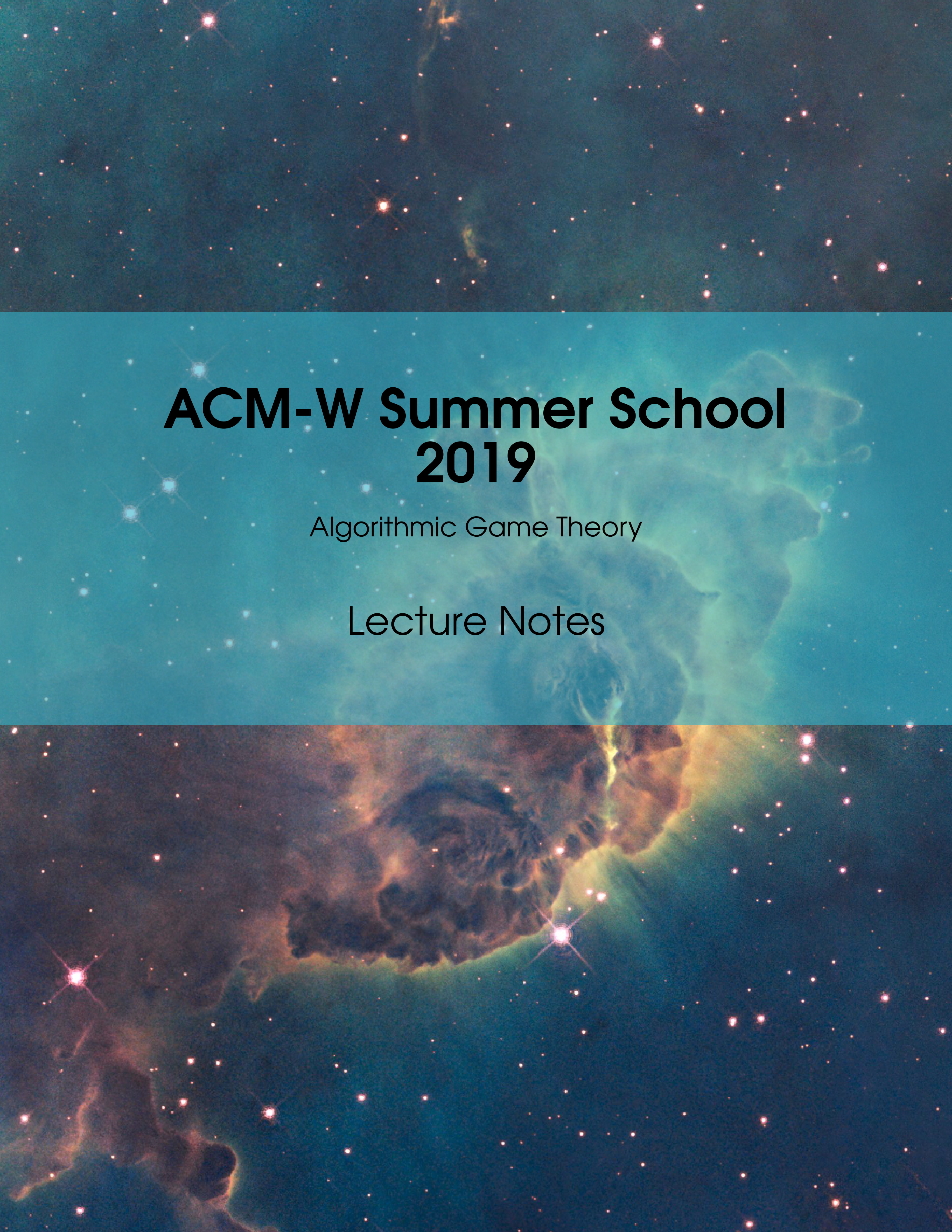


ACM-W Summer School 2019

Algorithmic Game Theory

Lecture Notes

ACM-W SUMMER SCHOOL ON ALGORITHMIC GAME THEORY

HTTP://EVENTS.IITGN.AC.IN/2019/ACMSUMMERSCHOOL/

Based on the lectures given in the ACM-W Summer School on Algorithmic Game Theory.

First release, August 2019

Contents

1. Matchings

1.1 Manipulation in Gale Shapley Algorithm

R Does anyone has an incentive to submit a false preference order?

Assumption 1 1) There is a central authority that executes a man proposing Gale Shaley Algorithm. 2) Men and Women have complete preference orders.

■ **Example 1.1** Take the example considered in Table 1.1, the GS algorithm for them will give the stable matching depicted in Figure 1.1. ■

a_1: b_4 b_2 b_1 b_3	b_1: a_1 a_3 a_2 a_4
a_2: b_1 b_2 b_3 b_4	b_2: a_3 a_1 a_2 a_4
a_3: b_1 b_2 b_3 b_4	b_3: a_1 a_2 a_3 a_4
a_4: b_4 b_1 b_2 b_3	b_4: a_4 a_1 a_2 a_3

Table 1.1: Example 1

1.1.1 Manipulation

Let's assume that a_1 changes its preference order a_1: b_4 b_2 b_3 b_1. If we check the matching using GS algorithm, we can see that M_a remains the same. Now, instead of a_1, if b_1 changes its preference order b_1: a_1 a_2 a_3 a_4. If we now check the matching using GS algorithm, we can see that M_a changes to M_b. Thus, it can be observed that when a woman changes her preference order, M_a stable matching turns out to work better for women.

Fact 1 There is no matching mechanism that ensures both stability and truthfulness.

(a) Man Proposing GS (b) Woman Proposing GS

Figure 1.1: Example 1

R Is it possible for a_1 to change its preference order to have a matching with b_4?

The answer to the above remark is no. This is because no matter how a_1 changes its preference order, b_4 will always prefer a_4 over a_1.

1.2 Cheating Strategy for Men

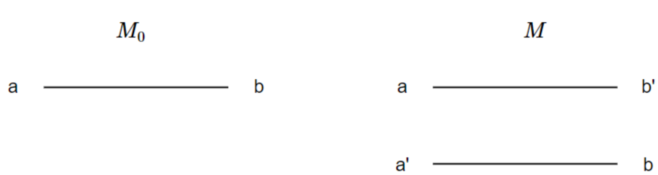
R Is it possible for all men to get strictly better partner than their respective man optimal partners?

To answer the above question, let us look at the lists and the true man optimal partners of all men. We have to check if there is a matching *wrt* true lists of men and women (possibly unstable) where every man gets a better partner than his man-optimal stable partner?

Proof. Let, b : last woman to get engaged (say with a). Therefore, this means that b gets only one proposal and hence does not reject anyone.

M_0 : Man optimal stable matching

M : Can be unstable, but every man gets a better partner than in M_0



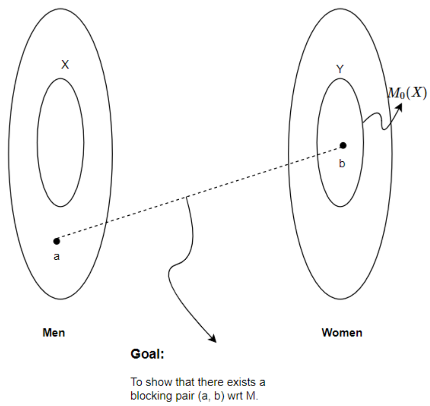
As seen in the above figure, in M_0, a is matched with b. It can be seen that in M, a is matched with b', a more preferred woman and a' is matched with b.

$$a = \dots b' \dots b$$

$$a' = \dots b \dots b''$$

But this will result in a contradiction, as we have already assumed that b can have only one proposal but here, b is having two proposals. Therefore, it is not possible for all men to get a better partner than his man-optimal stable partner. ■

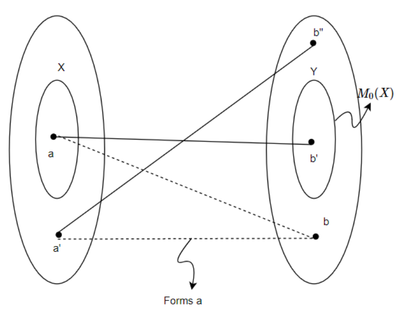
Theorem 1.2.1 There exists no such set of men such that they get better matchings than their man-optimal matchings?



Proof. Let, X : Set of men who get better partners by submitting false lists.
M_0 : Man-Optimal Stable Matching (with all true lists)
M : X get better partners
Y : Set of women who men get matched to in M_0

Goal: To show that there exists a blocking pair with respect to M.

Case 1 $\exists$ a women $b \notin$ Y such that some $a \in$ X has M(a) = b.



Let the preference order of a be:

$$a = \text{... b ... b'}$$

Therefore, b will have the following preference order, that is, it prefers some a' over a:

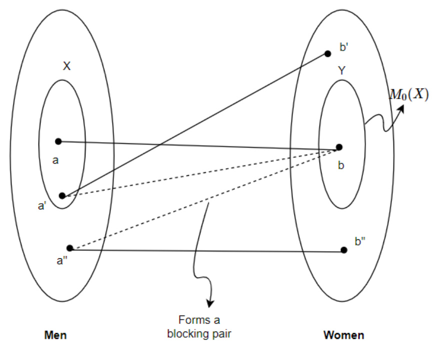
$$b = \text{... a' ... a}$$

else, (a,b) pair will become a blocking pair in M_0

a' is not a part of the set X as b $\notin$ Y. This means that a' will get the same or worse matching pair in M and b will be getting a worse pair in M. Therefore, (a', b) will become a blocking pair in M.

$$\textbf{Fact 2} \implies M_0(X) \neq M(X)$$

Case 2 Every man in X gets a partner in Y in M where M_0 is a man-optimal matching with true preferences.



Let (a, b) be the last pair formed between X and Y.

$$a' = \text{... b ... b'}$$

$$b = \text{... a ... a'}$$

But, if b rejects a' after forming a pair with a, that means (a, b) is not the last pair to be formed.

if b rejects a' because of a's proposal, then (a', b') pair will form after (a, b), which will lead to a contradiction.

$$b = \text{... a ... a" ... a'}$$

where, a' is M(b). a" will thus have the same or worse partner in M(X) as a" is not in X. a" prefers b over M(a") and b prefers a" over M(b). That means, a blocking pair (a", b) is formed in M. Thus, $\nexists$ a set of X of men which get a better matching than their man optimal matching. ■

$$\textbf{Fact 3} \implies M_0(X) = M(X)$$

Conclusion: In a man-proposing algorithm, a set of men can't get better partners than their total true optimal partners.

1.3 Gale Shapely Algorithm : What happens if Women Cheat?

In the above sections, we have already observed what happens when men behave strategically in a Man proposed Gale Shapely algorithm. But, what still remains an interesting question is that, what happens when Women tend to behave strategically in the settings of man proposed GS Algorithm?

Question : Can a woman submit a false preference list such that, she gets a better partner than an optimal partner in a man proposing G.S. Algorithm?

Turns out, we can approach this problem in either one of the following two approaches:

1. Woman/Women submit(s) an incomplete preference list.
2. Women are required to elicit complete preference lists in which they try to manipulate the outcomes by misrepresenting their preferences.

The first approach can further be studied according to the choice of the women over eliciting the incomplete preferences, which we study in the following sections.

1.3.1 Approach 1.1

Strategy : Women truncate their choices at the first/top choice itself.

So, what the women essentially do to manipulate the outcome is that they elicit only the their top choices and nobody else. This is a problematic scenario naturally, because in practice, many of women may have the same women as their top choice, and due to the fact that men elicit their complete preference list, some of the women might go unmatched, which is a worse off scenario as compared to being matched with someone who is relatively less preferred.

$$
\begin{array}{l|l}
a_1\colon b_4\ b_2\ b_1\ b_3 & b_1\colon a_1\ a_3\ a_2\ a_4 \\
a_2\colon b_1\ b_2\ b_3\ b_4 & b_2\colon a_3\ a_1\ a_2\ a_4 \\
a_3\colon b_1\ b_2\ b_3\ b_4 & b_3\colon a_1\ a_2\ a_3\ a_4 \\
a_4\colon b_4\ b_1\ b_2\ b_3 & b_4\colon a_4\ a_1\ a_2\ a_3
\end{array}
$$

Figure 1.2: Approach 1.1

A more optimized strategy, but relatively more difficult to accommodate and realize is as follows:

$$
\begin{array}{l|l}
a_1\colon b_4\ b_2\ b_1\ b_3 & b_1\colon a_1\ a_3\ a_2\ a_4 \\
a_2\colon b_1\ b_2\ b_3\ b_4 & b_2\colon a_3\ a_1\ a_2\ a_4 \\
a_3\colon b_1\ b_2\ b_3\ b_4 & b_3\colon a_1\ a_2\ a_3\ a_4 \\
a_4\colon b_4\ b_1\ b_2\ b_3 & b_4\colon a_4\ a_1\ a_2\ a_3
\end{array}
$$

As stated earlier, one of the women remains unmatched due to the rigid elicitation of preferences, and hence, even though this might not be the dominant strategy for the women for strategic behaviour because they end up getting a worse off result in this manner. Naturally, this leads us to think of another dominant strategy that has been described in the following section.

1.3.2 Approach 1.2

Strategy : Truncate the lists of women at women optimal stable partners.
Claim : This strategy always gives a woman optimal stable matching.

Proof. Say, M is a matching which is stable under original preferences. Our claim is that M continues to be stable after the lists are truncated at the woman optimal preferences.
An important question is that, whether the matching M still remains valid, that is, those edges which previously contributed to the stable matching in the underlying bipartite graph are still present or not.
Assuming that the stable matching M is still valid, following are two important observations:
Observation 1.3.1 No stable matching can give a better partner than every woman's optimal stable partner. (explanation required)
Observation 1.3.2 With truncated lists, woman optimal stable matching *in original instance* is still a valid stable matching *in the newly formed instance*.

This is because, none of the women can participate in a blocking pair in the newly formed instance, since all the preferences that are being truncated are those which enjoy a lower preference ordering as compared to those already present in the stable matching M.

Thus, owing to the above two crucial observations, we conclude that, none of the women get either a better preference nor any of them can be in a worse off condition than the original instance. Thus, a woman who truncates her list at her woman optimal stable partner gets her woman optimal partner even upon the execution of the man proposing Gale Shapely Algorithm. ■

1.3.3 Approach 2

Strategy : What if the complete lists are required?

As a customary practice and a hope to reduce algorithmic burden, say, the central authority imposes a compulsion to submit a complete list of preferences for the men as well as the women. In this approach, we study the strategic manipulation of outcomes by the women, as a result of misrepresenting their choices/preferences. The algorithm following the following example outlines the strategy for women to cheat.

Cheating Strategy : An Example

In this example, the woman b_1 tries to manipulate the outcome in her favour. Although her man optimal partner is a_5, she elicits a dicey preference order(I guess as follows : $b_1 : a_4a_1a_2a_3a_5$ and then $b_1 : a_4a_3a_1a_2a_5$) (chiefly constructed by inspecting the intermediate steps of the Gale Shapely Algorithm) and then tries to run the G.S. Algorithm, and hence finds that she ends up getting a better partner than the one she gets by eliciting truthfully.

Can a Woman ensure that she gets a woman optimal partner by submitting a false complete list?

As a matter of fact, however appealing it seems for a woman to try out all the experiments, there always exists an example of the following kind which does not allow woman to ensure a woman optimal partner by submitting a complete list.

a_1: b_1 b_2 b_3	b_1: a_2 a_3 a_1
a_2: b_2 b_3 b_1	b_2: a_3 a_1 a_2
a_3: b_3 b_1 b_2	b_3: a_1 a_2 a_3

Table 1.2: Ineffectiveness of Women's Choices

In a man proposed G.S. Algorithm, the decision of partnering would not even consult the women's preferences, since there does no conflict arises when we run the algorithm.
As you can observe, none of the women's choices are even considered here, and a straight forward allocation of women to men is made.

Can a woman get a worse off partner as a result of experimentation by some other woman?

The result was justified intuitively on a high level, although a hand wavy explanation is sufficient. So, if some woman does experimentation by juggling around its preferences so as to tailor them in order to obtain better preferences, as observed in the example, at least some of the other women get still more number of options to choose from. But, the other choices will be accepted by the women, if and only if they are strictly better than the current choice of the women and will get rejected if accepting them leaves the woman in a worse off condition than that in the original instance.
And hence, owing to the fact that even though a woman elicits strategically, the men continue to elicit truthfully, and hence, the woman can get either strictly better preference or the same, but not a worse off partner.
Observation 1.3.3 Even if one woman votes strategically, any other woman does not *suffer*, i.e. gets either a strictly more preferred partner, or gets the same partner, but not a worse partner.

As an attempt to formalize the cheating strategy for women, we present the following algorithm:

Algorithm Woman_Cheat :
Initialization : Say, the woman b tries to manipulate the outcome by misrepresenting its preferences.

1. Look at all the proposals that b receives.
2. For each man a who proposes to b,
   - put a in front of b's current list.
   - look at the new proposals received.
3. Let $\mathcal{N}$ be the set of men who ever proposed b in 1 and 2 above. The bet man for the woman b from $\mathcal{N}$ is the best partner that she can get by cheating with complete preference lists.

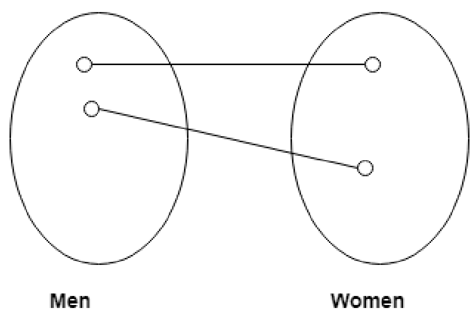
After arriving at results in which either one of the two sets plays strategically, it is but natural to ask whether or not, a specific set of women or men strategically collaborate to elicit their preferences strategically.

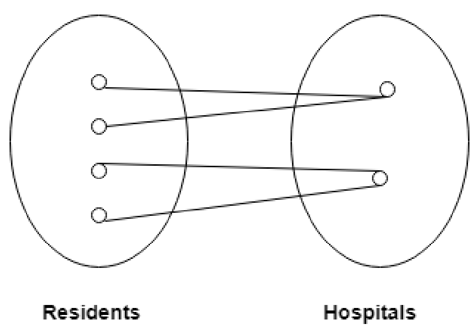
Question : Can a subset of men and women form a coalition and hence cheat strategically?

Although the answer to the above question is no, the formal proof cementing the answer's integrity is out of the scope of this lecture. Although, a hand-wavy explanation of the above question refers to the proof of the part in which we study strategic behaviour by men, one can exhibit a blocking pair among the truthful people.

1.4 MANY TO ONE STABLE MATCHING

In the previous sections, we studied the a broad application of matching between men and women, which is characterized by one woman accepting only one man and vice versa. Although widely in use, we also come across some famous examples, wherein, each agent of one set is matched on to more than one agents in the other set. Typically, such situations arise in cases where it is the responsibility of a central allocation authority to match these sets of agents.
A famous but easier to analyse version of this problem is called *many to one* matching. To make it easier to analyse the many to one model, we shall be studying a classical problem called the *Hospital Residents' problem (HR)* and try to figure out matching with some nice properties associated with it. Refer figure for a more semantic understanding of the one to one and many to one mappings.





Hospital Resident Matching Problem

An instance I of Hospital Resident Problem involves a set $R = \{r_1, \ldots, r_{n_1}\}$ of hospital *residents* and a set $H = \{h_1, \ldots, h_{n_2}\}$ a set of *hospitals*.
The overall setup of the problem is such that, each resident wants to get into one of his preferred hospitals for internship, and on the other hand, every hospital has a preference order on whom to select, and has a maximum capacity defined by the function c, such that,

$$\forall h \in H : \text{capacity} = c(h)$$

1.4.1 The Notion of Stability

Before we can further go into the study of stability for the HR Problem instance, we need to understand a definition, which gets us a command over drawing an analogy of the notion of stability between the one to one and many to one mapping.

Definition 1.4.1 — Under-subscription. A hospital h is said to be under-subscribed under a many one matching M if, it admits residents less than it total capacity.

Now, as you might recall, the notion of stability in the man woman matching was that, there does not exist any pair (a,b) which is a blocking pair for the matching M. Symmetrically, we need to tweak the definition of blocking pair just a bit to get a notion of stability in Hospital Residents (HR) Problem.

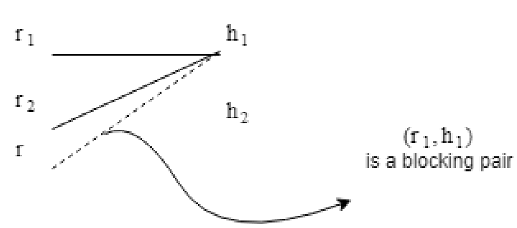
Definition 1.4.2 — Blocking HR Pair. A pair (r,h) blocks a matching M if:

1. r prefers h to $M(r)$.
2. h prefers r to at least 1 resident $M(h)$ (or) h is under subscribed i.e. not matched upto its capacity.

Thus in a stable matching, each resident gets one hospital and each hospital (h) gets $\leq c(h)$ residents, such that, none of them is a blocking pair.

r: h_1 h_2	h_1: r r_1 r_2
r_1: h_1	h_2: r_2
r_2: h_2	

Table 1.3: Example 1



1.4.2 Gale Shapely Algorithm : HR Problem

The G.S. Algorithm remains almost the same for the HR problem as we know from the man woman partnership problem, although, we need to tweak the details a notch up, so as to accommodate the many to one behaviour of the underlying bi-partite graph.

Modified Gale Shapely Algorithm

Residents propose in the order of their preferences as the men used to earlier. Hospitals accept the proposals if:

- they have a vacancy.
- they have a less preferred resident assigned to them, in which case, they kick out that resident, and accommodate the current resident.

Output : A stable matching, independent of the order of proposals

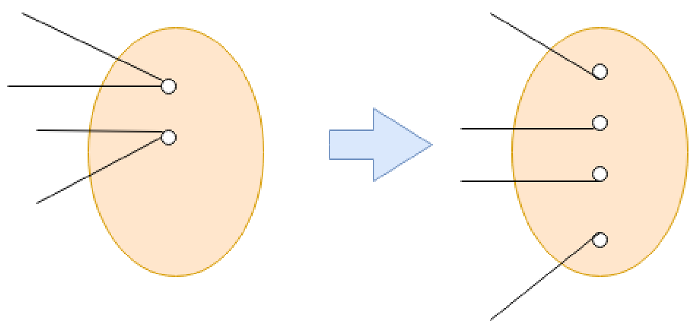
1.4.3 Properties and Some Nice Results Revisited

As an important result, in man - woman partnership, we asserted and proved that the same set of people in men as well as women are left out in all the stable matching possible. An analogous result can be proved in the settings of HR problem, which is that in every stable matching, the same set of residents remain unmatched and the same set of hospitals remain under-subscribed.

Question

Can a hospital get different number of residents in different stable matching?

Turns out, the answer to this question is a No. Even more so, this is an easy observation if we consider a simple modification to the underlying bipartite graph.



Thus, what we necessarily do is that, we break down each hospital into single capacity hospitals and hence get a $1-1$ matching.
In other words,

$$\forall h \in H : c(h) = k \implies h^1, \ldots, h^k \tag{1.1}$$

Where all $h^1, \ldots, h^k$ are single capacity dummy hospitals.

Such a reduced instance of the HR problem is called as a *blown out instance*. Effectively, running G.S. Algorithm on the equivalent new $1-1$ instance follows all the properties that a normal $1-1$ instance follows, thus answering the above question (because all the residents that remain unmatched in one stable matching remain so in all the other stable matchings. Thus, it is impossible for any hospital to get different number of residents in different stable matching.

As a matter of fact, all the instances of HR problem can be converted into equivalent $1-1$ matching, although the proof of which is out of the scope of this lecture.

Question

If every objective can be achieved by blowing up a HR instance into a $1-1$ matching, why not do so for every algorithm, and confer results?

The reason lies inside a facet of the problem which we have not yet concentrated upon, that is, the time complexity.
Say, in the underlying bipartite graph, there are m edges. Thus, running G.S. Algorithm gets us an asymptotic time complexity of $\mathscr{O}(m)$, which is quadratic on the number of vertices in the graph.
As for the analysis of the *blown up instance*,

$$c(h) \leq |R| \tag{1.2}$$

Where $|R|$ is the cardinality of the resident set.
$\therefore$ Blown up instance has $\leq m\times$(no.of residents' edges)
$\therefore$ G.S. for blown up instance = $\mathscr{O}(m.|R|)$

Theorem 1.4.1 — Rural Hospital's Theorem. If h is an under-subscribed hospital in a stable matching, then h gets the same set of residents in all stable matchings.

Observation 1.4.2 If all hospitals are under subscribed in a stable matching, then the instance has a unique stable matching. This implies that every resident gets her first choice because of the fact that the under subscribed hospitals are not supposed to reject anyone.

Observation 1.4.3 The property of unique stable matching in HR problem contrasts the one in $1-1$ as, in $1-1$ matching, if there exists a unique stable matching, it need not be necessary that all the residents/men get their top choices.

Proof. Let's say, M_0 is a Resident Optimal Stable Matching.
M is any other stable matching.
Say, for a resident r and a hospital h $M_0(r) = h$, h is under subscribed in M. Can r be matched to some other hospital in M, say,

$$h' = M(r) \tag{1.3}$$

We know that h is r's stable partner (due to G.S. Algorithm). Thus, r's preferences must look like the following:

$$r : ...h....h'.... \tag{1.4}$$

$\therefore (r,h)$ blocks $M \implies M_0(h) \subseteq M(h)$

Thus, $M(h)$ has to have those elements which are present in the set M_0 and it also may have more.
But,

$$|M_0(h)| = |M(h)|$$

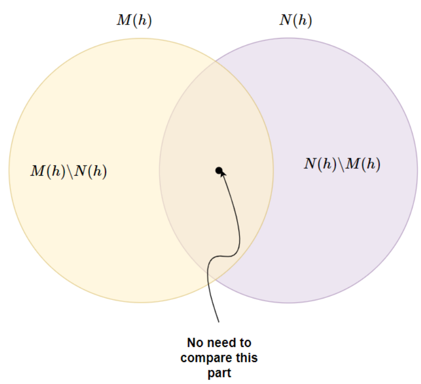
, (by blowing up the instance.)
$\therefore M_0(h) = M(h)$ ■

Question 1 Take a hypothetical situation in which a hospital h has the following preference order:

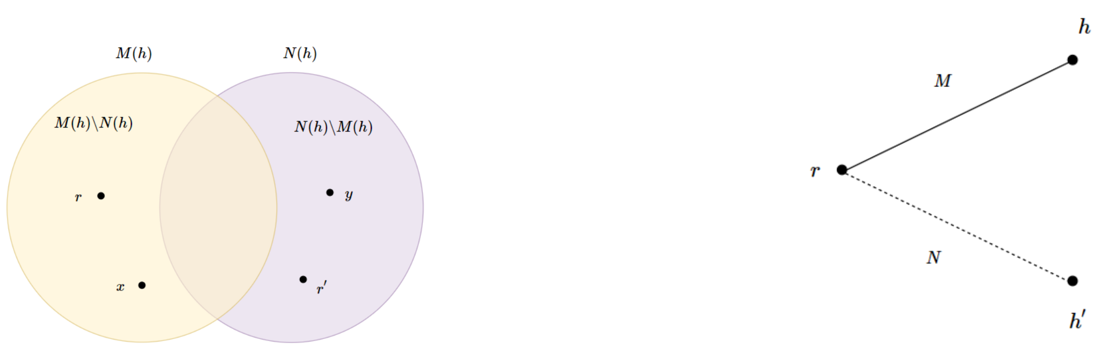
$$\text{h}: \quad r_1 \quad r_2 \quad r_3 \quad r_4$$

Then, is it possible that h gets the 1^{st} and 4^{th} preference in one stable matching and 2^{nd} and 3^{rd} preference in another stable matching? The following section discusses whether such a thing is possible or not.

Let M and N be two stable matchings where M(h) and N(h) denote the set of residents allotted to the hospital h. We will try to find a way to compare the residents in M(h) and N(h) with each other. We will also compare which matching is more preferred over other matching, i.e., which set of residents are more preferred by h. While comparing these two sets M(h) and N(h), it won't make sense to compare the common residents in both the matchings. Thus, we need to compare $M(h) \setminus N(h)$ with $N(h) \setminus M(h)$



Claim 1 If r prefers M over N then h prefers r' over r, where, r is the worst resident in hospital h in the stable matching $M(h) \setminus N(h)$ and r' is the worst resident in the hospital h in the stable matching $N(h) \setminus M(h)$.



(a) M, N : Stable Matchings

(b) Preference of r

Figure 1.3: Proof for Claim 1

Proof. Let us assume that there is a resident r which is in set $M(h) \backslash N(h)$ for hospital h. Also, r is the worst resident for h in matching M. Similarly, r' is the worst resident for matching N in set $N(h) \backslash M(h)$ for hospital h. As both r and r' come in $M(h) \backslash N(h)$ and $N(h) \backslash M(h)$ respectively, that means that both are not the residents in hospital h for both the stable matchings M and N. That means, is r is the resident of h in M, it might be a resident of some hospital h' in N. Similarly, r' is the resident of h in N, it might be a resident of some hospital h" in M.

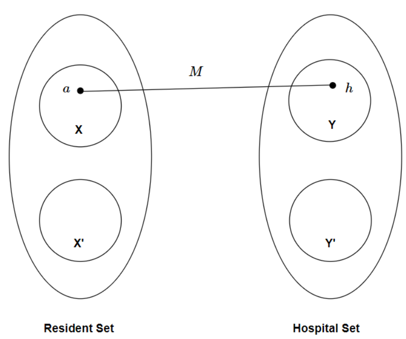
In M, r prefers h over h'. But if the preference of h is ... r ... r' ..., then it will prefer r over r' in N. That means, both h and r get their worse choices in N even though they have a possible better partner available. Thus, (r, h) pair will form a block in N. But, we have assumed that N is stable. Therefore, h will have to prefer r' over r.

■

Let X be a set of residents who prefer M over N, i.e., for some $x \in X$, M(x) = h. Let Y be the set of hospitals which prefer M over N, i.e., for some $y \in Y$, M(y) = r. Similarly, X' and Y' are sets of residents and hospitals respectively, which prefer N over M.

X : prefer M	Y : prefer M
X' : prefer N	Y' prefer N

Table 1.4: Preference Table



Question 2 If there exists a resident $a \in X$ and a hospital $h \in Y$, then, can $\exists\ (a,h) \in M$ such that (a,h) is a stable pair.

Proof. The proof for the above question is similar to the previous proofs. Let us assume that there indeed exists a pair (a, h) in stable matching M. Then, that means that h prefers resident a over some resident b. Similarly, a prefers hospital h over some hospital h'. This means that (a, h) will form a blocking pair in stable matching N as both a and h get the worse partners in N. This contradicts our previous assumption. That means, our assumption was wrong. Therefore, a pair (a, h) where $a \in X$ and $h \in Y$ cannot exist. ■

Fact 4 Anyone in X can't have M-Partner in Y. Similarly, anyone in X' can't have N-Partner in Y'.

$$M(X) \subseteq Y'$$
$$N(X') \subseteq Y$$

$$|X| \leq \sum_{h \in Y'} \text{no. of residents h gets in } M(h) \backslash N(h)$$

$$|X'| \leq \sum_{h \in Y} \text{no. of residents h gets in } N(h) \backslash M(h)$$

If we assume $X \cup X' = R$ and $Y \cup Y' = H$, then M(X) will have to be equal to Y' as it can't be a subset of Y and N(X') will have to be equal to Y. Therefore, our size of X and X' will be as follows:

$$|X| \leq \sum_{h \in Y'} \text{no. of residents h gets in } M(h) \backslash N(h)$$

Should be '='

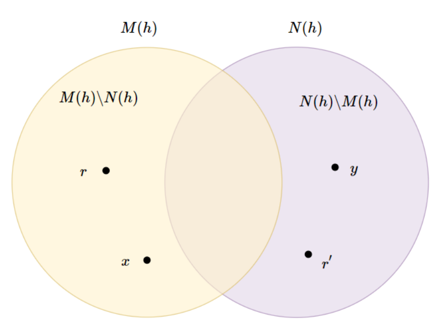
$$|X'| \leq \sum_{h \in Y} \text{no. of residents h gets in } N(h) \backslash M(h)$$

Therefore, |X| + |X'| represent the number of residents that get a different partner in M and N. This means that Rest of the residents $R \backslash X, X'$ will be the set of residents belong to the same hospital in both the stable matchings M and N.

Question 3 The next question which obviously comes to our mind is how will h compare x and y?

This leads us to the following theorem:

> **Theorem 1.4.4** If $M(h) \neq N(h)$, where M and N represent two stable matchings then every resident in $M(h) \backslash N(h)$ is a better partner for given hospital h than every resident in $N(h) \backslash M(h)$ or vice versa.



Proof. Let r : worst resident in $M(h)\backslash N(h)$,

r' : worst resident in $N(h)\backslash M(h)$.

M and N are a pair of stable matchings. M(h) and N(h) represents a resident in the resident set which is a partner of hospital h. Now, suppose, h like r more than r'. Then, that means,

$$h \in Y$$

$$y \in X'$$

We proved the above results which have been mentioned in Fact 4. The reason is that, otherwise a blocking pair will be formed either in M or N, which contradicts the stability of the matching.

Under the assumption that h prefers r more than r', how will h compare x and y? Let us suppose that h likes y more than x. That means that h like y more than some y' in M. This means, both h and y get a worse partner in M, which will result in a blocking pair. Thus, our assumption that h prefers y over x is wrong. This means that h prefers x over y, which can be proved for every x.

Fact 5 **Generalizing, if h likes r over r' then, this**

$\implies$ h likes everyone in $M(h)\backslash N(h)$ more than everyone in $N(h)\backslash M(h)$

■

1.5 Popular Matching

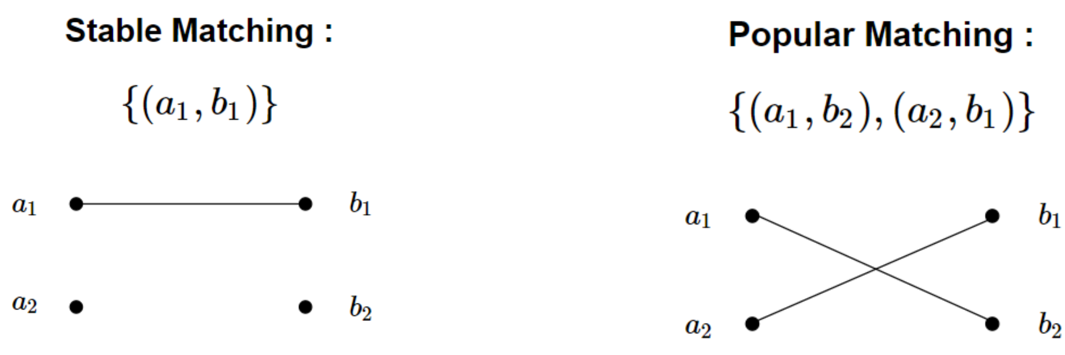
There arises a question whether stable matchings are always preferred. There may arise a situation in which a lot of men and/or a lot of women can remain unmatched, leading to overall dissatisfaction amongst the general public. This situation is one of the few cases which have a stable matching is a bad overall result.

This makes us question whether there is a matching which is "popular" amongst all the matchings, that is, is there a matching, which results in minimum dissatisfaction amongst men and women.

For example,

$a_1 : b_1\ b_2$	$b_1 : a_1\ a_2$
$a_2 : b_1$	$b_2 : a_1$
$a_3 : b_3\ b_4$	$b_3 : a_3\ a_4$
$a_4 : b_3$	$b_4 : a_3$
...	...
$a_{2k-1} : b_{2k-1}\ b_{2k}$	$b_{2k-1} : a_{2k-1}\ a_{2k}$
$a_{2k} : b_{2k-1}$	$b_{2k} : a_{2k-1}$

Table 1.5: Preference Table



(a) M,N : Stable Matching (b) Popular Matching

Figure 1.4: Drawback of Stable Matching

In the above example, when we apply the GS algorithm, we can see that the stable matching has a total of k sizes. This means, that a total of k matches are left unmatched. This leads to a dissatisfaction of size 2k. But, if we do the matchings, as shown in Fig 1.4(b), this will lead to a decrease in dis-satisfactions by a whole k. Though, the matching is obviously unstable, it is the most popular amongst all the involved.

Property of Stable Matching: People don't want to deviate from their partners.

Definition 1.5.1 — Popular Matching. If number of votes of M > number of votes of N, then M is said to be more popular than N. Thus, a matching is a popular matching, if $\nexists$ a matching M' which is more popular than M.

■ **Example 1.2 — Popular Matching.** Let us consider an example in which :

a_1 : b_1 b_2 and b_1 : a_1 a_2
a_2 : b_1 and b_2 : a_1

Then, there will be two possible matchings:
M : (a_1, b_1)
N : $\{(a_1, b_2), (a_2, b_1)\}$

vote	M	N
a_1	✓	
b_1	✓	
a_2		✓
b_2		✓

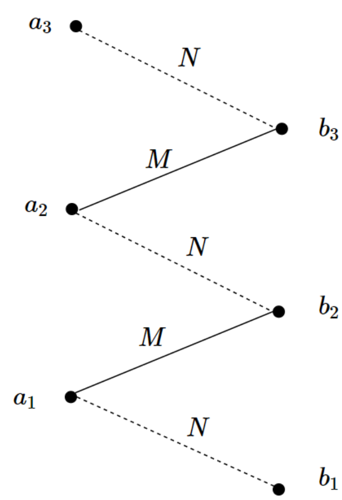
Table 1.6: Example of popular matching

The above table 1.6 describes the satisfaction of all men and women over the two matchings M and N. In this case, both are popular matchings as $\nexists$ a matching more popular than M or N. But, in this case, we will select a matching as popular according to its size, that is, N.

Popular Matching : {(a_1, b_2), (a_2, b_1)}

■

Question 4 Can a stable matching be a popular matching?



In the above figure, can a_3 and b_3 vote for the matching N? The answer to this is no, otherwise they will become a blocking pair for M, as seen in previous lectures.

$\therefore$ For every edge in N, atleast one end-point votes against N, i.e., for M. This means that the number of votes against N $\geq$ the number of votes in favor of N. That means M is a popular matching. But, M is a stable matching.

$\therefore$, **Stable Matchings are popular.**

Fact 6 This means that Stable Matchings are a subset of Popular Matchings.

1.5.1 Algorithm

We will present an algorithm for computing the largest possible popular matching in this section. The question is how to find one. Take the earlier example. In it, the stable matching, which is also a popular matching was $\{(a_1,b_1)\}$. In order to obtain the largest possible popular matching, we give the top priority to the unmatched men. Thus, this will result in breaking of a pair and result in matching of the unmatched man by making every woman prefer the man with the highest priority over one with normal priority. The unmatched man which is given the highest priority is denoted as $a_2 \rightarrow a_2^*$

Highest priority is given to a man who is unmatched and whose preference list is exhausted, i.e., he is rejected by every woman in his list.

The running time of the algorithm is linear time, i.e., $O(m)$

R It's critical to note that the highest priority can be given to atmost one man. It won't make any sense if multiple men are given such priority.

Algorithm 1 Popular Matching Algorithm

procedure START
  $\forall$ men, women $\rightarrow$ free
  if $\exists$ a free man 'a' who has exhausted his list **then**
    'a' proposes next women 'b' on his list
Usual as Accept/Reject
  if 'a' exhausted his list w/o * status: **then**
    a_* is put in queue of free men
Aftermath
  a_* is given highest priority by women.
  a_* list is renewed.

Question 5 Can there be a smaller matching (in comparison to stable matching) which is a popular matching?

1.6 One-Sided Preferences

Although there arise practical scenarios wherein, agents from both the sets have a preference ordering each other, there are also a lot of scenarios wherein, only one side of the agents are allowed to elicit their preferences over the other agents of other set.
For the illustration of such a scenario, we describe a problem of popularity in House Allocation Problem.

Popularity in House Allocation Problem

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*.
Each applicant a_i has a *preference list* in which she ranks her preferred plots/houses in some order. Houses do not have preference lists over applicants, and it is essentially this feature that distinguishes HA from the other problems that we have studied till now.

1.6.1 Assumptions

- We study this problem over full flexibility over incomplete preferences. Thus, we also welcome those problem instances having incomplete lists.
- Posts/Houses do not have a preference ordering over the applicants, which single handedly makes this an important problem to ponder over.
- Also, we assume that there are no ties in the preferences put forth by the applicants.

1.6.2 Preferences And Notion of Optimality

In the problems that we have studied before this, both the sides were encouraged to put forth their preferences, and we also saw relevant examples where this elicitation of information fits real world scenarios. But, as observed in the above problem, there are times when a set is indifferent about the agents in the other set. These settings are called as *One Sided Preference* settings.
Now, we put forth an example, which clarifies the notion of optimality, i.e. points us towards the

direction in which we should think so as to get the better judgement of which matching (stable or unstable) is more popular.

a_1: p_1 p_2 p_3

a_2: p_2 p_3 p_1

a_3: p_1 p_2 p_3

Table 1.7: Example

From the above example. we can figure out the following two matching, which we want to compare with each other.

M	N
$a_1 - p_1$ $a_2 - p_2$ $a_3 - p_3$	$a_1 - p_2$ $a_2 - p_3$ $a_3 - p_1$

Question : How to compare the above matchings?

We can do so by using the popularity technique established in the previous section.

	M	N
a_1:	✓	
a_2:	✓	
a_3:		✓

Table 1.8: Example

Once we fix the notion of comparison as Popularity, a natural question would be to check for existence of popular matching in the setting of one-sided preferences.

1.6.3 Existence

- How to compute popular matching?

Goal : Characterize popular matching and divide an efficient algorithm for computing a popular matching.

a_1	p_1	p_5
a_2	p_1	p_3
a_3	p_2	
a_4	p_4	p_6
a_5	p_4	p_3

Table 1.9: Caption

2. Voting

2.1 Axioms

In this section, we present a list of axioms which are important in the study of fairness notions in the social preference aggregation. We also observe that although all the axioms that we present here are quite intuitively appealing, there is a restriction upon designing voting mechanisms in which some axioms can coexist.

2.1.1 Preliminaries

Let $N = \{1,\ldots,n\}$ be a finite set of individuals (*agents* or *voters* and let A be a set of *alternatives* (or *candidates*). Lets say that we assume that the voters elicit a strict ordering over the alternatives, denoted by $\succ$. $\mathscr{L}(A)$ defines say, the preference ordering of each voter and the social preference aggregation is represented by say a weak ordering over the set of voters by $\mathscr{R}(A)$.

Definition 2.1.1 — Social Welfare Function. A *Social Welfare Function*(SWF) is a function of the form $f : \mathscr{L}(A)^n \mapsto \mathscr{R}(A)$. Which means f is accepting as input a preference profile and maps it to a single preference order which we can think of as a suitable compromise.

2.1.2 Desirable Properties of a SWF

As mentioned above, in the interest of finding out socially acceptable preference aggregation, we need to set some criteria, most appealing of which are the following:

- Weakly Paretian
- Independent of Irrelevant Alternatives

A Social Welfare Function is said to be *Weakly Paretian* if for any two alternatives $a, b \in A$, it is the case that, if $a \succ_i b$ for all individuals $i \in N$, then $a \succ b$. That is, if everyone strictly prefers a to b, then the social preference should rank a strictly above b.

A SWF f is *Independent of Irrelevant Alternatives* if for two alternatives $a, b \in A$ the relative rankings of a and b in the aggregated preference order depends only upon the relative rankings of a and b as provided by the individuals and not, for instance, on how the voters rank a and b.

Although both of these notions are simple to follow and promise a good and logically sound preference aggregation, there is a small glitch. A fundamental impossibility was proved by Kenneth Arrow (1951) and independently by Black, which restricts us from accommodating both of these nice axioms together.
Before moving on formally stating the results mentioned above, we shall be eliciting an infamous, but a strong SWF, called *dictatorship*.

Definition 2.1.2 We say that a SWF is a dictatorship if there exists an individual $i^* \in N$ (the dictator) such that, for all alternatives $a, b \in A$, it is the case that $a \succ_i b$ implies $a \succ b$.

Thus, f simply copies the (strict) preferences of the dictator, whatever the preferences of the other individuals.

Theorem 2.1.1 When there are three or more alternatives, then every SWF that is weakly Paretian and IIA must be a dictatorship.

It is not easy to see that every dictatorship is Weakly Paretian as well as IIA. Although the proof of the Arrow's theorem is quite involved and requires a little bit of maturity to understand its proof from the book, we link the proof of wikipedia, which is quite easy to understand, and is provided in a modular fashion. For the proof, click here [1].

Although Arrow's theorem establishes a fundamental impossibility on the aggregation of Social Choice, the next step is to figure out a way so as to reach as close as possible to the Ground Truth, that is, the almost perfect choice. Also, there are many aspects while choosing a SWF for preference aggregation like the ability of a SWF to avoid being manipulated i.e. *strategyproofness* of a SWF. Interestingly, a similar impossibility result exists in the case of manipulability of SWFs, which again directs us to an interesting question like, if not strategyproof, what about almost strategyproof, and a couple of other such questions, which have been addressed in the literature using various algorithmic tools.

[1] The examples of different voting rules following what criteria is given well in the pdf listed here (click here)