell} \geq MR_{f',\ell}$.

Since f prefers ℓ to ℓ' and is matched to ℓ' , it proposes to and is rejected by ℓ in some round k . In that round at least $MR_{f,\ell}$ families with a priority higher than f (and hence f') propose to ℓ . If f' is already proposing to ℓ , it is rejected. The fact that $f' \in \mathbf{F}_\ell(Y)$ implies that f' proposes to ℓ for the first time in some round $t > k$ and that, in this round, strictly fewer than $MR_{f',\ell}$ families propose to ℓ . This requires that strictly fewer than $MR_{f',\ell}$ families with a priority higher than f' are tentatively accepted by ℓ in Round $t - 1$. It follows that for some $k \leq k' < t$, ℓ tentatively accepts strictly fewer families with a higher priority than f' in Round $k' + 1$ than in Round k . We next show that this is impossible.

Suppose, towards a contradiction, that $n \geq MR_{f',\ell}$ families with a higher priority than f' are tentatively accepted by ℓ in Round k' . In Round $k' + 1$, $m \geq 0$ new families propose to ℓ so that in total $m + n$ families with a higher priority than f' propose to ℓ but only $p < n$ are accepted. Let \tilde{f} be the family with the $p + 1^{\text{st}}$ priority among these $m + n$ proposing families. Then $CR_{\tilde{f},\ell}^{k'+1} = p + 1 > MR_{\tilde{f},\ell}$ since \tilde{f} is rejected. Let \hat{f} be the family with the $p + 1^{\text{st}}$ priority among the n families proposing to ℓ in Round k' . Then $CR_{\hat{f},\ell}^{k'+1} = p + 1 \leq MR_{\hat{f},\ell}$ since \hat{f} is tentatively accepted. It follows that $MR_{\hat{f},\ell} > MR_{\tilde{f},\ell}$, however since the n families tentatively accepted in Round k all propose again in Round k' , either $\tilde{f} = \hat{f}$ or $\tilde{f} \pi_\ell \hat{f}$ hence $MR_{\hat{f},\ell} \leq MR_{\tilde{f},\ell}$, a contradiction. As the outcome is feasible and does not allow any (quasi-)blocking pairs, it is quasi-stable.

Strategy-proofness

We begin by proving two lemmas that, combined, yield the desired result.

Lemma 7. *Fix the capacity constraints, priorities, and reports by other families, and let $\ell \in L \cup \{\emptyset\}$ be the locality to which family f is matched if it reports \succ_f . If instead f reports \succ'_f where ℓ is listed as its first preference, then f is still matched to ℓ and all other families are weakly better off.*

Proof of Lemma 7. Let Y and Y' be the outcomes generated by the MRDA algorithm when f reports \succ_f , respectively \succ'_f . Y respects Maximum Ranks when f reports \succ'_f since the only difference is that f no longer reports preferring any locality to ℓ . By Lemma 6, Y' dominates Y given f 's new report. Since f listed ℓ first it remains matched to ℓ and since all other families kept the same report, they are all matched to either the same locality or to one they prefer more. \square

Let $\ell \in L \cup \{\emptyset\}$ be the locality to which f is matched if it reports its true preferences. An important consequence of Lemma 7 is that f can successfully misrepresent its preferences if and only if there exists $\ell' \succ_f \ell$ such that f is matched to ℓ' if it reports preference list $(\ell' \succ \emptyset)$.

For ease of exposition of the second lemma, we consider a slightly altered but equivalent algorithm to the MRDA algorithm. First, we run the MRDA algorithm pretending that f reported all localities to be unacceptable. Once an outcome is reached, f proposes to the locality listed first on its actual report. If the proposal is rejected, f proposes to its second choice. If it is tentatively accepted and no family is rejected as a result, the algorithm ends and the tentative outcome becomes final. If it is tentatively accepted and another family is rejected, a rejection chain starts. The rejected family proposes to other localities until one

accepts it. Another family may be rejected as a result and so on until either a family is accepted by a locality without triggering a rejection or a family is accepted by the locality to which f proposed and f is rejected as a result. In the former case, the algorithm ends and the tentative acceptances becomes permanent matches. In the latter case, f proposes to its second choice. The same process is followed for each of f 's choices, the algorithm always ends eventually as f proposes to the null object if it has been rejected by all acceptable localities. Any locality that rejects f in the MRDA algorithm rejects f in this *alternative MRDA algorithm* and vice versa so that same outcome is reached. Letting f propose after everyone else allows us to isolate the impact that f has on the outcome of the MRDA algorithm. We are now in a position to present the result.

Lemma 8. *Suppose f proposes to some family ℓ in the second part of the alternative MRDA algorithm and is matched to it. Then if f proposes to ℓ' instead, but it is rejected from ℓ' and subsequently proposes to ℓ , then it is matched to ℓ .*

Proof of Lemma 8. Consider first the case where f proposes to ℓ . By assumption, f is tentatively accepted and the rejection chain that follows ends when a family is tentatively accepted by a locality without triggering an additional rejection. The rejection chain is

$$f_0, \ell_0, f_1, \ell_1, f_2, \ell_2, f_3, \ell_3, \dots, f_n, \ell_n, \quad \text{where } f_0 \equiv f \text{ and } \ell_0 \equiv \ell \text{ and } n = 0, 1, 2, \dots$$

If no rejection is triggered the rejection chain ends here (in that case $n = 0$). If a rejection is triggered, Lemma 6 implies that only one family, say f_1 , is rejected. Family f_1 proposes to its next best choices in order of preference until a locality, say ℓ_1 , accepts it. Again another family, f_2 may be rejected and so on until a family f_n proposes to ℓ_n and is tentatively accepted without triggering a rejection. The algorithm ends and the tentative acceptances becomes permanent matches. This is the outcome that would have been found if we had run the MRDA algorithm with f reporting ℓ as its first preference. Observe that some families and localities may appear more than once in the rejection chain, however f does not appear again since by assumption ℓ does not reject it.

Consider now the case where f proposes to ℓ' . By assumption, f is rejected, which can happen in one of two ways. First, f may be immediately rejected by ℓ' when it proposes to it. In that case f 's proposal has not modified the outcome: If f then proposes to ℓ the same rejection chain occurs and f is permanently matched to ℓ . Second, f is tentatively accepted by ℓ' but the rejection chain that follows ends with ℓ' rejecting f . By Lemma 6, ℓ_n cannot appear in that rejection chain because if any family proposes to ℓ_n it is accepted without triggering a rejection. In that case the rejection chain ends without ℓ' rejecting f .

After f is rejected by ℓ' , it proposes to ℓ . Suppose it is tentatively accepted at first but triggers a rejection chain that ends with ℓ rejecting f . Then ℓ_n is not part of that rejection chain otherwise the chain ends without ℓ rejecting f . Therefore, irrespective of whether f is rejected immediately or after a rejection chain, ℓ_n has not received any proposal following f 's proposals to ℓ and ℓ' . It follows that f_n has not proposed to f_n and is tentatively matched to a locality it prefers to ℓ_n . Therefore if f reports $\ell \succ_f \emptyset$, f_n is permanently matched to ℓ_n but if f reports $\ell' \succ_f \ell \succ_f \emptyset$, f_n is matched to a locality it prefers, contradicting Lemma 7. □

We finally combine Lemmas 7 and 8 in order to obtain the desired result. Let f 's true preferences be $\ell^1 \succ_f \ell^2 \succ_f \dots \succ_f \ell^n$ with $\ell^n \equiv \emptyset$ and let ℓ^i ($i = 1, \dots, n$) be the locality to which f is matched if it reports truthfully. Suppose f can successfully misrepresent its preferences. Then, by Lemma 7, there exists $\ell \succ_f \ell^i$ such that f is matched to ℓ if it reports $\ell \succ_f \emptyset$. (This implies that $i \geq 2$ and $\ell \neq \emptyset$.) Let j be the index of the locality f prefers among those it can obtain by successfully misrepresenting its preferences. That is, the best f can do by misrepresenting its preferences is obtain ℓ^j and this can be achieved by reporting $\ell^j \succ_f \emptyset$.

If instead f reports $\ell^{j-1} \succ_f \ell^j \succ_f \emptyset$, Lemma 8 implies that f is matched to ℓ^j if it is rejected by ℓ^{j-1} . Lemma 7 implies that f is rejected by ℓ^{j-1} since by assumption it cannot be obtained by any successful misrepresentation. Consequently, f is matched to ℓ^j if it reports $\ell^{j-1} \succ_f \ell^j \succ_f \emptyset$. Suppose that, for some $k < j$, f is matched to ℓ^j when it reports $\ell^k \succ_f \dots \succ_f \ell^j \succ_f \emptyset$. If f instead reports $\ell^{k-1} \succ_f \ell^k \succ_f \dots \succ_f \ell^j \succ_f \emptyset$, Lemma 7 implies that ℓ^{k-1} rejects f since, by assumption, it cannot be obtained by any successful misrepresentation. Lemma 7 also implies that $\ell^k, \dots, \ell^{j-1}$ all reject f since they reject f when f reports $\ell^k \succ_f \dots \succ_f \ell^j \succ_f \emptyset$. Lemma 8 implies that ℓ^j accepts any proposal of f since it does when f reports $\ell^k \succ_f \dots \succ_f \ell^j \succ_f \emptyset$. It follows that f is permanently matched to ℓ^j if it reports $\ell^{k-1} \succ_f \ell^k \succ_f \dots \succ_f \ell^j \succ_f \emptyset$. By induction, f is permanently matched to ℓ^j when it reports $\ell^1 \succ_f \ell^2 \succ_f \dots \succ_f \ell^j \succ_f \emptyset$. As every family only proposes to the locality to which it is eventually permanently matched and the ones listed higher on its report, the localities f lists below ℓ^j do not have any impact on the outcome. Then f is matched to ℓ^j when it reports its true preferences $\ell^1 \succ_f \ell^2 \succ_f \dots \succ_f \ell^j \succ_f \dots \succ_f \emptyset$, a contradiction. \square

C The Top Choice algorithm

C.1 The Top-Down Bottom-Up (TDBU) algorithm

Example

In addition to the set-up of the [Example](#) and family preferences in the [Example for Section 5.1](#), we use the priorities of localities introduced in the [Example for Section 6.2.1](#)

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 3. Families are initially listed in order of priority for each locality, as can be seen in the top panel of Table 3. The second column indicates how each family ranks the locality in its preference list. We denote by ‘‘T’’ whether the family considers this locality to be its top choice.

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.) Family f_2 gets a guarantee because it is first in the priority list 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, ℓ_2) and (f_2, ℓ_4) are eliminated and f_2 stops contesting ℓ_2 and ℓ_4 . This is denoted by a ‘‘X’’ in the fourth columns of the rows devoted to f_2 and ℓ_2 , respectively ℓ_4 . Capacities for both services in locality ℓ_1 are sufficient to accommodate both f_1 and f_2 , however both families can only be assigned to h_{11} . As a result, it is impossible to match both f_1 and f_2 to locality ℓ_1 and f_1 does not get a guarantee. Family f_1 does not get rejected

Algorithm 6: TOP-DOWN BOTTOM-UP ALGORITHM

Remove unacceptable contracts: $Y_{(1,1)} := \{(f, \ell) \in F \times (L \cup \{\emptyset\}) \mid \ell \succ_f \emptyset\}$. Arbitrarily label the localities such that $L \equiv \{\ell_1, \ell_2, \dots, \ell_{|L|}\}$.

Round $i \geq 1$ (Locality round): We use subscripts modulo $|L|$.

Step 0:

If $i > |L|$ and $Y_{(i,j)} = Y_{(i-|L|,j)}$, the algorithm terminates and yields $\phi(Y) := Y_i$.

Else consider locality ℓ_i .

Step $j \geq 1$ (Family round):

Given $Y_{(i,j)}$, let F_{ℓ_i} be the set of families which can be accommodated in ℓ_i .

If $F_{\ell_i} \setminus \bigcup_{k \in \{1, \dots, j-1\}} \{f_{(i,k)}\} \neq \emptyset$, let $f_{(i,j)}$ be the family with the highest priority for ℓ_i within that set.

If $f_{(i,j)}$ receives a guarantee for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell) \in Y_{(i,j)} \mid \ell \in L \cup \{\emptyset\} \text{ s.t. } \ell \prec_{f_{(i,j)}} \ell_i\}$$

Else if $f_{(i,j)}$ receives a rejection for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell_i)\}$$

Else maintain $Y_{(i,j+1)} := Y_{(i,j)}$.

The algorithm continues to Step $j + 1$.

Else update $Y_{(i,j+1)} := Y_{(i+1,1)}$ and go to Round $i + 1$.

either since ℓ_1 is not f_2 's top choice. In contrast, f_4 does get a guarantee since it is possible to assign h_{11} to either f_2 or f_1 and h_{12} to f_4 without violating ℓ_1 's capacity constraint. Family f_4 stops contesting ℓ_2 and ℓ_3 , its third and fourth choices, as a result. Family f_5 does not get a guarantee since both houses will be used in the case both f_2 and f_4 are matched to ℓ_1 . It does not get a rejection either as it is possible that none of f_2 , f_1 , and f_4 end up matched to ℓ_1 , in which case f_5 could get matched to ℓ_1 . Family f_3 does not get a guarantee since it cannot be matched to 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 matched to ℓ_1 unless at least one of f_2 or f_4 is matched to ℓ_1 . As f_3 cannot be matched to ℓ_1 alongside either of these families, there is no situation where f_3 ends up in ℓ_1 without violating stability. Family f_3 stops contesting ℓ_1 and ℓ_2 becomes its third choice. The algorithm continues in Round 2, which focuses on ℓ_2 .

Round 2: We now look at ℓ_2 . Family f_5 gets a guarantee since it can be accommodated, consequently it will no longer contest ℓ_3 and ℓ_4 . Family f_1 is rejected because it cannot be assigned the only house in ℓ_2 . Family f_3 receives neither a guarantee nor a rejection since h_{21} may or may not be assigned to f_5 . Family 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: Family f_3 gets a guarantee at ℓ_3 but f_2 does not get a guarantee there since accommodating both families need three units of service s_2 and ℓ_3 only has two units available. Family f_2 is however not rejected since f_3 may give up its priority for ℓ_3 if it is matched to ℓ_4 . Family f_1 can be accommodated alongside f_3 but not alongside f_2 , as a result f_1 will be matched to ℓ_3 if and only if f_3 is. Family 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 . 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:

$$Y^{TDBU} = \{(f_1, \ell_3), (f_1, \ell_4), (f_2, \ell_1), (f_2, \ell_3), (f_3, \ell_3), \\ (f_3, \ell_4), (f_4, \ell_1), (f_4, \ell_4), (f_5, \ell_1), (f_5, \ell_2)\}$$

Since f_1 and f_2 cannot be accommodated together at ℓ_3 and f_3 and f_4 cannot be accommodated together at ℓ_4 , permanently matching all families to their top-choice localities 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.

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	X
f_5	T		f_4	3 rd	X	f_1	T		f_4	T	
f_3	3 rd	R X	f_2	3 rd	X	f_4	4 th	X	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	X	f_1	2 nd	
f_1	3 rd		f_1	4 th	R X	f_3	2 nd		f_5	3 rd	X
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	X	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	X				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 3: The TDBU algorithm: Example.

C.2 The Augmented Top-Down Bottom-Up (ATDBU) algorithm

Algorithm 7: THE AUGMENTED TOP-DOWN BOTTOM-UP (ATDBU) ALGORITHM

Let $Y_{(1,1)} := \tilde{Y}$. Define the sets of artificial guarantees G^A and artificial rejections R^A . Let $R_{(1,1)}^A := R^A$.

Round $i \geq 1$ (Locality Round): We use subscript modulo $|L|$.

Step 0:

If $i > |L|$, $R_{(i,1)}^A = R_{(i-|L|,1)}^A$ and $Y_{(i,j)} = Y_{(i-|L|,j)}$, the algorithm terminates and yields

$$\tilde{\phi}(G^A, R^A, \tilde{Y}) := (G^A, R_{(i,1)}^A, Y_{(i,1)})$$

Else if consider locality ℓ_i and continue to Step 1.

Step $j \geq 1$ (Family Round)

Given $X_{(i,j)}$, let F_{ℓ_i} be the set of families which can be accommodated in ℓ_i .

If $F_{\ell_i} \setminus \bigcup_{k \in \{1, \dots, j-1\}} \{f_{(i,k)}\} \neq \emptyset$, let $f_{(i,j)}$ be the family with the highest priority for ℓ_i within that set.

If $f_{(i,j)}$ receives a guarantee for ℓ_i , then

If $f_{(i,j)}$ also receives an artificial rejection for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell) \in Y_{(i,j)} \mid \ell \in L \cup \{\emptyset\} \text{ s.t. } \ell \prec_{f_{(i,j)}} \ell_i \text{ or } \ell = \ell_i\}$$

$$R_{(i,j+1)}^A := R_{(i,j)}^A \cup \{(f_{(i,j)}, \ell_i)\}$$

Else update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell) \in Y_{(i,j)} \mid \ell \in L \cup \{\emptyset\} \text{ s.t. } \ell \prec_{f_{(i,j)}} \ell_i\}$$

$$R_{(i,j+1)}^A := R_{(i,j)}^A$$

Else if $f_{(i,j)}$ receives a rejection for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell_i)\}$$

$$R_{(i,j+1)}^A := R_{(i,j)}^A$$

Else if $f_{(i,j)}$ receives an artificial rejection for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell_i)\}$$

$$R_{(i,j+1)}^A := R_{(i,j)}^A \cup \{(f_{(i,j)}, \ell_i)\}$$

Else maintain $Y_{(i,j+1)} := Y_{(i,j)}$ and $R_{(i,j+1)}^A := R_{(i,j)}^A$

Else if $Y_{(i,j+1)} \cap (\{f_{(i,j)}\} \times (L \cup \{\emptyset\})) = \emptyset$, the algorithm terminates and yields $\tilde{\phi}(G^A, R^A, \tilde{Y}) := (G^A, R_{(i,j+1)}^A, \emptyset)$.

Else continue to Step $j + 1$.

AUGMENTED TOP-DOWN BOTTOM-UP (ATDBU) ALGORITHM
(CONT.)

Else if $(R_{(i,j)}^A \cap (F \times \{\ell_i\})) \setminus \bigcup_{k \in \{1, \dots, j-1\}} \{f_{(i,k)}\} \neq \emptyset$, let $f_{(i,j)}$ be the family with the highest priority for ℓ_i within this set.

If $f_{(i,j)}$ receives a guarantee for ℓ_i , then update

$$Y_{(i,j+1)} := Y_{(i,j)} \setminus \{(f_{(i,j)}, \ell) \in Y_{(i,j)} \mid \ell \in L \cup \{\emptyset\} \text{ s.t. } \ell \prec_{f_{(i,j)}} \ell_i\}$$

$$R_{(i,j+1)}^A := R_{(i,j)}^A \setminus \{(f_{(i,j)}, \ell_i)\}$$

Else if $f_{(i,j)}$ receives a rejection for ℓ_i , then maintain $Y_{(i,j+1)} := Y_{(i,j)}$ and update $R_{(i,j+1)}^A := R_{(i,j)}^A \setminus \{(f_{(i,j)}, \ell_i)\}$.

Else maintain $Y_{(i,j+1)} := Y_{(i,j)}$ and $R_{(i,j+1)}^A := R_{(i,j)}^A$

If $Y_{(i,j+1)} \cap (\{f_{(i,j)}\} \times (L \cup \{\emptyset\})) = \emptyset$, the algorithm terminates and yields $\tilde{\phi}(G^A, R^A, \tilde{Y}) := (G^A, R_{(i,j+1)}^A, \emptyset)$.

Else continue to Step $j + 1$.

Else let $Y_{(i+1,1)} := Y_{(i,j)}$ and $R_{(i+1,1)}^A := R_{(i,j)}^A$. The algorithm continues in Round $i + 1$.

C.3 Depth-First Search (DFS)

The DFS is described in the main text and below we show how it applies to our Example and state it formally.

Example

In addition to the set-up of the [Example](#) and family preferences in the [Example for Section 5.1](#), we use the priorities of localities introduced in the [Example for Section 6.2.1](#)

We now illustrate how the DFS 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 Outcome

Suppose 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 its top choice that is we look for a stable outcome where f_1 is matched to 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 4](#). The artificial guarantee is denoted by ‘‘AG’’ in the fourth column of the relevant row. Nothing of interest occurs in Rounds 1 and 2, f_2 and f_5 receive a guarantee for ℓ_1 and ℓ_2 , respectively, but this does not allow us to eliminate any contracts. The action begins in Round 3, which focuses in ℓ_3 .

Round 3: As a consequence of f_2 's artificial guarantee at its top choice ℓ_3 , f_2 receives an artificial rejection.

Algorithm 8: DEPTH-FIRST SEARCH

Round 0:

Run the TDBU algorithm to obtain $Y_1 = \phi(F \times (L \cup \{\emptyset\}))$. Let the families be ordered from 1 to $|F|$ so that $F \equiv \{f_1, f_2, \dots, f_{|F|}\}$. Let $G_1^A := \emptyset$ and $R_1^A := \emptyset$. Additionally, initialize the family index $k = 1$ and the state $c^1 := \textit{artificial guarantee}$.

Round $i \geq 1$:

If $c^i = \textit{artificial guarantee}$

Run the ATDBU algorithm to obtain $\tilde{\phi}(G_i^A \cup \{(f_k, \bar{\ell}_{f_k})\}, R_i^A, Y_i)$.

If $\tilde{\phi}(G_i^A \cup \{(f_k, \bar{\ell}_{f_k})\}, R_i^A, Y_i) = (\cdot, \cdot, \emptyset)$, then let $G_{i+1}^A := G_i^A$, $R_{i+1}^A := R_i^A$, $Y_{i+1} := Y_i$ and $c^{i+1} := \textit{artificial rejection}$. Proceed to Round $i + 1$.

Else let $(G_{i+1}^A, R_{i+1}^A, Y_{i+1}) := \tilde{\phi}(G_i^A \cup \{(f_k, \bar{\ell}_{f_k})\}, R_i^A, G_i^A)$.

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

Else increase k by 1 and set $c^i := \textit{artificial guarantee}$. Proceed to Round $i + 1$.

Else $c^i = \textit{artificial rejection}$

Run the ATDBU algorithm to obtain $\tilde{\phi}(G_i^A, R_i^A \cup \{(f_k, \bar{\ell}_{f_k})\}, Y_i \setminus \{(f_k, \bar{\ell}_{f_k})\})$.

If $\tilde{\phi}(G_i^A, R_i^A \cup \{(f_k, \bar{\ell}_{f_k})\}, Y_i \setminus \{(f_k, \bar{\ell}_{f_k})\}) = (\cdot, \cdot, \emptyset)$, then

If $k > 1$, let $(G_{i+1}^A, R_{i+1}^A, Y_{i+1}) := (G_j^A, R_j^A, Y_j)$, where j is the last round dealing with family f_{k-1} . Set $c^{i+1} = \textit{artificial rejection}$. Proceed to Round $i + 1$.

Else $k = 1$, the set of stable outcomes is empty. The algorithm terminates and yields $\psi(Y) := \emptyset$.

Else let $(G_A^{i+1}, R_A^{i+1}, X^{i+1}) := \tilde{\phi}(G_i^A, R_i^A \cup \{(f_{k-1}, \bar{\ell}_{f_{k-1}})\}, Y_i \setminus \{(f_{k-1}, \bar{\ell}_{f_{k-1}})\})$. Set $c^{i+1} := \textit{artificial guarantee}$. Proceed to Round $i + 1$.

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 capacity from the locality to prevent f from being matched 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 does not receive a normal rejection but cannot be matched to a locality because families with a *lower* priority have received an artificial guarantee. In the second panel of Table 4, f_2 is not rejected since it is possible that f_3 will be matched to ℓ_4 , however f_2 can only be matched to ℓ_3 if f_1 is not. Matching f_2 to 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 on ℓ_3 's list. The reason for this is that f_2 could still get a guarantee if f_3 were to be matched to 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 receives an artificial guarantee, which means it will no longer require the artificial guarantee in the following rounds. Family f_1 is removed from ℓ_4 priority list due to its artificial guarantee at ℓ_3 and ℓ_1 is not f_2 's top choice after the latter was artificially rejected from ℓ_3 .

Round 4: Family f_4 receives an artificial guarantee for ℓ_4 and is removed from ℓ_1 's list. Family f_3 receives an artificial rejection since it cannot be accommodated at ℓ_4 alongside f_4 and the latter will now be matched to ℓ_4 in any stable matching. Family f_3 is removed from ℓ_4 's list and ℓ_3 becomes its top choice.

Round 5: The algorithm continues in Round 5, which focuses on ℓ_1 . Family f_2 receives a guarantee and, because ℓ_1 cannot accommodate both families, f_5 receives a rejection. Locality ℓ_2 becomes f_5 's top choice as a result.

Round 6: 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.

Round 7: Family f_3 receives an artificial 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 matched to ℓ_3 , f_2 receives an artificial 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 .

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 outcome has been found: f_1 and f_3 are permanently matched to ℓ_3 , f_2 in ℓ_1 , f_4 in ℓ_4 and f_5 in ℓ_2 . Hence:

$$Y^{TCA} = \{(f_2, \ell_1), (f_5, \ell_2), (f_1, \ell_3), (f_3, \ell_3), (f_4, \ell_4)\}$$

Second Stable Undominated Outcome If f_4 is at the top of the general priority, the ATDBU algorithm operates in an almost analogous way. Family f_4 's artificial guarantee for ℓ_4 means that f_3 is rejected and ℓ_3 becomes its new top choice. Family 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,

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	X
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	X				f_2		AR	f_3	T	R X
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 X				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 X			
						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 4: DFS with f_1 on Top of the General Priority.

f_1 and f_3 are matched to l_3 , f_2 to l_1 , f_4 to l_4 and f_5 to 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 is rejected from l_1 , yielding an outcome where f_1 and f_3 are matched to l_4 , f_2 in l_3 , f_4 in l_1 and f_5 in l_2 . Compared to the first stable undominated outcome, 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.

$$Y^{\dagger TCA} = \{(f_4, \ell_1), (f_5, \ell_2), (f_2, \ell_3), (f_1, \ell_4), (f_3, \ell_4)\}$$

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

The two outcomes 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 TDBU 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 show what happens when f_5 is at the top of the list, as we do now. The algorithm is summarized in Table 5.

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 5: DFS with f_5 on Top of the General Priority.

Round 1: Family f_5 's artificial guarantee yields an artificial rejection for f_2 and f_4 since neither of them can be matched to ℓ_1 alongside f_5 . Recall that f_2 and f_4 receive an artificial rejection rather than a rejection because they are rejected due to being 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 . Family f_2 and locality ℓ_1 will form a blocking pair unless f_2 is matched to a locality it prefers to ℓ_1 . In the Example, ℓ_3 is the only option. This 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.

Round 2: Nothing occurs in Round 2 since ℓ_2 does not have any contract associated with it left.

Round 3: Family 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 . Family f_1 receives a rejection from ℓ_3 since it has a lower priority than f_2 and the two families cannot both be matched to ℓ_3 . Locality 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.

Round 4: Family f_1 receives a guarantee and, since ℓ_4 is its top choice, the fact that f_1 and f_4 cannot both be matched to ℓ_4 means that f_4 is rejected. Family f_3 then gets a guarantee since it can be accommodated alongside f_1 .

Round 5: Family f_4 gets a guarantee 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 matched to 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 matched to ℓ_1 .

The DFS continues by going back to the outcome of the TDBU algorithm and removing f_5 from ℓ_1 's list. Family 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 outcome is found if either f_1 or f_4 is second on the general priority order and the second stable outcome is found if it is either f_2 or f_3 .

D Additional examples of PFDA and MRDA algorithms

D.1 Another example to compare PFDA and MRDA algorithms

There are eight families, three localities and two services. The preferences, priorities and service needs and capacities are displayed below.

Families						
Preferences		s_1	s_2	Preferences		s_1 s_2
f_1	$\ell_2, \ell_1, \emptyset$	2	1	f_5	$\ell_1, \ell_3, \ell_2, \emptyset$	3 1
f_2	$\ell_3, \ell_1, \ell_2, \emptyset$	2	0	f_6	$\ell_1, \ell_2, \emptyset$	1 1
f_3	$\ell_2, \ell_3, \emptyset$	0	2	f_7	$\ell_1, \ell_2, \emptyset$	3 3
f_4	$\ell_3, \ell_2, \ell_1, \emptyset$	2	3	f_8	$\ell_1, \ell_3, \ell_2, \emptyset$	1 0

Localities					
Priorities				s_1	s_2
ℓ_1	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8$	4	5		
ℓ_2	$f_5, f_2, f_6, f_8, f_3, f_4, f_1, f_7$	6	5		
ℓ_3	$f_2, f_3, f_6, f_1, f_7, f_8, f_5, f_4$	8	7		

FPDA algorithm

The FPDA algorithm lasts three rounds, which are displayed below:

Round 1				Round 2				Round 3			
f_1	\rightarrow	ℓ_2	✓	f_1	\rightarrow	ℓ_2	✓	f_1	\rightarrow	ℓ_2	✓
f_2	\rightarrow	ℓ_3	✓	f_2	\rightarrow	ℓ_3	✓	f_2	\rightarrow	ℓ_3	✓
f_3	\rightarrow	ℓ_2	✓	f_3	\rightarrow	ℓ_2	✓	f_3	\rightarrow	ℓ_2	✓
f_4	\rightarrow	ℓ_3	✓	f_4	\rightarrow	ℓ_3	✓	f_4	\rightarrow	ℓ_3	✓
f_5	\rightarrow	ℓ_1	✓	f_5	\rightarrow	ℓ_1	✓	f_5	\rightarrow	ℓ_1	✓
f_6	\rightarrow	ℓ_1	✓	f_6	\rightarrow	ℓ_1	✓	f_6	\rightarrow	ℓ_1	✓
f_7	\rightarrow	ℓ_1	✗	f_7	\rightarrow	ℓ_2	✗	f_7	\rightarrow	\emptyset	✓
f_8	\rightarrow	ℓ_1	✗	f_8	\rightarrow	ℓ_3	✓	f_8	\rightarrow	ℓ_3	✓

In Round 1, ℓ_1 tentatively accepts f_5 and f_6 since they have joint service needs (4, 2). Adding f_7 increases the total service needs to (7, 4), which ℓ_1 cannot provide. Families f_7 and f_8 are consequently rejected. Localities ℓ_2 and ℓ_3 do not reject any family since f_3 and f_1 joint service needs (2, 3) while f_2 and f_4 have joint service needs (4, 3). In Round 2, ℓ_1 does not receive any new proposals and tentatively accepts f_5 and f_6 . Locality ℓ_2 receives a proposal from f_7 as well as f_3 and f_1 . In total, these families have service needs (5, 6), which exceeds the locality's provision of service s_2 . Family f_7 is rejected. ℓ_3 receives a new proposal from f_8 so that the total service needs of families proposing to it is (5, 3). All families are tentatively accepted. In Round 3, all families propose to the same locality except for f_7 , which has been rejected from all localities and does not make any proposal. As a consequence, all localities receive the same proposals as in Round 2 and no family is rejected. The algorithm terminates and yields the following outcome: $\{(f_1, \ell_2), (f_2, \ell_3), (f_3, \ell_2), (f_4, \ell_3), (f_5, \ell_1), (f_6, \ell_1), (f_7, \emptyset), (f_8, \ell_3)\}$.

MRDA algorithm

Phase 1

The Maximum Ranks are displayed below:

ℓ_1				ℓ_2				ℓ_3			
f_1	∞	f_5	1	f_5	∞	f_3	3	f_2	∞	f_7	∞
f_2	∞	f_6	1	f_2	∞	f_4	2	f_3	∞	f_8	4
f_3	∞	f_7	1	f_6	∞	f_1	2	f_6	∞	f_5	3
f_4	2	f_8	1	f_8	3	f_7	1	f_1	∞	f_4	2

At ℓ_1 , f_1 , f_2 and f_3 have joint service needs (4, 3). As ℓ_1 can provide (4, 4), all three families get a Maximum Rank of ∞ . Family f_4 cannot be accommodated along these three families as their needs together would be (6, 6). A subset of $\{f_1, f_2, f_3\}$ containing two families may need up to (4, 3). (This is calculated as follows. f_1 and f_2 are the two families needing the most units of s_1 . They need 2 units each, hence a total of 4. Families f_1 and f_3 are the two families needing the most units of s_2 . They need respectively 2 and 1 units, hence a total of 3 units. Therefore f_4 could have to be accommodated alongside a subset of size 2 that needs up to 4 units of s_1 or to one that needs up to 3 units of s_2 .) Adding the needs of f_4 yields (6, 6), which exceeds the provision of ℓ_1 . We conclude that there exists a subset of size 2 alongside which f_4 cannot be accommodated, hence its Maximum Rank

is at most 2. A subset of size 1 may need up to $(2, 2)$. (Family f_1 has the highest need for s_1 with 2 units and f_3 the highest need of s_2 with 2 units.) Adding f_4 's needs yields $(4, 5)$, which lies within ℓ_1 's provision. It follows that f_4 's Maximum Rank is 2. Family f_5 's Maximum Rank can be at most 2 since it cannot be larger than the one of a family that has a higher priority. As f_4 and f_5 cannot be accommodated together (they would need $(5, 4)$, which exceeds ℓ_1 's provision of s_1), f_5 's Maximum Rank is 1. The Maximum Rank of f_6 , f_7 and f_8 can be at most 1. Since all these families can be individually accommodated at ℓ_1 , they all get a Maximum Rank of 1.

At ℓ_2 , f_5 , f_2 and f_6 all get a Maximum Rank of ∞ as they can be accommodated together. Family f_8 cannot be accommodated alongside them as this would need 7 units of s_1 . A subset of $\{f_5, f_2, f_6\}$ containing two families would need at most $(5, 2)$. Adding f_8 's needs yields $(6, 2)$, which lies within ℓ_2 's provision. Family f_8 can therefore be accommodated alongside any subset of size 2 and its Maximum Rank is 3. Any subset of $\{f_5, f_2, f_6, f_8\}$ containing two families again needs at most $(5, 2)$. Adding f_3 's needs gives $(5, 4)$, hence f_3 's Maximum Rank is also 3. Any subset of $\{f_5, f_2, f_6, f_8, f_3\}$ containing two families requires $(5, 3)$. As f_4 needs $(2, 3)$ its Maximum Rank is at most 2. It is in fact exactly 2 since any individual family in $\{f_5, f_2, f_6, f_8, f_3\}$ needs at most 3 units of s_1 and 2 units of s_2 . Adding f_4 's needs yields $(5, 5)$, which does not exceed ℓ_2 's provision. The same can be said of f_1 , however the fact that f_7 needs 3 units of s_2 means it cannot be accommodated alongside f_4 , consequently its Maximum Rank is 1.

At ℓ_3 , all families in $\{f_2, f_3, f_6, f_1, f_7\}$ together need $(8, 7)$, which is exactly the locality's provision. All five families get a Maximum Rank of ∞ . Clearly, no other family can be accommodated alongside them so f_8 's Maximum Rank will be finite. A subset of size 4 may need up to $(8, 7)$ as well, consequently f_8 's Maximum Rank is at most 4. A subset of size 3 may on the other hand only need up to $(7, 6)$, as a result f_8 can be accommodated alongside all of them and its Maximum Rank is 4. Subsets of $\{f_2, f_3, f_6, f_1, f_7, f_8\}$ containing three families also need at most $(7, 6)$, however f_5 's needs are $(3, 1)$, which means ℓ_3 cannot provide enough units of s_1 to guarantee that f_5 can be accommodated alongside any such subset. Subsets containing two families may however only need up to $(5, 5)$, which means that f_5 can be accommodated alongside all of them. Family f_5 's Maximum Rank is 3. Finally, any subset of $\{f_2, f_3, f_6, f_1, f_7, f_8, f_5\}$ containing three families may need up to $(6, 5)$. Adding f_4 's needs yields $(8, 8)$, which exceeds ℓ_3 's provision of s_2 . Family f_4 can, however, be accommodated alongside any other individual family, hence its Maximum Rank is 2.

Phase 2

The second phase of the MRDA algorithm lasts two rounds, which are summarized below:

Round 1			Round 2			Round 3			Round 4		
f_1	\rightarrow	ℓ_2 ✓	f_1	\rightarrow	ℓ_2 ✗	f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓
f_2	\rightarrow	ℓ_3 ✓	f_2	\rightarrow	ℓ_3 ✓	f_2	\rightarrow	ℓ_3 ✓	f_2	\rightarrow	ℓ_3 ✓
f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓
f_4	\rightarrow	ℓ_3 ✓	f_4	\rightarrow	ℓ_3 ✗	f_4	\rightarrow	ℓ_2 ✗	f_4	\rightarrow	ℓ_1 ✓
f_5	\rightarrow	ℓ_1 ✓	f_5	\rightarrow	ℓ_1 ✓	f_5	\rightarrow	ℓ_1 ✗	f_5	\rightarrow	ℓ_3 ✓
f_6	\rightarrow	ℓ_1 ✗	f_6	\rightarrow	ℓ_2 ✓	f_6	\rightarrow	ℓ_2 ✓	f_6	\rightarrow	ℓ_2 ✓
f_7	\rightarrow	ℓ_1 ✗	f_7	\rightarrow	ℓ_2 ✗	f_7	\rightarrow	\emptyset ✓	f_7	\rightarrow	\emptyset ✓
f_8	\rightarrow	ℓ_1 ✗	f_8	\rightarrow	ℓ_3 ✓	f_8	\rightarrow	ℓ_3 ✓	f_8	\rightarrow	ℓ_3 ✓

The outcome generated by the MRDA algorithm is $\{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2), (f_4, \ell_1), (f_5, \ell_3), (f_6, \ell_2), (f_7, \emptyset), (f_8, \ell_3)\}$.

Five families, f_1 , f_4 , f_5 , f_6 , and f_7 are better off under the PFDA algorithm compared to their outcomes under the MRDA algorithm.

D.2 Two examples of manipulability of the PFDA algorithm

We now present two examples of manipulation of the PFDA algorithm, which illustrate sufficient conditions for the construction of a quasi-stable, strategy-proof mechanism.

First example of manipulability of the PFDA algorithm

Consider a matching problem with three families (f_1 , f_2 , and f_3) and three localities (ℓ_1 , ℓ_2 , and ℓ_3). Preferences and priorities are as follows:

$$\begin{array}{ll}
f_1 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_1 : f_1, f_2, f_3 \\
f_2 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_2 : f_1, f_3, f_2 \\
f_3 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_3 : f_1, f_3, f_2
\end{array}$$

There is only one service, of which families f_1 and f_2 need two units and f_3 one unit. ℓ_1 provides three units and each of ℓ_2 and ℓ_3 provides two. The matrices of service needs and capacities are displayed below:

$$\nu = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \quad \kappa = \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}$$

The PFDA algorithm lasts three rounds, which are summarized below:

Round 1			Round 2			Round 3		
f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓
f_2	\rightarrow	ℓ_1 ✗	f_2	\rightarrow	ℓ_2 ✗	f_2	\rightarrow	ℓ_3 ✓
f_3	\rightarrow	ℓ_1 ✗	f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓

In Round 1, all families propose to ℓ_1 as it is their first preference. Family f_1 is at the top of ℓ_1 's priority list and is tentatively accepted as it needs two units and three are available. Family f_2 needs two units, which brings the total demand to four, as only three units are available, f_2 is rejected. Family f_3 is also rejected since its priority is lower than f_2 's.

In Round 2, f_1 continues to propose to ℓ_2 and is tentatively accepted. Families f_2 and f_3 both propose to ℓ_2 and since two units are available, only the family with the higher priority, f_3 , is tentatively accepted. Family f_2 is rejected and proposes to ℓ_3 in Round 3. All families propose to a different locality and are accepted, the algorithm ends. The outcome is $Y^{PFDA} \equiv \{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2)\}$.

Suppose now that f_2 changes its report to $\ell_2 \succ_{f_2} \ell_1 \succ_{f_2} \ell_3 \succ_{f_2} \emptyset$ (or equivalently to $\ell_2 \succ_{f_2} \emptyset$). In the first Round, f_2 is the only family to propose to ℓ_2 and is tentatively accepted. Families f_1 and f_3 propose to ℓ_1 and are both tentatively accepted since ℓ_1 can provide three units of the service. As no family is rejected, the algorithm ends and yields $Y^{\dagger PFDA} \equiv \{(f_1, \ell_1), (f_2, \ell_2), (f_3, \ell_1)\}$. By misrepresenting its preferences, f_2 clinches ℓ_2 , which it prefers to ℓ_3 . The problem that occurs in the counterexample presented is that a family can trigger the rejection of another family while being itself rejected. In Round 1, f_2 is rejected by ℓ_1 but as a consequence of its proposal f_3 is also rejected. In Round 2, ℓ_2 rejects f_2 because f_3 is also proposing. Family f_2 could have avoided this situation by proposing to ℓ_2 first.

Second example of manipulability of the PFDA algorithm

We conclude this section by presenting another counterexample to illustrate another instance where a misrepresentation may also be beneficial. Consider a matching problem with five families and four localities. The preferences and priorities are as follows:

$$\begin{array}{ll}
 f_1 : \ell_1 \succ \ell_4 \succ \emptyset & \ell_1 : f_4, f_1, f_2, f_3, f_5 \\
 f_2 : \ell_1 \succ \ell_3 \succ \emptyset & \ell_2 : f_3, f_4, f_1, f_2, f_5 \\
 f_3 : \ell_1 \succ \ell_2 \succ \emptyset & \ell_3 : f_2, f_5, f_1, f_3, f_4 \\
 f_4 : \ell_2 \succ \ell_1 \succ \emptyset & \ell_4 : f_5, f_1, f_2, f_3, f_4 \\
 f_5 : \ell_3 \succ \ell_4 \succ \emptyset &
 \end{array}$$

There is only one service. The needs of each family and capacities of each locality are as follows:

$$\nu = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad \kappa = \begin{pmatrix} 2 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

That is, family f_1 needs two units and all other families need one unit each. Localities ℓ_1 and ℓ_4 can provide up to two units each while ℓ_2 and ℓ_3 can provide at most one unit each. The PFDA algorithm last five rounds, which are displayed below.

Round 1	Round 2	Round 3	Round 4	Round 5
$f_1 \rightarrow \ell_1 \checkmark$	$f_1 \rightarrow \ell_1 \checkmark$	$f_1 \rightarrow \ell_1 \times$	$f_1 \rightarrow \ell_4 \times$	$f_1 \rightarrow \emptyset \checkmark$
$f_2 \rightarrow \ell_1 \times$	$f_2 \rightarrow \ell_3 \checkmark$	$f_2 \rightarrow \ell_3 \checkmark$	$f_2 \rightarrow \ell_3 \checkmark$	$f_2 \rightarrow \ell_3 \checkmark$
$f_3 \rightarrow \ell_1 \times$	$f_3 \rightarrow \ell_2 \checkmark$	$f_3 \rightarrow \ell_2 \checkmark$	$f_3 \rightarrow \ell_2 \checkmark$	$f_3 \rightarrow \ell_2 \checkmark$
$f_4 \rightarrow \ell_2 \checkmark$	$f_4 \rightarrow \ell_2 \times$	$f_4 \rightarrow \ell_1 \checkmark$	$f_4 \rightarrow \ell_1 \checkmark$	$f_4 \rightarrow \ell_1 \checkmark$
$f_5 \rightarrow \ell_3 \checkmark$	$f_5 \rightarrow \ell_3 \times$	$f_5 \rightarrow \ell_4 \checkmark$	$f_5 \rightarrow \ell_4 \checkmark$	$f_5 \rightarrow \ell_4 \checkmark$

In Round 1, f_1 takes up two units of capacity at ℓ_1 , which means that any family with a lower priority is rejected. This affects f_2 and f_3 . In Round 2, f_2 takes up one unit of capacity

at ℓ_3 , which means that f_5 is rejected. Similarly, f_3 takes up one unit of capacity at ℓ_2 and f_4 is rejected. In Round 3, f_5 proposes to ℓ_4 where it does not have any competition. Family f_4 proposes to ℓ_1 , resulting in f_1 being rejected. Family f_1 proposes to ℓ_4 in Round 4 and is again rejected, this time because of f_5 . Family f_1 finally proposes to the null object in Round 5, resulting in everyone else being permanently matched. The algorithm ends and yields $Y^{PFDA} \equiv \{(f_1, \emptyset), (f_2, \ell_3), (f_3, \ell_2), (f_4, \ell_1), (f_5, \ell_4)\}$. Suppose now that f_1 misrepresents its preferences and reports $\ell_4 \succ_{f_1} \emptyset$. Every family is accepted in Round 1 and thus matched to its first (reported) preference: $Y^{\dagger PFDA} \equiv \{(f_1, \ell_4), (f_2, \ell_1), (f_3, \ell_2), (f_4, \ell_2), (f_5, \ell_3)\}$. Family f_1 is now matched to ℓ_4 . In this example, no family triggers a rejection without being tentatively accepted, however f_1 triggers two rejections by proposing to ℓ_1 . The rejection chain that follows has two branches. The first one, initiated by f_3 , leads to f_1 being rejected by ℓ_1 . The second one, initiated by f_2 , leads to f_5 proposing to ℓ_4 . This prevents f_1 from getting its second choice after being rejected by ℓ_1 .

D.3 Example of MRDA algorithm producing a family-optimal outcome

Let us return to the first example of manipulability of the PFDA algorithm in Appendix D.2. As a reminder, the preferences and priorities are

$$\begin{array}{ll} f_1 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_1 : f_1, f_2, f_3 \\ f_2 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_2 : f_1, f_3, f_2 \\ f_3 : \ell_1 \succ \ell_2 \succ \ell_3 \succ \emptyset & \ell_3 : f_1, f_3, f_2 \end{array}$$

The matrices of service needs and capacities are

$$\nu = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \quad \kappa = \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}$$

We now run the MRDA algorithm.

Phase 1

As every family can be accommodated at every locality all Maximum Ranks will be at least 1. Family f_1 has the highest priority for all localities and therefore gets a Maximum Rank of ∞ for all of them. At ℓ_1 , f_2 cannot be accommodated alongside f_1 since both families need two units and ℓ_1 can only provide 3. Family f_2 's Maximum Rank for ℓ_1 is 1 and, consequently, so is f_3 's. At ℓ_2 and ℓ_3 , f_1 and f_3 cannot be accommodated together so the Maximum Rank of both f_2 and f_3 is 1.

The Maximum Ranks are displayed below:

ℓ_1
f_1 ∞
f_2 1
f_3 1

ℓ_2
f_1 ∞
f_3 1
f_2 1

ℓ_3
f_1 ∞
f_3 1
f_2 1

Phase 2

All families propose to ℓ_1 in Round 1. Family f_1 is tentatively accepted but f_2 and f_3 are rejected since both of their Maximum Ranks are 1. Families f_2 and f_3 propose to ℓ_2 in Round 2. Again their Maximum Ranks are 1 so f_3 , which has a higher priority at ℓ_2 , is tentatively accepted and f_2 is rejected. Family f_2 proposes to ℓ_3 in Round 3 and is tentatively accepted. The algorithm ends and yields $Y^{MRDA} \equiv \{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2)\}$.

Round 1			Round 2			Round 3		
f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓
f_2	\rightarrow	ℓ_1 ✗	f_2	\rightarrow	ℓ_2 ✗	f_2	\rightarrow	ℓ_3 ✓
f_3	\rightarrow	ℓ_1 ✗	f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓

Removing the incentive to manipulate in Phase 2

Phase 2 of the MRDA algorithm identical to the first example of manipulability of the PFDA algorithm in Appendix D.2 and consequently both algorithms generate the same outcome. Recall, however, that the PFDA algorithm allowed f_2 to obtain ℓ_2 by reporting $\ell_2 \succ_{f_2} \ell_1 \succ_{f_2} \ell_3 \succ_{f_2} \emptyset$. This is no longer possible in the MRDA algorithm. Given f_2 's new report, the three rounds of Phase 2 are displayed below:

Round 1			Round 2			Round 3		
f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓	f_1	\rightarrow	ℓ_1 ✓
f_2	\rightarrow	ℓ_2 ✓	f_2	\rightarrow	ℓ_2 ✗	f_2	\rightarrow	ℓ_3 ✓
f_3	\rightarrow	ℓ_1 ✗	f_3	\rightarrow	ℓ_2 ✓	f_3	\rightarrow	ℓ_2 ✓

In Round 1, f_2 is tentatively accepted by ℓ_2 since it is the only family to propose. Family f_1 is tentatively accepted by ℓ_1 as its Maximum Rank is ∞ , however ℓ_3 is rejected since it has a lower priority than f_1 and a Maximum Rank of 1. This is the key difference between the two algorithms in this example. The PFDA algorithm allows f_3 to be tentatively accepted since it can be accommodated alongside f_1 , this means that f_3 is rejected if f_2 also proposes but not otherwise. Family f_2 thus has an incentive not to propose to ℓ_1 so that ℓ_3 does not compete for ℓ_2 . This incentive no longer exists in the MRDA algorithm as f_3 is rejected irrespective of whether ℓ_2 proposes to ℓ_1 . In Round 2, as before, f_2 and f_3 propose to ℓ_2 and f_2 is rejected. Family f_2 proposes to ℓ_3 in Round 3 and the algorithm ends. The outcome produced is the same as before: $Y^{\dagger MRDA} \equiv \{(f_1, \ell_1), (f_2, \ell_3), (f_3, \ell_2)\}$. The outcome generated by the PFDA algorithm in this problem is $Y^{\dagger PFDA} \equiv \{(f_1, \ell_1), (f_2, \ell_2), (f_3, \ell_1)\}$. This example illustrates the trade-off between efficiency from the perspective of refugee families and strategy-proofness: Family f_2 's incentive to misreport its preferences is removed by ensuring that the outcome obtained under this misrepresentation is less attractive than it could be.

References

- Abdulkadiroğlu, A. (2005). College admissions with affirmative action. *International Journal of Game Theory* 33(4), 535–549.
- 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(5), 1954–1978.
- Abdulkadiroğlu, A. and T. Sönmez (1999). House allocation with existing tenants. *Journal of Economic Theory* 88(2), 233–260.
- Abdulkadiroğlu, A. and T. Sönmez (2003). School choice: A mechanism design approach. *American Economic Review* 93(3), 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. 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.
- Aygün, O. and I. Bó (2016, September). College admission with multidimensional privileges: The Brazilian affirmative action case. Mimeo.
- Azevedo, E. and J. W. Hatfield (2015, August). Existence of stable matchings in large markets with complementarities. Mimeo.
- 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(1), 73–94.
- BBC (2016a). Attitudes harden towards refugees from Syria and Libya, BBC poll suggests. *BBC* 4 February.

- BBC (2016b, 20 September). Obama: 50 countries to take in 360,000 refugees this year. *BBC*.
- Biró, P., T. Fleiner, R. W. Irving, and D. F. Manlove (2010). The college admissions problem with lower and common quotas. *Theoretical Computer Science* 411(34), 3136 – 3153.
- Blum, Y., A. E. Roth, and U. G. Rothblum (1997). Vacancy chains and equilibration in senior-level labor markets. *Journal of Economic Theory* 76(2), 362–411.
- Bodoh-Creed, A. L. (2016). Programming approaches to school choice problems. Mimeo.
- Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy* 119(6), 1061–1103.
- Cechlárová, K., T. Fleiner, D. F. Manlove, I. McBride, and E. Potpinková (2015). Modelling practical placement of trainee teachers to schools. *Central European Journal of Operations Research* 23(3), 547–562.
- Crawford, V. P. and E. M. Knoer (1981). Job matching with heterogeneous firms and workers. *Econometrica* 49(2), 437–50.
- 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. (2014, December). Matching with quantity.
- 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.
- Echenique, F. and M. B. Yenmez (2007). A solution to matching with preferences over colleagues. *Games and Economic Behavior* 59(1), 46–71.
- Echenique, F. and M. B. Yenmez (2015). How to control controlled school choice. *American Economic Review* 105(8), 2679–94.
- 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.
- Eurostat (2016, March). Statistics Explained (online). Asylum Statistics.

- Featherstone, C. R. (2014). Rank efficiency: Investigating a widespread ordinal welfare criterion. Technical report, Working Paper.
- Feywerda, J. and J. Gest (2016). Location, location: Refugee resettlement and integration outcomes in the United States. Mimeo.
- 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.
- Gale, D. and L. S. Shapley (1962). College admissions and the stability of marriage. *American Mathematical Monthly* 69(1), 9–15.
- 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.
- Haake, C.-J. and B. Klaus (2009). Stability and Nash implementation in matching markets with couples. *Theory and Decision* 69(4), 537–554.
- Hafalir, I. E., M. B. Yenmez, and M. A. Yildirim (2013). Effective affirmative action in school choice. *Theoretical Economics* 8(2), 325–363.
- Hatfield, J. W. and F. Kojima (2008). Matching with contracts: Comment. *American Economic Review* 98(3), 1189–1194.
- Hatfield, J. W. and P. Milgrom (2005). Matching with contracts. *American Economic Review* 95(4), 913–935.
- 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 (2015). Efficient matching under distributional constraints: Theory and applications. *American Economic Review* 105(1), 67–99.
- Kellerer, H., U. Pferschy, and D. Pisinger (2004). *Knapsack problems*. Springer, Berlin.
- Kelso, A. S. and V. P. Crawford (1982). Job matching, coalition formation, and gross substitutes. *Econometrica* 50(6), 1483–1504.
- Kesten, O. (2010). School choice with consent. *Quarterly Journal of Economics* 125(3), 1297–1348.
- Klaus, B. and F. Klijn (2005). Stable matchings and preferences of couples. *Journal of Economic Theory* 121(1), 75–106.
- Klaus, B., F. Klijn, and J. Massó (2007). Some things couples always wanted to know about stable matchings (but were afraid to ask). *Review of Economic Design* 11(3), 175–184.
- Kojima, F. (2012). School choice: Impossibilities for affirmative action. *Games and Economic Behavior* 75(2), 685–693.
- Kojima, F. (2015). Finding all stable matchings with couples. *Journal of Dynamics and Games* 2(3/4), 321–330.
- Kominers, S. D. and T. Sönmez (2016). Matching with slot-specific priorities: Theory. *Theoretical Economics* 11(2), 683–710.
- 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.
- McDermid, E. J. and D. F. Manlove (2010). Keeping partners together: algorithmic results for the hospitals/residents problem with couples. *Journal of Combinatorial Optimization* 19(3), 279–303.
- 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.
- Nguyen, T., A. Peivandi, and R. Vohra (2016). Assignment problems with complementarities. *Journal of Economic Theory* 165, 209–241.

- Pápai, S. (2000). Strategyproof assignment by hierarchical exchange. *Econometrica* 68(6), 1403–1433.
- 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(1), 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(6), 991–1016.
- Roth, A. E. (1984b). Stability and polarization of interests in job matching. *Econometrica* 52(1), 47–57.
- Roth, A. E. (1985). The college admissions problem is not equivalent to the marriage problem. *Journal of Economic Theory* 36(2), 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(4), 748–780.
- 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. *American Economic Review* 97(3), 828–851.
- Roth, A. E., T. Sönmez, and M. U. Ünver (2004). Kidney exchange. *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* 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(3), 559–570.

- Roth, A. E. and M. Sotomayor (1990). *Two-sided matching: A study in game-theoretic modeling and analysis*. Cambridge University Press.
- Sbihi, A. (2007). A best first search exact algorithm for the multiple-choice multidimensional knapsack problem. *Journal of Combinatorial Optimization* 13(4), 337–351.
- 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.
- 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. (2013). Bidding for army career specialties: Improving the ROTC branching mechanism. *Journal of Political Economy* 121, 186–219.
- Sönmez, T. and T. B. Switzer (2013). Matching with (branch-of-choice) contracts at United States Military Academy. *Econometrica* 81(2), 451–488.
- Sotomayor, M. (1996). A non-constructive elementary proof of the existence of stable marriages. *Games and Economic Behavior* 13(1), 135–137.
- UNHCR (2015, June). UNHCR Refugee Resettlement Trends 2015. Technical report, United Nations High Commissioner for Refugees, Geneva.
- UNHCR (2016, June). Global Resettlement Needs 2017. Technical report, United Nations High Commissioner for Refugees.
- UNHCR (2016). Global trends. Technical report, United Nations High Commissioner for Refugees, Geneva.
- UNHCR (2016, September). UK Resettlement of Syrian refugees on track thanks to welcome from British public. Press Release.
- Wells, R. (1999, April). UK Kosovo refugees settle in. *BBC Monday, April 26*.
- Westkamp, A. (2013). An analysis of the German university admissions system. *Economic Theory* 53(3), 561–589.
- White, D. (2016, 14 September). White House Plans to Allow 110,000 Refugees into U.S. in 2017. *Time Magazine*.
- Wu, Q. and A. E. Roth (2016, October). The lattice of envy-free matchings. Mimeo.