ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
DEPARTMENT OF MATHEMATICS

GRADUATE SEMINAR REPORT

ON

Matchings in Bipartite Graphs

(Submitted in partial fulfillment of M.Sc. Degree in Mathematics)

Compiled by : Baynesagn Mekonnen

Advisor : Dr. Seyoum Getu

June, 2010

Addis Ababa

ACKNOWLEGEMENT

First, I would like to thank my advisor Dr. Seyoum Getu who gave me constructive comments when I was writing this seminar report .I would also like to express my appreciation to my families for their continuous encouragement and commitment in every day life activities .Last, but certainly not least, I would like to extend many thanks to all my friends who have been helping me in all aspects.

Baynesagn Mekonnen

TTABLE OF CONTENTS

Contents ***page***

I. Introduction

Many combinatorial problems involve matching items, subject to certain restrictions. Examples for this are the problem of assigning airline pilots to flights (treated in section 2) and the problem of assigning professors to courses(treated in section 3).Bipartite graphs are useful for modeling matching problems.

This material is organized into four sections .The first section treats definitions and examples of bipartite graphs .The second section treats matchings .The third section treats systems of distinct representatives and the fourth section treats stable marriages.

Matchings in Bipartite Graphs

1. Definitions and examples

Definition. Let $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_n\}$ be two finite sets with m elements and n elements respectively. We assume that the sets X and Y have no elements in common, that is, $X \cap Y = \emptyset$

Let Δ be a collection of pairs e={x, y} where x is an element of X and y is an element of Y. The triple G=(X, Δ, Y) is called a **bipartite graph**. The elements of $X \cup Y$ are called the vertices of G and X, Y is called a bipartition (partition into two parts) of the vertices of G. We regard X, Y and Y, X as the same bipartition and thus do not distinguish between (X, Δ, Y) and (Y, Δ, X) although we usually write the vertices of X on the right and the vertices of Y on the left.

The pairs e={x, y} is a set of two vertices, one of which x, comes from X and the other of which y, comes from Y. We say that the edge e joins the vertices x and y, and that the vertices x and y meet the edge e. Thus a bipartite graph is prescribed by

(i) a set of vertices
(ii) a partition of the set of vertices into two parts; and
(iii) a set of edges joining a vertex in one part to a vertex in the other part.

Example1.1. Let $X=\{x_1, x_2, x_3, x_4\}$ and $Y=\{y_1, y_2, y_3\}$ and

Let $\Delta = \{\{x_1, x_2\}, \{x_1, y_3\}, \{x_2, y_1\}, \{x_3, y_2\}, \{x_3, y_3\}, \{x_4, y_3\}\}$

The vertex x_1 meets 2 edges, namely the edges $\{x_1, y_1\}$ and $\{x_1, y_3\}$. The vertex y_3 meets 3 edges. We can picture the bipartite graph G=(X, Δ, Y) as shown in Figure 1.1

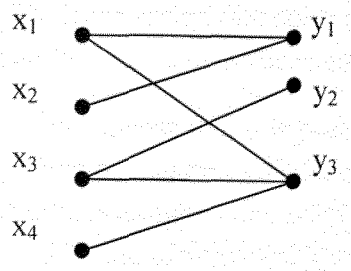


Figure 1.1

Example1.2. Are the graphs G and H bipartite?

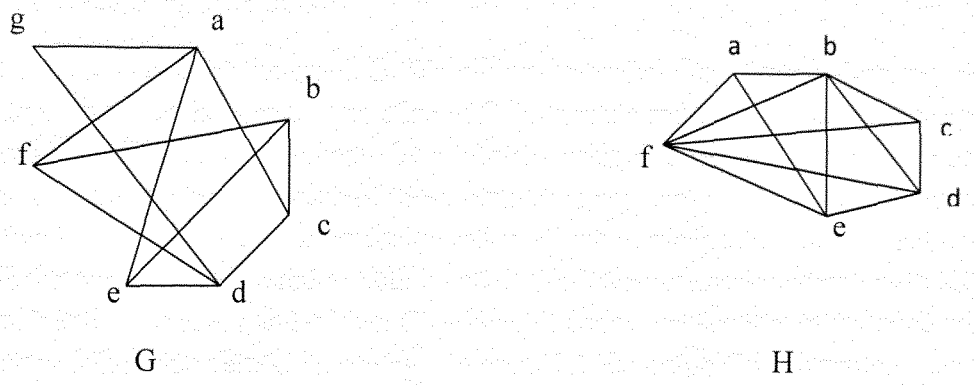


Figure 1.2

Solution: Graph G is bipartite because its vertex set is the union of two disjoint sets {a, b, d} and {c, e, f, g}, and each edge connects a vertex in one of these subsets to a vertex in the other subset (note that for G to be bipartite it is not necessary that every vertex in {a, b, d} be adjacent to every vertex in {c, e, f, g}). For instance, b and g are not adjacent.

Graph H is not bipartite because its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset.

Note. A graph G= (V, E) is called bipartite if $V=V_1 \cup V_2$ with $V_1 \cap V_2 = \emptyset$, and every edge of G is of the form {a, b} with $a \in V_1$ and $b \in V_2$.

Theorem1.1. A simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned the same color.

Proof: First, suppose that G= (V, E) is a bipartite simple graph (note that, a simple graph is a graph without loops and with at most one edge between any two vertices). Then $V= V_1 \cup V_2$, where V_1 and V_2 are disjoint sets and every edge in E connects a vertex in V_1 and a vertex in

V_2. If we assign one color to each vertex in V_1 and a second color to each vertex in V_2, then no two adjacent vertices are assigned the same color.

Now suppose that it is possible to assign colors to the vertices of the graph using just two colors so that no two adjacent vertices are assigned the same color. Let V_1 be the set of vertices assigned one color and V_2 be the set of vertices assigned the other color. Then, V_1 and V_2 are disjoint and $V = V_1 \cup V_2$. Furthermore, every edge connects a vertex in V_1 and a vertex in V_2 because no two adjacent vertices are either both in V_1 or both in V_2. Consequently, G is bipartite.

Example1.3. Using the above theorem determine whether the graphs in

example1.2, are bipartite or not.

Solution. First consider the graph G. We will try to assign one of two colors, say red and blue, to each vertex in G so that no edge in G connects a red vertex and a blue vertex. Without loss of generality we begin by arbitrarily assigning red to a. Then, we must assign blue to c, e, f and g because each of these vertices is adjacent to a. To avoid having an edge with two blue endpoints, we must assign red to all the vertices adjacent to c, e, f or g. This means that we must assign red to both b and d. We have now assigned colors to all vertices, with a, b and d red c, e, f and g blue. Checking all edges, we see that every edge connects a red vertex and a blue vertex. Hence by theorem 1.1 the graph G is bipartite.

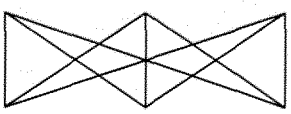
Next, we will try to assign either red or blue to each vertex in H so that no edge in H connects a red vertex and a blue vertex. Without loss of generality we arbitrarily assign red to a. Then, we must assign blue to b, e and f because each is adjacent to a. But this is not possible because e and f are adjacent, so both cannot be assigned blue. This argument shows that we cannot assign one of two colors to each of the vertices of H. So that no adjacent vertices are assigned the same color. It follows by theorem 1.1. that H is not bipartite.

Another useful criterion for determining whether a graph is bipartite is based on the notion of a path. A graph is bipartite if and only if it is not possible to start at a vertex and return to this vertex by traversing an odd number of distinct edges.

Complete bipartite graphs: The complete bipartite graph $K_{m,\,n}$ is the graph that has its vertex set partitioned into two subsets of m and n respectively. There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.

Examples for complete bipartite graphs

$K_{2,\,3}$



$K_{3,\,3}$

Figure1.3. Some complete bipartite graphs

2. Matchings

Definition2.1: Let G=(X, Δ, Y) be a bipartite graph. A matching of G is defined to be a subset M of the set Δ of edges with the property that no two of the edges of M have a common vertex thus if M is a matching, then each left vertex meets at most one edge of M and similarly each right vertex meets at most one edge of M.

For example, in the bipartite graph pictured in figure 1.1 the three edges $\{x_1, y_3\}$, $\{x_2, y_1\}$, $\{x_3, y_2\}$ form a matching of size 3.

If G is any bipartite graph we now define

$\rho(G) = \max\{|M|: M \text{ a matching}\}$ to be the largest number of edges in a matching M of G.

Consider a bipartite graph G=(X, Δ, Y) where $X= \{x_1, x_2,\ldots,x_m\}$ and $Y= \{y_1, y_2,\ldots,y_n\}$. The largest number of edges in a matching is denoted by $\rho(G)$. Our goal is to determine $\rho(G)$, more precisely to determine a matching M* with

$$|M^*|= \rho(G) \qquad (2.1)$$

By the pigeon-hole principle a matching can have at most m edges because if there were more than m edges, two edges would have to meet at the same left vertex. Similarly a matching can have at most n edges.

Thus we have $\rho(G) \leq \min\{m, n\}$

Each matching M satisfies $|M| \leq \rho(G)$. We call a matching M* satisfying (2.1), that is a matching M with the largest possible number of edges among all matchings in G, a maximum matching

A Matching Problem

The problem

An airline flying out of Addis Ababa has seven flights on its Monday morning schedule: Bahirdar, Jimma, Axum, Lalibela, Gondar, Asosa and Mekele.

Fortunately, seven capable pilots are available: Alemayehu, Taye, Jemal, Samuel, Mehari, Reta, and Abdisa. There is a complication, however, pilots are allowed to request particular destination, and these requests are to be honored if possible. The pilots and the cities they requested are as follows

Bahirdar: Taye, Jemal, Reta

Jimma: Alemayehu, Taye, Samuel, Mehari

Axum: Taye, Samuel, Reta

Lalibela: Alemayehu, Samuel, Reta, Abdisa

Gondar: Jemal, Mehari, Reta

Asosa: Jemal Abdisa

Mekele: Taye, Reta, Abdisa

This information could also be represented by a diagram

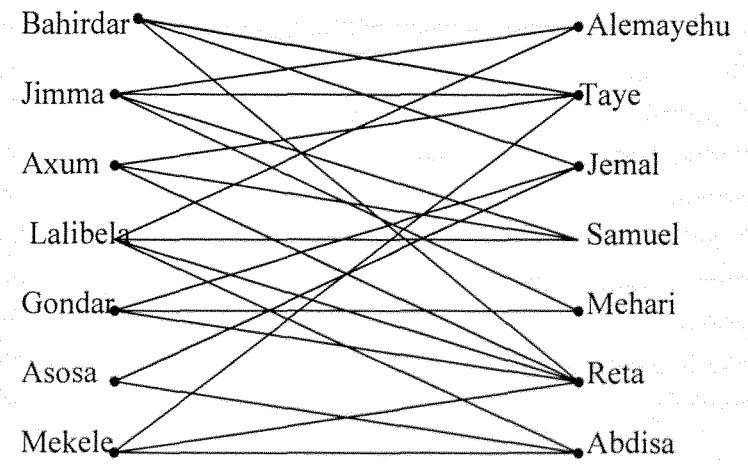


Figure 2.1

The person assigning the flights would like to please all the pilots if this can be done, and if not, would like to accommodate as many as possible. This may be thought to as an optimization problem. We desire a matching of pilots with flights such that the number of pilots who get flights they have requested is as large as possible.

Several questions come to mind concerning our plan for solving this problem;

(1) How much work will this be? In particular, how many arrangements will we have to check?
(2) How can we generate all possible arrangements so that we are sure we have not missed any?

To answer the first question we use a counting principle now we return to the problem of counting the number of ways the 7 flight can be assigned. Let us start with the Bahirdar flight. There are 7 pilots who can be assigned to it. We pick one, and turn to the Jimma flight. Now only 6 pilots are left. Choosing one of these leaves 5 from which to pick for the Axum flight. We can continue in this manner all the way to the Mekele flight, at which time only 1 pilot will be left. Thus, the total number of matchings we can devise will be $7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 7! = 5040$.

The same argument will work whenever we have the same number of flights and available pilots producing

$$n\,(n-1)\,(n-2)\,\ldots\,3 \cdot 2 \cdot 1 = n!$$

possible matchings if there are n flights and n pilots.

The practicality of our solution to the Airline problem

We were going to run through all the ways of assigning a pilot to each of the 7 flights to see which could please the most pilots. We now know the number of possible assignments is $7! = 5040$. This number is large enough to discourage us from trying this method by hand. If a computer were available, however, the method would look more promising. We would need a way to tell the computer how to generate these 5040 permutations, that is, an algorithm. This would amount to an explicit answer to question (2) asked earlier. Of course, 7 flights and 7 pilots are realistically small numbers. For example, if an average of more than 1000 airplanes take off every day, and consider the practicality of running through all possible assignments. To check how many pilots get their requested flights it may take several years even with a computer. So we need a much more efficient way to solve our matching problem.

Definition2.2 . Let u and v be any two vertices in the bipartite graph $G=(X, \Delta, Y)$. A chain joining u and v is a sequence of distinct vertices (except u may equal v).

$$\gamma: u=u_0, u_1, u_2, \ldots\ldots, u_{p-1}, u_p=v \qquad (2.2)$$

such that consecutive vertices are joined by an edge. Thus in order for (2.2) to be a chain,

$$\{u_0, u_1\}, \{u_1, u_2\}, \{u_2, u_3\}, \ldots, \{u_{p-1}, u_p\} \qquad (2.3)$$

must all be edges in Δ. The edges in (2.3) are called the edges of the chain γ. The length of the chain γ is the number p of its edges. The vertices u and v are called the end vertices of the chain γ. The vertices in a chain must be alternately left and right vertices. But the end vertices can be either both left vertices, both right vertices or one of each type. If u=v in the chain (2.2), then the chain is called a cycle. A cycle in a bipartite graph necessarily has even length.

Example2.1. Consider the following bipartite graph

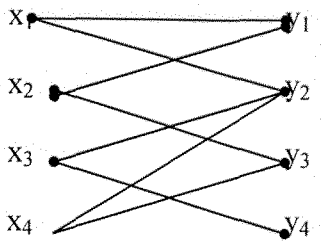


Figure 2.2

In Figure 2.2 x_1, y_2, x_3, y_4 is a chain of length 3 joining x_1 and y_4.

y_1, x_2, y_3, x_4 is a chain joining y_1 and x_4; and x_3, y_2, x_4, y_3, x_2 is a chain joining x_3 and x_2. Also $x_1, y_1, x_2, y_3, x_4, y_2, x_1$ is a cycle of length 6.

Now let M be a matching in the bipartite graph G=(X, Δ, Y). Let $\bar{M}$ be a matching of M, that is, the set of edges of G that do not belong to M. Let u and v be vertices where one of u and v is a left vertex and one is a right vertex. A chain γ joining u and v is an alternating chain with respect to the matching M, for brevity, an M alternating chain, provided the following properties hold:

(1) .The first, third, fifth,... edges of γ do not belong to the matching M (thus they belong to $\bar{M}$);

(2) .The second, fourth, sixth ,... edges of γ belong to the matching M;

(3) .Neither u nor v meets an edge of the matching M.

Notice that the length of an M-alternating chain γ is an odd number 2k+1 with $k \geq 0$, and that k+1 of the edges of γ are edges of $\bar{M}$ while k of the edges of γ are edges of M.

Furthermore:

M_γ denotes the set of edges of γ that belong to M; and

$\bar{M}_\gamma$ denotes the set of edges of γ that do not belong to M.

It follows that

$$|\overline{M}_\gamma| = |M_\gamma| + 1.$$

Example2.2. Consider the bipartite graph pictured in Figure 2.2.

The set $M = \{\{x_1, y_1\}, \{x_2, y_3\}, \{x_3, y_4\}\}$ is a matching of 3 edges.

The chain

γ: $u = x_4, y_3, x_2, y_1, x_1, y_2 = v$ is an M-alternating chain ,because the first ,third and fifth edges of γ ,that is,$\{x_4,y_3\}$,$\{x_2,y_1\}$ and $\{x_1,y_2\}$ do not belong to the matching M .The second and fourth edges of γ ,that is ,$\{x_2,y_3\}$ and $\{x_1,y_1\}$ belong to the matching M and neither $u=x_4$ nor $v=y_2$ belongs to the matching M.

Then we have

$$M_\gamma = \{\{x_2, y_3\}, \{x_1, y_1\}\}$$

and $$\overline{M}_\gamma = \{\{x_4, y_3\}, \{x_2, y_1\}, \{x_1, y_2\}\}.$$

If we remove the edges of M_γ from M and replace them with the edges of $\overline{M}_\gamma$, we obtain a matching

$M' = (M - M_\gamma) \cup \overline{M}_\gamma = \{\{x_3, y_4\}, \{x_4, y_3\}, \{x_2, y_1\}, \{x_1, y_2\}\}$ of 4 edges.

As illustrated in the previous example, if M is a matching and there is an M-alternating chain then

$$(M - M_\gamma) \cup \overline{M\gamma}$$

is a matching with one more edge than M and hence M is not a max-matching. We now show that the converse holds as well, that is, the only way a matching M cannot be a max-matching is for there to exist an M alternating chain.

Theorem2.1. Let M be a matching in the bipartite graph $G=(X, \Delta, Y)$. Then M is a max-matching if and only if there does not exist an M-alternating chain.

Proof. Suppose M is a max-matching as observed above there does not exist an M-alternating chain.

To establish the converse, we now assume that M is not a max-matching and prove that there exists an M-alternating chain.

Let M' be a matching satisfying

$$|M'|>|M|$$

We consider the bipartite graph

$$G^*=(X, \Delta^*, Y)$$

Where $\Delta^*= (M-M') \cup (M'-M)$

The bipartite graph G* has the same left and right vertices as G. The edges of G* are those edges of G which either belong to M but not to M′ (the edges of M-M′) or belong to M′ but not to M (the edges of M′-M). Thus to get G* we remove from G all edges that belong to both M and M′. Since $|M'|>|M|$, we have

$$|M'-M| > |M-M'| \qquad (2.4)$$

The bipartite graph G* has the property that each of its vertices meets at most two edges: each vertex meets at most one edge of M-M′since M′is a matching; and meets at most one edge of

M′–M since M′ is a matching. This property of G* implies that the set of edges of G* can be partitioned into chains and cycles. In each of the chains and cycles of this partition, the edges alternate between M-M′ and M′-M. These chains and cycles are of the following four types. A chain in this partition has the property that both its first and last vertex meet only 1 edge of G*.

Type1.A chain whose first and last edges are both in M′- M (see Figure 2.3, in this and the other figures the bold lines denote the edges of M). These chains have odd length and contain one more edge of M′ than they do of M. Included among the type1 chains are chains of length 1 in which the first edge is the same as the last edge.

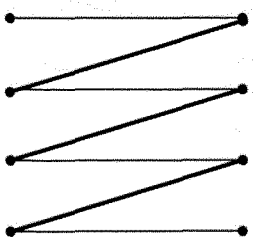


Figure 2.3

Type 2 A chain whose first and last edges are both in M-M$'$.

These chains also have odd length but they contain one more edge of M than they do of M$'$.

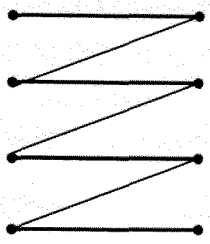


Figure 2.4

Type3. A chain whose first edge is in M-M$'$ and whose last edge is in M$'$- M (or vice-versa). These chains have even length and contain as many edges of M as they do of M$'$.

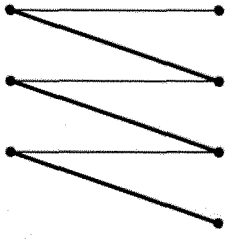


Figure 2.5

Type 4. A cycle. These cycle have even length and contain as many edges of M as they do of M$'$.

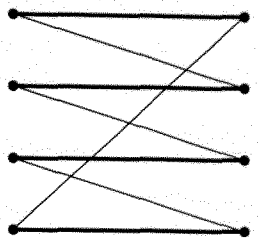


Figure 2.6

There are more edges of $M - M'$ than of $M'-M$ in a chain of type2, and the same number of edges of $M-M'$ as of $M-M'$ in a chain of type 3 and a cycle of type 4. In a chain of type 1 there are more edges of $M'-M$ than of $M-M'$. By (2.4) $M'-M$ has more edges than $M-M'$ does. Hence in G^* there must be a chain of type1 but a chain of type1 is by definition an M-alternating chain. Thus if a matching M is not a max-matching, there is an M-alternating chain.

We cannot expect to examine all possible chains in order to determine whether among them there is an M-alternating chain. Such a task would require in general too much time and effort. What we seek is some way of establishing that a matching is a max-matching that is easy to check. In other words, we seek an easily verifiable certification that a matching is a max-matching. We now discuss such a certification.

Definition2.3. Let $G=(X, \Delta, Y)$ be a bipartite graph. A subset S of the set $X \cup Y$ of vertices of G is called a cover provided each edge of G has at least one of its two vertices in S;

$\{x, y\} \cap S \neq \emptyset$ for all $\{x, y\}$ in Δ.

The set X of left vertices of G is a cover since each edge has a left vertex. The set Y of right vertices is also a cover. Indeed the set $X \cup Y$ of all vertices of G is a cover. However our interest lies in small covers.

Example 2.3. Let G be a bipartite graph pictured below

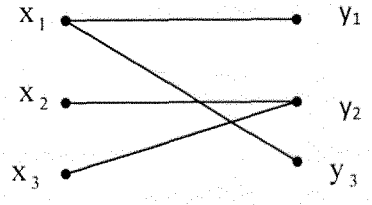


Figure 2.7

In addition to the covers $\{x_1, x_2, x_3\}$ and $\{y_1, y_2, y_3\}$ we have the cover $S=\{x_1, y_2\}$ with only 2 vertices. The fact that S is a cover means that there is no edge whose left vertex is one of $\{x_2, x_3\}$ and whose right vertex is one of $\{y_1, y_3\}$, and this is easily checked by inspection.

We define the cover number of G to be

$c(G)= \min\{|X|: X \text{ a cover of } G\}$, the smallest number of vertices in a cover of G. Every cover S satisfies $|S| \geq c(G)$. We call a cover S of G which Satisfies

$|S|= c(G)$, that is, a cover with the smallest number of vertices, a min-cover (minimum cover).

Lemma 2.2. If G is a bipartite graph, then

$$\rho(G) \leq c(G) \tag{2.5}$$

That is, the largest number of edges in a matching of G does not exceed the smallest number of vertices in a cover of G.

Proof. Let G be the bipartite graph (X, Δ, Y) and let S^* be a cover satisfying $|S^*|= c(G)$, let M be a matching. Since S^* is a cover, each edge of M has at least one of its vertices in S^*, suppose that $|M| > |S^*|$. Then by the pigeon-hole principle, two different edges in M contain the same vertex of S^*. But this contradicts the fact that M is a matching. Hence

$$|M| \leq |S^*|= c(G)$$

Lemma 2.2 has the following consequence. Suppose that in some way or other way we have found a matching M in a bipartite graph G which we think might be a max-matching. If we can find a cover S such that

$|M|=|S|$, then M is a max-matching (and S is a min-cover). This fact is a consequence of the inequalities

$$c(G) \leq |S|=|M| \leq \rho(G) \tag{2.6}$$

Which imply $c(G) \leq \rho(G)$

Applying (2.5) we conclude that $c(G) = \rho(G)$. Now (2.6) implies that $|M|= \rho(G)$ and $|S|=c(G)$, that is, M is a max-matching and S is a min-cover. Thus in this case S acts as a certification that there is no matching with a larger number of edges than M.

Example2.4. In the bipartite graph given above (Figure 2.7), we see that

$M= \{\{x_1, y_1\}, \{x_2, y_2\}\}$ is a matching of 2 edges. As already observed $S=\{x_1, y_2\}$ a cover of 2 vertices thus

$$2 = |M| \leq \rho(G) \leq c(G) \leq |S| = 2$$

We have equality throughout, and hence M is a max-matching, S is a min-cover, and

$$\rho(G) = c(G) = 2.$$

We now turn to showing that we can always find a matching M and a minimum-cover S satisfying.

$$|M| = |S| \qquad (2.7)$$

From which we will be able to conclude as above that $\rho(G) = c(G)$. M is a max-matching and S is a min-cover. Thus our sought-after certification is a cover S with the same size as the matching M.

Let $G = (X, \Delta, Y)$ be a bipartite graph and let M be a matching in G. We describe an algorithm which is a systematic search for an M-alternating chain. Either the algorithm produces an M-alternating chain or fails to produce an M-alternating chain but does produce a min-cover S with $|M| = |S|$ (and we thus conclude that M is a max-matching and S is a certification for M; thus the algorithm didn't produce an M-alternating chain because no such alternating chain exists).

Matching Algorithm

Let $G=(X, \Delta, Y)$ be a bipartite graph where $X=\{x_1, x_2, \ldots, x_m\}$ and $Y= \{y_1, y_2, \ldots, y_n\}$. Let M be any matching in G.

(0) Begin by labeling with (*) all vertices in X that do not meet any edge in M and call all vertices unscanned. Go to (1).

(1) If in the previous step, no new label has been given to a vertex of X, then stop. Otherwise go to (2).

(2) While there exists a labeled, but unscanned vertex of X, select a labeled but unscanned vertex of X, say x_i, and label with (x_i) all vertices in Y, which are joined to x_i by an edge not belonging to M and which have not been previously labeled. The vertex x_i is now scanned. Go to (3).

(3) If in step (2), no new label has been given to a vertex of Y, then stop. Otherwise go to (4).

(4) While there exists a labeled, but unscanned vertex of Y, select a labeled but unscanned vertex of Y, say y_j, and label with (y_j) any vertex of X which is joined to y_j by an edge belonging to M and which has not been previously labeled. The vertex y_j is now scanned. Go to (1).

Since each vertex receives at most one label and since each vertex is scanned at most once; the matching algorithm halts after a finite number of steps. There are two possibilities to consider.

Breakthrough: there is a labeled vertex of Y which does not meet an edge of M.

Non-breakthrough: the algorithm has come to a halt and breakthrough has not occurred, that is, each vertex of Y which is labeled also meets some edge of M.

In the case of breakthrough, the matching algorithm has succeeded in finding an M-alternating chain γ. One end vertex of γ is the vertex v of Y which is labeled but does not meet any edge of M. The other end vertex of γ is a vertex u of X with label (*). (And which therefore does not meet any edge of M). The M-alternating chain γ can be constructed by starting at v and working backwards through the labels until a vertex u with label (*) is found. With breakthrough and the M-alternating chain γ we can obtain a matching with one more edge than M.

If non-breakthrough occurs, we show that it is because M is a max-matching, that is, by theorem 2.1 there isn't any M-alternating chain. Thus breakthrough occurs exactly when M is not a max-matching, and when breakthrough occurs, we have a way of obtaining an M-alternating chain and hence a matching with one more edge than M.

Theorem 2.3. Assume non-breakthrough occurs in the matching algorithm. Let X^{un} consist of all the unlabeled vertices of X and let Y^{lab} consist of all the labeled vertices of Y. Then both of the following hold.

(i) $S= X^{un} \cup y^{lab}$ is a min-cover of the bipartite graph G;
(ii) $|M| =|S|$ and M is a max-matching.

Proof. We show that S is a cover by assuming there is an edge e = {x, y} neither of whose vertices belongs to S and obtaining a contradiction. Thus assume that x is in $x\text{-}x^{un}$ and y is in $y - y^{lab}$ and e= {x, y} is an edge. Since x is not in x^{un}, x is labeled; since y is not in y^{lab}, y is unlabeled. Either e belongs to M or it does not. If e does not belong to M. then in applying step (2) of the algorithm, y receives the label (x), a contradiction. We now assume that e belongs to M. since x meets the edge of M, it follows from step (0) that the label of x is not (*). But then it follows from the algorithm that x has label (y) (see step (4)). By the algorithm again, vertex y can give label (y) to a vertex of x only if y is already labeled. Since y is not labeled; we have a contradiction again. Since both possibilities lead to a contradiction, we conclude that S is a cover.

We complete the proof of the theorem by showing that $|M| = |S|$. As we have already demonstrated, this equality also implies that S is a min-cover and M is a max-matching. We establish a one-to-one correspondence between the vertices in S and the edges in M there by proving $|M| = |S|$. Let y be a vertex in y^{lab} so that y is labeled. Since breakthrough has not occurred, y meets an edge of M and hence exactly one edge, say the edge {x, y} of M. By step (4) of the algorithm, x gets the label (y) and hence x is not in X^{un}. Thus each vertex of y^{lab} meets any edges of M whose other vertex belongs to $X\text{-}X^{un}$. Now consider a vertex x' in X^{un}. Since x' is

not labeled, it follows from step (0) that x' meets an edge of M (otherwise x' would have the label (*)), and hence exactly one edge, say $\{x', y'\}$, of M. the vertex y' cannot be in Y^{lab}, since we have shown above that the unique edge of M meeting a vertex in Y^{lab} has its other vertex in X- X^{un}. Thus we have shown that for each vertex of $X^{un} \cup Y^{lab}$ there is a unique edge of M containing it and all these edges are distinct. Hence

$$|S| = |X^{un} \cup Y^{lab}| \geq |M|,$$

and we conclude that $|S| = |M|$.

Corollary 2.4: Let G= (X, Δ, Y) be a bipartite graph.

Then $\rho(G) = c(G)$,

that is, the largest number of edges in a matching equals the smallest number of vertices in a cover.

Proof .By lemma 2.2

$$\rho(G) \leq c(G).$$

by theorem 2.3 there is a matching M and a cover S such that $|M| = |S|$. Hence

$$\rho(G) \geq |M| = |S| \geq c(G),$$

and we conclude that $\rho(G) = c(G)$.

The matching algorithm can be applied to determine a max-matching in a bipartite graph (X, Δ, Y) as follows. We first choose a matching in a greedy fashion: we pick any edge e_1, then any edge e_2 that does not meet e_1, then any edge e_3 that does not meet e_1 or e_2, and continue like this until we run out of choices. We call the resulting matching M^1 and apply the matching algorithm to it. If non-breakthrough occurs, then by theorem 2.2 M^1 is a max-matching. If breakthrough occurs, then we obtain a matching M^2 with more edge than M^1. We now apply the matching algorithm to M^2. In this way we obtain a sequence of matching M^1, M^2, M^3, … each with more edges than the preceding one. After a finite number of application of the matching algorithm we obtain a matching M^k for which the matching algorithm results in non-breakthrough and hence M^k is a max-matching.

Example 2.5 . We determine a max-matching in the bipartite graph

G= (X, Δ, Y) in Figure 2.8

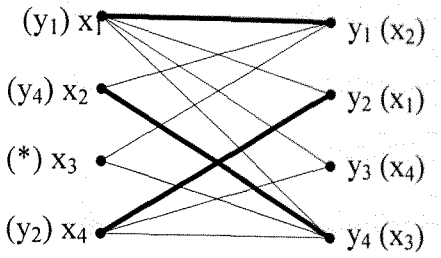


Figure 2.8

We choose the edges $\{x_1, y_1\}$, $\{x_2, y_4\}$, $\{x_4, y_2\}$ and obtain a matching M^1 of size 3. We now apply the matching algorithm to M^1 and the results as shown in Figure 2.8 are as follows;

(i) step(0): the vertex x_3 does not meet an edge of M^1 is labeled (*)
(ii) step(1): we scan x_3 and label y_4 with (x_3)
(iii)step(2): we scan the vertex y_4 labeled in (ii), and label x_2 with (y_4)
(iv)step(3): we scan the vertex x_2 labeled in (iii), and label y_1 with (x_2)
(v) step(4): we scan the vertex y_1 labeled in (iv), and label x_1 with (y_1)
(vi)step(5): we scan the vertex x_1 labeled in (v), and label y_2 with (x_1)
(vii) step(6): we scan the vertex y_2 labeled in (vi), and label x_4 with (y_2)
(viii) step(7): we scan the vertex x_4 labeled in (vii), and label y_3 with (x_4)
(ix)Step (8): we scan the vertex y_3 labeled in (viii), and find that no new labels are possible.

The algorithm has now come to an end, since we have labeled a vertex of Y which does not meet an edge of M^1, in fact, y_3 has this property, we have achieved breakthrough. If we trace backwards from y_3 using the labels as a guide, we find the M^1-alternating chain.

γ : $y_3, x_1, y_1, x_2, y_4, x_3$

We have

$$M_\gamma^1 = \{\{x_1, y_1\}, \{x_2, y_4\}\}$$

And

$$\overline{M}_\gamma^1 = \{\{y_3, x_1\}, \{y_1, x_2\}, \{y_4, x_3\}\}$$

Then $M^2 = (M^1 - M_\gamma^1) \cup (\overline{M}_\gamma^1)$

$= \{\{x_4, y_2\}\} \cup \{\{y_3, x_1\}, \{y_1, x_2\}, \{y_4, x_3\}\}$

$= \{\{x_4, y_2\}, \{y_3, x_1\}, \{y_1, x_2\}, \{y_4, x_3\}\}$ is a matching of 4 edges.

We now apply the matching algorithm to M^2. The resulting labeling of the vertices is shown in Figure 2.9. In this case break-through has not occurred. By theorem 2.3; M^2 is a max-matching of size 4, and the set

$S= \{y_1, y_2, y_3, y_4\}$ of size 4 consisting of the unlabeled vertices of X, that is, $\emptyset$ and the labeled vertices of Y ,that is, $\{y_1, y_2, y_3, y_4\}$ is a min-cover.

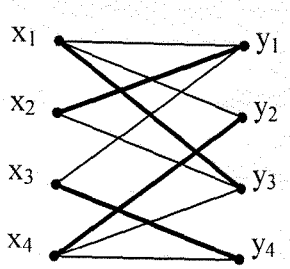


Figure 2.9

Now we return to the matching problem of the airline problem. We can solve it by using the matching algorithm as follows

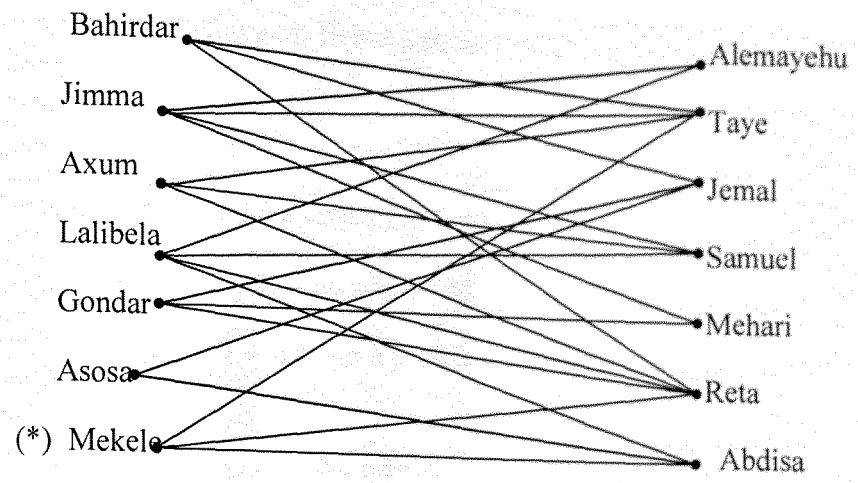


Figure 2.10

We choose the edges {Bahirdar, Taye}, {Jimma, Alemayehu}, {Axum, Samuel}, {Lalibela, Reta}, {Gondar, Jemal} and {Asosa, Abdisa} and obtain a matching M^1 of size 6. We now apply the matching algorithm by labeling (*) the vertex Mekele which does not meet an edge of M^1 at the first step and the algorithm will come to an end, by labeling the vertex Mehari who does not meet an edge of M^1. Then we have achieved break-through. If we trace backwards from Mehari using the labels as a guide, we find the M^1-alternating chain.

γ: Mehari, Jimma, Alemayehu, Lalibela, Reta, Mekele.

We have

M^1_γ = {{Jimma, Alemayehu}, {Lalibela, Reta}}

And

$\bar{M}^1_\gamma$= {{Mehari, Jimma}, {Alemayehu, Lalibela}, {Reta, Mekele}}.

Then $M^2 = (M^1 - M^1_\gamma) \cup (\bar{M}^1_\gamma)$

= {{Bahirdar, Taye}, {Axum, Samuel}, {Gondar, Jemal}, {Asosa, Abdisa}, {Mehari, Jimma}, {Alemayehu, Lalibela}, {Reta, Mekele}}

Therefore the matching problem developed in this section is solved.

Let G=(X, Δ, Y) be a bipartite graph such that the set X of left vertices and the set Y of right vertices have the same size n. A matching in G can contain at most n edges. A matching M in G with n edges is called a perfect matching. Each vertex in X and each vertex in Y meets exactly

one edge of a perfect matching M. Hence a perfect matching M establishes a one-to-one correspondence

$$f: X \to Y$$

between the vertices in X and the vertices in Y where

$$f(x) = y \text{ if } \{x, y\} \text{ is an edge of M.}$$

It follows from corollary 2.4 that G has a perfect matching if and only if no set of fewer than n vertices covers all the edges of G.

A bipartite graph $G=(X,\Delta, Y)$ is called regular of degree p provided each of its vertices meets exactly p edges. If G is regular of degree $p \geq 1$, then X and Y must have the same size n. This is because counting the number of edges by looking at the left vertices we see that the total number of edges of G is $p|X|$, while counting by looking at right vertices we see that the total number is $p|Y|$.

Equating those two counts we get

$$p|X| = p|Y|.$$

And since $p \neq 0$ we obtain $|X|=|Y|$

Theorem2.5. A bipartite graph $G=(X, \Delta, Y)$ which is regular of degree $p \geq 1$ always has a perfect matching.

Proof: Let X and Y each contain n vertices. Let S be any cover of G, and let $S_1 = S \cap X$ be the vertices of S that belongs to Y. Because S is a cover, every edge of G meets at least one vertex of S. Since G is regular of degree p each vertex of S meets exactly p edges. Hence the total number of edges of G is at most $p|S|$. But the total number of edges of G is pn. Hence

$$p|S| \geq pn$$

and thus $|S| \geq n$.

Therefore every cover of G has at least n vertices and by corollary 2.4, G has a perfect matching.

3. Systems of Distinct Representatives.

Definition3.1 Let Y be a finite set and let $A=(A_1,A_2,\ldots,A_n)$ be a family of subsets of Y .A family $(e_1,e_2,\ldots,e_n)$ of elements of Y is called a system of representatives of A provided

e_1 is in A_1, e_2 is in A_2, …, e_n is in A_n.

In a system of representatives the element e_i belongs to A_i and thus 'represents' the set A_i. If in a system of representatives the elements $e_1,e_2,\ldots,e_n$ are all different, then $(e_1,e_2,\ldots,e_n)$ is called a system of distinct representatives abbreviated SDR.

Example3.1. Let (A_1, A_2, A_3, A_4) be the family of subsets of the set Y= {p, q, r, s, t} defined by

A_1= {p, q, r}, A_2= {q, s}, A_3= {p, q, s}, A_4= {q, s}

Then (p, q, q, s) is a system of representatives, and (r, q, p, s) is an SDR.

A family $A= (A_1, A_2,\ldots,A_n)$ of nonempty sets always has a system of representatives. We need only pick one element from each of the sets to obtain a system of representatives. However, the family A need not have SDR even though all the sets in the family are nonempty. For instance, if there are two sets in the family, say A_1 and A_2, each containing only one element and the element in A_1 is the same as the element in A_2, that is,

$$A_1= \{a\}, A_2= \{a\}$$

Then the family A does not have an SDR. This is because in any system of representatives a has to represent both A_1 and A_2, thus no SDR exists (no matter what $A_3,\ldots,A_n$ equal)

However, a family A can fail to have an SDR for somewhat more complicated reasons.

Example3.2. Let the family $A= (A_1, A_2, A_3, A_4)$ be defined by

A_1= {x, y}, A_2={x, y}, A_3={x, y}, A_4={x, y, s, t}

Then A does not have an SDR because in any system of representatives, A_1 has to be represented by x or y, A_2 has to be represented by x or y, and A_3 has to be represented by x or y. So we have two elements, namely, x and y, from which the representative of three sets, namely A_1, A_2, and A_3 have to be drawn. By the pigeon-hole principle, two of the three sets A_1, A_2 and A_3 have to be represented by the same element.

Lemma 3.1 .In order for the family $A= (A_1, A_2,\ldots, A_n)$ of sets to have an SDR it is necessary that the following condition hold:

(MC): for each k=1, 2… n and each choice of k distinct indices $i_1, i_2\ldots i_k$ from {1, 2… n},

$$|Ai_1 \cup Ai_2 \cup \ldots \cup Ai_k| \geq k \qquad (3.1)$$

That is, let $A=(A_1, A_2, \ldots, A_n)$ be a family of sets. Let k be an integer with $1 \leq k \leq n$. In order for A to have an SDR it is necessary that the union of every k sets of the family A contain at least k elements.

Proof. Suppose to the contrary that there are k sets, say $A_1, A_2, \ldots, A_k$, which together contain fewer than k elements; $A_1 \cup A_2 \cup \ldots \cup A_k = F$ where

$$|F| < k$$

Then the representatives of each of the k sets $A_1, A_2, \ldots, A_k$ have to be drawn from the elements of the set F. Since F has fewer than k elements, it follows from the pigeon-hole principle that two of the k sets $A_1, A_2, \ldots, A_k$ have to be represented by the same element. Hence there can be no SDR.

The marriage condition (3.1) is not only a necessary condition for the existence of SDR but a sufficient condition as well. It thus provides a characterization for the existence of an SDR.

We associate a bipartite graph $G=(X, \Delta, Y)$ to each family $A=(A_1, A_2, \ldots, A_n)$ of subsets of a set $Y=\{y_1, y_2, \ldots, y_n\}$ we take X equal to the set $\{1, 2, \ldots, n\}$, the set indexing the members of the family A and define the set of edges Δ by

$$\Delta = \{\{i, y_j\} : y_j \text{ is in } A_i\}$$

Thus the vertex i and y_j are joined by an edge in G if and only if y_j is an element of the set A_i. Put another way, vertex i is joined by an edge to those elements of Y which can serve as representatives of A_i. A system of representatives of A corresponds to a set of n edges, one meeting each vertex of X, but there may be more than one edge meeting a vertex of Y, since in a system of representative the same element may represent two different sets. An SDR corresponds to a set of n edges, one meeting each vertex of X and at most one meeting each vertex of Y. We thus conclude that: the family of sets $A=(A_1, A_2, \ldots, A_n)$ has an SDR if and only if the associated bipartite graph G has a matching M of n edges, since X has only n vertices, G cannot have a matching of more than n edges, thus A has an SDR if and only if $\rho(G) = n$.

Theorem 3.2 .The family $A=(A_1, A_2, \ldots, A_n)$ of sets has an SDR if and only if the marriage condition (MC) holds.

Proof. By lemma 3.1 we know that the marriage condition holds if A has an SDR. We now assume that the marriage condition holds and show that A has an SDR.

Let $G=(X, \Delta, Y)$ be the bipartite graph associated with the family A as in the paragraph preceding the theorem.

We need to show that $\rho(G) = n$. By corollary 2.4,we can conclude that $\rho(G)=n$ if we show that c(G)=n, that is, if we show that there is no cover of G consisting of fewer than n vertices. Suppose to the contrary that there is a cover S of G with $|S| < n$.

Let $S=S_1 \cup S_2$

Where $S_1= S \cap X$ are the left vertices in S and $S_2= S \cap Y$ are the right vertices in S. Since $|S| < n$, We have

$$|S_1| + |S_2| < n \qquad (*)$$

Because S is a cover there is no edge joining a vertex in $X-S_1$ to a vertex in $Y- S_2$.
Let $k=|X-S_1|=n-|S_1|$

and let $X- S_1 = \{i_1, i_2 \ldots i_k\}$

Since there is no edge joining a vertex in $X - S_1$ to a vertex in $Y- S_2$, Ai_1, $Ai_2 \ldots$ and Ai_k are all subsets of S_2.

Hence

$$Ai_1 \cup Ai_2 \cup \ldots \cup Ai_k \subseteq S_2$$

And thus

$$Ai_1 \cup Ai_2 \cup \ldots \cup Ai_k \leq |S_2|$$

By (*), $|S_2| < n - |S_1|=k$

And therefore

$|Ai_1 \cup Ai_2 \cup \ldots \cup Ai_k| < k$,

Contradicting the marriage condition, hence there is no cover S of G with fewer than n vertices, $\rho(G) = n$, and A has an SDR.

Example 3.3 Consider the summer schedule of classes in the mathematics department at a small college. There is a demand for 6 courses. To keep things simple, we will call these course 1, course 2,…, course 6. Certain professors A, B, C, D, E, F & G are available to teach each course, as given in the following table.

Course	Professor
1	A, C, F
2	C, D, E, G
3	A, C
4	A, F
5	B, E, G
6	C, F

In order to distribute the summer teaching jobs as fairly as possible; it is decided that no professor should teach more than one course. The question is whether all courses can be taught, subject to this restriction. If not, what is the maximum number of courses that can be taught?

This is a problem of exactly the same sort as that of assigning airline pilots in section 2 with only 6 courses and 7 professors; we could probably find the answer by considering all possible matchings. One systematic way of doing this is the following: let P_1 denote the set of professors available to teach course 1, P_2 the set of professors available to teach course 2, etc.

If we forget for the moment the restriction that no professor may teach more than one course, then a possible assignment of a professor to each course consists of 6-tuple $(x_1, x_2, x_3, x_4, x_5, x_6)$ where $x_1 \in P_1$, $x_2 \in P_2$, etc. This is an element of the Cartesian product

$P_1 \times P_2 \times P_3 \times P_4 \times P_5 \times P_6$; which has $3 \cdot 4 \cdot 2 \cdot 2 \cdot 3 \cdot 2 = 288$ elements.

We need to know whether any of these 288, 6-tuples has all its entries distinct (so that no professor teaches more than one course). Checking this without the help of a computer would be possible, but extremely tedious. As in the case of the pilot assignment problem, however, such crude methods of searching for a solution quickly get beyond the capability of even a computer as the number of items to be matched gets larger.

Now we are looking for a system of distinct representatives for the sequence $P_1, P_2, \ldots, P_6$

The problem seems small enough that we might expect to find the solution, if there is one, by simply trying different combinations. Yet perhaps the best we can come up with is to cover 5 of the 6 courses.

For example, we might assign the first 5 courses as in the following list:

Course 1 to A

Course 2 to D

Course 3 to C

Course 4 to F

Course 5 to B

We might suspect that it is not possible to do better than this but it is difficult to be certain. We would like a way to convince ourselves that no assignment of all 6 courses is possible without going through all 288 possibilities.

If we could discover some collection of sets chosen from P_1 through P_6, the union of which contained fewer elements than the number of sets in the collection, then we would know that a system of distinct representative was impossible.

The collection we have in mind is P_1, P_3, P_4 and P_6. Notice that $P_1 \cup P_3 \cup P_4 \cup P_6 = \{A, C, F\}$

Suppose we had a system of distinct representatives $X_1, X_2, \ldots, X_6$ then X_1, X_3, X_4 and X_6 would be 4 distinct elements lying in the union of the sets P_1, P_3, P_4 and P_6. But that is impossible, because this union contains only 3 elements. There are only 3 professors (A, C, F) available to teach 4 of the courses, and so an assignment where no professor teaches more than 1 course cannot be made.

We have found a general principle, which could be stated as follows; suppose $S_1, S_2, \ldots, S_n$ is a finite sequence of sets, and suppose I is a subset of $\{1,2, .. ,n\}$ such that the union of the sets S_i for $i \in I$ contains fewer elements than the set I does. Then $S_1, S_2, \ldots, S_n$ has no SDR. In our example (taking $S_i = P_i$ for $i=1,\ldots,6$, then the set I is $\{1,3,4,6\}$).Finding such a set I enables us to be sure no SDR exists.

4. Stable Marriages

There are n women and n men in a community. Each woman ranks each man in accordance with her preference for that man as a spouse. The preferences are to be purely ordinal and thus each woman ranks the men in the order 1,2,…,n. Similarly, each man ranks the women in the order 1,2,…,n. There are n! ways in which the women and men can be paired so that a complete marriage takes place, we say that a complete marriage is unstable provided there exist two women A and B and two men a and b such that

(i) A and a get married;
(ii) B and b get married;
(iii) A prefers (i.e. ranks higher) b to a;
(iv) b prefers A to B.

A complete marriage is called stable provided it is not unstable .The question that arises first is: does there always exist a stable complete marriage?

We set up a mathematical model for this problem by using a bipartite graph. Let G=(X, Δ, Y) be a bipartite graph in which

$$X= \{w_1, w_2, \ldots, w_n\}$$

is the set of n women and

$$Y= \{m_1, m_2, \ldots, m_n\}$$

is the set of n men. We join each woman-vertex to each man-vertex. The resulting bipartite graph is complete in the sense that it contains all possible edges between its two sets of vertices. Corresponding to each edges $\{w_i, m_j\}$ there is a pair p, q of numbers where p denotes the position of m_j in w_i's ranking of the men and q denoted the position of w_i in m_j's ranking of the women. A complete marriage of the women and men corresponds to a perfect matching (of n edges) in this bipartite graph G.

It is more convenient for notational purposes to use the model afforded by the preferential ranking matrix. This matrix is an n by n array of n rows, one corresponding to each of the n women $w_1, w_2, \ldots, w_n$, and n columns, one corresponding to each of the n men $m_1, m_2, \ldots m_n$. In the position at the intersection of row i and column j we place the pair p, q of numbers representing, respectively, the ranking of m_j by w_i and the ranking w_i by m_j. A complete marriage corresponds to a set of n positions of the matrix which includes exactly one position from each row and one position from each column.

Example4. 1. Let n=2, and let the preferential ranking matrix be

$$\begin{array}{c} \\ w1 \\ w2 \end{array}\begin{array}{c} \begin{array}{cc} m1 & m2 \end{array} \\ \begin{bmatrix} 1,2 & 2,2 \\ 2,1 & 1,1 \end{bmatrix} \end{array}$$

Thus, for instance, the entry 1,2 in the first row and first column means that w_1 has put m_1 first on her list and m_1 has put w_1 second on his list. There are two possible complete marriages.

1, $w_1 \leftrightarrow m_1$, $w_2 \leftrightarrow m_2$

2, $w_1 \leftrightarrow m_2$, $w_2 \leftrightarrow m_1$

The first is readily seen to be stable. The second is unstable since w_2 prefers m_2 to her spouse m_1, and m_2 prefers w_2 to his spouse w_1.

Theorem 3.1. For each preferential ranking matrix there is a stable complete marriage.

Proof. First for the existence of a complete marriage we define an algorithm.

Deferred acceptance algorithm also called Gale-Shapley algorithm begin with every woman marked as rejected. While there exists a rejected woman, do;

(1). Each woman marked as rejected chooses the man whom she ranks highest among all those men who have not yet rejected her.

(2). Each man picks out the woman he ranks highest among all those women who have chosen him and whom he has not yet rejected, defers decision on her and now rejects the others.

It follows from the description of the algorithm that as soon as there are no rejected women then each man is engaged to exactly one woman, and since there are as many men as women, each women is engaged to exactly one man. We now pair each man with the woman to whom he is engaged and obtain a complete marriage.

We now show that this marriage is stable.

Consider women A and B and men a and b such that A is paired with a and B is paired with b, but A prefers b to a. We show b cannot prefer A to B. since A prefers b to a during some stage of the algorithm, A chose b but A was rejected by b for some woman he ranked higher. But the woman b eventually gets paired with is at least as high on his list as any woman that he rejected during the course of the algorithm. Since A was rejected by b, b must prefer B to A. Thus there is no unstable pair and this completed marriage is stable

Example 4.2. We apply the deferred acceptance algorithm to the preferential ranking matrix designating the women as P, Q, R, S and the men as p, q, r, s.

$$A = \begin{array}{c} \\ P \\ Q \\ R \\ S \end{array} \begin{array}{c} \begin{array}{cccc} p & q & r & s \end{array} \\ \begin{bmatrix} 1,2 & 2,1 & 3,2 & 4,1 \\ 2,4 & 1,2 & 3,1 & 4,2 \\ 2,1 & 3,3 & 4,3 & 1,4 \\ 1,3 & 4,4 & 3,4 & 2,3 \end{bmatrix} \end{array}$$

The results of the algorithm are:

(i) P choose p, Q choose q, R choose s, S chooses p; p rejects S
(ii) S choose s; s rejects R
(iii) R chooses p; p rejects P
(iv) P chooses q; q rejects Q
(v) Q chooses p; p rejects Q
(vi) Q chooses r.

In (vi) there are no rejections, and

P ↔ q, Q ↔ r, R ↔ p, S ↔ s

is a stable complete marriage

If in the deferred acceptance algorithm we interchange the roles of the women and men and have the men choose women according to their rank of preferences, we obtain a stable complete marriage which may, but need not, differ from the one obtained by having the women choose men.

Example 4.3 We apply the deferred acceptance algorithm to the preferential ranking matrix in example 4.2. Where the men choose the women.

The results are:

(i) p chooses R, q chooses P, r chooses Q, s chooses P; P rejects s
(ii) s chooses Q; Q rejects s
(iii) s chooses S
In (iii) there are no rejections, and the complete marriage

p ↔ R, q ↔ P, r ↔ Q, s ↔ S

is stable. This is the same complete marriage obtained by applying the algorithm the other way around.

A stable complete marriage is called optimal for a woman provided that woman gets as a spouse a man whom she ranks at least as high as the spouse she obtains in every other stable complete marriage. In other words there is no stable complete marriage in which the woman gets a spouse which is higher on her list. A stable complete marriage is called women-optimal provided it is optimal for each woman. In a similar way we define a men-optimal stable complete marriage. It is not obvious that there exist women-optimal and men-optimal stable complete marriages. In fact it is not even obvious that if each women is independently given the best partner that she has in all the stable complete marriages, then this results in a pairing of the women and the men (it is conceivable that two women might end up with the same man in this way) clearly there can be only one women-optimal complete marriage and only one men-optimal complete marriage.

Example4. 4 Use the deferred acceptance algorithm to obtain both the women-optimal and men-optimal stable complete marriages for the preferential ranking matrix.

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1,3 & 2,3 & 3,2 & 4,3 \\ 1,4 & 4,1 & 3,3 & 2,2 \\ 2,2 & 1,4 & 3,4 & 4,1 \\ 4,1 & 2,2 & 3,1 & 1,4 \end{bmatrix} \end{array}$$

Conclude that for the given preferential ranking matrix there is only one stable complete marriage.

Solution. First denote the women by A,B,C and D and denote the men by a, b, c and d.

Now apply the deferred acceptance algorithm where women choose the men.

The results of the algorithm are :

(i) A chooses a, B chooses a, C chooses b, D chooses d; a rejects B

(ii) B chooses d; d rejects D

(iii) D chooses b ; b rejects C

(iv) C chooses a; a rejects A

(v) A chooses b; b rejects A

(vi) A chooses c

In (vi) there are no rejections and **B ↔ d, D ↔ b, C ↔ a, A ↔ c** is a stable complete marriage.

Where the men choose the women .

The results are:

(i) a chooses D, b chooses B ,c chooses D, d chooses C; D rejects a

(ii) a chooses C;C rejects d

(iii) d chooses B;B rejects b

(iv) b chooses D;D rejects c

(v) c chooses A

In (v) there are no rejections and **a ↔ C, d ↔ B, b ↔ D , c ↔ A** is a stable complete marriage.

Conclusion: In both cases we get the stable complete marriage

A ↔ c , B ↔ d , C ↔ a , D ↔ b.

REFERENCES

- Alan Tucker, **Applied combinatorics, 2nd edition, New York, 1984**
- Gary Froelich, Nancy Crisler, **Discrete Mathematics through applications**, 3rd edition, W.H.Freeman and company, New York, 2006
- John M Harris,Jeffry L.Hirst, Michael J.Hossinghoff, **Combinatorics and Graph theory**, Springer-Verlag New York, Inc., 2000
- Ralph P. Grimaldi, **Discrete and Combinatorial Mathematics (An applied introduction)**, 3rd edition , Rose-Hulman Institute of Technology, Addison-Wesley Publishing Company,Inc., 1994
- Richard A. Brualdi, **Introductory Combinatorics**, 4th edition , University of Wisconsin-Madison, 2004
- Vanden Eynden, Spence, Otto, Dossey, **Discrete Mathematics**, 5th edition ,Illionis State University, 2006
- *mathworld.wolfram.com › ... › Bicolorable Graphs*
- *en.wikipedia.org/wiki/Stable_marriage_problem*