MATCHINGS AND FACTORS

Submitted by

DANIYA JOSE

Register No: 170021032405

DANNY PETER

Register No: 170021032406

MARIA LAYA

Register No: 170021032419

Under the guidance of

Prof. Valentine D'Cruz

In partial fulfilment of the requirement for the award of
BACHELOR DEGREE OF SCIENCE in MATHEMATICS

2017-2020

DEPARTMENT OF MATHEMATICS

ST. PAUL'S COLLEGE, KALAMASSERY

(AFFILIATED TO M G UNIVERSITY, KOTTAYAM)

CERTIFICATE

This is to certify that the project report titled "MATCHINGS AND FACTORS" submitted by DANIYA JOSE(Reg no. 170021032405), DANNY PETER(Reg no. 170021032406) and MARIA LAYA(Reg. no:170021032419) towards partial fulfilment of the requirements for the award of Degree of Bachelor of Science in Mathematics is a bonafide work carried out by them during the academic year 2017-2020.

Supervising Guide	Head of the Department
Prof. Valentine D'Cruz	**Dr. Savitha K S**
Principal	
Associate professor	**Assistant Professor**
Department of Mathematics	**Department of Mathematics**

Examiner:

Place: Kalamassery

Date:

DECLARATION

We ,DANIYA JOSE (Reg. no:170021032405), DANNY PETER (Reg. no:170021032406) and MARIA LAYA(Reg. no:170021032419) hereby declare that this project entitled "MATCHINGS AND FACTORS" is an original work done by us under the supervision and guidance of prof. Valentine D'Cruz, faculty, Department of Mathematics in St. Paul's college Kalamassery in partial fulfilment for the award of The Degree of Bachelor of Science in Mathematics under Mahatma Gandhi University. We further declare that this project is not partly or wholly submitted for any other purpose and the data included in the project is collected from various sources and are true to the best of our knowledge.

Place: Kalamassery DANIYA JOSE

Date: DANNY PETER

MARIA LAYA

ACKNOWLEDGEMENT

For any accomplishment or achievement, the prime requisite is the blessing of the Almighty and it's the same that made this world possible. We bow to the lord with a grateful heart and prayerful mind.

It is with great pleasure that we express our sincere gratitude to our beloved teacher prof. Valentine D'Cruz, Department of Mathematics, St. Paul's College, for her overwhelming support, motivation and encouragement.

We would like to acknowledge our deep sense of gratitude to Dr. Savitha K S, Head of Department of Mathematics and all the faculty members of the department and our friends who helped us directly and indirectly through their valuable suggestions and self-criticisms, which came a long way in ensuring that this project becomes a success.

We also express our sincere gratitude to Prof. Valentine D'Cruz, Principal, St. Paul's College, Kalamassery for the support and inspiration rendered to us in this project report.

MATCHINGS AND FACTORS

TEAM MEMBERS;

MARIA LAYA

DANIYA JOSE

DANNY PETER

CONTENTS

INTRODUCTION

Graph theory is a delightful playground for the exploration of proof techniques in discrete mathematics and its results have applications in many areas computing, social and natural sciences. In this project we will examine about an important branch of graph theory called **Matchings and Factors**. Due to the mature techniques and wide range of application, matchings and factors became a useful tool in investigation of many theoretical problems and practical issues.

Graphs are a powerful tool used in various subfields of science and engineering. When graphs are used for the representation of structured objects, the problem of measuring object similarity turns into the problem of computing the similarity of the graphs. Which is also known as graph matching.

A graph matching is nothing but a subgraph of a graph with a set of edges that doesn't have a set of common vertices. A factor of graph 'G' is a spanning subgraph. ie, a subgraph that has the same vertex set as G. Matching has many applications in flow networks, scheduling and planning, stable marriage problems, neural networks in artificial intelligence and many more.

CHAPTER 1

PRELIMINARY DEFINITIONS

- Let ƒ(G) denote the size of maximum matching in G .A Vertex v € V(G) is called ***critical i***f ƒ(G)>ƒ(G-v)
- We define three special subsets of the vertex set of G
  D(G)={u:u is not critical}
  A(G)={u:u is critical and has non critical neighbour }
  C(G)={u:uis critical and all neighbours of u are critical }
  Set A(G) is called the Tutte set of G clearly,the set D(G) ,A(G) and C(G) partition the vertex set of G .

- Let L={M_1,..........M_d} be a family of equisized matching in G . we again define the three subsets of G using the family L .

  D(L)={u:there exist i € [d]such that u is free in M_i }
  A(L)={u:u$\notin$ D(L) and u has a neighbor in D(L)}
  C(L)=V(G)\(D(L)UA(L))

- Let G =(V,E) be a graph M ⊆E is called as ***matching*** OF G if for all v €V we have | {e€M:v is incident on e€E}|≤1
- A matching M of G is said to be ***maximal*** if for all e€E\M the set of edges given by MU{e} is not a matching of G .
- The ***size*** of a matching M of G is the number of edges it contains and is denoted by |M|
- A Matching M of G is said to be **maximum** if for all matching M' of G we have |M|≥|M'|.A maximum matching is always maximal but not vice –versa .
- Let M be matching of G . a vertex v€V is said to be ***M-saturated*** if M contains an edge incident on v .otherwise v is said to be ***M –unsaturated***.
- A matching M of G is said to be ***perfect*** if all vertices of G are M –saturated . A graph with an odd number of vertices can never admit a perfect matching .
- A matching M of G is said to be ***A-perfect*** if each vertex in A is M –saturated.A perfect matching is a ***V-perfect*** matching.

CHAPTER 2

MATCHINGS AND COVERS

DEFINITION : A **Matching** in a graph *G* is a set of non-loop edges with no shared endpoints. The vertices incident to the edges of a matching *M* are **saturated** by *M* ; the others are **unsaturated** (we say *M* - saturated and *M* - unsaturated). A **PerfectMatching** in a graph is a matching that saturates every vertex.

SECTION 2.1

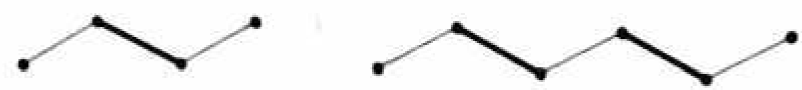
MAXIMUM MATCHINGS

A matching is a set of edges, so its size is the number of edges. We can seek a large matching by iteratively selecting edges whose endpoints are not used by the edges already selected, until no more are available. This yields a maximal matching but maybe not a maximum matching.

2.1.1. Definition. A maximal matching in a graph is a matching that cannot be enlarged by adding an edge. A maximum matching is a matching of maximum size among all matchings in the graph.

A matching M is maximal if every edge not in M is incident to an edge already in M. Every maximum matching is a maximal matching, but the converse need not hold.

2.1.2. Example. Maximal ≠ maximum. The smallest graph having a maximal matching that is not a maximum matching is P4. If we take the middle edge, then we can add no other, but the two end edges form a larger matching. Below we show this phenomenon in P4 and in P6



In Example 2.1.2, replacing the bold edges by the solid edges yields a larger watching. This rives us way to look for larger matching

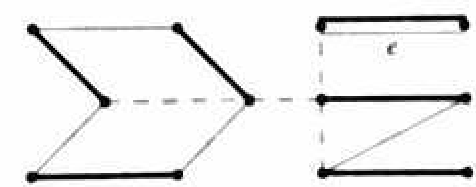
2.1.3. Definition. Given a matching M, an M-alternating path is a path that alternates between edges in M and edges not in M . An M-alternating path whose endpoints are unsaturated by M is an M-augmenting path

Given an M-augmenting path P, we can replace the edges of M in P with the other edges of P to obtain new matching M′ with one more edge. Thus when M is a maximum matching, there is no M-augmenting path

In fact, we prove next that maximum matchings are characterized by the absence of augmenting paths. We prove this by considering two matching and examining the set of edges belonging to exactly one of them. We define this operation for any two graphs with the same vertex set (the operation is defined in general for any two sets: see Appendix A).

2.1.4. Definition. If G and H are graphs with vertex set V, then the symmetric difference GΔH is the graph with vertex set V whose edges are all those edges appearing in exactly one of G and H. We also use this notation for sets of edges, in particular, if M and M′ are matchings, then MΔM′ = (M-M′) U (M′-M)

2.1.5. Example. In the graph below, M is the matching with five solid edges, M′ is the one with six bold edges, and the dashed edges belong to neither M nor M′. The two matchings have one common edge e, it is not in their symmetric difference. The edges of MΔM′ form a cycle of length 6 and a path of length 3.



2.1.6. Lemma. Every component of the symmetric difference of two matchings is a path or an even cycle.

Proof: Let M and M′ be matchings, and let F = MΔM′. Since M and M′ are matchings, every vertex has at most one incident edge from each of them. Thus F has at most two edges at each vertex. Since $\Delta(F)\leq 2$,every component of F is a path or a cycle. Furthermore, every path or cycle in F alternates between edges of M-M′ and edges of M′-M. Thus each cycle has even length, with an equal number of edges from M and from M′

2.1.7. Theorem. (Berge [1957) A matching M in a graph G is a maximum matching in G if and only if G has no M-augmenting path.

Proof: We prove the contrapositive of each direction: G has a matching larger than M if and only if G has an M-augmenting path. We have observed that an M-augmenting path can be used to produce a matching larger than M.

For the converse, let M′ be a matching in G larger than M. We construct an M-augmenting path. Let F=MΔM′ By Lemma 3.1.6 , F consists on path and even cycles the cycles have the same number of edges from M and M′. Since $|M'|>|M|$,F must have a component with more edges of M′ than of M . Such a component can only be a path that starts and ends with an edge of M′ , thus it is an M-augmenting path in G .

SECTION 2.2

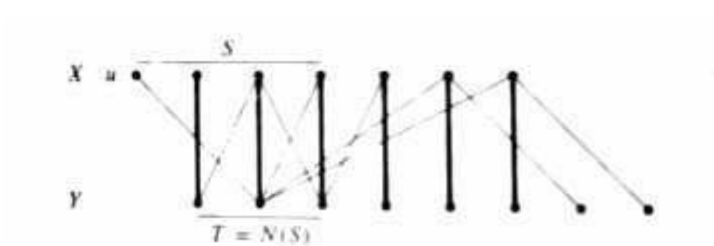
HALL'S MATCHING CONDITION

When we are filling jobs with applicants there may be many more applicants than jobs: successfully filling the jobs will not use all applicants. To model this problem, we consider an XY-graph , and we seek a matching that saturates X.

If a matching M saturates X. then for every S⊆X there must be at least |S| vertices that have neighbors in S because the vertices matched to S must be chosen from that set. We use NG(S) or simply N(s) to denote a set of vertices having a neighbour in S. Thus |N(S)|≥|S| is a necessary condition.
The condition "For all S⊆X, |N(S)|≥|S is Hall’s condition.Hall’s proved that the obvious necessary condition is also sufficient (TONCAS)

2.2.1. Theorem. (Hall's Theorem-P Hall [1935]) An X,Y-bigraph G has a matching that saturates X if and only if |N(S)|≥|S|for all S⊆X.

Proof: Necessity. The |S| vertices matched to S must lie in N(S).
Sufficient. To prove that Hall's Condition is sufficient, we prove the contrapositive. If M is a maximum matching in G and does M does not saturate X, then we obtain set S⊆X such that |N(S)|<|S|.Let $\mu \in$ X be a vertex unsaturated by M. Among all the vertices reachable from u by M-alternating paths in G, let S consist of those in X, and let T consists of those in Y.Note μ ∈S



We claim that M matches T with S-{ μ}. The M-alternating paths from u reach Y along edges not in M and return to X along edges in M. Hence every vertex of S-{ μ} is reached by an edge in M from a vertex in T. Since there is no M-augmenting path, every vertex of T is saturated; thus an M-alternating path reaching y ∈ T extends via M to a vertex of S. Hence these edges of M yield a bijection from T to S-{ μ}, and we have |T|=|S-{ μ}|,

The matching between T and S-{ μ} yields T ⊆N(S). In fact , T=N(S). Suppose that y ∈ Y-T has a neighbour *v* ∈S. The edge *vy* cannot be *in M,*since u is unsaturated and the rest of S is matched to T by M. Thus adding uy to an M-alternating path reaching v yields an M-alternating path to y. This contradicts y ∉ T, and hence *vy* cannot exist.

With T =N(S),we have proved that |N(S)|=|T|=|S|-1 < |S| for this choice of S. This complete the proof of contrapositive

One can also prove sufficient by assuming Hall's Condition, supposing that no matching saturates X and obtaining a contradiction. As we have seen, lack of a matching saturating Xyields a violation of Hall's Condition. Contradicting the hypothesis usually means that the contrapositive of the desired implication has been proved Thus we have stated the proof in that language

2.2.2. Remark. Theorem 3:1 1 implies that whenever an X Y bigraph has no matching saturating X. we can verify this by exhibiting a subset of X with too few neighbours. Note also that the statement and proof permit multiple edges.

Many proofs of Hall's Theorem have been published: see Milky [1971, p38] and Jacobs (1969) for summaries. A proof by M. Hall (1948) leads to is lower bound on the number of matchings that saturates X, as function of the vertex degrees. We consider algorithmic aspects in Section 3.2

When the sets of the bipartite have the same size. Hall's Theorem is the Marriage Theorem, proved originally by Frobenius (1917). The name arises from the setting of the compatibility relation between a set of n men and a set of n women. If every man is compatible with k women and every woman is compatible with k men, then a perfect matching must exist. Again multiple edges are allowed, which enlarges the scope of application.

2.2.3. Corollary. For $k > 0$. every k-regular bipartite graph has n perfect matching

Proof: Let G be a k-regular X, Y-bigraph. Counting the edges by endpoints in X and by endpoints in Y shows that $k\,|X| = k\,|Y|$, so $|X|= |Y|$. Hence it suffices to verify Hall's Condition, a matching that saturates X will also saturate Y and be a perfect matching

Consider $S \subseteq X$, Let m be the number of edges from S to N(S). Since G is k- regular $m=k|S|$. These m edges are incident to N(S), so $m \le k\,|N(S)|$. Hence $k|S|\le k\,|N(S)|$,which yields $|N(S)|\ge|S|$ when $k > 0$. Having chosen $S \subseteq X$ arbitrarily, we have established Hall's condition

One can also non contradiction here. Assuming that G has no perfect matching yields a set $S \subseteq X$ such that $|N(S)|<|S|$. The argument obtaining a contradiction amount to a rewarding of the direct proof given above .

SECTION 2.3

MIN-MAX THEOREMS

When a graph G does not have a perfect matching. Theorem 3.1.6 allows us to prove that M is a maximum matching by proving that G has no M-augmenting path. Exploring all M-alternating paths to eliminate the possibility of augmentation could take a long time.

We faced a similar situation when proving that a graph is not bipartite.Instead of checking all possible partitions, we can exhibit an odd cycle. Here again, instead of exploring all M-alternating paths, we would prefer to exhibit an explicit structure in G that forbids a matching larger than M.

2.3.1. Definition. A vertex cover of a graph G is a set $Q\subseteq V(G)$ that contains at least one endpoint of every edge. The vertices in Q cover E(G).

In a graph that represents a road network (with straight roads and no isolated vertices), we can interpret the problem of finding a minimum vertex cover as the problem of placing the minimum number of policemen to guard the entire road network. Thus "cover" means "watch" in this context.

Since no vertex can cover two edges of a matching, the size of every vertex cover is at least the size of every matching. Therefore, obtaining a matching and a vertex cover of the same size PROVES that each is optimal. Such proofs exist for bipartite graphs, but not for all graphs.

2.3.2. Example. Matching and vertex covers. In the graph on the left below we mark a vertex cover of size 2 and show a matching of size 2 in bold. The vertex cover of size 2 prohibits matchings with more than 2 edges, and the matching of size 2 prohibits vertex covers with fewer than 2 vertices. As illustrated on the

right, the optimal values differ by 1 for an odd cycle. The difference can be arbitrarily large.

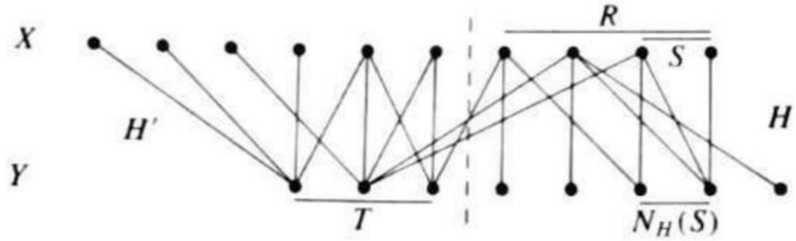
2.1.16. Theorem. (König[1931], Egerváry[1931]) If G is a bipartite graph, then the maximum size of a matching in G equals the minimum size of a vertex cover of G.

Proof: Let G be an X,Y-graph. Since distinct vertices must be used to cover the edges of a matching, $|Q| \geq |M|$ whenever Q is vertex cover and M is a matching in G. Given a smallest vertex cover Q of G, we construct a matching of size $|Q|$ to prove that equality can always be achieved.

Partition Q by letting $R = Q \cap X$ and $T = Q \cap Y$. Let H and H' be the subgraphs of G induced by $R \cup (Y - T)$ and $T \cup (X - R)$, respectively. We use Hall's Theorem to show that H has a matching that saturates R into Y - T and H has a matching that saturates T. Since H and H’ are disjoint, the two matchings together form a matching of size |Q| in G.

Since RUT is a vertex cover, G has no edge from Y-T to X-R For each S ⊆ R, we consider NH(S), which is contained in Y-T . If |NH(S)|<|S|, then we can substitute NH(S) for S in Q to obtain a smaller vertex cover, since NH(S) covers all edges incident to S that are not covered by T.

The minimality of Q thus yields Hall's Condition in H, and hence H has a matching that saturates R. Applying the same argument to H' yields the matching that saturates T.



As graph theory continues to develop, new proofs of fundamental results like the König-Egerváry Theorem appear; see Rizzo [2000].

2.3.3. Remark. A min-max relation is a theorem stating equality between the answers to a minimization problem and a maximization problem over a class of instances. The König-Egerváry Theorem is such a relation for vertex covering and matching in bipartite graphs.

For the discussion in this text, we think of a dual pair of optimization problems as a maximization problem M and a minimization problem N, defined on the same instances (such as graphs), such that for every candidate solution M to **M** and every candidate solution N to**N**, the value of **M** is less than or equal to the value of N. Often the "value" is cardinality, as above where **M** is maximum matching and **N** is minimum vertex cover.

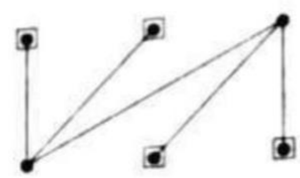
When **M** and **N** are dual problems, obtaining candidate solutions M and N that have the same value PROVES that M and N are optimal solutions for that instance. We will see many pairs of dual problems in this book. A min-max relation states that, on some class of instances, these short proofs of optimality exist. These theorems are desirable because they save work! Our next objective is another such theorem for independent sets in bipartite graphs.

SECTION 2.4

INDEPENDENT SETS AND COVERS

We now turn from matchings to independent sets. The independence number of a graph is the maximum size of an independent set of vertices.

2.4.1. Example. The independence number of a bipartite graph does not always equal the size of a partite set. In the graph below, both partite sets have size 3, but we have marked an independent set of size 4.



No vertex covers two edges of a matching. Similarly, no edge contains two vertices of an independent set. This yields another dual covering problem.

2.4.2.. Definition. An edge cover of G is a set L of edges such that every vertex of G is incident to some edge of L.

We say that the vertices of G are covered by the edges of L. In Example3.4.1, the four edges incident to the marked vertices form an edge cover; the remaining two vertices are covered "for free".

Only graph without isolated vertices have edge covers. A perfect matching forms an edge cover with n(G)/2 edges. In general, we can obtain an edge cover by adding edges to a maximum matching.

2.4.3. Definition. For the optimal size of the sets in the independence and covering problems we have defined, we use the notation below.

maximum size of independent set	α(G)
maximum size of matching	α'(G)
minimum size of vertex cover	β(G)
minimum size of edge cover	β'(G)

A graph may have many independent sets of maximum size (C5 has five of them), but the independent numberα(G) is a single integer (α(C5) = 2). The notation treats the numbers that answer these optimization problems as graph parameters, like the order, size, maximum degree, diameter, etc. Our use of α'(G) to count the edges in a maximum matching suggests a relationship with the parameter α(G) that counts the vertices in a maximum independent set.We explore this relationship in Section 7.1.

We use β(G) for minimum vertex cover due to its interaction with maximum matching. The "prime" goes on β'(G) rather than on β(G) because β(G) counts a set of vertices and β'(G) counts a set of edges.

In this notation, the König-Egerváry Theorem states that α'(G) = β(G) for every bipartite graph G. We will prove that also α(G) = β'(G) for bipartite graphs without isolated vertices. Since no edge can cover two vertices of an independent set, the inequality β'(G) ≥ α(G) is immediate. (When S ⊆ V(G), we often use s^- to denote V(G)- S, the remaining vertices).

2.4.4. Lemma. In a graph G, S ⊆ V(G) is an independent set if and only if s^- is a vertex cover, and hence α(G)+ β(G) = n(G)

Proof: If S an independent set, then every edge is incident to at least one vertex of s^-. Conversely, if s^- coversall the edges, then there are no edge joining vertices of S. Hence every maximum Independent set in the complement of a minimum vertex cover, and α(G)+ βIG) =n(G).

The relationship between matchings and edge covering is more subtle Nevertheless, a similar formula holds

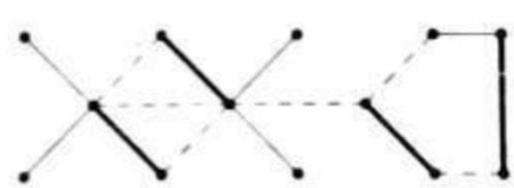
2.4.5. Theorem. (Gallai[1959]) If G in a graph without isolated vertices, then α'(G)+ β'(G)=n(G)

Proof: From a maximum matching M. we will construct an edge cover of size n(G)-|M|. Since a smallest edge cover is no bigger than this cover, this will imply that β(G)≤n(G)- α'(G). Also, from a minimum edge cover L, we will construct a matching of size n(G)-|L|.

Let M be a maximum matching in G. We construct an edge cover of G by adding to Mone edge incident to each unsaturated vertex We have used one edge for each vertex, except that each edge of M takes care of two vertices, so the total size of this edge cover is n(G)-|M|, as desired.

Now let L be a minimum edge cover. If both endpoints of an edge *e* belong to edges in L other than *e*, then *e* ∉ L, sinceL- (*e*) is also an edge cover. Hence each component formed by edges of L has at most one vertex of degree exceeding 1 and is a star (a tree with at most one non-leaf). Let K be the number of these components. Since L has one edge for each non-central vertex in each star, we have |L|= n(G)-k, We form a matching M of size k = n(G)-|L|by choosing one edge from each star in L

2.4.6. Example. The graph below has 13 vertices. A matching of size 4 appears in bold, and adding the solid edges yields an edge cover of size 9. The dashed edges are not needed in the cover. The edge cover consists of four stars; from each we extract one edge (bold) to form the matching.



2.4.7. Corollary, (König[1916]) If G is a bipartite graph with no isolated vertices, then α(G) = β'(G)

Proof: By Lemma 3.4.4 and Theorem 3.4.5, α(G) + β(G) = α'G) + β'(G).Subtracting the König-Egerváry relation α'(G)= βIG) completes the proof.

5ECTION 2.5

DOMINATING SETS

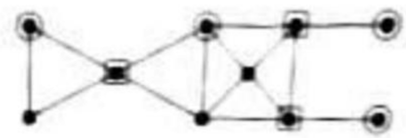
The edges incident to The edges covered by any vertex in a vertex cover are the edges incident they form a star The vertex cover problem can be described as cover edge set with the fewest stars. Sometimes we instead want to cover the vertex with fewest stars. This is equivalent to our next graph parameter.

2.5.1. Example. A company wants to establish transmission towers in a remote region. The towers are located at inhabited buildings, and each inhabit building must be reachable. If a transmitter at x can reach y, then also one at y can reach x .Given the pairs that can reach each other, how many transmitters are needed to cover all the buildings

A similar problem comes from recreational mathematics: How many queens are needed to attack all squares on a chessboard?

2.5.2. Definition. In a graph G, a set S⊆V(G) is a dominating set if every vertex not in S has a neighbour in S. The domination number (G) is the minimum size of a dominating set in G

2.5.3. Example. The graph G below has a minimal dominating set of size 4 circles and a minimum dominating set of size 3 (squares): γ(G)) = 3.



Berge (1962) introduced the notion of domination. Ore (1962) coined this terminology, and the notation (G) appeared in an early survey (CockayneHedetniemi [1977). An entire book (Haynes-Hedetniemi-Slater (1998)) is devoted to domination and its variations

2.5.4. Example. Covering the vertex set with stars may not require as many stars as covering the edge set. When a graph G has no isolated vertices, every vertex cover is a dominating set, so (G) ≤ (G). The difference can be large,(Kn) = 1, but β (Kn)=n-1

When studying domination as an extremal problem, we try to obtain bounds in terms of other graph parameters, such as the order and the minimum degree. A vertex of degree k dominates itself and k other vertices; thus every dominating set in a k-regular graph G has size at least n(G)/(k+1). For every graph with minimum degree k, a greedy algorithm produces a dominating set not too much bigger than this.

2.5.5. Definition. The closed neighbourhood N[*v*]of a vertex v in a graph is N(*v*)∪ [*v*].It is the set of vertices dominated by *v.*

2.5.6. Theorem. (Arnautovic (1974), Payan (1975)) Every n-vertex graph with minimum degree k has a dominating set of size at most n (1+1n(k+1)/k+1

Proof Alon (1990)) Let G be graph with minimum degree k. Given S⊆ V(G). Let *U*be the set of vertices not dominated by S . We claim that some vertex v outside S dominates at least |*U*|(k+1)/n vertices of U. Each vertex in *U has at least k neighbor,so* ∑*|*N[v] ≥ */*U*/*(*k+1).* Each vertex of G is counted atmost n times by these|U|sets,so some vertex y appears atleast |U|(k+1)/n times and satisfies the claim.

We iteratively select a vertex that dominates the most of the remaining undominated vertices. We have proved that when r undominated vertices remain, after the next selection at most r(1-(k+1)/n) undominated vertices remain hence after a n (n(k+1)/k+1 steps the number of undominated vertices is at most n(1-k+1/n)ln(k+1)/k+1< ne ln(k+1) = n/k+1

The selected vertices and these remaining undominated vertices together a dominating set of size at most n(1+ln(k+1)/k+1

2.5.7. Remark. This bound is also proved by probabilistic methods in Theorem 8.5.10 Caro-Yuster-West (2000) showed that for large the total domination number satisfies a bound asymptotic to this. Alan 1990] uod probabilistic methods to show that this bound is asymptotically sharp when is large Exact bounds remain of interest for small k Among connected n -vertex graphs δ(G) ≥2 implies yG) ≤ 2 n/5 (McCuaig-Shepherd (1989), with seven small exceptions, andδ(G) ≥ 3 implies y(G) ≤ 3n/8 Reed (1996), Exercise 53 requests constructions achieving these bounds

Many variations on the concept of domination are studied In Example3.1.25, for example, one might want the transmitters to be able to communicate with each other, which requires that they induce a connected subgraph

2.5.8. Definition. A dominating set 5 in G is a connected dominating set if GIS) is connected an independent dominating set if GIS is independent, and a total dominating set if GIS has no isolated vertex

Each variation add a constraint, so dominating sets of these types are at least as y(G).Exercises 54-60 explore these variations Studying independent dominating sets amounts to studying maximal independent set This leads to a nice result about claw-free graphs.

2.5.9. Lemma. A set of vertices in a graph is an independent dominating set if and only if it is a maximal independent set.

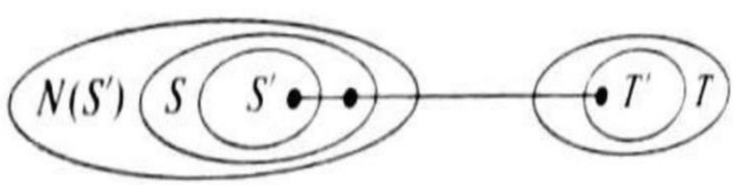
Proof: Among independent setS is maximal if and only if every has a neighbor in S, which is the condition for S to be a dominating set

2.5.10. Theorem. (Bollobås-Cockayne [1979]) Every claw-free graph he independent dominating set of size y(G).

Proof: Let S be a minimum dominating set in a claw-free graph G. Let S' be a maximal independent subset of S.Let T = V(G)- N(S'). Let T' be a maximal independent subset of S.

Since T' contains no neighbor of S', S'UT' is independent. Since S meximal to S we have S ⊆N(S'). Since T' is maximal in T, T' dominates Hence S'UT' is a dominating set.

It remains to show that |SUT'|≤ y(G). Since S' is meximal in S,T independent, and G is claw-free, each vertex of S-S' has at most one neighbor in T'. Since S is dominating, each vertex of T has at least one neighbor in S-S. Hence |T'|≤|S -S'| , which yields |S'UT'| = |S|= y(G).



CHAPTER 3

3.1 MATCHINGS IN GENERAL GRAPHS

When discussing perfect matchings in graphs,it is natural to consider more general subgraphs.

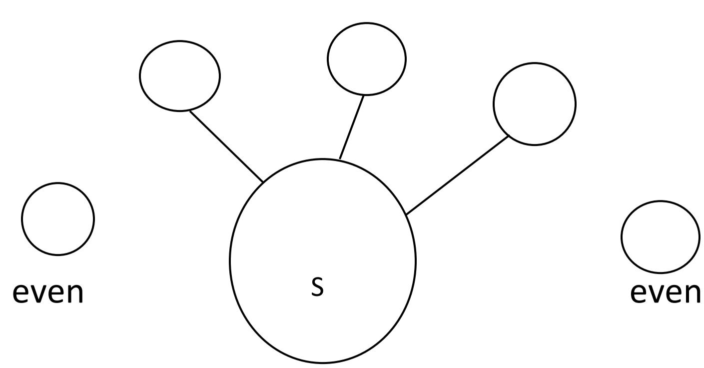
Definition.A ***Factor*** of a graph *G* is a spanning subgraph of *G* .A *k-factor* is a spanning *k*-regular subgraph .An ***odd component*** of a graph is a component of odd order ; the number of odd components of H is o(H).

Remark. A 1-factor and a perfect matching are almost the same thing.The precise distinction is that "1-factor" is a spanning 1-regular subgraph of *G* ,while "perfect matching" is the set of edges in such a subgraph.

A 3-regular graph that has a perfect matching decomposes into a 1-factor and a 2-factor .

3.2 TUTTE'S 1-FACTOR THEOREM

Tutte found a necessary and sufficient condition for which graphs have 1-factors .If *G* has a 1-factor and we consider a set $S \subseteq V(G)$,then every odd component of *G*-*S* has a vertex matched to something outside it,which can only belong to *S* . since these vertices of *S* must be distinct ,$o(G-S) \leq |S|$.



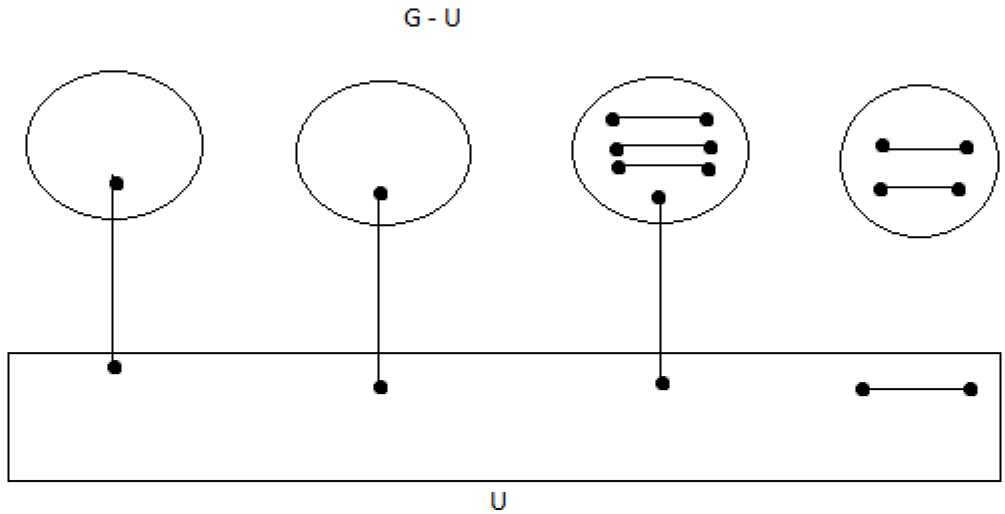
The condition “for all S⊆V(G),o(G-S) ≤│S│” is **Tutte’s Condition.** Tutte proved that his obvious necessary condition is also sufficient(TONCAS).

3.2.1 THEOREM (Tutte 1947) A graph *G* has a 1-factor if and only if o(G-S)≤│S│for every *S⊆V(G)*

PROOF Necessity the odd component G-S must have vertices matched to distinct vertices of S .

Sufficiency when we add an edge joining two components of G-S ,the number of odd components does not increase (odd and even together become one odd omponent ,two components of the same party become one even component).H ence Tutte’s condition is preserved by addition of edges : if G’=G+e and S⊆V(G),then o(G’-S) ≤O(G-S) ≤│S│.Also if G’=G+e has no 1-factor ,then G has no 1-factor . Therefore, the theorem holds unless there exists a simple graph G such that G satisfies Tutte's Condition, G has no 1-factor,and adding any missing edge to G yields a graph with a 1-factor. Let G be such a graph. We obtain a contradiction by showing that G actually does contain a 1-factor.
Let U be the set of vertices in G that have degree *n*(G) - 1.

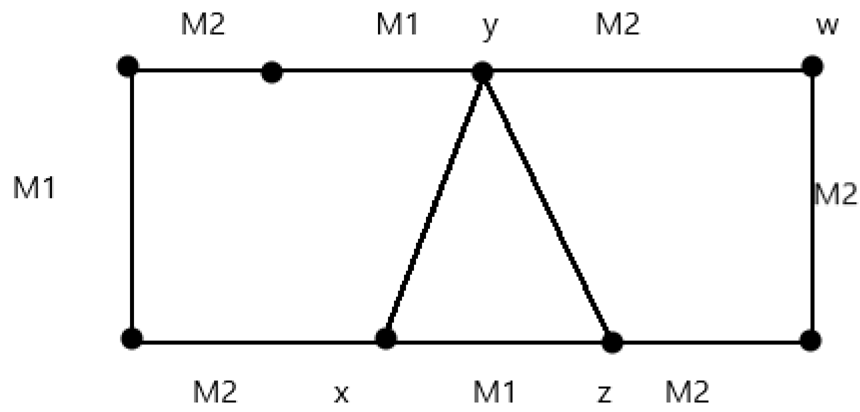
Case 1: G-U consists of disjoint complete graphs. In this case, the vertices each component of G - U can be paired in any way, with one extra in the 0dd components. Since o(G - U) ≤│U│ and each vertex of U is adjacent to all of G-U, We can match the leftover vertices to vertices of U.
The remaining vertices are in U, which is a clique. To complete the 1-factor, we need only show that an even number of vertices remain in U. we have matched an even number , so it suffices to show that n(G) is even this follows by invoking tutte’s condition for S=ϕ,since a graph of odd order would have a component of odd order.



Case 2: G- U is not a disjoint union of cliques. In this case, G- U has two vertices at distance 2; these are nonadjacent vertices x, z with a common neighbor y ∉ U . Furthermore, G- U has another vertex w not adjacent to y, since y∉ U. By the choice of G, adding an edge to G creates a 1-factor; let $M_1$and M_2be 1-factors in G +xz and G+ yw, respectively. It suffices to show that $M_1 \Delta M_2$ contans a 1-factor avoiding xz and yw, because this will be a 1-factor in G.

Let F=$M_1 \Delta M_2$. Since xz € M_1 - M_2 and yw € M_2- M_1both xz and yw are in F . Since every vertex of G has degree 1 in each of $M_1$and M_2 , every vertex of G has degree 0 or 2 in F. Hence the components of F are even cycles and isolated vertices Let C be the cycle of F containing xz .

If C contains both yw and xz, as shown below, then to avaid them we use yx or yz n the portion of C starting from y along yw, we use edges of M_1 to avoid using yw .When we reach(x,z) , we use zy if we arrive at z (as shown) otherwise, we use xy. In the remainder of C we use the edges of M_2. We have produced a 1-factor of C that does not use xz or yw Combined with M_1 or M_2 outside C we have a 1-factor of G.

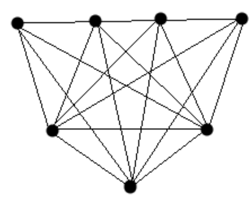


3.2.2 Remark. Like other characterization theorems, Theorem 3.2.1 yields short verifcaions both when theproperty holds and when it doesn't exist ,. We prove that G has a l-factor exists by exhibiting one. When it doesn't exist, Theorem 3.2.1 guarantees that we Canexhibit a set whose deletion leaves too many odd compornents.

3.2.3 Remark. For a graph G and any S ⊆ V(G), counting tha vertices modulo 2 shows that |S| +o(G-S) has the same parity as n(G). Thus also the difference of (G-S) -|S| has the same parity as n(G). We conclude that l1 n(G) is even and G has no 1-factor, then o(G-S) exceeds |S| by at least 2 for some S.

For non-bipartite graphs (such as odd cycles), there may be a gap between α'(G) and β(G) . Nevertheless, another minimization problem yields a min-max relation for .α'(T) in general graphs.This min-max relation generalizes Remark 3.2.3 The proof uses a graph transtormation that involves a general graph operation.

3.2.4. Definition. The join of simple graphs G and H, written G v H, is the graph obtained from the disjoint union G+ H by adding the edges [xy :x € V(G), y€ V(H).



$P_4 \vee k_3$

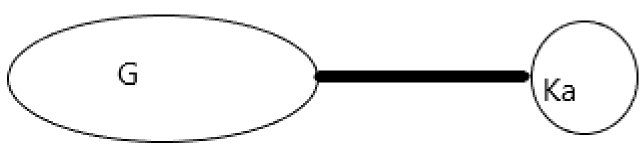
3.2.5. Corollary. (Berge-Tutte Formula-Berge [1958) The largest number of vertices saturated by a matching in G is minS⊆V(G){n(G)-d(S),where d(S) = o(G- S)-|S|

Proof: Given S⊆ V(G), at most |S| edges can match vertices of S to vertices in odd components of G-S, so every matching has at least o(G-S)-|S|unsaturated vertices. We want to achieve this bound.

Let d= max{o(G- S)-|S|: S V(G)}. The case S =ϕ yields d ≥ 0. LetG' =GvK. Since d(S) has the same parity as n(G) for each S, we know that n(G') is even. If G' satisfies Tutte's Condition, then we obtain a matching ot the desired Size in G from a perfect matching in G', because deleting the *d* added vertices eliminates edges that saturate at most *d* vertices of G.

The condition o(G'-S') ≤ |S| holds for S'= ϕ because n(G') is even.

If S' is nonempty but does not contain all of Ka, then G'- S' has only one component, and 1≤ |S'|. Finally, when K CS', we let S = S'- V(K_d). We have G'-S' = G--S, so o(G'-S') = o(G-S)≤ |S| +d = |S'|We have verified that G' satisfies Tutte's Condition.



Corollary 3.2.5**.** guarantees that there is a short PROOF that a maximum matching indeed nas maximum size by exhibiting a vertex set S whose deletion leaves the appropriate nunmber of odd components.

Most applications of Tutte's Theorem involve showing that some other condition implies Tutte' s Condition and hence guarantees a 1-factor Some were proved by other means long before Tutte's Theorem was available.

3.2.6 Corollary. (Petersen [1891) Every 3-regular graph with no cut-edge has a 1-factor.

Proof: Let G be a 3-regular graph with no cut-edge. We prove that G satisties Tutte's Condition. Given SCV(G), we count the edges between S and the odd components of G-S . Since G is 3-regular, each vertex of S is incident to at most three such edges. If each odd component H of G-S is incident to at least three such edges, then 3o(G- S) ≤ 3 |S| and hence o(G-S) ≤ |S|, as desired.

Let m be the number of edges from S to H. The sum of the vertex degrees in H is 3n(H)- *m.* Since H is a graph, the sum of its vertex degrees must be even. Since n(H) is odd, we conclude that *m* must also be odd. Since G has no cut-edge, *m* cannot equal 1. We conclude that there are at least three edges from S to H, as desired.

Proof by contradiction would also be natural here. Assuming o(G-S)>|S| also leads to o(G -S)≤ IS|, so we rewrite the proof directly. Corollary 3.2.6 is best possible; the Petersen graph satisfies the hypothesis but does not have two edge-disjoint 1-factors (Petersen |1898|).

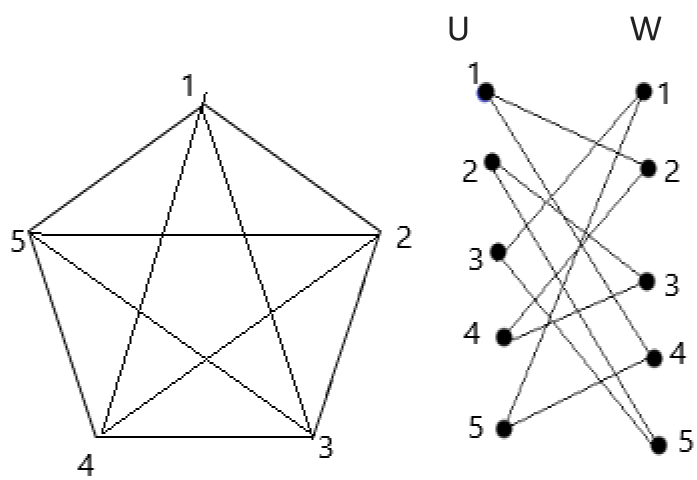
Petersen also proved a sufficient condition for 2-factors. A connected graph with even vertex degrees is Eulerian and decomposes into edge-disjoint cycles . For regular graphs of even degree, the cycles in 8ome decomposition can be grouped to form 2-factors.

3.2.7 Theorem. (Petersen (1891)) Every regular graph of even degree has a 2-factor.

Proof: Let G be a 2k-regular graph with vertices $v_1,\dots\dots,v_n$Every component of G is Eulerian, with some Eulerian eircuit C. For each component, define a bipartite graph H with vertices $u_1,\dots\dots u_n$ and $w_1,\dots\dots w_n$.by putting $u_i \leftrightarrow w_j$ if v_1, immediately follows u_1, somewhere on C. Because C enters and exits each vertex k times, H is k-regular. (Actuaiiy, H is the split of the digraph obtainedby orienting G in accordance with C.

Being a regular bipartite graph, H has a 1-fáctor M . The edge incident to w_1 in H corresponds to an edge entering $v_1$in C. The edge incident to u_i, in H corresponds to an edge exiting v_j. Thus the 1-factor in H transforms into a 2-regular spanning subgraph of this component of G. Doing this for each component of G yields a 2-factor of G.

3.2.8. Example. Construction of a 2-factor Consider the Eulerian circuit in G = k_5 that successively visits 1231425435. The corresponding bipartite graph H is on the right. For the 1-factor whose u, w-pairs are 12, 43, 25, 31, 54, the resulting 2-factor is the cycle (1,2, 5, 4, 3). The remaining edges form another 1-factor, which corresponds to the 2-tactor (, 4, 2,3, 5) that remains in G.



f-FACTORS OF GRAPHS

A factor is a spanning subgraph of G; we ask about existence of factors of special types A k-factor is a k-regular factor we have studied 1-factors and

2-factors. We can try to specity the degree at each vertex.

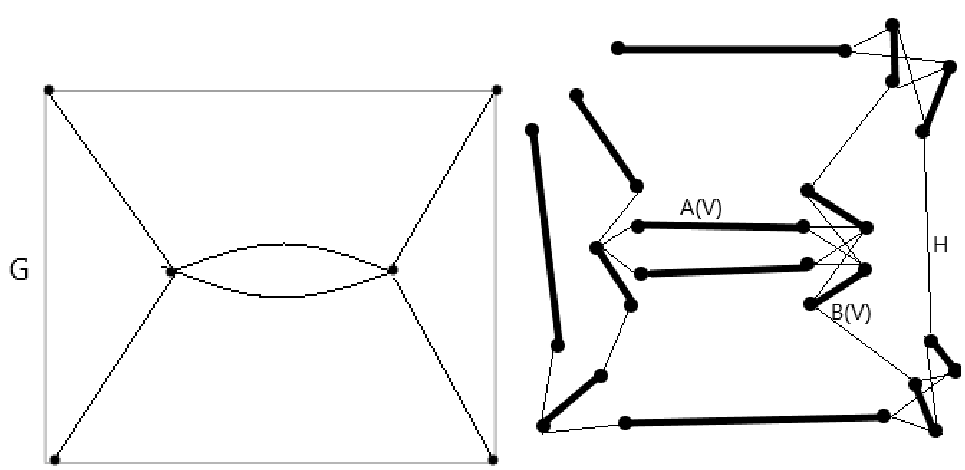
3.2.9. Definition. Given a function f: V(G) → NU{0}, an f-**factor** of a graph G is a subgraph H such that da(v)= f(v) for all v €V(G)

Tutte 1962) proved a necessary and sufficient condition for a graph G to have an f-factor . He later reduced the problom to checking for a 1-tactor in a related simple graph. We describe this reduction; it is a beautiful exampie Ot ranstorming a graph problem into a previously solved problem.

3.2.10 Example. A graph transformation (Tutte [1954a)). We assume that f(w)≤d(w) for all w; otherwise G has too few edges at w to have an f-factor We then construct a graph H that has a 1-factor if and only if G has an f-factor Let e(w)= d(w)-f(w); this is the exces degree at w and is nonnegative.

To construct H, replace each vertex with a biclique $k_{d(v),e(v)}$ having partite sets A(v) of size d(v) and B(v) of size e(v). For each vw € E(G), add an edge JOining One vertex of A(v) to one vertex of A (w). Each vertex of A(v) participatesin one such edge.

The figure below shows a graph G, vertex labels given by f, and the resulting simple graph H. The bold edges in H form a 1-factor that correspondsto an f-tfactor of G. In this example, the f-factor is not unique.



3.2.11. Theorem. A graph G has an f-factor if and only if the graph H constructed from G and f has a 1-factor.

Proof: Necessity. If G has an f-factor, then the corresponding edges in H leave e(v) vertices of A(v) unmatched; match them arbitrarily to the vertices of B(v) to obtain 1-factor of H.

Sufficiency. From a 1-factor of H, deleting B(v) and the vertices of A(v) matched into B(v) leaves f(v) edges at v. Doing this for each v and merging the remaining f(v) vertices of each A(v) yields a subgraph of G with degree f(v)at v. It is an f-factor of G.

Tutte's Condition for a 1-factor transforms into a necessary and sufficient condition for an f-factor in G. Among the applications is a proof of the Erdős-Gallai [1960] characterization of degree sequences of simple graphs

Given an algorithm to find a 1-factor, the correspondence in provides an algorithmic test for an f-factor. Instead of just seeking a 1-factor (that is, a perfect matching), we next consider the more general problem of finding a maximum matching in a graph.

APPLICATIONS

. Street Sweeping and the Transportation Problem. A cleaning machine sweeping a curb must move in the same direction as traffic. This yields a digraph; a two-way street generates two oppositely directed edges, while a one-way street generates two edges in the same direction. We consider a simple version of the Street Sweeping Problem, discussed in more detail in Roberts (1978) as based on Tucker-Bodin [1976).

In New York City, parking is prohibited from some curbs each day to allow for street sweeping. For each day, this defines a sweep subgraph G of the full digraph H of curbs, consisting of those available for sweeping. Each e € E(H) has a deadheading time t(e) needed to travel it without sweeping.

The question is how to sweep G while minimizing the total deadheading time spent without sweeping. This is a generalization of a directed version of the Chinese Postman Problem. If indegree equals outdegree at each vertex of G, then no deadheading is needed. Otherwise, we duplicate edges of G or add edges from H to obtain an Eulerian digraph G' containing G.

Let X be the set of vertices with excess indegree; let σ(x) = d_G^-(x) -d_G^-(x) fer x ϵ X. Let Y be the set with excess outdegree; let ∂(y)= d_G^- (y) -d_G^+(y) for yϵ Y. Note that $\Sigma_{x\epsilon X}\sigma(x)=\Sigma_{y\epsilon Y}\partial(y)$. To obtain G' from G, we must add σ(x) edges with tails at x € X and ∂(y) edges with heads at y € Y. Since G needs net outdegree 0 at each vertex, the additions form paths from X to Y. The cost c(xy) of an x,y-path is the distance from x to y in the weighted digraph H which can be found by Dijkstra's Algorithm.

This yields the **Transportation Problem**. Given supply σ(x) for xϵ X, detnand ∂(y) for y ϵ Y, cost c(xy) per unit sent from x to y, andΣσ (x) = Σ∂(y), we want to satisfy the demands at least total cost. A version of the problem was introduced by Kantorovich [1939]; the form above arose (with a constructive solution) in Hitcheock (1941] (see also Koopmans [1947]). The problem is discussed at length in Ford-Fulkerson [1962, p93-130].

When the supplies and demands are rational, the Assignment Problem can be applied. First scale up to obtain integer supplies and demands. Next define a matrix with Σ σ(x) rows and columns. For each x ϵ X, create σ(x) rows. For each y ϵ Y, create ∂(y) columns. When row i and column j represent x and y . let $w_{i,j}$ = M –c(xy) where M = $\max_{x,y}$.c(xy). A maximum weight matching now yields a minimum cost solution to the Transportation Problem

Graph matching has applications in *flow networks*, *scheduling and planning, modeling bonds in chemistry*, *graph coloring*, *the stable marriage problem*, neural networks in artificial intelligence and more.

Conclusion

Project was done on Matchings and Factors. In this project, we have seen how matchings and factors is related to filling jobs with applicants , how a perfect matching is done in many fields , how it is applicable in street sweeping and the transportation problem.

We conclude from the project that Matchings and Factors is an important and useful concept in graph theory. A Matching in a graph is a set of non-loop edges with no shared endpoints. A Factor of a graph G is a spanning subgraph of G.

REFERENCES

- *AN INTRODUCTION TO GRAPH THEORY* –K BALAKRISH & R RANGANATH
- *BASIC GRAPH THEORY* –JOHN CLARK &DEREK ALLAN HOLTON
- *INTRODUCTION TO GRAPH THEORY* ;SECOND EDITION -DOUGLAS B. WEST