CHAPTER 14

Matching under Preferences

Bettina Klaus, David F. Manlove, and Francesca Rossi

14.1 Introduction and Discussion of Applications

Matching theory studies how agents and/or objects from different sets can be matched with each other while taking agents' preferences into account. The theory originated in 1962 with a celebrated paper by David Gale and Lloyd Shapley (1962), in which they proposed the Stable Marriage Algorithm as a solution to the problem of two-sided matching. Since then, this theory has been successfully applied to many real-world problems such as matching students to universities, doctors to hospitals, kidney transplant patients to donors, and tenants to houses. This chapter will focus on algorithmic as well as strategic issues of matching theory.

Many large-scale centralized allocation processes can be modelled by matching problems where agents have preferences over one another. For example, in China, over 10 million students apply for admission to higher education annually through a centralized process. The inputs to the matching scheme include the students' preferences over universities, and vice versa, and the capacities of each university.[1] The task is to construct a matching that is in some sense optimal with respect to these inputs.

Economists have long understood the problems with decentralized matching markets, which can suffer from such undesirable properties as *unravelling*, *congestion* and *exploding offers* (see Roth and Xing, 1994, for details). For centralized markets, constructing allocations by hand for large problem instances is clearly infeasible. Thus centralized mechanisms are required for automating the allocation process.

Given the large number of agents typically involved, the *computational efficiency* of a mechanism's underlying algorithm is of paramount importance. Thus we seek polynomial-time algorithms for the underlying matching problems. Equally important are considerations of strategy: an agent (or a coalition of agents) may manipulate their input to the matching scheme (e.g., by misrepresenting their true preferences or underreporting their capacity) in order to try to improve their outcome. A desirable

[1] In fact, students are first assigned to universities and then to their programme of study within the university (see, e.g., Zhu, 2014).

property of a mechanism is *strategyproofness*, which ensures that it is in the best interests of an agent to behave truthfully.

The study of matching problems involving preferences was begun in 1962 with the seminal paper of Gale and Shapley (1962) who gave an efficient algorithm for the so-called *Stable Marriage problem* (which involves matching men to women based on each person having preferences over members of the opposite sex) and showed how to extend it to the *College Admissions problem*, a many-to-one extension of the Stable Marriage problem which involves allocating students to colleges based on college capacities, as well as on students' preferences over colleges, and vice versa. Their algorithm has come to be known as the *Gale–Shapley algorithm*.

Since 1962, the study of matching problems involving preferences has grown into a large and active research area, and numerous contributions have been made by computer scientists, economists, and mathematicians, among others. Several monographs exclusively dealing with this class of problems have been published (Knuth, 1976; Gusfield and Irving, 1989; Roth and Sotomayor, 1990; Manlove, 2013).

A particularly appealing aspect of this research area is the range of practical applications of matching problems, leading to real-life scenarios where efficient algorithms can be deployed and issues of strategy can be overcome. One of the best-known examples is the National Resident Matching Program (NRMP) in the United States, which handles the annual allocation of intending junior doctors (or *residents*) to hospital posts. In 2014, 40,394 aspiring junior doctors applied via the NRMP for 29,671 available residency positions (NRMP, 2014). The problem model is very similar to Gale and Shapley's College Admissions problem, and indeed an extension of the Gale–Shapley algorithm is used to construct the allocation each year (Roth, 1984a; Roth and Peranson, 1997). Similar medical matching schemes exist in Canada, Japan, and the United Kingdom. As Roth argued, the key property for a matching to satisfy in this context is *stability*, which ensures that a resident and hospital do not have the incentive to deviate from their allocation and become matched to one another.

Similar applications arise in the context of School Choice (Abdulkadiroğlu and Sönmez, 2003). For example in Boston and New York, centralized matching schemes are employed to assign pupils to schools on the basis of the preferences of pupils (or more realistically their parents) over schools, and pupils' *priorities* for assignment to a given school (Abdulkadiroğlu et al., 2005a, 2005b). A school's priority for a pupil might include issues such as geographical proximity and whether the pupil has a sibling at the school already, among others.

Kidney exchange (Roth et al., 2004, 2005) is another application of matching that has grown in importance in recent years. Sometimes, a kidney patient with a willing but incompatible donor can swap their donor with that of another patient in a similar position. Efficient algorithms are required to organize kidney "swaps" on the basis of information about donor and patient compatibilities. Such swaps can involve two or more patient–donor pairs, but usually the maximum number of pairs involved is three. Also altruistic donors can trigger "chains" involving swaps between patient–donor pairs. These allow for a larger number of kidney transplants (compared to those one could perform based on deceased donors only) and thus more lives saved. Centralized clearinghouses for kidney exchange are in operation on a nationwide scale in a number

of countries including the United States (Roth et al., 2004, 2005; Ashlagi and Roth, 2012), the Netherlands (Keizer et al., 2005), and the United Kingdom (Johnson et al., 2008). The problem of maximizing the number of kidney transplants performed through cycles and chains is NP-hard (Abraham et al., 2007a), though algorithms based on Mixed Integer Programming have been developed and are used to solve this problem at scale in the countries mentioned (Abraham et al., 2007a; Dickerson et al., 2013; Manlove and O'Malley, 2012; Glorie et al., 2014).

The importance of the research area in both theoretical and practical senses was underlined in 2012 by the award of the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel (commonly known as the Nobel Prize in Economic Sciences) to Alvin Roth and Lloyd Shapley for their work in "the theory of stable allocations and the practice of market design." This reflects both Shapley's contribution to the Stable Marriage algorithm among other theoretical advances, and Roth's application of these results to matching markets involving the assignment of junior doctors to hospitals, pupils to schools, and kidney patients to donors. The Nobel prize rules state that the prize cannot be awarded posthumously and hence David Gale (1921–2008) could not be honoured for his important contributions.

Matching problems involving preferences can be classified as being either *bipartite* or *nonbipartite*. In the former case, the agents are partitioned into two disjoint sets A and B, and the members of A have preferences over only the members of B (and possibly vice versa). In the latter case we have one single set of agents, each of whom ranks some or all of the others in order of preference. For space reasons we will consider only bipartite matching problems involving preferences in this chapter.

Bipartite problems can be further categorized according to whether the preferences are *two-sided* or *one-sided*. In the former case, members of both of the sets A and B have preferences over one another, whereas in the latter case only the members of A have preferences (over the members of B). Bipartite matching problems with two-sided preferences arise in the context of assigning junior doctors to hospitals, for example, while one-sided preferences arise in applications including the assignment of students to campus housing and reviewers to conference papers.

Our treatment covers *ordinal preferences* (where preferences are expressed in terms of first choice, second choice, etc.) rather than *cardinal utilities* (where preferences are expressed in terms of real-numbered valuations). In their simplest form, models of kidney exchange problems can involve *dichotomous preferences* (a special case of ordinal preferences, where an agent either finds another agent acceptable or not, and is indifferent among those it does find acceptable), on the basis of whether a patient is compatible with a potential donor. However, in practice, models of kidney exchange are more complex, typically involving cardinal utilities rather than ordinal preferences, and therefore the matching problems defined in this chapter do not encompass theoretical models of kidney exchange.

The problems considered in this chapter sit strongly within the field of computational social choice. This field lies at the interface of economics and computer science, and our approach will involve interleaving key aspects that have hitherto been considered by the two communities in bodies of literature that have largely pertained to the two disciplines separately. Such key considerations involve the existence of structural

results and efficient algorithms, and the derivation of strategyproof mechanisms. These topics will be reviewed in each of the cases of bipartite matching problems with two-sided and one-sided preferences. Although space restrictions have necessarily limited our coverage, we have tried to include the results that we feel will be of most importance to the readership of this handbook.

The structure of this chapter is as follows. In Section 14.2, we focus on bipartite matching problems where both sides have preferences. Here the most important property for a matching to satisfy is *stability*. In Section 14.2.1 we define the key matching problems in this class, most notably the *Hospitals / Residents problem*, and we also define stability in this context. We then state fundamental structural and algorithmic results concerning the existence, computation, and structural properties of stable matchings, in Section 14.2.2. Issues of strategy, and in particular the existence (or otherwise) of strategyproof mechanisms, are dealt with in Section 14.2.3. Next, in Section 14.2.4, we outline some further algorithmic results, including decentralized algorithms for computing stable matchings, variants of the Hospitals/Residents problem involving ties and couples, and many-to-many extensions.

Bipartite matching problems where only one side of the market has preferences are considered in Section 14.3. The fundamental problems in this class are the *House Allocation problem* and its extension to *Housing Markets*. We define these problems together with key properties of matchings, including *Pareto optimality* and membership of the *core*, in Section 14.3.1. Section 14.3.2 describes some important mechanisms that can be used to produce Pareto optimal matchings and matchings in the core. Strategyproofness is considered in Section 14.3.3, and then further algorithmic results are described in Section 14.3.4, including the computation of *maximum Pareto optimal*, *popular*, and *profile-based optimal matchings*.

Finally, in Section 14.4 we give some concluding remarks and list some further sources of reading.

14.2 Two-Sided Preferences

14.2.1 Introduction and Preliminary Definitions

The *Hospitals/Residents problem*[2] (HR) (Gale and Shapley, 1962; Gusfield and Irving, 1989; Roth and Sotomayor, 1990; Manlove, 2008) was first defined by Gale and Shapley in their seminal paper "College Admissions and the Stability of Marriage" (Gale and Shapley, 1962).

An instance I of HR involves a set $R = \{r_1, \ldots, r_{n_1}\}$ of *residents* and a set $H = \{h_1, \ldots, h_{n_2}\}$ of *hospitals*. Each hospital $h_j \in H$ has a positive integer *capacity*, denoted by c_j, indicating the number of *posts* that h_j has. Also there is a set $E \subseteq R \times H$ of *acceptable* resident–hospital pairs. Let $m = |E|$. Each resident $r_i \in R$ has an *acceptable* set of hospitals $A(r_i)$, where $A(r_i) = \{h_j \in H : (r_i, h_j) \in E\}$. Similarly each hospital $h_j \in H$ has an acceptable set of residents $A(h_j)$, where $A(h_j) = \{r_i \in R : (r_i, h_j) \in E\}$.

[2] The Hospitals/Residents problem is sometimes referred to as the College (or University or Stable) Admissions problem, or the Stable Assignment problem.

The *agents* in I are the residents and hospitals in $R \cup H$. Each agent $a_k \in R \cup H$ has a *preference list* in which she/it ranks $A(a_k)$ in strict order. Given any resident $r_i \in R$, and given any hospitals $h_j, h_k \in H$, r_i is said to *prefer* h_j to h_k if $\{h_j, h_k\} \subseteq A(r_i)$ and h_j precedes h_k on r_i's preference list; the *prefers* relation is defined similarly for a hospital.

An *assignment* M in I is a subset of E. If $(r_i, h_j) \in M$, r_i is said to be *assigned to* h_j, and h_j is *assigned* r_i. For each $a_k \in R \cup H$, the set of assignees of a_k in M is denoted by $M(a_k)$. If $r_i \in R$ and $M(r_i) = \emptyset$, r_i is said to be *unassigned*, otherwise r_i is *assigned*. Similarly, a hospital $h_j \in H$ is *undersubscribed* or *full* according as $|M(h_j)|$ is less than or equal to c_j, respectively. A *matching* M in I is an assignment such that $|M(r_i)| \leqslant 1$ for each $r_i \in R$ and $|M(h_j)| \leqslant c_j$ for each $h_j \in H$. For notational convenience, given a matching M and a resident $r_i \in R$ such that $M(r_i) \neq \emptyset$, where there is no ambiguity the notation $M(r_i)$ is also used to refer to the single member of the set $M(r_i)$.

Given an instance I of HR and a matching M, a pair $(r_i, h_j) \in E \backslash M$ *blocks* M (or is a *blocking pair* for M) if (i) r_i is unassigned or prefers h_j to $M(r_i)$ and (ii) h_j is undersubscribed or prefers r_i to at least one member of $M(h_j)$. M is said to be *stable* if it admits no blocking pair. If a resident–hospital pair (r_i, h_j) belongs to some stable matching in I, r_i is called a *stable partner* of h_j, and vice versa.

Example 14.1 (HR instance). Consider the following HR instance:

$$\begin{array}{ll} r_1 : h_1 \;\; h_2 & h_1 : 1 : r_3 \;\; r_2 \;\; r_1 \;\; r_4 \\ r_2 : h_1 \;\; h_2 \;\; h_3 & h_2 : 2 : r_2 \;\; r_3 \;\; r_1 \;\; r_4 \\ r_3 : h_2 \;\; h_1 \;\; h_3 & h_3 : 1 : r_2 \;\; r_3 \\ r_4 : h_2 \;\; h_1 & \end{array}$$

Here, r_1 prefers h_1 to h_2 and does not find h_3 acceptable. Also, h_1 has capacity 1 and prefers r_3 to r_2, and so on. $M = \{(r_2, h_1), (r_3, h_2), (r_4, h_2)\}$ is a matching in which each resident is assigned except for r_1, and both h_1 and h_2 are full while h_3 is undersubscribed. M is not stable because (r_1, h_2) is a blocking pair.

The *Stable Marriage problem with Incomplete lists* (SMI) (Gale and Shapley, 1962; Knuth, 1976; Gusfield and Irving, 1989; Roth and Sotomayor, 1990; Irving, 2008) is an important special case of HR in which $c_j = 1$ for all $h_j \in H$, and residents and hospitals are more commonly referred to as *men* and *women* respectively. The classical *Stable Marriage problem* (SM) is the restriction of SMI in which $n_1 = n_2$ and $E = R \times H$.

Finally, the *School Choice problem* (SC) (Balinski and Sönmez, 1999; Abdulkadiroğlu and Sönmez, 2003) is a one-sided preference version of HR where students and schools replace residents and hospitals respectively, and schools are endowed with *priorities* over students instead of preferences. A school's priority ranking over students may reflect a school district's policy choice (e.g., by giving students who are within walking distance or have a sibling in the same school a higher priority) or they may be based on other factors (e.g., grades in an entrance exam, time spent on a waiting list). For SC, schools are not considered to be economic agents: they neither strategize nor is their welfare measured and taken into account. Many results can easily be translated from HR to SC, but often the interpretation changes. For instance, the notion of stability can be interpreted as the elimination of *justified envy* (Balinski and Sönmez, 1999): a

student can justifiably envy the assignment of another student to a school if he likes that school better than his own assignment and he has a higher priority (with a lower priority, envy might be present as well but is not justifiable). Two recent and exhaustive surveys on school choice have been written by Abdulkadiroglu (2013) and Pathak (2011).

14.2.2 Classical Results: Stability and Gale-Shapley Algorithms

Gale and Shapley (1962) showed that every instance I of HR admits at least one stable matching. Their proof of this result was constructive, that is, they described a linear-time algorithm for finding a stable matching in I. Their algorithm is known as the *resident-oriented Gale-Shapley algorithm* (or RGS algorithm for short), because it involves residents applying to hospitals. Given an instance of HR,

(1) at the first step of the RGS algorithm, every resident applies to her favourite acceptable hospital. For each hospital h_j, the c_j acceptable applicants who have the highest ranks according to h_j's preference list (or all acceptable applicants if there are fewer than c_j) are placed on the waiting list of h_j, and all others are rejected;

(l) at the lth step of the RGS algorithm, those applicants who were rejected at step $l-1$ apply to their next best acceptable hospital. For each hospital h_j, the c_j acceptable applicants among the new applicants and those on the waiting list who have the highest ranks according to h_j's preference list (or all acceptable applicants if there are fewer than c_j) are placed on the waiting list of h_j, and all others are rejected.

Example 14.2 (RGS algorithm). We now illustrate an execution of the RGS algorithm for the HR instance shown in Example 14.1. In the first step, each of r_1 and r_2 applies to h_1, and each of r_3 and r_4 applies to h_2. Whilst h_2 accepts each of r_3 and r_4, h_1 can only accept r_2 (from among r_1 and r_2). Thus r_1 is rejected by h_1 and applies to the next hospital in his preference list, namely, h_2, at the second step. At this point, h_2 accepts r_1, keeps r_3, and rejects r_4. In the third step, r_4 applies to h_1 and is rejected again. Now the algorithm terminates because each resident is either assigned to a hospital or has applied to every hospital on his preference list. The resulting matching is thus $M' = \{(r_1, h_2), (r_2, h_1), (r_3, h_2)\}$, and the reader may verify that M' is stable.

The RGS algorithm is well-defined and terminates with the unique *resident-optimal* stable matching M_a that assigns to each resident the best hospital that she could achieve in any stable matching, while each unassigned resident is unassigned in every stable matching (Gale and Shapley, 1962; Gusfield and Irving, 1989, Section 1.6.3).

It is instructive to give a short sketch of the proof illustrating why M_a is stable. For, consider any resident r_i and suppose that h_j is any hospital that r_i prefers to $M_a(r_i)$ (if r_i is assigned in M_a) or h_j is any hospital that r_i finds acceptable (if r_i is unassigned in M_a). Then r_i applied to h_j during the execution of the RGS algorithm, and was rejected by h_j. This could only happen if h_j was full and preferred its worst assignee to h_j at that point. But h_j cannot subsequently lose any residents and indeed can only potentially gain better assignees. Hence in M_a, h_j is full and prefers its worst assigned resident to r_i. Thus (r_i, h_j) cannot block M_a, and because r_i and h_j were arbitrary, M_a is stable.

Furthermore, M_a is worst-possible for the hospitals in a precise sense: if M is any other stable matching then every hospital $h_j \in H$ prefers each resident in $M(h_j)$ to each resident in $M_a(h_j) \backslash M(h_j)$ (Gusfield and Irving, 1989, Section 1.6.5).

Theorem 14.3 (Gale and Shapley, 1962; Gusfield and Irving, 1989). *Given an* HR *instance, the RGS algorithm constructs, in $O(m)$ time, the unique resident-optimal stable matching, where m is the number of acceptable resident–hospital pairs.*

A counterpart of the RGS algorithm, known as the *hospital-oriented Gale–Shapley algorithm*, or HGS algorithm for short, involves hospitals offering posts to residents. The HGS algorithm terminates with the unique *hospital-optimal* stable matching M_z. In this matching, every full hospital $h_j \in H$ is assigned its c_j best stable partners, while every undersubscribed hospital is assigned the same set of residents in every stable matching (Gusfield and Irving, 1989, Section 1.6.2). Furthermore, M_z assigns to each resident the worst hospital that she could achieve in any stable matching, while each unassigned resident is unassigned in every stable matching (Gusfield and Irving, 1989, Theorem 1.6.1).

Theorem 14.4 (Gusfield and Irving, 1989). *Given an instance of* HR*, the HGS algorithm constructs, in $O(m)$ time, the unique hospital-optimal stable matching, where m is the number of acceptable resident–hospital pairs.*

Note that the RGS / HGS algorithms are often referred to as *deferred acceptance algorithms* by economists (Roth, 2008).

It is easy to check that for Example 14.2, $M_a = M' = M_z$. In general there may be other stable matchings—possibly exponentially many (Irving and Leather, 1986)—between the two extremes given by M_a and M_z. However some key structural properties hold regarding unassigned residents and undersubscribed hospitals with respect to all stable matchings in I, as follows.

Theorem 14.5 (Rural Hospitals Theorem: Roth, 1984a; Gale and Sotomayor, 1985; Roth, 1986). *For a given instance of* HR*, the following properties hold:*

1. *the same residents are assigned in all stable matchings;*
2. *each hospital is assigned the same number of residents in all stable matchings;*
3. *any hospital that is undersubscribed in one stable matching is assigned exactly the same set of residents in all stable matchings.*

The term "Rural Hospitals Theorem" stems from the tendency of rural hospitals to have problems in recruiting residents to fill all available slots. The theorem's name then indicates the importance of the result to the rural hospitals' recruitment problem: under stability one can never choose matchings to help undersubscribed rural hospitals to recruit more or better residents. Additional background to the Rural Hospitals Theorem for HR is given by Gusfield and Irving (1989, Section 1.6.4).

A classical result in stable matching theory states that, for a given instance of SM, the set of stable matchings forms a distributive lattice; Knuth (1976) attributed this result to John Conway (see also Gusfield and Irving, 1989, Section 1.3.1). In fact such a structure is also present for the set of stable matchings in a given instance I of HR

(Gusfield and Irving, 1989, Section 1.6.5). To describe this structure, we will define some preliminary notation and terminology.

Let $\mathcal{S}$ denote the set of stable matchings in I and let $M, M' \in \mathcal{S}$. We say that $r_i \in R$ *prefers* M to M' if r_i is assigned in both M and M', and r_i prefers $M(r_i)$ to $M'(r_i)$. Also, we say that r_i *is indifferent between* M and M' if either (i) r_i is unassigned in both M and M', or (ii) r_i is assigned in both M and M', and $M(r_i) = M'(r_i)$. Then, M *dominates* M', denoted $M \succeq M'$, if each resident either prefers M to M', or is indifferent between them.

For $M, M' \in \mathcal{S}$ we denote by $M \wedge M'$ (respectively $M \vee M'$) the set of resident-hospital pairs in which either (i) r_i is unassigned if she is unassigned in both M and M', or (ii) r_i is given the better (respectively poorer) of her partners in M and M' if she is assigned in both stable matchings. It turns out that each of $M \wedge M'$ and $M \vee M'$ is a stable matching in I, representing the *join* and the *meet* of M and M' respectively (Gusfield and Irving, 1989, Section 1.6.5). These operations give rise to a lattice structure for $\mathcal{S}$, as the following result indicates.

Theorem 14.6 (Gusfield and Irving, 1989). *Let I be an instance of* HR*, and let $\mathcal{S}$ be the set of stable matchings in I. Then $(\mathcal{S}, \succeq)$ forms a distributive lattice, with $M \wedge M'$ representing the meet, and $M \vee M'$ the join, for two stable matchings $M, M' \in \mathcal{S}$, where $\succeq$ is the dominance partial order on $\mathcal{S}$.*

14.2.3 Strategic Results: Strategyproofness

Note that both the RGS and HGS algorithms are described in terms of agents taking actions based on their preference lists (one side proposes and the other side tentatively accepts or rejects these proposals). However, unless agents have an incentive to truthfully report their preferences, any preference-based requirement (such as stability) might lose some of its meaning. The following theorem demonstrates that in general, stability is not compatible with the requirement that for all agents truth telling is a weakly dominant strategy (strategyproofness).

To be more precise, we call a function that assigns a matching to each instance of HR (or SMI/SM) a *mechanism*. A mechanism that assigns only stable matchings is called *stable*. The mechanism that always assigns the resident-optimal (hospital-optimal) stable matching is called the *RGS* (*HGS*) *mechanism*.

A mechanism for which no single agent can ever benefit from misrepresenting her/its preferences is called *strategyproof*, that is, in game-theoretic terms, it is a weakly dominant strategy for each agent to report her/its true preference list. If we restrict preference misrepresentations to one type of agents only, we obtain the one-sided versions of strategyproofness: a mechanism for which no single resident can ever benefit from misrepresenting her preferences is called strategyproof for residents. Strategyproofness for hospitals is similarly defined.

Theorem 14.7 (Impossibility Theorem: Roth, 1982b). *There exists no mechanism for* SMI *that is stable and strategyproof.*

As SMI is a special case of HR, Theorem 14.7 clearly extends to the HR case. The proof of Theorem 14.7 can be shown with the following example.

Example 14.8 (Impossibility). Consider the following instance:

$$\begin{array}{ll} r_1 : h_1 \;\; h_2 & \qquad h_1 : r_2 \;\; r_1 \\ r_2 : h_2 \;\; h_1 & \qquad h_2 : r_1 \;\; r_2 \end{array}$$

The two stable matchings for this instance are $M_a = \{(r_1, h_1), (r_2, h_2)\}$ and $M_z = \{(r_1, h_2), (r_2, h_1)\}$. Assume that the mechanism picks stable matching M_a. Then, if h_1 pretended that only r_2 is acceptable, M_a is not stable anymore and the stable mechanism would have to pick the only remaining stable matching M_z, which h_1 would prefer; a contradiction to strategyproofness. Similarly, if the mechanism picks stable matching M_z, r_1 could manipulate by declaring h_1 uniquely acceptable.

The intuition behind this impossibility result is that an agent who is assigned to a stable partner that is not her/its best stable partner can improve her/its outcome by truncating the preference list just below the best stable partner: this unilateral manipulation will result in the assignment of the best stable partner to the agent who misrepresented her/its preference list. Alcalde and Barberà (1994) and Takagi and Serizawa (2010) further strengthened the impossibility result by considerably weakening the stability requirement.

On the positive side, stable mechanisms that respect strategyproofness for all residents exist.

Theorem 14.9 (Roth, 1985). *The RGS mechanism for* HR *is strategyproof for residents.*

As HR is a generalization of each of SM and SMI, clearly Theorem 14.9 also holds in these latter contexts. This theorem for HR is an extension of an earlier corresponding theorem for SM (Dubins and Freedman, 1981; Roth, 1982a). Strategyproofness for all residents also turns out to be a key property in characterizing the RGS mechanism (Ehlers and Klaus, 2014): almost all real-life mechanisms used in variants of HR (including SC)—including the large classes of priority mechanisms and linear programming mechanisms—satisfy a set of simple and intuitive properties, but once strategyproofness is added to these properties, the RGS mechanism is the only one surviving (and characterized by the respective properties including strategyproofness). For SC, since residents (aka students) are the only economic agents, Theorem 14.9 in fact establishes a possibility result. For HR, the negative result of Theorem 14.7 persists even if restricting attention only to hospitals.

Theorem 14.10 (Roth, 1986). *There exists no mechanism for* HR *that is stable and strategyproof for hospitals.*

This result implies that even when the HGS mechanism is used, hospitals might have an incentive to misrepresent their preferences.

Once the incompatibility of stability and strategyproofness is established (Theorems 14.7 and 14.10), the question arises as to whether we can at least find stable mechanisms that are *resistant* to strategic behavior, meaning that it is computationally difficult (i.e., NP-hard) for agents to behave strategically. This approach is typical in voting theory, which is the subject of Chapter 6 on barriers to manipulation, because no voting rule is strategyproof (Arrow et al., 2002; Bartholdi et al., 1989a). It is possible to exploit such results to define stable mechanisms that are resistant to

strategic behaviour. Pini et al. (2011a) showed how to take voting rules that are resistant to strategic behaviour and use them to define stable mechanisms with the same property.

Besides worst-case analysis, we may also consider the occurrence and impact of strategic behavior when stable matching mechanisms are used in real-world instances of HR. Roth and Peranson (1999) showed that, for data from the NRMP, only a few participants could improve their outcomes by changing their preference list. They also showed via simulations that the opportunities for manipulation diminish when the instances of HR grow larger in population. Since then, various articles have provided theoretical explanations for this phenomenon for large population instances of SMI or HR (Immorlica and Mahdian, 2005; Kojima and Pathak, 2009; Lee, 2014).

14.2.4 Further Algorithmic Results

Decentralized Algorithms for SMI

In Section 14.2.2 we described the Gale-Shapley algorithm, which can be regarded as a centralized algorithm for HR. There has also been much interest in the study of decentralized algorithms for finding stable matchings. In particular, Roth and Vande Vate (1990) studied a mechanism for SMI that involves starting from some initial matching M_0 (which need not be stable) and constructing a random sequence of matchings $M_0, M_1, M_2 \ldots$, where for each $i \geqslant 1$, M_i is obtained from M_{i-1} by satisfying a blocking pair (m, w) of M_{i-1} (that is, the partners of m and w in M_{i-1}, if they exist, are both single in M_i, and (m, w) is added to M_i). The blocking pair that is satisfied at each step is chosen at random, subject to the constraint that there is a positive probability that any particular blocking pair (from among those that exist at a given step) is chosen. Roth and Vande Vate (1990) showed that this random sequence converges to a stable matching with probability 1. The algorithm underlying their result became known as the *Roth-Vande Vate Mechanism*. The special case of this mechanism in which $M_0 = \emptyset$ (and some other subtle modifications are made) has been referred to as the *Random Order Mechanism* (Ma, 1996).

When satisfying a blocking pair (m, w), if the "divorcees" ($M(w)$ and $M(m)$) are required to marry one another then the situation is very different. In this case there are SM instances and initial matchings M_0 such that it is not possible to transform M_0 to a stable matching by satisfying a sequence of blocking pairs (Tamura, 1993; Tan and Su, 1995).

Ackermann et al. (2011) categorized decentralized algorithms for SMI into *better response dynamics* and *best response dynamics*. The former description applies to mechanisms that are based on satisfying blocking pairs, while the latter refers to a more specific mechanism where, should a blocking pair be satisfied, it is the best blocking pair for the *active* agent (i.e., the agent who makes the proposal). The authors also considered *random better response dynamics* and *random best response dynamics*. In the former case, a blocking pair is chosen uniformly at random, while in the latter case, a blocking pair that corresponds to the best blocking pair for a given proposer is selected uniformly at random. The authors gave exponential lower bounds for the convergence time of both approaches in uncoordinated markets.

Both sequential and parallel local search algorithms, based on the approach of Roth and Vande Vate (1990), have been implemented and tested on large SMI problem instances, showing a very efficient behavior (Gelain et al., 2013; Munera et al., 2015).

Hospitals/Residents Problem with Ties

In the context of centralised clearinghouses for junior doctor allocation, often large hospitals have many applicants and may find it difficult to produce a strict ranking over all these residents. In practice a hospital may be indifferent between batches of residents, represented by *ties* in its preference list. This naturally leads to the *Hospitals/Residents problem with Ties* (HRT), the generalization of HR in which the preference lists of both residents and hospitals can contain ties.

In the HRT context, several stability definitions have been formulated in the literature, with varying degrees of strength. A matching M is *weakly stable* if there is no resident–hospital pair (r, h), such that by coming together, each would be strictly better off than their current situation in M. In the case of *strong stability*, in a blocking pair (r, h) it is enough for one of (r, h) to be strictly better off, while the other must be no worse off, by forming a partnership. Finally, in the case of *super-stability*, all we require is that each of (r, h) must be no worse off.

Example 14.11 (HRT instance). To illustrate these stability concepts, we insert some ties into the preference lists in the HR instance shown in Example 14.1. The resulting instance of HRT is

$$\begin{array}{ll} r_1 : h_1 \;\; h_2 & h_1 : 1 : r_3 \;\; (r_2 \;\; r_1) \;\; r_4 \\ r_2 : h_1 \;\; h_2 \;\; h_3 & h_2 : 2 : r_2 \;\; (r_3 \;\; r_1 \;\; r_4) \\ r_3 : h_2 \;\; (h_1 \;\; h_3) & h_3 : 1 : r_2 \;\; r_3 \\ r_4 : h_2 \;\; h_1 & \end{array}$$

Here, parentheses indicate ties in the preference lists, so for example, r_3 prefers h_2 to each of h_1 and h_3, and is indifferent between the latter two hospitals. The matchings $\{(r_1, h_2), (r_2, h_1), (r_3, h_2)\}$ and $\{(r_1, h_1), (r_2, h_2), (r_3, h_3), (r_4, h_2)\}$ are both weakly stable, but the instance admits no strongly stable matching, and hence no super-stable matching either.

We continue by considering algorithmic results for HRT under weak stability. Firstly, an HRT instance is bound to admit a weakly stable matching, and such a matching can be found in linear time (Irving et al., 2000). Recall from Theorem 14.5 that all stable matchings in an HR instance have the same size. However in the case of HRT, weakly stable matchings may have different sizes, as illustrated by Example 14.11. Often in the case of centralized clearinghouses, an important consideration is to match as many participants as possible. This motivates MAX HRT, the problem of finding a maximum weakly stable matching, given an HRT instance. This problem is NP-hard (Iwama et al., 1999; Manlove et al., 2002) even if each hospital has capacity 1, and also even under severe restrictions on the number, length and positions of the ties (Manlove et al., 2002). A succession of approximation algorithms has been proposed in the literature for various restrictions of MAX HRT, culminating in the best current bound of $3/2$ for the general problem (McDermid, 2009; Király, 2013; Paluch, 2014).

Although an HRT instance I is bound to admit a weakly stable matching as mentioned above, by contrast a strongly stable matching or a super-stable matching in I may not exist (Irving et al., 2000, 2003). However there is an efficient algorithm to find a strongly stable matching or report that none exists (Kavitha et al., 2007). A faster and simpler algorithm exists in the case of super-stability (Irving et al., 2000). Moreover an analogue of Theorem 14.5 holds in HRT under each of the strong stability and super-stability criteria (Scott, 2005; Irving et al., 2000).

Hospitals/Residents Problem with Couples

Another variant of HR that is motivated by practical applications arises in the presence of *couples*. These are pairs of residents who wish to be jointly assigned to hospitals via a common preferences list over pairs of hospitals, typically in order to be geographically close to one another. Each couple (r_i, r_j) has a preference list over a subset of $H \times H$, where each pair (h_p, h_q) on this list represents the joint assignment of r_i to h_p and r_j to h_q. (There may be single residents in addition, as before.) We thus obtain the *Hospitals/Residents problem with Couples* (HRC).

Relative to a suitable stability definition, Roth (1984a) showed that an HRC instance need not admit a stable matching. Ng and Hirschberg (1988) and Ronn (1990) independently showed that the problem of deciding whether an HRC instance admits a stable matching is NP-complete, even if each hospital has capacity 1 and there are no single residents.

McDermid and Manlove (2010) considered a variant of HRC in which each resident (whether single or in a couple) has a preference list over individual hospitals, and the joint preference list of each couple (r_i, r_j) is *consistent* with the individual lists of r_i and r_j in a precise sense. Relative to Roth's stability definition (Roth, 1984a), they showed that the problem of deciding whether a stable matching exists is NP-complete. However if instead we enforce classical (Gale–Shapley) stability on a given matching relative to the individual lists of residents, then the problem of finding a stable matching or reporting that none exists is solvable in polynomial time (McDermid and Manlove, 2010).

Biró et al. (2011) developed a range of heuristics for the problem of finding a stable matching or reporting that none exists in a given HRC instance, and subjected them to a detailed empirical evaluation based on randomly generated data. They found that a stable matching is very likely to exist for instances where the ratio of couples to single residents is small and of the magnitude typically found in practical applications.

Ashlagi et al. (2014) studied large random matching markets with couples. They introduced a new matching algorithm and showed that if the number of couples grows slower than the size of the market, a stable matching will be found with high probability. If, however, the number of couples grows at a linear rate, with constant probability (not depending on the market size), no stable matching exists.

Further results for HRC are described in the survey paper of Biró and Klijn (2013).

Many-to-Many Stable Matching

Many-to-many extensions of SM (and by implication HR) have been considered in the literature (Roth, 1984b; Roth and Sotomayor, 1990; Sotomayor, 1999; Baïou and

Balinski, 2000; Fleiner, 2003; Martínez et al., 2004; Echenique and Oviedo, 2006; Bansal et al., 2007; Kojima and Ünver, 2008; Eirinakis et al., 2012, 2013; Klijn and Yazıcı, 2014). These matching problems tend to be described in the context of assigning *workers* to *firms*, where each agent can be multiply assigned (up to a given capacity). We will discuss the two main models of many-to-many matching in the literature.

The first version we consider, which we refer to as the *Workers/Firms problem, Version 1*, denoted by WF-1, involves each worker ranking in strict order of preference a set of individual acceptable firms, and vice versa for each firm. Baïou and Balinski (2000) generalized the stability definition for SM to the WF-1 case. They showed that every instance I of WF-1 has a stable matching and such a matching can be found in $O(n^2)$ time, where $n = \max\{n_1, n_2\}$, n_1 is the number of workers and n_2 is the number of firms in I. They also generalized Theorems 14.5 and 14.6 to the WF-1 context. Additional algorithms have been given for computing stable matchings with various optimality properties in WF-1 (Bansal et al., 2007; Eirinakis et al., 2012, 2013).

In the second version, which we refer to as the *Workers/Firms problem, Version 2*, denoted by WF-2, each worker ranks in strict order of preference acceptable subsets of firms, and vice versa for each firm. Two main forms of stability have been studied in the context of WF-2, namely, *pairwise stability* and *setwise stability*.

A matching M in a WF-2 instance is *pairwise stable* (Roth, 1984b) if it cannot be undermined by a single worker–firm pair acting together. A WF-2 instance need not admit a pairwise stable matching (Roth and Sotomayor, 1990, Example 2.7). However Roth (1984b) proved that, given an instance of WF-2 where every agent's preference list satisfies so-called *substitutability* (Kelso and Crawford, 1982), a pairwise stable matching always exists, and he gave an algorithm for finding one. Martínez et al. (2004) gave an algorithm for finding all pairwise stable matchings.

A more powerful definition of stability is *setwise stability*. Informally, a matching M is *setwise stable* (Sotomayor, 1999) if it cannot be undermined by a coalition of workers and firms acting together. More precisely, several definitions of setwise stability have been given in the literature (Sotomayor, 1999; Echenique and Oviedo, 2006; Konishi and Ünver, 2006); the various alternatives were formally defined and analyzed by Klaus and Walzl (2009).

Bansal et al. (2007) noted that, generally speaking, WF-1 has been studied mainly by the computer science community, while the economics community has mainly focused on WF-2. One reason for this is that WF-2 suffers from the drawback that the length of an agent's preference list is in the worst case exponential in the number of agents. A consequence of this is that the practical applicability of any algorithm for WF-2 would be severely limited in general, however, this problem does not arise with WF-1.

14.3 One-Sided Preferences

14.3.1 Introduction and Preliminary Definitions

Many economists and game theorists, and increasingly computer scientists in recent years, have studied the problem of allocating a set H of indivisible goods among a set A of applicants (Shapley and Scarf, 1974; Hylland and Zeckhauser, 1979; Deng

et al., 2003; Fekete et al., 2003). Each applicant a_i may have ordinal preferences over a subset of H (the *acceptable* goods for a_i). Many models have considered the case where there is no monetary transfer. In the literature the situation in which each applicant initially owns one good is known as a *Housing Market* (HM)[3] (Shapley and Scarf, 1974; Roth and Postlewaite, 1977; Roth, 1982b). When there are no initial property rights, we obtain the *House Allocation problem* (HA) (Hylland and Zeckhauser, 1979; Zhou, 1990; Abdulkadiroğlu and Sönmez, 1998). A mixed model, in which a subset of applicants initially owns a good has also been studied (Abdulkadiroğlu and Sönmez, 1999).

House Allocation Problems

Formally, an instance I of the *House Allocation problem* (HA) comprises a set $A = \{a_1, a_2, \ldots, a_{n_1}\}$ of *applicants* and a set $H = \{h_1, h_2, \ldots, h_{n_2}\}$ of *houses*. The *agents* in I are the applicants and houses in $A \cup H$. There is a set $E \subseteq A \times H$ of *acceptable* applicant–house pairs. Let $m = |E|$. Each applicant $a_i \in A$ has an *acceptable* set of houses $A(a_i)$, where $A(a_i) = \{h_j \in H : (a_i, h_j) \in E\}$. Similarly each house $h_j \in H$ has an acceptable set of applicants $A(h_j)$, where $A(h_j) = \{a_i \in A : (a_i, h_j) \in E\}$.

Each applicant $a_i \in A$ has a *preference list* in which she ranks $A(a_i)$ in strict order. Given any applicant $a_i \in A$, and given any houses $h_j, h_k \in H$, a_i is said to *prefer* h_j to h_k if $\{h_j, h_k\} \subseteq A(a_i)$, and h_j precedes h_k on a_i's preference list. Houses do not have preference lists over applicants, and it is essentially this feature that distinguishes HA from SMI.

HA is a very general problem model and any application domain having an underlying matching problem that is bipartite, where agents in only one of the sets have preferences over the other, can be viewed as an instance of HA. These include the problems of allocating graduates to trainee positions, students to projects, professors to offices, clients to servers, and so on. The literature concerning HA has largely described this problem model in terms of assigning applicants to houses, so for consistency we also adopt this terminology.

An *assignment* M in I is a subset of E. The definitions of the terms *assigned to*, *assigned*, *unassigned* and *assignees* relative to M are analogous to the same definitions in the HR case (see Section 14.2.1). A *matching* M in I is an assignment such that, for each $p_k \in A \cup H$, the set of assignees of p_k in M, denoted by $M(p_k)$, satisfies $|M(p_k)| \leqslant 1$. For notational convenience, as in the HR case, if p_k is assigned in M then where there is no ambiguity the notation $M(p_k)$ is also used to refer to the single member of the set $M(p_k)$. Let $\mathcal{M}$ denote the set of matchings in I.

Given two matchings M and M' in $\mathcal{M}$, we say that an applicant $a_i \in A$ *prefers* M' *to* M if either (i) a_i is assigned in M' and unassigned in M, or (ii) a_i is assigned in both M and M', and a_i prefers $M'(a_i)$ to $M(a_i)$. We say that M' *Pareto dominates* M if (i) some applicant prefers M' to M and (ii) no applicant prefers M to M'. A matching $M \in \mathcal{M}$ is *Pareto optimal* if there is no matching $M' \in \mathcal{M}$ that Pareto dominates M. Intuitively M is Pareto optimal if no applicant a_i can be better off without requiring

[3] This problem is also referred to as the *House-swapping Game* in the literature.

another applicant a_j to be worse off. For example, M is not Pareto optimal if two applicants could improve by swapping the houses that they are assigned to in M.

Housing Markets

An instance I of a *Housing Market* (HM) comprises an HA instance I where $n_1 = n_2$, together with a matching M_0 in I (the *initial endowment*) such that $|M_0| = n_1$. A matching M in I is *individually rational* if, for each applicant $a_i \in A$, either a_i prefers $M(a_i)$ to $M_0(a_i)$, or $M(a_i) = M_0(a_i)$. Since we are only interested in individually rational matchings, we assume that $M_0(a_i)$ is the last house on a_i's preference list, for each $a_i \in A$. Clearly then, any individually rational matching M in I satisfies $|M| = n_1$.

The notion of Pareto optimality in HA is closely related to the concept of *core* matchings in the HM context (Roth and Postlewaite, 1977): let I be an instance of HM where M_0 is the initial endowment, and let M be an individually rational matching in I. Let M' be a matching in I, and let S be the set of applicants who are assigned in M'. Then M' *weakly blocks* M with respect to the *coalition* S if:

- (i) the members of the coalition are only allowed to improve by exchanging their own resources (via their initial endowment M_0): $\{M'(a_i) : a_i \in S\} = \{M_0(a_i) : a_i \in S\}$;
- (ii) some member of the coalition $a_i \in S$ is better off in M': some $a_i \in S$ prefers $M'(a_i)$ to $M(a_i)$;
- (iii) no member of the coalition $a_i \in S$ is worse off in M' than in M: no $a_i \in S$ prefers $M(a_i)$ to $M'(a_i)$.

M is a *strict core matching*, or M is *in the strict core*, if there is no other matching in I that weakly blocks M. Also M' *strongly blocks* M with respect to S if Condition (i) is satisfied, and in addition, every $a_i \in S$ prefers $M'(a_i)$ to $M(a_i)$. M is a *weak core matching*, or M is *in the weak core*, if there is no other matching in I that strongly blocks M.

Note that M is Pareto optimal if and only if M is not weakly blocked by any matching M' such that $|M'| = n_1$ (here the coalition comprises all applicants and is referred to as the *grand coalition*). Hence a strict core matching is Pareto optimal.

Example 14.12 (HM instance). Consider the following HM instance in which the initial endowment is $M_0 = \{(a_1, h_4), (a_2, h_3), (a_3, h_2), (a_4, h_1)\}$.

$$\begin{array}{l} a_1 : h_1 \;\; h_2 \;\; h_3 \;\; h_4 \\ a_2 : h_1 \;\; h_2 \;\; h_4 \;\; h_3 \\ a_3 : h_4 \;\; h_1 \;\; h_3 \;\; h_2 \\ a_4 : h_4 \;\; h_3 \;\; h_2 \;\; h_1 \end{array}$$

Now define the matchings $M = \{(a_1, h_4), (a_2, h_3), (a_3, h_1), (a_4, h_2)\}$, $M' = \{(a_1, h_3), (a_2, h_2), (a_3, h_4), (a_4, h_1)\}$ and $M'' = \{(a_1, h_1), (a_2, h_2), (a_3, h_3), (a_4, h_4)\}$. Then M' strongly blocks M with respect to the coalition $S = \{a_1, a_2, a_3\}$, while M'' is a strict core matching and hence Pareto optimal.

We call a function that assigns a matching to each instance of HA (or HM) a *mechanism*. A mechanism that assigns only Pareto optimal matchings is called *Pareto optimal*.

14.3.2 Classical Structural and Algorithmic Results

House Allocation Problems

All Pareto optimal matchings can be constructed using a classical algorithm called the *Serial (SD) Dictatorship Algorithm* (see Theorem 14.14). For any fixed order of applicants $f = (i_1, i_2, \ldots, i_{n_1})$, the SD algorithm is a straightforward greedy algorithm that takes each applicant in turn and assigns her to the most-preferred available house on her preference list. The associated mechanism is called the *Serial Dictatorship* (*SD*) *mechanism*. The order in which the applicants are processed will, in general, affect the outcome. If a uniform lottery is used in order to determine the applicant ordering, then we obtain a random mechanism called the *Random Serial Dictatorship Mechanism* or *RSD mechanism* (Abdulkadiroğlu and Sönmez, 1998).

Often, the fixed order of applicants used for the SD mechanism is determined in some objective way. Roth and Sotomayor (1990, Example 4.3) remark that when the U.S. Naval Academy matches graduating students to their first posts as naval officers using an approach based on the SD algorithm, students are considered in nondecreasing order of graduation results. Clearly the SD algorithm may be implemented to execute in $O(m)$ time (m being the number of acceptable applicant–house pairs).

Strictly speaking RSD produces a probability distribution over matchings, and its output can be regarded as a bi-stochastic $n_1 \times n_2$ matrix M in which entry (i, j) gives the probability of applicant a_i receiving house h_j. Independently, Aziz et al. (2013a) and Saban and Sethuraman (2013) proved that computing M is #P-complete. Saban and Sethuraman (2013) also proved the surprising result that determining whether a given entry (i, j) in M has positive probability is NP-complete. This implies NP-completeness for the problem of determining whether, given an applicant a_i and house h_j, there exists a Pareto optimal matching containing (a_i, h_j).

Krysta et al. (2014) gave an $O(n_1^2\gamma)$ strategyproof adaptation of RSD to the more general extension of HA in which preference lists may include ties, where γ is the maximum length of a tie in any applicant' s preference list.

Housing Markets

For a somewhat more general housing market model that allows for indifferences in preference lists, Shapley and Scarf (1974) showed that the weak core is always nonempty by constructing a weak core matching using Gale's *Top Trading Cycles* or *TTC algorithm* (the authors attributed the now famous TTC algorithm to David Gale). They also showed that the weak core matching constructed is a competitive allocation,[4] the strict core may be empty and the nonempty weak core may exceed the (not necessarily singleton) set of competitive allocations. Note that for our housing market model with strict preferences, the weak and the strict core coincide. Given an instance of HM with initial endowment M_0,

[4] While housing markets are modelled as pure exchange economies, a competitive allocation of a housing market can be defined using fiat money. Then, an allocation is competitive if there exists a price for each house such that, by selling his house at the given price, each agent can afford to buy his most-preferred house (i.e., market clearance ensues).

(1) at the first step of the TTC algorithm, every applicant points to the owner of her favourite house (possibly to herself). Because there are finitely many applicants, there is at least one cycle (where a cycle is an ordered list $(i_1, i_2, \ldots, i_k)$, $1 \leqslant k \leqslant n_1$, of applicants with each applicant pointing to the next applicant in the list and applicant a_{i_k} pointing to applicant a_{i_1}; $k = 1$ is the special case of a self-loop where an applicant points to herself). In each cycle the implied cyclical exchange of houses is implemented and the algorithm continues with the remaining applicants and houses;

(l) at the lth step of the TTC algorithm, every remaining applicant points to the owner of her favourite remaining house (possibly to herself). Again, there is at least one cycle and in each cycle the implied cyclical exchange of houses is implemented and the algorithm continues with the remaining applicants and houses, and terminates when no applicants remain.

Note that there is an equivalent two-sided formulation of the TTC algorithm in which agents point to houses, as specified previously, and houses will always point to their owners. The TTC algorithm can be implemented to run in $O(m)$ time (m being the number of acceptable applicant–house pairs) (Abraham et al., 2004). Roth and Postlewaite (1977) demonstrated that the matching found by the TTC algorithm is the unique strict core allocation as well as the unique competitive allocation. The mechanism that assigns to each instance of HM the strict core matching obtained by the TTC algorithm is called the *Core Mechanism* or sometimes simply the *Core*.

Example 14.13. We apply the TTC algorithm to the HM instance shown in Example 14.12. The initial directed graph has four nodes (representing all applicants) where each applicants points to the owner (in M_0) of its most preferred house. Hence there is a directed arc from a_1 to a_4, from a_2 to a_4, from a_3 to a_1, and from a_4 to a_1. Because there is a cycle involving a_1 and a_4, we swap their houses, and thus a_1 receives h_1 and a_4 receives h_4. Now we delete a_1 and a_4 from the graph, as well as their houses from the HM instance. We are thus left with a_2 and a_3, with an arc from a_2 to a_3 (because after having deleted h_1, the most preferred house of a_2 is h_2, owned by a_3) and similarly an arc from a_3 to a_2. Thus we swap their houses and the algorithm stops, returning the matching $M'' = \{(a_1, h_1), (a_2, h_2), (a_3, h_3), (a_4, h_4)\}$ as in Example 14.12.

Recall that the only difference between an instance of HA and an instance of HM is that in the latter case an initial endowment matching M_0 is given as well. Hence, we could define a mechanism for HA that fixes an initial endowment matching M_f and then uses the Core mechanism for the obtained instance of HM. We call such a mechanism a *Core from Fixed Endowments* or *CFE mechanism*. If now a uniform lottery is used in order to determine the initial endowment matching, then we obtain a random mechanism called the *Core from Random Endowments* or *CRE mechanism* (Abdulkadiroğlu and Sönmez, 1998). Abdulkadiroğlu and Sönmez (1998) proved that the two random mechanisms we have introduced are equivalent.

Theorem 14.14 (Abdulkadiroğlu and Sönmez, 1998).

1. *All SD mechanisms for* HA *are Pareto optimal. For each Pareto optimal matching M of an instance of* HA*, there exists an order of applicants such that the corresponding SD mechanism assigns M.*

2. *All Core mechanisms for* HM *are Pareto optimal. For each Pareto optimal matching M of an instance of* HA*, there exists an initial endowment matching M_f such that the CFE mechanism assigns M.*
3. *The CRE and the RSD mechanisms for* HA *are equivalent.*

Hylland and Zeckhauser (1979) had already shown that the RSD mechanism is ex-post Pareto optimal, that is, the final matching that is chosen by the RSD lottery is Pareto optimal. Bogomolnaia and Moulin (2001) showed that the RSD mechanism, however, is not ex ante or ordinally efficient (Pareto optimal), that is, for some lotteries chosen by the RSD mechanism there exist Pareto dominating lotteries (with stochastic dominance being used to formulate the dominance relation). They also suggested a new random mechanism, the *Probabilistic Serial mechanism*, that satisfies ex ante efficiency.

14.3.3 Strategic Results: Strategyproofness

As in Section 14.2.1, a mechanism for which no single applicant can ever benefit from misrepresenting her preferences is called *strategyproof* (i.e., in game-theoretic terms, it is a weakly dominant strategy for each applicant to report her true preference list). All mechanisms introduced so far in this section are strategyproof, as the following results indicate.

Theorem 14.15 (Hylland and Zeckhauser, 1979). *The SD mechanisms for* HA *are strategyproof.*

Theorem 14.16 (Roth, 1982b). *The Core mechanism for* HM *is strategyproof. Hence, all CFE mechanisms for* HA *are strategyproof.*

In addition, the Core and CFE mechanisms are *group strategyproof* (i.e., no coalition of applicants can jointly misrepresent their true preferences in order for at least one member of the coalition to improve, while no other coalition member is worse off; see, e.g., Svensson, 1999). Strategyproofness is also one of the properties that characterize the Core mechanism.

Theorem 14.17 (Ma, 1994). *The Core mechanism for* HM *is the only mechanism that is Pareto optimal, individually rational, and strategyproof.*

Abdulkadiroğlu and Sönmez (1999) extended Ma's characterization result to a mixed model that combines HA and HM: in the House Allocation problem with Existing Tenants, a subset of applicants initially owns a house. They defined mechanisms that combine elements of SD as well as Core mechanisms based on the so-called YRMH-IGYT (You Request My House—I Get Your Turn) algorithm. All YRMH-IGYT mechanisms are strategyproof, Pareto optimal, and individually rational (in the sense that no existing tenant receives a house inferior to his own).

In Section 14.2.1 we introduced SC as a one-sided preference variant of HR, but we could also introduce this class of problems as a variant of HA with the additional properties that objects (i.e., houses/schools) have priorities over students, and objects can be multiply assigned up to some capacity. Either way, the RGS mechanism can be used to find a matching for each instance of SC. This mechanism is then strategyproof

$a_1 : h_1 \ h_2$	$a_1 : h_1 \ h_2 \ h_3$	$a_1 : h_1 \ h_3$
$a_2 : h_1$	$a_2 : h_1 \ h_2 \ h_3$	$a_2 : h_2 \ h_1$
	$a_3 : h_1 \ h_2 \ h_3$	$a_3 : h_2$
(a)	(b)	(c)

Figure 14.1. (a) HA instance I_1; (b) HA instance I_2; (c) HA instance I_3.

(by Theorem 14.9) and stable (Gale and Shapley, 1962), but it is not Pareto optimal. In fact, no mechanism is both stable and Pareto optimal (Balinski and Sönmez, 1999). However it turns out that no other stable mechanism would do better in the following sense.

Theorem 14.18 (Balinski and Sönmez, 1999). *The RGS mechanism for* SC *Pareto dominates any other stable mechanism.*

Finally, when focusing on strategyproofness and Pareto optimality only, no better mechanism than the RGS mechanism emerges.

Theorem 14.19 (Kesten, 2010). *The RGS mechanism for* SC *is not Pareto-dominated by any other Pareto optimal mechanism that is strategyproof.*

14.3.4 Further Algorithmic Results

Pareto Optimal Matchings

For a given instance of HA, Pareto optimal matchings may have different sizes, as illustrated by Figure 14.1a: for the instance I_1 shown, matchings $M_1 = \{(a_1, h_1)\}$ and $M_2 = \{(a_1, h_2), (a_2, h_1)\}$ are both Pareto optimal. In many applications we seek to match as many applicants as possible. This motivates the problem of finding a Pareto optimal matching of maximum size, which we refer to as a *maximum Pareto optimal matching*.

Toward an algorithm for this problem, Abraham et al. (2004) gave a characterization of Pareto optimal matchings in a given HA instance I. A matching M in I is *maximal* if there is no pair $(a_i, h_j) \in E$, both of which are unassigned in M. Also M is *trade-in-free* if there is no pair $(a_i, h_j) \in E$ such that h_j is unassigned in M, and a_i is assigned in M and prefers h_j to $M(a_i)$. Finally M is *cyclic coalition-free* if M admits no cyclic coalition, which is a sequence of applicants $C = \langle a_{i_0}, a_{i_1}, \ldots, a_{i_{r-1}} \rangle$, for some $r \geqslant 2$, all assigned in M, such that a_{i_j} prefers $M(a_{i_{j+1}})$ to $M(a_{i_j})$ $(0 \leqslant j \leqslant r-1)$ (with subscripts taken modulo r). Abraham et al. gave the following necessary and sufficient conditions for a matching to be Pareto optimal in terms of these concepts:

Proposition 14.20 (Abraham et al., 2004). *Let I be an instance of* HA *and let M be a matching in I. Then M is Pareto optimal if and only if M is maximal, trade-in-free and coalition-free. Moreover there is an $O(m)$ algorithm for testing M for Pareto optimality, where m is the number of acceptable applicant–house pairs in I.*

Abraham et al. also gave a three-phase algorithm for finding a maximum Pareto optimal matching in I, with each phase enforcing one of the conditions for Pareto optimality given in Proposition 14.20. In Phase 1 they construct a maximum matching

M in the *underlying graph* of I, which is the bipartite graph with vertex set $A \cup H$ and edge set E. This step can be accomplished in $O(\sqrt{n_1}m)$ time and ensures that M is maximal. Phase 2 is based on an $O(m)$ algorithm in which assigned applicants repeatedly trade in their own house in M for any preferred vacant house. Once this step terminates, M is trade-in-free. Finally, cyclic coalitions are eliminated during Phase 3, which is based on an $O(m)$ implementation of the TTC algorithm. Putting these three phases together, they obtained the following result.

Theorem 14.21 (Abraham et al., 2004). *Let I be an instance of* HA. *A maximum Pareto optimal matching in I can be found in $O(\sqrt{n_1}m)$ time, where n_1 is the number of applicants and m is the number of acceptable applicant–house pairs in I.*

Popular Matchings

Pareto optimality is a fundamental solution concept, but on its own it is a relatively weak property. A stronger notion is that of a *popular matching*. Intuitively a matching M in an HA instance I is *popular* if there is no other matching that is preferred to M by a majority of the applicants who are not indifferent between the two matchings. This concept was first defined by Gärdenfors (1975) (using the term *majority assignment*) in the context of SMI.

To define the popular matching concept more formally, let $M, M' \in \mathcal{M}$, and let $P(M, M')$ denote the set of applicants who prefer M to M'. We say that M' is *more popular than* M, denoted $M' \succ M$, if $|P(M', M)| > |P(M, M')|$. Define a matching $M \in \mathcal{M}$ to be *popular* (Abraham et al., 2007b) if M is $\succ$-maximal (i.e., there is no other matching $M' \in \mathcal{M}$ such that $M' \succ M$).

Clearly a matching M is Pareto optimal if there is no other matching M' such that $|P(M, M')| = 0$ and $|P(M', M)| \geqslant 1$. Hence a popular matching is Pareto optimal. However in contrast to the case for Pareto optimal matchings, an HA instance need not admit a popular matching. To see this, consider the HA instance I_2 shown in Figure 14.1b. It is clear that a matching in I_2 cannot be popular unless all applicants are assigned. The unique matching up to symmetry in which all applicants are assigned is $M = \{(a_i, h_i) : 1 \leqslant i \leqslant 3\}$, however, $M' = \{(a_2, h_1), (a_3, h_2)\}$ is preferred by two applicants, which is a majority. The relation $\succ$ in this case cycles, hence the absence of a $\succ$-maximal solution (therefore, in general, $\succ$ is not a partial order on $\mathcal{M}$).

The potential absence of a popular matching in a given HA instance can be related all the way back to the observation of Condorcet (1785) that, given k voters who each rank n candidates in strict order of preference, there may not exist a "winner," namely, a candidate who beats all others in a pairwise majority vote. See also Chapter 2.

Abraham et al. (2007b) derived a neat characterization of popular matchings, leading to an $O(m)$ algorithm to check whether a given matching M in I is popular. The same characterization also led naturally to an $O(n + m)$ algorithm for finding a popular matching or reporting that none exists, where $n = n_1 + n_2$. We remark that popular matchings in I can have different sizes, and the authors showed how to extend their algorithm in order to find a maximum popular matching without altering the time complexity. This discussion can be summarized as follows.

Theorem 14.22 (Abraham et al., 2007b). *Let I be an instance of* HA. *There is an $O(n+m)$ algorithm to find a maximum popular matching in I or report that no popular matching exists, where n is the number of applicants and houses, and m is the number of acceptable applicant–house pairs.*

A more complex algorithm, with $O(\sqrt{n}m)$ complexity, can be used to find a maximum popular matching in I or report that no popular matching exists, in the case that preference lists include ties (Abraham et al., 2007b).

McDermid and Irving (2011) showed that the set of popular matchings in an HA instance can be characterized succinctly via a structure known as the *switching graph*. Using this representation they showed that a number of problems can be solved efficiently, including counting popular matchings, sampling a popular matching uniformly at random, listing all popular matchings and finding various types of "optimal" popular matchings.

As a given HA instance need not admit a popular matching, it is natural to weaken the notion of popularity, and seek matchings that are "as popular as possible" in cases where a popular matching does not exist. To this end, McCutchen (2008) defined two versions of "near-popular" matchings, namely, a *least unpopularity factor matching* and a *least unpopularity margin matching*. Also Kavitha et al. (2011) studied the concept of a *popular mixed matching*, which is a probability distribution over matchings that is popular in a precise sense.

Profile-Based Optimal Matchings

Further notions of optimality are based on the *profile* $p(M)$ of a matching M in an HA instance I. Informally, $p(M)$ is an r-tuple whose ith component is the number of applicants who have their ith-choice house, where r is the maximum length of an applicant's preference list.

A matching M is *rank-maximal* (Irving et al., 2006) if $p(M)$ is lexicographically maximum, taken over all matchings in $\mathcal{M}$. Intuitively, in such a matching, the maximum number of applicants are assigned to their first-choice house, and subject to this condition, the maximum number of applicants are assigned to their second-choice house, and so on. A rank-maximal matching need not be of maximum cardinality. To see this, consider the HA instance I_3 in Figure 14.1c and the following matchings in I_3: $M_1 = \{(a_1, h_1), (a_2, h_2)\}$ and $M_2 = \{(a_1, h_3), (a_2, h_1), (a_3, h_2)\}$. Clearly M_1 is rank-maximal and $|M_1| = 2$, whereas $|M_2| = 3$.

In many applications we seek to assign as many applicants as possible. With this in mind, consider $\mathcal{M}^+$, the set of maximum matchings in a given HA instance I. A *greedy maximum matching* is a matching $M \in \mathcal{M}^+$ such that $p(M)$ is lexicographically maximum, taken over all matchings in $\mathcal{M}^+$. Both rank-maximal and greedy maximum matchings maximize the number of applicants with their sth-choice house as a higher priority than maximizing the number of those with their tth-choice house, for any $1 \leqslant s < t \leqslant r$. As a consequence, both of these types of matchings could end up assigning applicants to houses relatively low down on their preference lists.

Consequently, define a *generous maximum matching* to be a matching $M \in \mathcal{M}^+$ such that $p^R(M)$ is lexicographically minimum, taken over all matchings in $\mathcal{M}^+$,

where $p^R(M)$ is the reverse of $p(M)$. That is, M is a maximum cardinality matching that assigns the minimum number of applicants to their rth-choice house, and subject to this, the minimum number to their $(r-1)$th-choice house, and so on.

We collectively refer to rank-maximal, greedy maximum and generous maximum matchings as *profile-based optimal matchings*. Returning to instance I_3 shown in Figure 14.1c, the matching M_2 defined previously is the unique maximum matching and is therefore both a greedy maximum matching and a generous maximum matching.

The following results indicate the complexity of the fastest current algorithms for constructing rank-maximal, greedy maximum and generous maximum matchings in a given HA instance.

Theorem 14.23 (Irving et al., 2006). *Let I be an instance of* HA. *A rank-maximal matching M in I can be constructed in $O(\min(n_1 + r^*, r^*\sqrt{n_1})m)$ time, where n_1 is the number of applicants, m is the number of acceptable applicant–house pairs, and r^* is the maximum rank of an applicant's house in M.*

Theorem 14.24 (Huang and Kavitha, 2012). *Let I be an instance of* HA. *A greedy maximum matching M in I can be constructed in $O(r^*\sqrt{n}m \log n)$ time, where n is the number of applicants and houses, m is the number of acceptable applicant–house pairs, and r^* is the maximum rank of an applicant's house in M. The same time complexity holds for computing a generous maximum matching.*

The algorithms referred to in Theorems 14.23 and 14.24 are also applicable in the more general case that preference list contain ties.

14.4 Concluding Remarks and Further Reading

In this chapter we have tried to cover some of the most important results on matching problems with preferences. However the literature in this area is vast, and due to space limitations, we could only cover a subset of the main results in a single survey chapter. Chapter 11 introduces some of our matching problems within the context of fair resource allocation, namely, *object allocation problems* (HA), *priority-augmented object allocation problems* (SC), and *matching agents to each other* (SMI and HR). The following nonexhaustive list of articles contains normative results for these problems and basic axioms of fair allocation as introduced in Chapter 11 (e.g., resource-monotonicity, population-monotonicity, consistency, converse consistency): Ehlers and Klaus (2004, 2007, 2011), Ehlers et al. (2002), Ergin (2000), Kesten (2009), Sasaki and Toda (1992), and Toda (2006).

One obvious omission has been the Stable Roommates problem (SR), a non-bipartite generalization of SM. However a wider class of matching problems, known as *hedonic games*, which include SR as a special case, are explored in Chapter 15.

Looking ahead, it seems likely that the level of interest in matching under preferences will show no sign of diminishing, and if anything we predict that this field will continue to grow. This is due in no small part to the exposure that the research area has had on a global stage following the award of the Nobel Prize in Economic Sciences to Alvin Roth and Lloyd Shapley in 2012. Another contributing factor is the increasing engagement

by more and more elements of society in forms of electronic communication, thereby easing preference elicitation and centralization of allocation processes.

To conclude, we give some sources for further reading. For more details on structural and algorithmic aspects of SM, HR and SR, we recommend Gusfield and Irving (1989). The second author's monograph (Manlove, 2013) provides an update to Gusfield and Irving (1989) and also expands the coverage to include HA. It expands on the algorithmic results presented in this chapter in particular. For more depth from an economic and game-theoretic viewpoint, the reader is referred to Roth and Sotomayor (1990), which considers issues of strategy in SM and HR in much more detail, and also covers monetary transfer and the Assignment Game. Finally, more recent results that also include economic applications (e.g., school choice and kidney exchange) are reviewed by Sönmez and Ünver (2011) and Vulkan et al. (2013).

Acknowledgments

Bettina Klaus acknowledges financial support from the Swiss National Science Foundation (SNFS). David Manlove is supported by grant EP/K010042/1 from the Engineering and Physical Sciences Research Council. Francesca Rossi is partially supported by the project "KIDNEY—Incorporating patients' preferences in kidney transplant decision protocols," funded by the University of Padova. The authors would like to thank Peter Biró, Felix Brandt, Vincent Conitzer, and Ulle Endriss for detailed comments, which have helped us to improve the presentation of this chapter.