Models in Microeconomic Theory

Expanded Second Edition

Martin J. Osborne Ariel Rubinstein





https://www.openbookpublishers.com

© 2020, 2023 (expanded second edition) Martin J. Osborne and Ariel Rubinstein



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs license (CC BY-NC-ND 4.0). This license allows you to share, copy, distribute, and transmit the work providing you do not modify the work, you do not use the work for commercial purposes, you attribute the work to the authors, and you provide a link to the license. Attribution should not in any way suggest that the authors endorse you or your use of the work and should include the following information.

Martin J. Osborne and Ariel Rubinstein, *Models in Microeconomic Theory* (Expanded Second Edition), Cambridge, UK: Open Book Publishers, 2023, https://doi.org/10.11647/0BP.0361.

To access detailed and updated information on the license, please visit https://www.openbookpublishers.com/product/1171#copyright.

For further details on CC BY-NC-ND licenses, see https://creativecommons.org/ licenses/by-nc-nd/4.0/. All external links were active at the time of publication unless otherwise stated and have been archived via the Internet Archive Wayback Machine at https://archive.org/web.

Updated digital material and resources associated with this book are available at https: //www.openbookpublishers.com/product/1171#resources.

Every effort has been made to identify and contact copyright holders and any omission or error will be corrected if notification is made to the publisher.

ISBN Paperback: 978-1-80511-121-4 ISBN Hardback: 978-1-80511-122-1 ISBN Digital (PDF): 978-1-80511-123-8 DOI: 10.11647/OBP.0361

This book is available in two versions, one that uses feminine pronouns and one that uses masculine pronouns. This version uses feminine pronouns.

Version: 2023.5.30 (s)

18 Matching

Consider the following problem. Some individuals in a society are *X*'s and others are *Y*'s. Every individual of each type has to be matched with one and only one individual of the other type. For example, managers have to be matched with assistants, or pilots have to be matched with copilots. Each *X* has preferences over the *Y*'s and each *Y* has preferences over the *X*'s. Every individual prefers to be matched than to remain unmatched. We look for matching methods that result in sensible outcomes given any preferences.

18.1 The matching problem

We denote the set of *X*'s by *X* and the set of *Y*'s by *Y* and assume that they have the same number of members. Each $x \in X$ has a preference relation over the set *Y* and each $y \in Y$ has a preference relation over the set *X*. We assume that all preferences are strict (no individual is indifferent between any two options).

Definition 18.1: Society and preference profile

A *society* (*X*, *Y*) consists of finite sets *X* and *Y* (of individuals) with the same number of members. A *preference profile* (\succeq^i)_{$i \in X \cup Y$} for the society (*X*, *Y*) consists of a strict preference relation \succeq^i over *Y* for each $i \in X$ and a strict preference relation \succeq^i over *X* for each $i \in Y$.

A matching describes the pairs that are formed. Its definition captures the assumption that each individual has to be matched with exactly one individual of the other type.

Definition 18.2: Matching

A *matching* for a society (X, Y) is a one-to-one function from X to Y. For a matching μ and $x \in X$ we refer to $(x, \mu(x))$ as a *match*.

We discuss matching methods, which map preference profiles into matchings. That is, a matching method specifies, for each preference profile, who is matched with whom.

Chapter of *Models in Microeconomic Theory* by Martin J. Osborne and Ariel Rubinstein. Version 2023.5.30 (s). (c) 2023 Martin J. Osborne and Ariel Rubinstein CC BY-NC-ND 4.0. https://doi.org/10.11647/OBP.0361.18

Definition 18.3: Matching method

A *matching method* for a society is a function that assigns a matching to each preference profile for the society.

The following example treats one side (the *Y*'s), like the houses in the models of Chapter 8, and takes into account only the preferences of the *X*'s.

```
Example 18.1: Serial dictatorship
```

The *X*'s, in a pre-determined order, choose *Y*'s, as in the serial dictatorship procedure. Each *X* chooses from the *Y*'s who were not chosen by previous *X*'s. This procedure always results in a matching, and thus defines a matching method.

Here are two more examples of matching methods.

Example 18.2: Minimizing aggregate rank

For any pair (x, y) consisting of an X and a Y, let $n_x(y)$ be y's rank in x's preferences and let $n_y(x)$ be x's rank in y's preferences. Attach to each pair (x, y) a number $I(x, y) = \alpha(n_x(y), n_y(x))$, where α is a function increasing in both its arguments (for example $\alpha(n_1, n_2) = n_1 + n_2$). The number $\alpha(n_x(y), n_y(x))$ is a measure of the dissatisfaction of individuals x and y with their match. The matching method chooses the matching that minimizes the sum of I(x, y) over all pairs (x, y) (or one such matching if more than one exists).

Example 18.3: Iterative selection of the best match

Start by choosing a pair for which the value of I(x, y) defined in Example 18.2 is minimal over all pairs (x, y). Remove the members of the chosen pair from *X* and *Y* and choose a pair for which the value of I(x, y) is minimal over the smaller sets. Continue iteratively in the same way.

18.2 Gale-Shapley algorithm

We now consider a matching method that has an interesting description and some attractive properties. The algorithm that defines it has two versions, one in which the *X*'s initiate matches, and one in which the *Y*'s do so. We describe the former.



Figure 18.1 An example of the first stages of the Gale-Shapley matching method. The *X*'s are indicated by a green background and the *Y*'s by an orange background.

The algorithm proceeds in a series of stages. At the end of each stage, some pairs are tentatively matched; at the end of the procedure, the existing tentative matches become final.

Stage 1

Each *X* chooses her favorite *Y*. If every *Y* is chosen by exactly one *X* then the algorithm ends with that matching. If some *Y* is chosen by more than one *X*, then every such *Y* chooses her favorite *X* from among those who chose her and is tentatively matched with this *X*. She rejects the other *X*'s who chose her. All *X*'s who were not chosen by a *Y* are left unmatched. (See Figure 18.1 for an illustration of the start of the algorithm in the case that **1** is the favorite of **1** and **2** and **4** is the second best of **1**; **4** is the favorite of **3** and **4** and **2** is the second best of **3**; and **1** prefers **2** to **1** and **4** prefers **4** to **3**.)

Stage t + 1 for $t \ge 1$

At the start of stage t + 1, tentative matches exist from stage t, and some individuals are unmatched. Each unmatched X chooses her favorite Y among those who have not rejected her in the past. Some of the Y's thus chosen may already be tentatively matched with X's. (In the example in Figure 18.1, that is the case in Stage 2 for 4.) Each Y chooses her favorite X from the set consisting of the X with whom she was tentatively matched at the end of stage tand the unmatched X's who chose her, resulting in new tentative matches.

Stopping rule

The process ends when every X is tentatively matched with a Y, in which case the tentative matches become final.

The following definition specifies the algorithm formally, but if you find the previous description clear you may not need to refer to it.

Procedure: Gale-Shapley algorithm

Given a society (X, Y), the *Gale-Shapley algorithm for* (X, Y) *in which X*'s *initiate matches*, denoted GS^X , has as input a preference profile $(\succeq^i)_{i \in X \cup Y}$ for (X, Y) and generates a sequence $(g_t, R_t)_{t=0,1,\dots}$ where

- $g_t : X \to Y \cup \{unmatched\}$ is a function such that no two X's are mapped to the same Y
- R_t is a function from *X* to subsets of *Y*.

A pair (g_t, R_t) is the state of the algorithm at the end of stage t. If $g_t(x) \in Y$ then $g_t(x)$ is the Y with whom x is tentatively matched and $R_t(x)$ is the set of all Y's who rejected x through stage t.

```
Definition of (g_0, R_0)
```

Every *X* is *unmatched* and every set $R_0(x)$ is empty:

 $g_0(x) = unmatched$ for all $x \in X$ $R_0(x) = \emptyset$ for all $x \in X$.

```
Definition of (g_{t+1}, R_{t+1}) given (g_t, R_t)
```

For each $y \in Y$ let

 $A_{t+1}(y) = \{x \in X : y \text{ is best in } Y \setminus R_t(x) \text{ according to } \succeq^x\}.$

That is, $A_{t+1}(y)$ is the set of the *X*'s who choose *y* in stage t + 1. (Note that if $g_t(x) = y$ then $x \in A_{t+1}(y)$.)

Now

$$g_{t+1}(x) = \begin{cases} y & \text{if } x \text{ is best in } A_{t+1}(y) \text{ according to } \succeq^y \\ unmatched & \text{otherwise} \end{cases}$$

and

$$R_{t+1}(x) = \begin{cases} R_t(x) & \text{if } g_{t+1}(x) \in Y \\ R_t(x) \cup \{y \in Y : x \in A_{t+1}(y)\} & \text{if } g_{t+1}(x) = unmatched. \end{cases}$$

Thus $R_{t+1}(x)$ is equal to $R_t(x)$ unless x is rejected at stage t by some $y \in Y$, in which case that y (and only her) is added to $R_t(x)$.

Stopping rule

The process ends at stage *T* if $g_T(x) \in Y$ for all $x \in X$.

The description of the algorithm talks of individuals choosing matches. But that language should not be taken literally. We use it simply to describe the algorithm attractively (as we did for serial dictatorship). The Gale-Shapley algorithm simply defines a function that attaches a matching to every preference profile. Soon we prove that the algorithm indeed always ends with a matching, but first we give an example.

Example 18.4

Consider the society with four *X*'s and four *Y*'s and the following preference profiles, where *X*'s are shown in green and *Y*'s in orange.

1:	$1 \succ 4 \succ 2 \succ 3$	1:	$3 \succ 1 \succ 2 \succ 4$
2:	$2 \succ 3 \succ 1 \succ 4$	2:	$3 \succ 2 \succ 4 \succ 1$
3:	$4 \succ 2 \succ 3 \succ 1$	3:	$4 \succ 3 \succ 2 \succ 1$
4:	$4 \succ 3 \succ 1 \succ 2$	4:	$1 \succ 4 \succ 3 \succ 2$

X's initiate matches We first apply GS^X (where *X*'s initiate matches) to this preference profile.

Stage 1: 1 chooses 1, 2 chooses 2, and both 3 and 4 choose 4. 4 prefers 4 to 3 and thus rejects 3 and keeps 4. That is, $g_1(1) = 1$, $g_1(2) = 2$, $g_1(3) = unmatched$, and $g_1(4) = 4$, and $R_1(1) = \emptyset$, $R_1(2) = \emptyset$, $R_1(3) = \{4\}$, and $R_1(4) = \emptyset$.

1	\longrightarrow	1
2	\longrightarrow	2
3		3
4		4

Stage 2: 3 chooses 2, who is tentatively matched with 2. 2 prefers 3 to 2 and so rejects 2. That is, $g_2(1) = 1$, $g_2(2) = unmatched$, $g_2(3) = 2$, and $g_2(4) = 4$, and $R_2(1) = \emptyset$, $R_2(2) = \{2\}$, $R_2(3) = \{4\}$, and $R_2(4) = \emptyset$.

1	\longrightarrow	1
2		2
3		3
4	\longrightarrow	4

Stage 3: 2 chooses 3. Every *Y* is now matched with a unique *X*, and the process ends. We have $g_3(1) = 1$, $g_3(2) = 3$, $g_3(3) = 2$ and $g_4(4) = 4$.

1	\longrightarrow	1
2	\checkmark	2
3	\frown	3
4	\longrightarrow	4

Y's initiate matches Now we apply GS^{Y} to the profile.

Stage 1: 1 and 2 choose 3, 3 chooses 4, and 4 chooses 1. 3 prefers 2 to 1 and so rejects 1.



Stage 2: 1 chooses 1, who prefers 1 to 4 and so rejects 4.



Stage 3: 4 chooses 4, who prefers 4 to 3 and so rejects 3.



Stage 4: 3 chooses 3, who prefers 2 to 3 and so rejects 3.

1	←]
2		2
3	£	3
4	←	4

Stage 5: **3** chooses **2**. Every *X* is now matched with a unique *Y*, and the process ends.

1	←	1
2	\checkmark	2
3	\sim	3
4	←	4

Note that the matchings in these two examples are the same. But don't jump to conclusions: for many preference profiles the matchings generated by GS^X and GS^Y differ.

We now show that for every profile of preferences the algorithm is well defined and eventually terminates in a matching.

Proposition 18.1: Gale-Shapley algorithm yields a matching

For any society and any preference profile for the society, the Gale-Shapley algorithm is well defined and generates a matching.

Proof

We consider the algorithm GS^X , in which *X*'s initiate matches. The argument for GS^Y is analogous.

We first show that the algorithm is always well defined. That is, we argue that for no preference profile does GS^X have a stage t at which some $x \in X$ has been rejected by all Y's (that is, $R_t(x) = Y$). Note that when a Y rejects an X, she remains tentatively matched with some other X. Thus if some $x \in X$ has been rejected after stage t by every Y, then every Y is tentatively matched with an X. But the number of members of X is the same as the number of members of Y, so it must be that x is tentatively matched (that is $g_t(x) \in Y$), and in particular has not been rejected by $g_t(x)$, a contradiction.

We now show that the algorithm terminates. At each stage at which the algorithm continues, at least one X is rejected. Thus if the algorithm did not stop, we would reach a stage at which one of the X's would have been rejected by every Y, which we have shown is not possible.

Finally, the algorithm terminates when no *X* is unmatched, so that the outcome is a matching.

18.3 Gale-Shapley algorithm and stability

We now consider properties of the matching generated by the Gale-Shapley algorithm. We classify a matching as unstable if there are two individuals who prefer to be matched with each other than with the individuals with whom they are matched. That is, matching μ is unstable if for some $x \in X$ and $y \in Y$ with $y \neq \mu(x)$, x prefers y to $\mu(x)$ (the Y with whom x is matched) and y prefers x to $\mu^{-1}(y)$ (the X with whom y is matched). In this case both x and y want to break the matches assigned by μ and match with each other. A matching is stable if no such pair exists.

Definition 18.4: Stable matching

For a society (X, Y) and preference profile $(\succeq^i)_{i \in X \cup Y}$, a matching μ is *stable* if there is no pair $(x, y) \in X \times Y$ such that $y \succ^x \mu(x)$ and $x \succ^y \mu^{-1}(y)$.

To illustrate the concept of stable matching we give an example showing that the serial dictatorship algorithm, in which the members of *X* sequentially choose members of *Y* from those who were not chosen previously, may generate an unstable match.

Example 18.5

Consider a society ({1,2,3}, {1,2,3}) with the following preferences.

1:	$1 \succ 2 \succ 3$	1:	$2 \succ 3 \succ 1$
2:	$1 \succ 2 \succ 3$	2:	$3 \succ 2 \succ 1$
3:	$1 \succ 2 \succ 3$	3:	$1 \succ 3 \succ 2$

Apply the serial dictatorship algorithm in which the *X*'s choose *Y*'s in the order 1, 2, 3. The algorithm yields the matching in which each $i \in X$ is matched to $i \in Y$. This matching is unstable because $2 \in X$ and $1 \in Y$ both prefer each other to the individual with whom they are matched.

We now show that the Gale-Shapley algorithm generates a stable matching.

Proposition 18.2: Gale-Shapley algorithm yields a stable matching

For any society and any preference profile for the society, the Gale-Shapley algorithm generates a stable matching.

Proof

Consider a society (X, Y) and preference profile $(\succeq^i)_{i \in X \cup Y}$. Denote by μ the matching generated by the algorithm GS^X . Assume that for $x \in X$ and $y \in Y$, individual x prefers y to $\mu(x)$ ($y \succ^x \mu(x)$). Then at some stage before x chose $\mu(x)$, she must have chosen y and have been rejected by her in favor of another X. Subsequently, y rejects an X only in favor of a preferred X. Thus it follows that y prefers $\mu^{-1}(y)$, the X with whom she is eventually matched, to x. We conclude that no pair prefers each other to the individual with whom she is assigned by μ , so that μ is stable.

Many matchings may be stable. The GS^X algorithm finds one stable matching and GS^Y finds another, possibly the same and possibly not. Is it better to be on the side that initiates matches? Is any stable matching better for one of the *X*'s than the matching generated by GS^X ? The next result answers these questions: for each $x \in X$ the matching generated by GS^X is at least as good for *x* as any other stable matching, and in particular is at least as good as the matching generated by GS^Y .

Proposition 18.3: *GS^X* algorithm yields best stable matching for *X*'s

For any society and any preference profile for the society, no stable matching is better for any X than the matching generated by the GS^X algorithm.

Proof

Let μ be the (stable) matching generated by GS^X for the society (X, Y) and preference profile $(\succeq^i)_{i \in X \cup Y}$. Suppose that ψ is another stable matching and $\psi(x) \succ^x \mu(x)$ for some $x \in X$. Let X^* be the set of all $x \in X$ for whom $\psi(x) \succ^x \mu(x)$. For each $x \in X^*$ denote by t(x) the stage in the GS^X algorithm at which x chooses $\psi(x)$ and is rejected by her. Let $x_0 \in X^*$ for whom $t(x_0)$ is minimal among the $x \in X^*$. Let $y_0 = \psi(x_0)$ and let $x_1 \in X$ be the individual y_0 chose when she rejected x_0 in the GS^X algorithm. That is, at stage $t(x_0)$ individuals x_0 and x_1 choose y_0 , who rejects x_0 in favor of x_1 ($x_1 \succ^{y_0} x_0$). Let $\psi(x_1) = y_1$.

$$t(x_0)$$

$$x_0 \xrightarrow{----} y_0 = \psi(x_0)$$

$$x_1 \xrightarrow{\cdots} y_1 = \psi(x_1)$$

If $y_0 \succ^{x_1} y_1$ then each member of the pair (x_1, y_0) prefers the other member to the individual assigned her by ψ , contradicting the stability of ψ . Therefore $y_1 \succ^{x_1} y_0$. But then in the GS^X algorithm x_1 must choose y_1 and be rejected before stage $t(x_0)$, so that $t(x_1) < t(x_0)$, contradicting the minimality of $t(x_0)$.

We complement this result with the observation that for each Y no stable matching is worse than the matching generated by the GS^X algorithm.

Proposition 18.4: GS^X algorithm yields worst stable matching for *Y*'s

For any society and any preference profile for the society, no stable matching is worse for any Y than the matching generated by the GS^X algorithm.

Proof

Consider a society (X, Y) and suppose that for the preference profile $(\succeq^i)_{i \in X \cup Y}$ the algorithm GS^X leads to the (stable) matching μ . Suppose another stable matching ψ is worse for some $y \in Y$: $x_2 = \psi^{-1}(y) \prec^y \mu^{-1}(y) = x_1$. Let $y_1 = \psi(x_1)$.

$$\frac{\mu \qquad \psi}{\substack{y \leftrightarrow x_1 \qquad y \leftrightarrow x_2 \\ y_1 \leftrightarrow x_1}}$$

By Proposition 18.3, $y = \mu(x_1) \succ^{x_1} \psi(x_1) = y_1$. Given $x_1 \succ^y \psi^{-1}(y) = x_2$, x_1 and y prefer each other to the individuals with whom they are matched in ψ , contrary to the assumption that ψ is stable.

Problems

- 1. Stable matching and Pareto stability.
 - *a.* Show that any stable matching is Pareto stable. That is, for no preference profile is another matching (stable or not) better for one individual and not worse for every other.
 - b. Give an example of a Pareto stable matching that is not stable.
- 2. *Pareto stability for the X's.* Show that the outcome of the GS^X algorithm is weakly Pareto stable for the *X*'s. That is, no other matching is better for every *X*.
- 3. *All X's have the same preferences*. Assume that all *X*'s have the same preferences over the *Y*'s.
 - *a.* How many stages does the GS^X algorithm take?
 - *b*. Which serial dictatorship procedure yields the same final matching as the GS^X procedure?
 - *c*. Show that in this case GS^X and GS^Y lead to the same matching.
- 4. *Matching with unequal groups*. Group *A* has *m* members and group *B* has *n* members, with m < n. Each individual has strict preferences over the members of the other group. Let GS^A be the Gale-Shapley algorithm in which the *A*'s initiate the matches and let GS^B be the algorithm in which the *B*'s initiate the matches.
 - *a.* Show that if m = 1 then GS^A and GS^B yield the same matching.
 - *b*. Explain why GS^A and GS^B do not necessarily yield the same matching if m = 2.
 - *c*. Show that in both GS^A and GS^B every individual in *A* is matched with one of her *m* most preferred members of *B*.
- 5. *Clubs*. Assume that 3*n* students have to be allocated to three programs, each with capacity *n*. Each student has strict preferences over the set of programs and each of the programs has strict preferences over the students. Describe an algorithm similar to the Gale-Shapley algorithm, define a notion of stability, and show that for any preference profile your algorithm ends with a stable matching.
- 6. *Manipulation*. The Gale-Shapley algorithm is not strategy-proof and thus is not immune to manipulation. Specifically, if each individual is asked to

Notes

report preferences and the GS^X algorithm is run using the reported preferences, then for some preference profiles some individual is better off, according to her true preferences, if she reports preferences different from her true preferences.

To see this possibility, consider the following preference profile for a society with three *X*'s and three *Y*'s.

1:	$1 \succ 2 \succ 3$	1:	$2 \succ 1 \succ 3$
2:	$2 \succ 1 \succ 3$	2:	$1 \succ 2 \succ 3$
3:	$1 \succ 2 \succ 3$	3:	$1 \succ 2 \succ 3$

Show that **1** can benefit by reporting preferences different from her true preferences.

7. *The roommate problem*. A society contains 2n individuals. The individuals have to be partitioned into pairs. Each individual has a (strict) preference over the other individuals. An assignment μ is a one-to-one function from the set of individuals to itself such that if $\mu(i) = j$ then $\mu(j) = i$. An assignment is stable if for no pair of individuals does each individual prefer the other member of the pair to her assigned partner. Construct an example of a preference profile (with four individuals) for which no assignment is stable.

Notes

The chapter is based on Gale and Shapley (1962).