10.3 House Allocation Problem

The House allocation problem is a model for understanding the allocation of indivisible goods. It involves a set N of n agents, each owning a unique house and a strict preference ordering over all n houses. The objective is to reallocate the houses among the agents in an appropriate way. A modern version of the same would replace houses by kidneys.

While any possible (strict) preference ordering over the homes is permitted, the set of preferences over allocations is restricted. In particular, an agent is indifferent between all allocations that give her the same house. Therefore the Gibbard–Satterthwaite Theorem does not apply in this setting.

One could select an allocation of homes in a variety of ways, perhaps so as to optimize some function of the preferences and then investigate if the resulting allocation rule is *strategy-proof*. However, this ignores an important feature not present in earlier examples. In this environment, agents control the resources to be allocated. Therefore an allocation can be subverted by a subset of agents who might choose to break away and trade among themselves. For this reason it is natural to focus on allocations that are invulnerable to agents opting out.

Number each house by the number of the agent who owns that house. An allocation is an n vector a whose ith component, a_i, is the number of the house assigned to agent i. If a is the initial allocation then $a_i = i$. For an allocation to be feasible, we require that $a_i \neq a_j$ for all $i \neq j$. The preference ordering of an agent i will be denoted $\succ_i$ and $x \succ_i y$ will mean that agent i ranks house x above house y. Denote by A the set of all feasible allocations. For every $S \subseteq N$ let $A(S) = \{z \in A : z_i \in S \ \forall i \in S\}$ denote the set of allocations that can be achieved by the agents in S trading among themselves alone. Given an allocation $a \in A$, a set S of agents is called a **blocking coalition** (for a) if there exists a $z \in A(S)$ such that for all $i \in S$ either $z_i \succ_i a_i$ or $z_i = a_i$ and for at least one $j \in S$ we have that $z_j \succ_j a_j$. A blocking coalition can, by trading among themselves, receive homes that each strictly prefers (or is equivalent) to the home she receives under a, with at least one agent being strictly better off. The set of allocations that is not blocked by any subset of agents is called the **core**.

The reader will be introduced to the notion of the core in Chapter 15 (Section 15.2) where it will be defined for a cooperative game in which utility is transferable via money (a TU game). The house allocation problem we consider is an example of a cooperative game with nontransferable utility (an NTU game). The definition of the core offered here is the natural modification of the notion of TU core to the present setting.

The theorem below shows the core to be nonempty. The proof is by construction using the top trading cycle algorithm (TTCA).

Definition 10.5 (Top Trading Cycle Algorithm) Construct a directed graph using one vertex for each agent. If house j is agent i's kth ranked choice, insert a directed edge from i to j and color the edge with color k. An edge of the form (i, i) will be called a loop. First, identify all directed cycles and loops consisting only of edges colored 1. The strict preference ordering implies that the set of such cycles and loops is node disjoint. Let N_1 be the set of vertices (agents) incident to these cycles. Each cycle implies a sequence of swaps. For example,

suppose $i_1 \to i_2 \to i_3 \to \cdots \to i_r$ is one such cycle. Give house i_1 to agent i_r, house i_r to agent i_{r-1}, and so on. After all such swaps are performed, delete all edges colored 1. Repeat with the edges colored 2 and call the corresponding set of vertices incident to these edges N_2, and so on. The TTCA yields the resulting matching.

This algorithm is used to prove the following result.

Theorem 10.6 *The core of the house allocation problem consists of exactly one matching.*

PROOF We prove that if a matching is in the core, it must be the one returned by the TTCA.

Under the TTCA, each agent in N_1 receives his favorite house, i.e., the house ranked first in his preference ordering. Therefore, N_1 would form a blocking coalition to any allocation that does not assign to all of those agents the houses they would receive under the TTCA. That is, any core allocation must assign N_1 to houses just as the TTCA assigns them.

Given this fact, the same argument applies to N_2: Under the TTCA, each agent in N_2 receives his favorite house *not including* those houses originally endowed by agents in N_1. Therefore, if an allocation is in the core and the agents in N_1 are assigned each other's houses, then agents in N_2 must receive the same houses they receive under the TTCA.

Continuing the argument for each N_k proves that if an allocation is in the core, then it is the one determined by the TTCA. This proves that there is at most one core allocation.

To prove that the TTCA allocation is in the core, it remains to be shown that there is no other blocking coalition $S \subseteq N$. This is left to the reader. □

To apply the TTCA, one must know the preferences of agents over homes. Do they have an incentive to truthfully report these? To give a strongly positive answer to this question, we first associate the TTCA with its corresponding direct revelation mechanism. Define the **Top Trading Cycle (TTC) Mechanism** to be the function (mechanism) that, for each profile of preferences, returns the allocation computed by the TTCA.

Theorem 10.7 *The TTC mechanism is* strategy-proof.

PROOF Let π be a profile of preference orderings and a the allocation returned by TTCA when applied to π. Suppose that agent $j \in N_k$ for some k misreports her preference ordering. Denote by π' the new profile of preference orderings. Let a' the allocation returned by TTCA when applied to π'. If the TTCA is not *strategy-proof* $a'_i >^i a_i$. Observe that $a_i = a'_i$ for all $i \in \bigcup_{r=1}^{k-1} N_r$. Therefore, $a'_i \in N \setminus \{\bigcup_{r=1}^{k-1} N_r\}$. However, the TTCA chooses a_i to be agent i's top ranked choice from $N \setminus \{\bigcup_{r=1}^{k-1} N_r\}$ contradicting the fact that $a'_i >^i a_i$. □

If we relax the requirement that preferences be strict, what we had previously called a blocking set is now called a **weakly** blocking set. What we had previously called the

core is now called the *strict* core. With indifference, a **blocking** set S is one where *all* agents in S are *strictly* better off by trading among themselves. Note the requirement that all agents be strictly better off. The *core* is the set of allocations not blocked by any set S.

When preferences are strict, every minimal weakly blocking set is a blocking set. To see this, fix a weakly blocking set S. An agent in S who is not made strictly better off by trade among agents in S must have been assigned their own home. Remove them from S. Repeat. The remaining agents must all be allocated houses that make them strictly better off. Hence, when preferences are strict the core and strict core coincide. With indifference permitted, the strict core can be different from the core. In fact, there are examples where the strict core is empty and others where it is not unique. Deciding emptiness of the strict core is polynomial in $|N|$.

Another possible extension of the model is to endow the agents with more than one good. For example, a home and a car. Clearly, if preferences over pairs of goods are sufficiently rich, the core can be empty. It turns out that even under very severe restrictions the core can still be empty. For example, when preferences are separable, i.e., one's ranking over homes does not depend on which car one has.

10.4 Stable Matchings

The stable matching problem was introduced as a model of how to assign students to colleges. Since its introduction, it has been the object of intensive study by both computer scientists and economists. In computer science it used as vehicle for illustrating basic ideas in the analysis of algorithms. In economics it is used as a stylized model of labor markets. It has a direct real-world counterpart in the procedure for matching medical students to residencies in the United States.

The simplest version of the problem involves a set M of men and a set W of women. Each $m \in M$ has a strict preference ordering over the elements of W and each $w \in W$ has a strict preference ordering over the men. As before the preference ordering of agent i will be denoted $\succ_i$ and $x \succ_i y$ will mean that agent i ranks x above y. A **matching** is an assignment of men to women such that each man is assigned to at most one woman and vice versa. We can accommodate the possibility of an agent choosing to remain single as well. This is done by including for each man (woman) a dummy woman (man) in the set W (M) that corresponds to being single (or matched with oneself). With this construction we can always assume that $|M| = |W|$.

As in the house allocation problem a group of agents can subvert a prescribed matching by opting out. In a manner analogous to the house allocation problem, we can define a blocking set. A matching is called **unstable** if there are two men m, m' and two women w, w' such that

(i) m is matched to w,
(ii) m' is matched to w', and
(iii) $w' \succ_m w$ and $m \succ_{w'} m'$

The pair (m, w') is called a **blocking pair**. A matching that has no blocking pairs is called **stable**.

Example 10.8 The preference orderings for the men and women are shown in the table below

$\succ_{m_1}$	$\succ_{m_2}$	$\succ_{m_3}$	$\succ_{w_1}$	$\succ_{w_2}$	$\succ_{w_3}$
w_2	w_1	w_1	m_1	m_3	m_1
w_1	w_3	w_2	m_3	m_1	m_3
w_3	w_2	w_3	m_2	m_2	m_2

Consider the matching $\{(m_1, w_1), (m_2, w_2), (m_3, w_3)\}$. This is an unstable matching since (m_1, w_2) is a blocking pair. The matching $\{(m_1, w_1), (m_3, w_2), (m_2, w_3)\}$, however, is stable.

Given the preferences of the men and women, is it always possible to find a stable matching? Remarkably, yes, using what is now called the deferred acceptance algorithm. We describe the male-proposal version of the algorithm.

Definition 10.9 (Deferred Acceptance Algorithm, male-proposals) First, each man proposes to his top-ranked choice. Next, each woman who has received at least two proposals keeps (tentatively) her top-ranked proposal and rejects the rest. Then, each man who has been rejected proposes to his top-ranked choice among the women who have not rejected him. Again each woman who has at least two proposals (including ones from previous rounds) keeps her top-ranked proposal and rejects the rest. The process repeats until no man has a woman to propose to or each woman has at most one proposal. At this point the algorithm terminates and each man is assigned to a woman who has not rejected his proposal. Notice that no man is assigned to more than one woman. Since each woman is allowed to keep only one proposal at any stage, no woman is assigned to more than one man. Therefore the algorithm terminates in a matching.

We illustrate how the (male-proposal) algorithm operates using Example 10.8 above. In the first round, m_1 proposes to w_2, m_2 to w_1, and m_3 to w_1. At the end of this round w_1 is the only woman to have received two proposals. One from m_3 and the other from m_2. Since she ranks m_3 above m_2, she keeps m_3 and rejects m_2. Since m_3 is the only man to have been rejected, he is the only one to propose again in the second round. This time he proposes to w_3. Now each woman has only one proposal and the algorithm terminates with the matching $\{(m_1, w_2), (m_2, w_3), (m_3, w_2)\}$. It is easy to verify that the matching is stable and that it is different from the one presented earlier.

Theorem 10.10 *The male propose algorithm terminates in a stable matching.*

PROOF Suppose not. Then there exists a blocking pair (m_1, w_1) with m_1 matched to w_2, say, and w_1 matched to m_2. Since (m_1, w_1) is blocking and $w_1 \succ_{m_1} w_2$, in the proposal algorithm, m_1 would have proposed to w_1 before w_2. Since m_1 was not matched with w_1 by the algorithm, it must be because w_1 received a proposal from a man that she ranked higher than m_1. Since the algorithm matches her to m_2 it follows that $m_2 \succ_{w_1} m_1$. This contradicts the fact that (m_1, w_1) is a blocking pair. □

One could just as well have described an algorithm where the women propose and the outcome would also be a stable matching. Applied to the example above, this would produce a stable matching different from the one generated when the men propose. Thus, not only is a stable matching guaranteed to exist but there can be more than 1. If there can be more than one stable matching, is there a reason to prefer one to another? Yes. To explain why, some notation.

Denote a matching by μ. the woman assigned to man m in the matching μ is denoted $\mu(m)$. Similarly, $\mu(w)$ is the man assigned to woman w. A matching μ is **male-optimal** if there is no stable matching ν such that $\nu(m) \succ_m \mu(m)$ or $\nu(m) = \mu(m)$ for all m with $\nu(j) \succ_j \mu(j)$ for at least one $j \in M$. Similarly define **female-optimal**.

Theorem 10.11 *The stable matching produced by the (male-proposal) Deferred Acceptance Algorithm is male-optimal.*

PROOF Let μ be the matching returned by the male-propose algorithm. Suppose μ is not male optimal. Then, there is a stable matching ν such that $\nu(m) \succ_m \mu(m)$ or $\nu(m) = \mu(m)$ for all m with $\nu(j) \succ_j \mu(j)$ for at least one $j \in M$. Therefore, in the application of the proposal algorithm, there must be an iteration where some man j proposes to $\nu(j)$ before $\mu(j)$ since $\nu(j) \succ_j \mu(j)$ and is rejected by woman $\nu(j)$. Consider the first such iteration. Since woman $\nu(j)$ rejects j she must have received a proposal from a man i she prefers to man j. Since this is the first iteration at which a male is rejected by his partner under ν it follows that man i ranks woman $\nu(j)$ higher than $\nu(i)$. Summarizing, $i \succ_{\nu(j)} j$ and $\nu(j) \succ_i \nu(i)$ implying that ν is not stable, a contradiction. □

Clearly one can replace the word "male" by the word "female" in the statement of the theorem above. It is natural to ask if there is a stable matching that would be optimal with respect to both men and women. Alas, no. The example above has two stable matchings: one male optimal and the other female optimal. At least one female is strictly better off under the female optimal matching than the male optimal one and no female is worse off. A similar relationship holds when comparing the two stable matchings from the point of view of the men.

A stable matching is immune to a pair of agents opting out of the matching. We could be more demanding and ask that no subset of agents should have an incentive to opt out of the matching. Formally, a matching μ' **dominates** a matching μ if there is a set $S \subset M \cup W$ such that for all $m, w \in S$, both (i) $\mu'(m), \mu'(w) \in S$ and (ii) $\mu'(m) \succ_m \mu(m)$ and $\mu'(w) \succ_w \mu(w)$. Stability is a special case of this dominance condition when we restrict attention to sets S consisting of a single couple. The set of undominated matchings is called the **core** of the matching game. The next result is straightforward.

Theorem 10.12 *The core of the matching game is the set of all stable matchings.*

Thus far we have assumed that the preference orderings of the agents is known to the planner. Now suppose that they are private information to the agent. As before we can associate a direct revelation mechanism with an algorithm for finding a stable matching.

Theorem 10.13 *The direct mechanism associated with the male propose algorithm is* strategy-proof *for the males.*

PROOF Suppose not. Then there is a profile of preferences $\pi = (\succ_{m_1}, \succ_{m_2}, \ldots, \succ_{m_n})$ for the men, such that man m_1, say, can misreport his preferences and obtain a better match. To express this formally, let μ be the stable matching obtained by applying the male proposal algorithm to the profile π. Suppose that m_1 reports the preference ordering $\succ_*$ instead. Let ν be the stable matching that results when the male-proposal algorithm is applied to the profile $\pi^1 = (\succ_*, \succ_{m_2}, \ldots, \succ_{m_n})$. For a contradiction, suppose $\nu(m_1) \succ_{m_1} \mu(m_1)$. For notational convenience we will write $a \succeq_m b$ to mean that $a \succ_m b$ or $a = b$.

First we show that m_1 can achieve the same effect by choosing an ordering $\succ$ where woman $\nu(m_1)$ is ranked first. Let $\pi^2 = (\succ, \succ_{m_2}, \ldots, \succ_{m_n})$. Knowing that ν is stable with respect to the profile π^1 we show that it is stable with respect to the profile π^2. Suppose not. Then under the profile π^2 there must be a pair (m, w) that blocks ν. Since ν assigns to m_1 its top choice with respect to π^2, m_1 cannot be part of this blocking pair. Now the preferences of all agents other than m_1 are the same in π^1 and π^2. Therefore, if (m, w) blocks ν with respect to the profile π^2, it must block ν with respect to the profile π^1, contradicting the fact that ν is a stable matching under π^1.

Let λ be the male propose stable matching for the profile π^2. Since ν is a stable matching with respect to the profile π^2. As λ is male optimal with respect to the profile π^2, it follows that $\lambda(m_1) = \nu(m_1)$.

Thus we can assume that $\nu(m_1)$ is the top-ranked woman in the ordering $\succ_*$. Next we show that the set $B = \{m_j : \mu(m_j) \succ_{m_j} \nu(m_j)\}$ is empty. This means that all men, not just m_1, are no worse off under ν compared to μ. Since ν is stable with respect to the original profile, π this contradicts the male optimality of μ and completes the proof.

Suppose $B \neq \emptyset$. Therefore, when the male proposal algorithm is applied to the profile π^1, each $m_j \in B$ is rejected by their match under μ, i.e., $\mu(m_j)$. Consider the first iteration of the proposal algorithm where some m_j is rejected by $\mu(m_j)$. This means that woman $\mu(m_j)$ has a proposal from man m_k that she ranks higher, i.e., $m_k \succ_{\mu(m_j)} m_j$. Since m_k was not matched to $\mu(m_j)$ under μ it must be that $\mu(m_k) \succ_{m_k} \mu(m_j)$. Hence $m_k \in B$, otherwise

$$\mu(m_j) \succeq m_k \nu(m_k) \succeq_{m_k} \mu(m_k) \succ_{m_k} \mu(m_j),$$

which is a contradiction.

Since $m_k \in B$ and m_k has proposed to $\mu(m_j)$ at the time man m_j proposes, it means that m_k must have been rejected by $\mu(m_k)$ prior to m_j being rejected, contradicting our choice of m_j. □

The mechanism associated with the male propose algorithm is not *strategy-proof* for the females. To see why, it is enough to consider example. The male propose algorithm returns the matching $\{(m_1, w_2), (m_2, w_3), (m_3, w_1)\}$. In the course of the algorithm the only woman who receives at least two proposals is w_1. She received proposals from m_2 and m_3. She rejects m_2 who goes on to propose to w_3 and the algorithm terminates.

Notice that w_1 is matched with her second choice. Suppose now that she had rejected m_3 instead. Then m_3 would have gone on to proposes to w_2. Woman w_2 now has a choice between m_1 and m_3. She would keep m_3 and reject m_1, who would go on to propose to w_1. Woman w_1 would keep m_1 over m_2 and in the final matching be paired with a her first-rank choice.

It is interesting to draw an analogy between the existence of stable matchings and that of Walrasian equilibrium. We know (Chapter 6) that Walrasian equilibria exist. Furthermore, they are the solutions of a fixed point problem. In the cases when they can be computed efficiently it is because the set of Walrasian equilibria can be described by a set of convex inequalities. The same can be said of stable matchings. The set of stable matchings is fixed points of a nondecreasing function defined on a lattice. In addition, one can describe the set of stable matchings as the solutions to a set of linear inequalities.

10.4.1 A Lattice Formulation

We describe a proof of the existence of stable matchings using Tarski's fixed point theorem. It will be useful to relax the notion of a matching. Call an assignment of women to men such that each man is assigned to at most one woman (but a woman may be assigned to more than one man) a **male semimatching**. The analogous object for women will be called a **female semimatching**. For example, assigning each man his first choice would be a male semimatching. Assigning each woman her third choice would be an example of a female semimatching.

A pair of male and female semimatchings will be called a **semimatching** which we will denote by μ, ν, etc. An example of a semi-matching would consist of each man being assigned his first choice and each woman being assigned her last choice.

The woman assigned to the man m under the semi-matching μ will be denoted $\mu(m)$. If man m is assigned to no woman under μ, then $\mu(m) = m$. Similarly for $\mu(w)$. Next we define a partial order over the set of semimatchings. Write $\mu \succeq \nu$ if

(i) $\mu(m) \succ_m \nu(m)$ or $\mu(m) = \mu(m)$ for all $m \in M$ and
(ii) $\mu(w) \prec_w \nu(w)$ or $\mu(w) = \nu(w)$ for all $w \in W$.

Therefore $\mu \succeq \nu$ if all the men are better off under μ than in ν and all the women are worse off under μ than in ν.

Next we define the meet and join operations. Given two semimatchings μ and ν define $\lambda = \mu \vee \nu$ as follows:

(i) $\lambda(m) = \mu(m)$ if $\mu(m) \succ_m \nu(m)$ otherwise $\lambda(m) = \nu(m)$,
(ii) $\lambda(w) = \mu(w)$ if $\mu(w) \prec_w \nu(w)$ otherwise $\lambda(w) = \nu(w)$.

Define $\lambda' = \mu \wedge \nu$ as follows:

(i) $\lambda'(m) = \mu(m)$ if $\mu(m) \prec_m \nu(m)$ otherwise $\lambda(m) = \nu(m)$,
(ii) $\lambda(w) = \mu(w)$ if $\mu(w) \succ_w \nu(w)$ otherwise $\lambda(w) = \nu(w)$.

With these definitions it is easy to check that the set of semimatchings forms a compact lattice.