Chapter 10

Two-Sided Matching

10.1 Quick Introduction: Two-Sided Matching Problems 149
10.2 Narrative Description of Two-Sided Matching Problems 150
10.3 Representing the Problem 152
10.4 Stable Matches and the Deferred Acceptance Algorithm 154
10.5 Once More, in More Depth 155
10.6 Generalization: Matching in Centralized Markets 156
10.7 Discussion: Complications 157
10.8 For More Information 159

10.1 Quick Introduction: Two-Sided Matching Problems

This section gives a very quick introduction to two-sided matching problems. Subsequent sections tell the story in more detail, at times repeating some information presented here.

A *match* μ between two sets or sides $\mathcal{A}$ and $\mathcal{B}$ can be represented nicely in tabular form. We pick one set to be the row (player), R, and the other to be the column (player), C, and we impose an arbitrary ordering on them individually. Then in the table element (r_i, c_j)=1 if there is a match—a pairing up—between $r_i \in R$ and $c_j \in C$; otherwise the entry is 0.

In the simple case of what we call a *conventional marriage matching problem* there are n agents on each side and each agent is matched to exactly one agent on the other side. The table then looks like this:

	1	2	3	4	5
1	0	0	0	1	0
2	0	1	0	0	0
3	0	0	0	0	1
4	1	0	0	0	0
5	0	0	1	0	0

That is, this match table or matrix represents one of 5!=120 possible matches between the 5 agents on each side. Taking the column's perspective, a more compact form of this table would be

$$\begin{pmatrix} 4 & 2 & 5 & 1 & 3 \end{pmatrix} \tag{10.1}$$

which we call a *match vector*. Its interpretation is direct: column 1 is matched with row 4, column 2 with row 2, column 3 with row 5, column 4 with row 1, and column 5 with row 3. Notice that in this special case, the match vector is a *permutation* of the numbers 1 through 5. A matrix that is all 0s except for a single 1 in each row and column, as in the match table above, is thus called a *permutation matrix.* For those familiar with matrix

multiplication notice that:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 5 & 1 & 3 \end{pmatrix} \tag{10.2}$$

Points arising:

1. The square matrix arrangement is characteristic of the two-sided matching problem known as the (conventional) *marriage problem.*
2. In the marriage problem, then, a match, μ, is essentially a permutation, of which there are $n!$. This is too many matches to search exhaustively for any but the smallest problems. Even 20! is larger than 10^{18}.
3. The entries in the permutation matrix are 0–1 decision variables. Equivalently, the entries in the associated match vector (e.g., expression (10.1)) are decision variables whose values cover the index values of the rows. So far, we have not specified an objective function for these decision variables (but we will, soon). For the moment, consider that we seek a (solution) procedure for finding a decision (setting of the decision variables in either the match array, or the match vector) that has nice properties, which are yet to be specified.
4. The representation is richer and more flexible than is perhaps evident at first glance. For example, if we have n "men" and n "women" we could add two dummies to each list and specify that if a man (or woman) ranked the dummy above any woman (man), then the man (woman) would prefer not to be "married" at all to being married to a woman (man) ranked below the dummy. And so on.
5. In the marriage problem we say that each agent has a quota of 1. We can easily let one side have agents with quotas larger than 1; we can even let different agents have different quotas. When we do this, we get what is called an *admissions problem.* Think of students (quotas of 1) applying for admission to schools (quotas of hundreds or even thousands).
6. Matching problems, even the special case of marriage problems, occur in practice quite often, as we shall see.

10.2 Narrative Description of Two-Sided Matching Problems

Consider, prototypically, the following problem. We have n men and n women, all of whom would like to get married. We consider no other individuals than those before us, the n men and the n women. Our job is to find a match, μ, in which each man has be paired with one woman and each woman has been paired with one man. The men have rated the women and the women have rated the men. How should we pair them up, assign them to each other, make the match μ? This is called the *marriage problem* (or something similar) in an extensive literature. Cheerfully yielding to modern sentiments, we prefer to call it the *conventional marriage problem,* and shall later consider variations on it. For the present,

however, the mathematical properties of the conventional problem hold plenty of interest for us.

The (conventional) marriage problem is prototypical in two ways. First, there are many purposes, besides marrying off men and women, for which the problem is apt, such as matching roommates in a dormitory, partners at a dance, workers for a task, and so on. Second, the marriage problem is an instance of the much larger class of *2-sided matching problems* in which individuals from two collections are to be paired up. For this more general class we do not require that the two collections be equally sized or even non-overlapping (although most authors do require the latter). Further, we do not require that all of the individuals in either set be matched. And so on.

Indefinitely many complications may legitimately arise in two-sided matching and do in real-world applications. Student applicants and schools are matched, but the schools have a limit on how many students they are willing to admit. Similarly for interns and hospitals. In a variant of the marriage problem, some individuals may prefer to remain single than to be paired with individuals below a certain level of attractiveness. And so on.

And why limit matching to two sides? The interesting possibilities are manifold. But first things first.

The first question we need to ask in dealing with the (conventional) marriage problem is: On what basis should we create matches? Here the key thing to note is that this is (assumed to be) a strategic problem, in distinction to a non-strategic (aka: parametric) problem. That is, we assume that the men and women (and more generally the agents in any n-sided matching problem) have a degree of autonomy in their preferences and their actions. In particular, it will not do just to pair them up willy-nilly. Unhappy couples will find other partners, undoing our match.

Reflections of just this sort led Gale and Shapley, the authors of the seminal paper on two-sided matching [53], to propose *stability* as a requirement for any adequate match. Part of what makes the paper seminal is that an extensive literature since agrees with, or rather presupposes, the requirement of stability.

A match μ is *unstable* if there are two matched couples, call them (r_1, c_1) and (r_2, c_2) such that r_1 prefers to be matched with c_2 rather than c_1 and c_2 prefers to be matched with r_1 rather than r_2. The presumption is they—r_1 and c_2—will tend to get their way, making the match unstable. If a match is not unstable—if there are no such pairs of frustrated couples, more generally called *blocking pairs*—then the match is said to be *stable.*

Note that here we are foreshadowing the notational conventions we shall use later. Instead of men and women to be matched, we shall often prefer to have two sets to be matched, the row set, of which the r_is are members, and the column set, containing the c_is. In any particular case, we may think of the men as being assigned to the row set and the women to the column set, or vice versa. (It will matter which is which.)

We have thus bumped up against issues of problem representation, and shall now address it explicitly in the next section. To foreshadow what is to come, our aim for this chapter is to explore in some depth the concept of stability in the context of the marriage problem. What are its operating characteristics? Does it capture everything we want or need to assess a match or matching procedure? These and other key questions for any application of the marriage problem, and more generally matching problems, will exercise us for the remainder of the chapter.

10.3 Representing the Problem

In the original paper [53], Gale and Shapley assumed that each of the men individually ranked in order of preference each of the women and vice versa. The ranks were 1, 2, ... n, with 1 being the best, most preferred, value and n the least. We shall abide by these assumptions until further notice. (The literature varies on this crucial but inconsequential convention.)

Tables are natural and convenient representation forms for preference ranking information. In an *individual preference table* the rows correspond to individuals of one side or another and the columns to their rank scores for the individuals on the other side. For example, this table

	Women			
Men	1	2	3	4
1	1	2	3	4
2	1	4	3	2
3	2	1	3	4
4	4	2	3	1

shows the "men's" preferences for women in an $n = 4$ marriage problem. Man 2 ranks woman 1 the highest, woman 4 the next highest, then woman 3, and woman 2 the lowest. "Women's" preferences are indicated in a second table

	Men			
Women	1	2	3	4
1	3	4	2	1
2	3	1	4	2
3	2	3	4	1
4	3	2	1	4

where we see that woman 2 likes man 2 the most, followed by man 4, then man 1, and finally man 3.

A *ranking table* combines the individual preference tables. Using the examples above, the resulting ranking table when the men are assigned to be row players and the women to be column players is [53, Example 2]:

	1	2	3	4
1	1,3	2,3	3,2	4,3
2	1,4	4,1	3,3	2,2
3	2,2	1,4	3,4	4,1
4	4,1	2,2	3,1	1,4

If instead we assign the women the role of row players and the men that of column players we get this as our ranking table:

	1	2	3	4
1	3,1	4,1	2,2	1,4
2	3,2	1,4	4,1	2,2
3	2,3	3,3	4,3	1,3
4	3,4	2,2	1,4	4,1

(The one is easily transformed into the other by reversing the pairs in each cell and then taking the transpose.)

We can also represent a matches as a table: 1 if the pair is matched, 0 if not. For example

	Women			
Men	1	2	3	4
1	0	0	1	0
2	0	0	0	1
3	1	0	0	0
4	0	1	0	0

indicates matches between row player (in this case, man) 1 and column player (here, woman) 3. Compressing this we say (1,3). The other matches are (2,4), (3,1) and (4,2).

Notice that a more compact representation of a match is possible (as discussed in §10.1). Choosing to take the column perspective, we can represent the match above as a vector or 1-dimensional table: [3,4,1,2] or

3	4	1	2

This has the additional advantage of letting us see immediately how many possible matches there are for a given ranking matrix: $n!$ where n is the number of agents to a side. In consequence, the search space of matches for a given ranking matrix of size n grows very rapidly with n

n	$n!$
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880
10	3628800
11	39916800
12	479001600

so that anything larger than 12 or so cannot as a practical matter be generated and visited (Knuth's felicitous terms) on a contemporary personal computer.

These are just the counts for the number of possible matches for a given ranking matrix. Given n, how many ranking matrices are there? On the row side, each player can independently choose among $n!$ preference rankings for itself. Since there are n row players, there are $(n!)^n$ possible individual preference matrices for the rows. Similarly for the columns, yielding $(n!)^{2n}$ possible ranking matrices of size n. This number grows even faster than $n!$ (one wants to add another "!").

n	$(n!)^{2n}$
2	16
3	46,656
4	11,007,531,4176
5	6.1917364224e+20

Of course there will be "duplications" of sorts among the $(n!)^{2n}$ possibilities. With $n = 2$, for example, we will have both

	1	2
1	1,1	2,2
2	1,1	2,2

and

	1	2
1	2,2	1,1
2	2,2	1,1

Why not just switch the women's names (column IDs) and call these the same? An argument could be made for this, but we are unpersuaded and shall not pursue this prospect for—rather modest—reduction of complexity.[1]

With so many possible matches for a given ranking table, $n!$, and given the problem of choosing a match, how are we to find a good one? Well, before we can say how to find a good one, we perhaps should say something about what would count as a good one. To that now, plus an algorithm for finding good matches on this criterion.

10.4 Stable Matches and the Deferred Acceptance Algorithm

As we noted above, Gale and Shapley in their paper [53] propose *stability* as a, even the, requirement for any adequate match. Again,

> A match μ is *unstable* if there are two couples, call them (r_1, c_1) and (r_2, c_2) such that r_1 prefers being matched with c_2 rather than c_1 and c_2 prefers to be matched with r_1 rather than r_2. The presumption is they—r_1 and c_2—will tend to get their way, making the match unstable. If a match is not unstable—if there are no such frustrated couples—then the match is said to be *stable.*

In the related literature that has extended this work, stability has continued to figure centrally in evaluation of procedures for two-sided matching. It is easy to see why. Besides its obvious stability property, Gale and Shapley showed that for the marriage problem (and others) there always exists at least one stable matching. In addition, they presented a fast algorithm for finding a stable match. The algorithm, called the *deferred acceptance algorithm*, requires at most $n^2 - 2n + 2$ passes, each of which is relatively cheap. So, while $n!$ is huge (for $n > 12$ or so), $n^2 - 2n + 2$ is not. Even for $n = 1000$, $n^2 - 2n + 2$ is still under a million, at 998002.

The procedure, more precisely, the deferred acceptance algorithm as applied to the marriage problem, is quite simple. Here is Gale and Shapley's original description of it.

> To start, let each boy propose to his favorite girl. Each girl who receives more than one proposal rejects all but her favorite from among those who have proposed to her. However, she does not accept him yet, but keeps him on a string to allow for the possibility that someone better may come along later.
>
> We are now ready for the second stage. Those boys who were rejected now propose to their second choices. Each girl receiving proposals chooses her favorite from the group consisting of the new proposers and the boy on her string, if any. She rejects all the rest and again keeps the favorite in suspense.

[1] Nevertheless the mathematical problem of defining what counts as a duplication and why, and then counting the number of possible ranking matrices, excluding duplications, is itself a delightful one.

> We proceed in the same manner. Those who are rejected at the second stage propose to their next choices, and the girls again reject all but the best proposal they have had so far.
>
> Eventually (in fact, in at most $n^2 - 2n + 2$ stages) every girl will have received a proposal, for as long as any girl has not been proposed to there will be rejections and new proposals, but since no boy can propose to the same girl more than once, every girl is sure to get a proposal in due time. As soon as the last girl gets her proposal the "courtship" is declared over, and each girl is now required to accept the boy on her string. [53, pages 12–3]

10.5 Once More, in More Depth

Again, now with more technical detail, in a two-sided matching problem we are given two sets (sides) of agents, X and Y, and are asked to find a match, μ, consisting of a decision (in or out?) for each pair $(x, y), x \in X, y \in Y$. It is helpful to view a match as represented by a table, $\mathbf{M}$, of size $|X| \times |Y|$, based upon arbitrary orderings of X and Y. The element $m_{i,j}$ of $\mathbf{M}$ equals 1, if $x_i \in X$ is matched with $y_j \in Y$; otherwise the element is 0. Thus the element $m_{i,j}$ of $\mathbf{M}$ represents the *pair* (x_i, y_j). Matchings pair up agents from X and Y, which need not in general be the size.

Particular matching problems come with particular requirements on μ (or $\mathbf{M}$) as well as X and Y. For example in the conventional marriage matching problem (as discussed above), we require that $|X| = |Y| = n$; the number of "men" equals the number of "women." We further require of any (valid) match that each man (or member of X (Y)) be paired (or matched) with exactly one woman (or member of Y (X)), and vice versa. In terms of $\mathbf{M}$, this means that there is one 1 in each row and one 1 in each column. $\mathbf{M}$ is thus a permutation matrix and the number of possible valid matches is $n!$. In *admissions* matching problems, which are used to model, for example, interns applying to hospitals and students applying to schools, one side of the problem, say X, is much larger than the other. There are more doctors than hospitals, more students than schools. Unlike conventional marriage problems, however, one side will have *quotas* larger than 1. Each doctor and each student will have a quota of 1, but each hospital and each school may have a much larger quota and admit many doctors or students. Thus, in a valid match for an admissions problem, each agent on one side (X or Y) is paired to one or more agents on the counter party side, not to exceed the counter party's quota. With students as X, and schools as Y, $\mathbf{M}$ will have one 1 in each row, and each column will have a number of 1s not to exceed the quota of the corresponding school.

Many other variations are possible and are met in practice for two-sided matching problems, a topic we take up in the sequel. For now we focus on a more technical question. How do and how should centralized market agencies produce matches? In practice, some form of, variation on, the deferred acceptance algorithm of Gale and Shapley [53] is used to find a *stable* match, which is then used. A match is *unstable* if there is a pair of matched pairs—(x_i, y_j) and (x_k, y_l)—such that x_i prefers to be matched with y_l over being matched with y_j and y_l prefers to be matched with x_i to being matched with x_k. Stable matches are the ones that are not unstable.

Although the procedure was discovered and used independently before, the deferred acceptance algorithm (DAA) was first published in a paper by Gale and Shapley [53], quoted above. Figure 10.1 (after [53]) presents the DAA in pseudocode.

1. Each $x \in X$ ranks each $y \in Y$, and each $y \in Y$ ranks each $x \in X$.
2. *Matched* $\longleftarrow \emptyset$, *Unmatched* $\longleftarrow \emptyset$.
3. For each y, *string.y* $\longleftarrow []$.
4. Each $x \in X$ proposes to its most preferred y, appending x to *string.y*.
5. Each y with *length*(*string.y*) > 1 (i.e., with more than one proposal), retains in the string its most preferred member of the string, and removes the rest, adding them to *Unmatched*.
6. Each x remaining on some string is added to *Matched*.
7. Do while *Unmatched* $\neq \emptyset$:
   (a) *Matched* $\longleftarrow \emptyset$, *Unmatched* $\longleftarrow \emptyset$.
   (b) Each $x \in$ *Unmatched* proposes to its most preferred y, among the ys that have not already rejected x, appending x to *string.y*.
   (c) Each y with *length*(*string.y*) > 1 (i.e., with more than one proposal), retains in the string its own most preferred member of the string, and removes the rest, adding them to *Unmatched*.
   (d) Each x remaining on some string is added to *Matched*.
8. Stop. Each x is matched to a distinct y, which has x as the sole member of its string.

FIGURE 10.1: Pseudocode for the deferred acceptance algorithm (DAA) for the simple marriage matching problem, Xs proposing to Ys.

10.6 Generalization: Matching in Centralized Markets

In the usual case, markets are *distributed*, with buyers and sellers mostly on their own in finding each other and in negotiating terms of trade. Distributed markets may fail in one way or another, however. A common response is to create a *centralized* market organized by a third party whose responsibility it is to set the conditions of trade, for example the price, based on the bids and asks from the buyers and sellers. Many electricity markets are organized in this way. Deregulated electricity producers, for example, offer supply schedules to a third party, often called the *independent systems operator* or ISO, who aggregates the supply schedules, observes the market demand, and sets the price of electricity (for a given period of time).

Quite a number of labor markets are similarly centralized, most famously, markets in which physicians are matched to hospitals for internships [136]. Roughly speaking, the individual doctors submit their rankings of hospitals, the hospitals submit their rankings of doctors, and a third party organization undertakes to match doctors with hospitals. This is an example of a *two-sided matching* problem, which problems are the subject of this chapter.

As Roth notes [136], two-sided matching models are often natural for representing markets, in which agents need to be paired up and in which price alone is insufficient to govern

the exchanges. Thus, matching markets are inherently different from the usual commodity markets of textbook economics.

> The price does all the work [in a commodity market], bringing the two of you [buyer and seller] together at the price at which supply equals demand. On the NYSE, the price decides who gets what.
>
> But in *matching markets,* prices don't work that way. [For example]...In the working world, courtship often goes both ways, with employers offering good salaries, perks, and prospects for advancement, and applicants signaling their passion, credentials, and drive. College admissions and labor markets are more than a little like courtship and marriage: each is a two-sided matching market that involves searching and wooing on both sides. A market involves matching whenever price isn't the only determinant of who gets what. [137, pages 5–6]

Men and women want to find partners, workers want to find employment and employers want to find workers, and so on. Moreover, many of these markets, decentralized or free markets, experience failure and unsatisfactory performance in practice. They experience unraveling, e.g., offers to students are made earlier and earlier; congestion, e.g., offerers have insufficient time to make new offers after candidates have rejected previous offers; and participants engage in disruptive strategic behavior, so that behaving straightforwardly with regard to one's preferences becomes risky, e.g., in scheduling offers and responding to them [17, 136, 137, 138]. In consequence, there is a large and growing number of applications of two-sided matching in which decentralized markets have been replaced by centralized ones, in which a coordinating agency undertakes periodic matching between two sides of a specific market. [136] lists over 40 labor markets, mostly in medical fields; schools in New York and Boston are using centralized markets to match students to schools; [137] describes a number of applications, including kidney exchanges; see also [17].

In brief, matching markets occur naturally and just as naturally often fail, leading (again naturally) to institution of centralized exchanges for them. How can, how do, the exchanges do their jobs? Using the deferred acceptance algorithm or some variant.

10.7 Discussion: Complications

A presumption in matching problems (as distinguished from assignment problems, Chapter 5, which are treated in operations research and employ non-strategic decision making) is that both sides consist of agents who have interests of their own and capacities to act on them. Consequently, matches are ordinarily evaluated in terms of *stability.* Matching problems are inherently strategic, or game-theoretic, and stability is the accepted equilibrium concept. This is in contrast to most or all of the other problems we consider in this book. A match is said to be stable, again, if there is no pair of matched couples in it containing individuals who would prefer to be matched to each other but are not. The thought is that, with stability, unraveling will not happen and without stability it is difficult to prevent unraveling and concomitant market failure. Requiring matches to be stable in the first place will, it is thought, prevent breakup and reformation among pairs and its attendant costs.

The point of departure for this section is the observation that two-sided matches can be evaluated—and for many applications should be evaluated—according to several objectives, particularly stability, equity, and social welfare. For present purposes, by stability we mean the count of unstable pairs of matched couples in a match. This should be minimized and

at 0 the match is stable.[2] By equity we mean the sum of the absolute differences in the preference scores of each matched pair. We will be scoring preference on a ranking 1 to n scale (1 = most preferred, n = least preferred), so this too should be minimized. Finally, by social welfare we mean the sum of the agent scores in the match. Again, since scoring is from low to high, this quantity should also be minimized. (To illustrate, if agents i and j are paired in a match, and i's preference score for j is 5 and j's preference score for i is 3, then the matched pair's contribution to social welfare is the sum, 8, and their contribution to equity is the absolute value of the difference of the scores, $|5 - 3| = 2$.)

Given that we would consider designing or even centralizing a matching market (as is widely done in practice), the question arises of how best to provide the market operators and users with match options that honor these three objectives, and inform the decision makers about them.

Several points by way of additional background on the deferred acceptance algorithm. First, as proved by Gale and Shapley [53], under the special assumptions they made (e.g., preference ranking by agents), the stable marriage problem and the admissions problem (see above) have stable matches and the DAA will find one and will find one quickly ($O(n^2)$). Second, the DAA is asymmetric. One side proposes, the other disposes. Focusing now on the marriage problem, if the men propose, they obtain a stable match that is male optimal in the sense that no man in this match does better in any other stable match. Conversely, the match is female pessimal in the sense that no woman is worse off in any other stable match. And vice versa if the women propose [53, 73, 107].

This asymmetry is a general characteristic of the DAA in its various forms. It occasions the important question of whether better matches exist and can be found. To this end, we will want to look at stable matches that may be preferable to the matches found by the DAA. And as announced above, we want to examine both social welfare and equity.

These issues could be neatly resolved by finding all of the stable solutions for a given problem and comparing them with respect to equity, social welfare, and whatever other measures of performance are relevant. Predictably, however, this is an intractable problem. Irving and Leather [79] have shown that the maximum number of stable matches for the simple marriage matching problem grows exponentially in n (see also [73, 75]). Further, they provide a lower bound on the maximum by problem size. Remarkably, for a problem as small as $n = 32$, the lower bound is $104,310,534,400$ [79]. Further, they establish that the problem of determining the number of stable matches is #P-complete, and hence in a theoretically very difficult complexity class. These are, of course, extreme-case results, but very little is known about average cases. So we are left to rely upon heuristics, and indeed on metaheuristics.

The first complication for the deferred acceptance algorithm thus is the fact that there can be very many stable matches and those found by the algorithm may not be very attractive on grounds other than mere stability. Simulation results with artificial data, e.g., [74, 91, 99, 110], indicate that indeed both evolutionary algorithms and agent-based models can find stable and nearly stable decisions that are superior on other grounds to those found by the deferred acceptance algorithm. Unfortunately, real world matching data is hard to come by and it remains to be seen whether real world matching problems can yield attractive alternative matches with these methods.

There is a second complication pertaining to the deferred acceptance algorithm: beyond the simple cases discussed by Gale and Shapley [53], when such real world considerations as "two body problems" arise (e.g., two partners who wish to be placed in internships near one another, siblings (perhaps more than two) who wish to attend the same school or schools

[2]If the count of unstable pairs is 1, there is no guarantee that if the two pairs rematch by exchanging partners the resulting match would be stable. In fact, it could have a higher count of unstable pairs [107].

near one another, etc.), then the deferred acceptance algorithm may fail in various ways and there may not even be any stable matches to be found. Such problems are handled somewhat in an ad hoc fashion in existing applications [137]. Whether and to what extent metaheuristics or other algorithms can contribute usefully to more complex existing or potential new matching markets is an intriguing possibility that lies on the frontier of current knowledge.

10.8 For More Information

The original paper by Gale and Shapley [53] was published in a journal aimed at high school teachers of mathematics. It remains a delightful exemplar of accessible mathematical insight and clarity. Alvin Roth has made something of a career designing matching markets, for which he shared the Nobel Prize with Lloyd Shapley. Roth's popular account of matching markets [137] is readable and quite accessible, although lacking in technical details and fine points of matching problems (e.g., there is no mention of asymmetry and pessimal matches).

At an intermediate level of mathematical discussion, Knuth's book [107] is a gem. Roth and Sotomayor's book [138] is an essential read from an economic perspective, while the book by Gusfield and Irving [73] is essential from a computer science perspective.

See [5] for a related treatment using agent-based modeling.

This page intentionally left blank