# Chapter 7
# Matchings and r-Factors

### Section 7.0 Introduction

Suppose you have your own company and you have several job openings to fill. Further, suppose you have several candidates to fill these jobs and you must somehow decide which candidates are to fill which jobs. Let's try to model this problem using graphs. The most natural model takes the form of a bipartite graph. Suppose that corresponding to each of *m* open jobs we associate a vertex and say we call these vertices $j_1, j_2, \ldots, j_m$. Also, corresponding to each of *n* job applicants we associate a vertex, say $a_1, \ldots, a_n$. Now, we join vertex $a_i$ to vertex $j_k$ if, and only if, applicant $a_i$ is qualified for job $j_k$. We clearly have created a bipartite graph. A solution to our hiring dilemma is to find a set of edges that "match" each job to some distinct applicant. Clearly, our problem would make even more sense if we were to somehow rate the applicants and their "suitability" to handle each job. That is, we associate a measure of suitability (or unsuitability) with each edge in our model. An optimal solution, then, would be to find a set of job assignments that maximizes (or minimizes) the sum of these measures (generally called weights). We will consider this enhancement later. For now, we will be satisfied with merely finding suitable pairings.

We shall begin with a detailed investigation of such pairings in bipartite graphs. Our goal is to find an effective method of determining the best possible pairing, whether it be in terms of most edges used or in terms of optimizing some weight function. We shall investigate both theoretic and algorithmic approaches. We shall ultimately see that this area is a meeting point for many different ideas in discrete mathematics. This will provide us with a chance to use diverse techniques and apply our results in many interesting and unusual ways.

### Section 7.1 Matchings and Bipartite Graphs

More formally, two distinct edges are *independent* if they are not adjacent. A set of pairwise independent edges is called a *matching.* Thus, to solve our job assignment problem, we seek a matching with the property that each job $j_i$ is incident to an edge of the matching. In most situations, it is not merely a matching that we want, but the largest possible matching with respect to some measurable quantity. Here, we wish the maximum number of jobs to be filled, but in other situations there may be better ways to measure how successfully we have formed our matching. In $G$, a matching of maximum cardinality is called a *maximum matching* and its cardinality is denoted $\beta_1(G)$. A matching that pairs all the vertices in a graph is called a *perfect matching.*

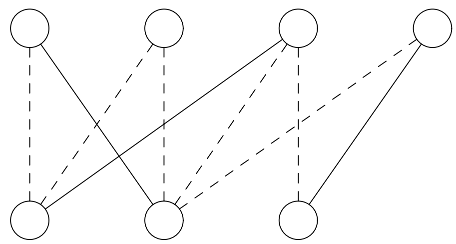


**Figure 7.1.1.** The solid edges form a maximum matching.

In a study of matchings, several useful observations will actually take us a long way toward our goal. Berge [2] made perhaps the most applicable of these observations. Following his terminology, we define an edge to be *weak with respect to a matching M* if it is not in the matching. A vertex is said to be *weak with respect to M* if it is only incident to weak edges. An *M-alternating path* in a graph $G$ is a path whose edges are alternately in a matching $M$ and not in $M$ (or conversely). An *M-augmenting path* is an alternating path whose end vertices are both weak with respect to $M$. Thus, an $M$-augmenting path both begins and ends with a weak edge. If it is clear what matching we are using, we will simply say alternating path or augmenting path. The graph of Figure 7.1.2 contains a matching $M$ with edges 23, 54 and 78. An augmenting path containing these edges is shown with nonmatching edges dashed. With this terminology in mind, we will find the following lemma extremely useful.

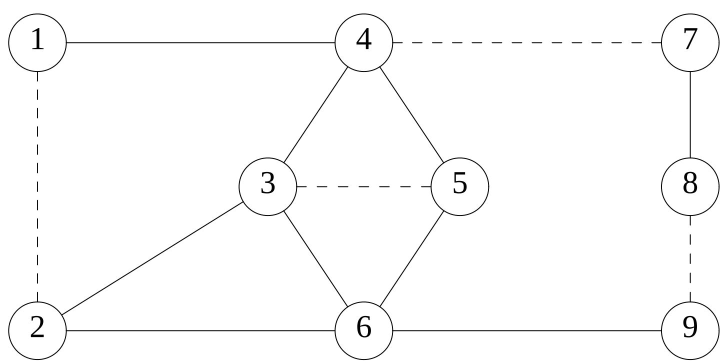



**Figure 7.1.2.** Augmenting path 1, 2, 3, 5, 4, 7, 8, 9 for $M$

**Lemma 7.1.1** Let $M_1$ and $M_2$ be two matchings in a graph $G$. Then each component of the spanning subgraph $H$ with edge set

$$E(H) = (M_1 - M_2) \cup (M_2 - M_1)$$

is one of the following types:

1. An isolated vertex.
2. An even cycle with edges alternately in $M_1$ and $M_2$.
3. A path whose edges are alternately in $M_1$ and $M_2$ and such that each end vertex of the path is weak with respect to exactly one of $M_1$ and $M_2$.

**Proof.** It is easily seen that $\Delta(H) \leq 2$, since no vertex can be adjacent to more than one edge from each matching. Thus, the possible components are paths, cycles or isolated vertices.

Now, consider a component in $H$ that is not an isolated vertex. It is easily seen that in any such component, the edges must alternate, or the definition of matching would be violated. Hence, if the component is a cycle, it must be even and alternating. Finally, assume the component is a path. Then, we must only show that each end vertex is weak with respect to exactly one of the matchings. Clearly, each end vertex is already adjacent to an edge of one of the matchings. Suppose it was adjacent to an edge $e$ from the other matching, without loss of generality, say $e \in M_1 - M_2$. Since we can now extend the path in question, we violate the fact that our vertex was an end vertex of a path and that this path was a component of H. Thus, we must have one of the three possibilities listed above. □

We now present Berge's [2] characterization of maximum matchings.

**Theorem 7.1.1** A matching $M$ in a graph $G$ is a maximum matching if, and only if, there exists no $M$-augmenting path in $G$.

**Proof.** Let $M$ be a matching in $G$ and suppose that $G$ contains an $M$-augmenting path

$$P\colon v_0, v_1, \ldots, v_k,$$

where $k$ is clearly odd. If $M_1$ is defined to be

$$\begin{aligned} M_1 = (M - \{ v_1 v_2, v_3 v_4, \ldots, v_{k-2} v_{k-1} \}) \\ \cup \{ v_0 v_1, v_2 v_3, \ldots, v_{k-1} v_k \}, \end{aligned}$$

then $M_1$ is a matching in $G$, and it contains one more edge than $M$; thus, $M$ is not a maximum matching.

Conversely, suppose that $M$ is not a maximum matching and there does not exist an $M$-augmenting path and let $M_1$ be a maximum matching in $G$. Now, consider the spanning subgraph $H$, where $E(H)$ is the symmetric difference of $M$ and $M_1$ (that is, $(M - M_1) \cup (M_1 - M)$). By Lemma 7.1.1, we know the possibilities for the components of $H$. By our earlier observations, we know that some alternating path in $H$ must contain more edges of $M_1$ than $M$, since $M_1$ contains more edges than $M$. But, then, this path must be an $M$-augmenting path in $G$, contradicting our assumptions that there were no augmenting paths in $G$. □

The situation presented in the job assignment problem is very common. One often wishes to find a matching that uses every vertex in some set. Given a matching $M$, we will say that a set $S$ *is matched under* $M$ if every vertex of $S$ is incident to an edge in $M$. For bipartite graphs, Hall [11] first determined necessary and sufficient conditions under which a set could be matched.

**Theorem 7.1.2** Let $G = (X \cup Y, E)$ be a bipartite graph. Then $X$ can be matched to a subset of $Y$ if, and only if, $|N(S)| \geq |S|$ for all subsets $S$ of $X$.

**Proof.** Suppose that $X$ can be matched to a subset of $Y$. Then, since each vertex of $X$ is matched to a distinct vertex of $Y$, it is clear that $|N(S)| \geq |S|$ for every subset $S$ of $X$.

Conversely, suppose that $G$ is bipartite and that $X$ cannot be matched to a subset of $Y$. We wish to construct a contradiction to the assumed neighborhood conditions. Thus, consider a maximum matching $M$ in $G$. By our assumptions, the edges of $M$ are not incident with all the vertices of $X$. Let $u$ be a vertex that is weak with respect to $M$ and let $A$ denote the set of all vertices of $G$ connected to $u$ by an $M$-alternating path. Since $M$ is a maximum matching, it follows from Berge's theorem (Theorem 7.1.1) that $u$ is the only weak vertex of A. Let $S = A \cap X$ and $T = A \cap Y$.

Clearly, the vertices of $S - \{u\}$ are matched with vertices of $T$; therefore, $|T| = |S| - 1$ and $T \subseteq N(S)$. In fact, $T$ must equal $N(S)$ since every vertex in $N(S)$ is connected to $u$ by an alternating path. But then $|N(S)| = |S| - 1 < |S|$, contradicting our neighborhood assumption. □

An easy and well-known corollary to Hall's theorem can now be presented.

**Corollary 7.1.1** If $G$ is a $k$-regular bipartite graph with $k > 0$, then $G$ has a perfect matching.

**Proof.** Let $G = (X \cup Y, E)$ be a $k$-regular bipartite graph. Then $k\,|X| = k\,|Y| = |E|$, and since $k > 0$, we see that $|X| = |Y|$. For any $A \subseteq V(G)$, let $E_A$ be the set of edges of $G$ incident with a vertex of $A$. Let $S \subseteq X$ and consider $E_S$ and $E_{N(S)}$. By the definition of $N(S)$, we see that $E_S \subseteq E_{N(S)}$. Thus,

$$k\,|N(S)| = |E_{N(S)}| \geq |E_S| = k\,|S|,$$

and so $|N(S)| \geq |S|$. Thus, by Hall's theorem, $X$ can be matched to a subset of $Y$. But since $|X| = |Y|$, we see that $G$ must contain a perfect matching. □

Hall's theorem is a very flexible and useful result. It can be seen from many different points of view, and it can be stated in many ways. We shall now state it in set theoretic terms. To do this, we need some terminology. Given sets $S_1, \ldots, S_k$, we say any element $x_i \in S_i$ is a *representative* for the set $S_i$ which contains it.

Our purpose is to find a collection of distinct representatives for the sets $S_1, \ldots, S_k$. This collection is usually known as a *system of distinct representatives* or a *transversal* of the sets. From a graph point of view, we could use a vertex $s_i$ to represent each set $S_i$. We could also use a distinct vertex $u_j$ to represent each of the elements $x_j$ in each of the sets. We then join vertices $s_i$ and $u_j$ if, and only if, the element $x_j$ is in the set $S_i$. In this way, we see that $N(s_i) = \{ u_j \mid x_j \in S_i \}$. It is now easy to see that finding a system of distinct representatives is equivalent to finding a matching of the $s_i$'s into a subset of the $u_j$'s. We now restate Hall's theorem in set terms.

**The SDR Theorem** A collection $S_1, S_2, \ldots, S_k$, $k \geq 1$ of finite nonempty sets has a system of distinct representatives if, and only if, the union of any $t$ of these sets contains at least $t$ elements for each $t$, $(1 \leq t \leq k)$.

Another popular version of Hall's theorem takes the form of a statement on marriage. Our goal this time is to match as many men to women as possible so that the maximum number of couples can be married. This matching of men to women is the reason Hall's theorem is often called the marriage theorem.

**The Marriage Theorem** Given a set of $n$ men and a set of $n$ women, let each man make a list of the women he is willing to marry. Then each man can be married to a woman on his list if, and only if, for every value of $k$ $(1 \leq k \leq n)$, the union of any $k$ of the lists contain at least $k$ names.

We now consider a related result from König [12] and Egerváry [6]. A set $C$ of vertices is said to *cover* the edges of a graph $G$ (or be an *edge cover*), if every edge in $G$ is incident to a vertex in $C$. The minimum cardinality of an edge cover in $G$ is denoted $\alpha(G)$. In Figure 7.1.3 we see a bipartite graph with a matching (dashed edges). The solid vertices form a cover in this graph.

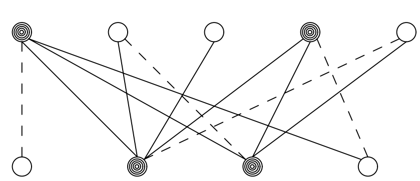


**Figure 7.1.3.** A matching and cover in a graph.

The König-Egerváry theorem relates matchings and covers. The proof technique is reminiscent of those already seen in this section.

**Theorem 7.1.3** If $G = (X \cup Y, E)$ is a bipartite graph, then the maximum number of edges in a matching in $G$ equals the minimum number of vertices in a cover for $E(G)$, that is, $\beta_1(G) = \alpha(G)$.

**Proof.** Let a maximum matching in $G$ contain $\beta_1(G) = m$ edges and let a minimum cover for $E(G)$ contain $\alpha(G) = c$ vertices. Note that $c \geq m$ always holds.

Let $M$ be a maximum matching in $G$. Also let $W$ be those vertices of $X$ that are weak with respect to $M$. Note that $|M| = |X| - |W|$. Let $S$ be those vertices of $G$ that are connected to some vertex in $W$ by an alternating path. Define $S_X = S \cap X$ and $S_Y = S \cap Y$.

From the definition of $S$ and the fact that no vertex of $S_X - W$ is weak, we see that $S_X - W$ is matched under $M$ to $S_Y$ and that $N(S_X) = S_Y$. Since $S_X - W$ is matched to $S_Y$, we see that $|S_X| - |S_Y| = |W|$.

Let $C = (X - S_X) \cup S_Y$. Then $C$ is a cover for $E(G)$, for if it were not, there would be an edge $vw$ in $G$ such that $v \in S_X$ and $w \notin S_Y = N(S_X)$. Hence,

$$|C| = |X| - |S_X| + |S_Y| = |X| - |W| = |M|.$$

Thus, $c = m$, and the proof is complete. □

The form of the König-Egerváry theorem should by now be a tipoff that something deeper is going on here. The min-max form that we saw in Menger's theorem and in the max flow-min cut theorem is once again present. Thus, we should expect that these results are closely related (see the exercises) and that flows could be used to prove results about matchings. We now investigate this connection.

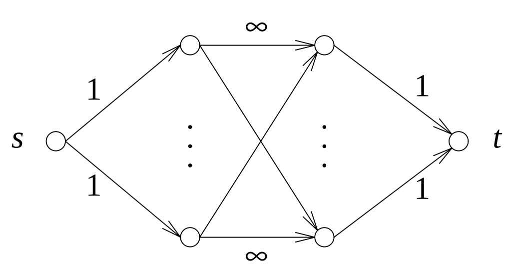



**Figure 7.1.4.** The network $N_G$.

Given a bipartite graph $G = (X \cup Y, E)$, we construct a network $N_G$ (see Figure 7.1.4) corresponding to $G$ by first orienting all edges of $G$ from $X$ to $Y$. Now, insert a source vertex $s$ with arcs to all vertices of $X$ and a sink vertex $t$ with arcs from all vertices of $Y$. We assign the capacity of all arcs out of $s$ or into $t$ as 1. The capacities of all arcs from $X$ to $Y$ are set to $\infty$. With this network in mind, we are now able to show a connection between matchings and flows.

**Theorem 7.1.4** In a bipartite graph $G = (X \cup Y, E)$, the number of edges in a maximum matching equals the maximum flow in the network $N_G$.

**Proof.** Let $M$ be a maximum matching in $G$. For each edge $xy$ in $M$, we use the directed path $s, x, y, t$ to flow 1 unit from $s$ to $t$ in $N_G$. It is clear that these paths are all disjoint except for $s$ and $t$. Thus, $F \geq |M| = \beta_1(G)$.

Now let $f$ be an integral flow function on the network $N_G$ corresponding to $G$. All the directed paths between $s$ and $t$ have the form $s, x, y, t$. If such a path is used to carry flow from $s$ to $t$, then no other arc can be used to carry flow to $y$. Also, no other arc can be used to carry flow out of $x$. Then the set of edges $xy$ for which $f(x \to y) = 1$ determines a matching in $G$. Thus, $\beta_1(G) = |M| \geq F$, and this, combined with our previous observations, shows that $\beta_1(G) = |M| = F$. □

It is a simple matter now to deduce from the max-flow min-cut theorem and Theorem 7.1.3 that $\alpha(G)$ must equal the capacity of a minimum cut. But we can do more than just state this equality; we can use cuts to determine the cover. Suppose that $(C, \overline{C})$ is a cut of minimum capacity in $N_G$. If we let $A = X \cap \overline{C}$ and $B = Y \cap C$, then it is easy to see that $A \cup B$ is a cover for $G$. Further, since

$$c(s, A) + c(B, t) = |A \cup B|$$

(that is, since the capacity of the arcs from $s$ to $A$ and those from $B$ to $t$ total $|A \cup B|$), we see that $A \cup B$ must be a minimum cover. Thus, we can use flows and cuts to find not only maximum matchings but also minimum covers as well. Does all this remind you of the way we selected the cover in the proof of Theorem 7.1.3?

## Section 7.2 Matching Algorithms and Marriage

It turns out that in the setting of bipartite graphs, we can easily apply Berge's ideas and use augmenting paths to build maximum matchings. We now present a labeling algorithm to find a maximum matching in a bipartite graph. This algorithm assumes a given matching $M$ is known and attempts to extend $M$ by finding an augmenting path. This is done by trying to follow all possible augmenting paths. As we go, we "mark" vertices using edges not in $M$ while walking from $X$ to $Y$, and we mark vertices using edges in $M$ while walking from $Y$ to $X$. Hence, we essentially trace all possible alternating paths as we go. This algorithm is a special case of the network flow algorithm of Ford and Fulkerson [9].

**Algorithm 7.2.1 A Maximum Matching in a Bipartite Graph.**

**Input:** Let $G = (X \cup Y, E)$ be a bipartite graph and suppose that $X = \{ x_1, \ldots, x_m \}$ and $Y = \{ y_1, \ldots, y_n \}$. Further, let $M$ be any matching in $G$ (including the empty matching).

**Output:** A matching larger than $M$ or the information that the present matching is maximum.

**Method:** We now execute the following labeling steps until no step can be applied.

1. Label with an * all vertices of $X$ that are weak with respect to $M$. Now, alternately apply steps 2 and 3 until no further labeling is possible.
2. Select a newly labeled vertex in $X$, say $x_i$, and label with $x_i$ all unlabeled vertices of $Y$ that are joined to $x_i$ by an edge weak with respect to $M$. Repeat this step on all vertices of $X$ that were labeled in the previous step.
3. Select a newly labeled vertex of $Y$, say $y_j$, and label with $y_j$ all unlabeled vertices of $X$ which are joined to $y_j$ by an edge in $M$. Repeat this process on all vertices of $Y$ labeled in the previous step.

Notice that the labelings will continue to alternate until one of two possibilities occurs:

$E1$: A weak vertex in $Y$ has been labeled.

$E2$: It is not possible to label any more vertices and $E1$ has not occurred.

If ending $E1$ occurs, we have succeeded in finding an $M$-augmenting path, and we can construct this path by working backwards through the labels until we find the vertex of $X$ which is labeled *. The purpose of the labels is to allow us to actually determine an $M$-augmenting path. We can then extend our matching as in Theorem 7.1.1 and repeat the algorithm on our new matching. Our next theorem shows that if $E2$ occurs, $M$ is already a maximum matching. The proof is reminiscent of that of the König-Egerváry theorem.

**Theorem 7.2.1** Suppose that Algorithm 7.2.1 has halted with ending $E2$ occurring and having constructed matching $M$. Let $U_X$ be the unlabeled vertices in $X$ and $L_Y$ the labeled vertices in $Y$. Then $C = U_X \cup L_Y$ covers the edges of $G$, $|C| = |M|$, and $M$ is a maximum matching in $G$.

**Proof.** Suppose that $C$ does not cover the edges of $G$. Then there must exist an edge from $L_X = X - U_X$ to $U_Y = Y - L_Y$. Suppose there was such an edge, call it $e = xy$, where $x \in L_X$ and $y \in U_Y$. If $e$ is not in $M$, then since $x$ is labeled, it follows from step 2 that $y$ is labeled, and this condition contradicts the fact that $L_Y$ contains all the labeled vertices of $Y$. Thus, $e \in M$, and so it follows from step 3 that the label on $x$ is y. It also follows from the algorithm that $y$ must be labeled; and that in fact, it must have received that label prior to $x$ receiving its label. But this condition again contradicts the fact that

$L_Y$ contains all the labeled vertices of $Y$. Thus, we conclude there are no edges from $X - U_X$ to $Y - L_Y$, and so it must be the case that $C$ covers all the edges of $G$.

Now, consider $y \in L_Y$. Since $y$ is labeled and $E1$ has not happened, $y$ must be incident with an edge of $M$ (exactly one such edge since $M$ is a matching). Suppose that $xy$ is this edge. By step 3, the vertex $x$ must be labeled, so $x$ is not in $U_X$. Consider some $x_1 \in U_X$. Since $x_1$ is not labeled, it must be incident with an edge of $M$, or it would have received the label * in step 1. Since $M$ is a matching, $x_1$ is incident with exactly one edge of $M$. Let this edge be $x_1 y_1$. If $y_1$ were labeled, by step 3 we would see that $x_1$ would also be labeled, but $x_1 \in U_X$. Then $y_1$ must be unlabeled, and thus, none of the edges of $M$ which are incident to vertices in $U_X$ are the same as any of the edges of $M$ with incidences in $L_Y$. Since every edge of $M$ has an end vertex in either $U_X$ or $L_Y$, there must be as many edges in $M$ as vertices in $C$; that is, $|C| = |M|$. Since $C$ covers the edges of $G$, by Theorem 7.1.3, $M$ must be a maximum matching, and so the proof is complete. □

**Example 7.2.1.** We now apply Algorithm 7.2.1 to the bipartite graph of Figure 7.2.1.

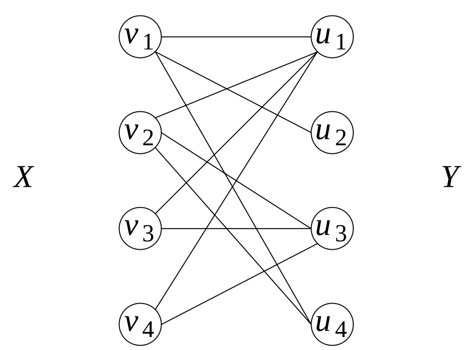



**Figure 7.2.1.** A bipartite graph $G = (X \cup Y, E)$.

We select the edge $v_1 u_1$ as our initial matching $M$. We now apply Algorithm 7.2.1.
Step 1: Label $v_2, v_3, v_4$ with *.
Step 2: Select $v_2$ and label $u_1, u_3, u_4$ with $v_2$.
Step 3: Select $u_1$ and label $v_1$ with $u_1$.
Step 2: Select $v_1$ and label $u_2$ with $v_1$.
Note that no other labeling is possible.

Since the labeling included weak vertices in $Y$, condition $E1$ holds. Note that the path $P: v_2, u_3$ is augmenting; thus our new matching is now $M = \{v_1 u_1, v_2 u_3\}$, and we repeat Algorithm 7.2.1 on this $M$.

To see what this algorithm is really doing, we can trace what happens in each pass of the algorithm. We began labeling $v_2$ and followed by labeling the weak neighbors $u_1, u_3$ and $u_4$. From these vertices we looked instead for edges in the matching $M$ and labeled $v_1$. Then, we again reversed our thinking and looked for weak neighbors of this vertex. In Figure 7.2.2 we picture the situation after the labeling was completed. Note the layering of vertices and the fact that edges between consecutive layers were introduced in the same step of the algorithm. The tree that has been "grown" by the algorithm has the property that each path from the initial vertex to a leaf is an alternating path. When the tree has been grown to its utmost, the algorithm halts, and any path from the root (the vertex labeled *) to a weak leaf is augmenting. We retrace any such path by following the labels assigned to the vertices. It has become customary to call such a tree a *hungarian tree*. Note that such a tree has been started for every vertex labeled *, but not all have been successful in finding an augmenting path.

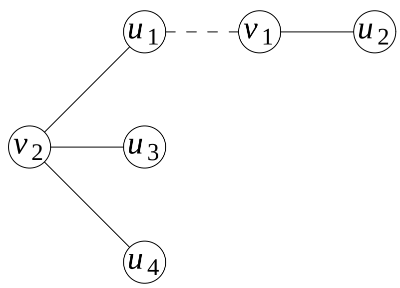



**Figure 7.2.2.** A hungarian tree grown in pass one.

The second pass of Algorithm 7.2.1 (see Figure 7.2.3) produces:
Step 1: Label $v_3$, $v_4$ with *.
Step 2: Select $v_4$ and label $u_1$, $u_3$ with $v_4$.
Select $v_3$; nothing more can be labeled.
Step 3: Select $u_1$ and label $v_1$ with $u_1$. Select $u_3$ and label $v_2$ with $u_3$.
Step 2: Select $v_1$ and label $u_2$, $u_4$ with $v_1$.

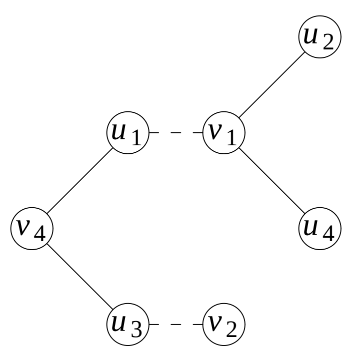



**Figure 7.2.3.** The hungarian tree rooted at $v_4$ in pass two.

The labeling halts, and we have found an augmenting path, namely $P$: $u_4$, $v_1$, $u_1$, $v_4$. Using $P$, we extend the matching to

$$M = \{ v_2u_3, u_4v_1, u_1v_4 \},$$

and we repeat Algorithm 7.2.1 again on $M$.

Step 1: Label $v_3$ with *.
Step 2: Select $v_3$ and label $u_1$, $u_3$ with $v_3$.
Step 3: Select $u_1$ and label $v_4$ with $u_1$;
then select $u_3$ and label $v_2$ with $u_3$.
Step 2: Select $v_2$ and label $u_4$ with $v_2$;
then select $v_4$ and note that no further labeling can be done.
Step 3: Select $u_4$ and label $v_1$ with $u_4$.
Step 2: Select $v_1$ and label $u_2$ with $v_1$.
Step 3: No labeling is possible, and the algorithm halts.

We now interchange edges on the path $P$: $u_2, v_1, u_4, v_2, u_3, v_3$ to obtain the maximum matching $M = \{ u_1v_4, v_3u_3, v_2u_4, v_1u_2 \}$ (see Figure 7.2.4).

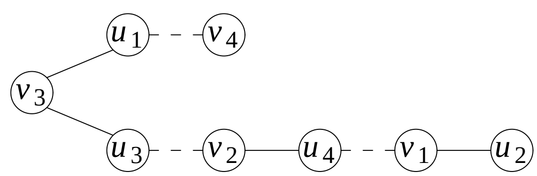



**Figure 7.2.4.** The hungarian tree from pass three.

We are now ready to consider a strengthening of the job assignment problem. We wish to include information about the relative suitabilities of the job candidates for the various jobs. The problem to be considered will be restricted to the case in which there are $n$ candidates for $n$ jobs, and each candidate has a measure of suitability for each job. That is, we have assigned a weight function to the edges of $K_{n,n}$. It is clear that it may not be possible to assign each applicant to the job he or she is best suited for, since two applicants might be best suited for the same job. Thus, our goal is to find the overall best solution, that is, the solution with the optimal sum of the weights assigned to the edges of the matching.

To attack this problem, we will find it more convenient to have our weight function represent a measure of the applicant's unsuitability for the job. Then, the larger the weight, the more unsuitable the applicant is for the job. For any matching $M$, we define the *weight of the matching* to be $W(M) = \sum_{e \in M} w(e)$. Thus, an optimal solution will be a perfect matching with $W(M)$ a minimum. We now present an algorithm for finding such a solution.

We begin by representing our graph in matrix form, $U = [\ w_{i,k}\ ]$ where $w_{i,k}$ is the weight of the edge joining $j_i$ and $a_k$ (that is, the unsuitability of applicant $k$ to job $i$). An example of such a matrix is now given.

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | 4 | 6 | 14 | 11 |
| $j_2$ | 7 | 2 | 8 | 9 |
| $j_3$ | 3 | 13 | 1 | 4 |
| $j_4$ | 5 | 2 | 0 | 13 |

It is important to note that our solution is unchanged if we subtract the same number from all members of some row or some column. This follows since only one entry will be selected from any row or column; hence, the value of $W(M)$ for any matching $M$ will be reduced by the same amount. Thus, we can make the entries in our unsuitability matrix easier to deal with by first subtracting from each row the minimum entry in that row. The resulting matrix still has all nonnegative entries, which we hope are smaller than before. Our example matrix thus becomes:

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | 0 | 2 | 10 | 7 |
| $j_2$ | 5 | 0 | 6 | 7 |
| $j_3$ | 2 | 12 | 0 | 3 |
| $j_4$ | 5 | 2 | 0 | 13 |

Now, subtract from each column the smallest entry in that column to obtain a further reduced unsuitability matrix. Our example is then:

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | $0^*$ | 2 | 10 | 4 |
| $j_2$ | 5 | $0^*$ | 6 | 4 |
| $j_3$ | 2 | 12 | 0 | $0^*$ |
| $j_4$ | 5 | 2 | $0^*$ | 10 |

Our problem now is to select numbers from the table, no two in the same row or column, with as small a sum as possible. Since our entries are all nonnegative, the smallest sum we could hope for is zero. Thus, if $n$ zeros can be found, no two in the same row or column, an optimal solution will be obtained. In our example, a solution is easily found. We select the entries starred above.

The suspicious reader is now asking what happens if at this stage we cannot find a suitable set of $n$ "independent zeros," or must this always be the case. The answer is that we are not always sure of having enough zeros at this stage to represent a perfect matching in our graph. Sometimes, further adjustments must be made. Consider the following unsuitability matrix.

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | 6 | 8 | 2 | 7 |
| $j_2$ | 5 | 8 | 13 | 9 |
| $j_3$ | 2 | 8 | 10 | 9 |
| $j_4$ | 4 | 12 | 8 | 11 |

Then, after reducing the rows followed by the columns, we are left with the following matrix.

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | 4 | 3 | 0 | 1 |
| $j_2$ | 0 | 0 | 8 | 0 |
| $j_3$ | 0 | 3 | 8 | 3 |
| $j_4$ | 0 | 5 | 4 | 3 |

This matrix does not contain four independent zeros since all the zeroes are contained in the first column and the first two rows. This can be seen by crossing with a line the rows and columns containing zeros. In the graph, then, the independent zeros represent the edges of the matching, while the lines drawn show the vertices "covered" by the vertex corresponding to the row or column in which the line was drawn. Our adjustment procedure is as follows:

1. Let $m$ be the smallest number that is not included in any of our crossed rows or columns.
2. Subtract $m$ from all uncrossed numbers.
3. Leave numbers which are crossed once unchanged.
4. Add $m$ to all numbers which are crossed twice.

This procedure produces at least one more zero in the uncrossed portion of our matrix and leaves all the zeros unchanged, unless they happen to be crossed twice. Can you explain why this adjustment procedure works? Our example becomes:

| U | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| $j_1$ | 7 | 3 | 0 | 1 |
| $j_2$ | 3 | 0 | 8 | 0 |
| $j_3$ | 0 | 0 | 5 | 0 |
| $j_4$ | 0 | 2 | 1 | 0 |

The procedure described here will always yield a set of $n$ independent zeros after a finite number of repetitions. The algorithm presented above to solve our optimal matching problem is usually known as the hungarian algorithm, in honor of König and Egerváry. An alternate form of the König-Egerváry theorem can now be stated. We derive its proof from the first form.

**Theorem 7.2.2** Let $S$ be any $m \times n$ matrix. The maximum number of independent zeros which can be found in $S$ is equal to the minimum number of lines (either rows or columns) which together cover all the zeros of $S$.

**Proof.** Construct a bipartite graph $G = (X \cup Y, E)$ modeling our matrix as follows. Let the vertices of $X$ correspond to the rows of our matrix and the vertices of $Y$ to the columns. We join $x_i$ and $y_j$ if, and only if, entry $i, j$ of our matrix is zero. Then, a maximum independent set of zeros corresponds to a maximum matching of $G$, and a minimum set of lines covering all the zeros corresponds to a minimum covering of $G$. Thus, by Theorem 7.1.3 the result follows. □

Suppose we now consider the job assignment problem from the greedy point of view. Can we simply begin with the edge of minimum cost and somehow extend to a matching of minimum cost? The answer is that we can if we are careful about our point of view. Rather than build a matching by greedily taking edges, we shall set our view on the vertices involved. Given a bipartite graph $G = (V_1 \cup V_2, E)$, we say a subset $I$ of $V_1$ is *matching-independent* for matchings of $V_1$ into $V_2$ if there is a matching which matches all the elements of $I$ to elements of $V_2$. We wish to build a maximum sized matching-independent set in a greedy fashion. Thus, if we have a set $I$ that is matching-independent, we would add to $I$ the vertex $x$ of $V_1$ having cheapest incident edge that still allows us to match $I \cup \{ x \}$ into $V_2$. When we can no longer do this, we stop. The question of interest now is: How do we know that we have formed a maximum sized matching-independent set when this process halts? That this is indeed the case can be concluded from the next result.

**Theorem 7.2.3** Matching-independent sets for matchings of $V_1$ into $V_2$ satisfy the following rule:

If $I$ and $J$ are matching-independent subsets of $V_1$ and $|I| < |J|$, then there is an element $x$ of $J$ such that $I \cup \{ x \}$ is matching-independent.

**Proof.** Suppose that $M_1$ is a matching of $I$ into $V_2$ and $M_2$ is a matching of $J$ into $V_2$. Then, by Lemma 7.1.1, the spanning subgraph $H$ with $E(H) = (M_1 - M_2) \cup (M_2 - M_1)$ has connected components of only three possible types. Since $|M_2| > |M_1|$, at least one of these components must be of type 3 in Lemma 7.1.1. Thus, there is a path $P$ whose edges are alternately in $M_2$ and $M_1$ and whose first and last edges are in $M_2$. Each vertex of $P$ incident to an edge from $M_1$ is also incident to an edge from $M_2$. Further, there is a vertex $x$ in $V_1$ (and $J$) incident to an edge from $M_2$, and $x$ is not incident to any edge from $M_1$. Now the set of edges

$$M = (M_1 - E(P)) \cup (E(P) - M_1)$$

forms a matching with one more edge than $M_1$. Also, $M$ is a matching of $I \cup \{ x \}$ into $V_2$. Thus, $I \cup \{ x \}$ is matching-independent, and since $x \in J$, the proof is complete.

□

To actually use the greedy approach to construct a maximum sized matching of minimum weight, we must determine a method that allows us to select the vertex $x$ we wish to add to our matching-independent set. To do this, we must also keep track of the edges that are presently matching $I$ into $V_2$. Otherwise, we would face the possibility of having to check all possible subsets of $V_2$ in a search for the matching. Since this is clearly an exponential process, the bookkeeping of the intermediate matchings is necessary. Applying our methods of finding alternating paths allows us to construct the maximum sized matching-independent set, and it is an exercise to show that the corresponding matching is of minimum cost.

Can we vary the assignment problem somewhat? For a suitability weight function $w$, we can change the function we are optimizing from

$$\sum_{e \in M} w(e) \text{ to } \min_{e \in M} \{ w(e) \}.$$

That is, suppose we try to maximize the minimum weight of an edge in the matching. This is the mathematical version of the old proverb that the strength of a chain equals the strength of its weakest link. This is known as the *bottleneck assignment problem.* It turns out that we can solve the bottleneck assignment problem by repeated applications of any algorithm for finding matchings in bipartite graphs. Suppose we begin with any matching $M$ in the bipartite graph $G$. We can easily find the minimum weight of an edge in $M$, say $b$. We form a new graph $G_b$ from $G$ by removing all edges from $G$ with weight $b$ or less. If we now find a maximum matching in $G_b$, and if it is a perfect matching, then each of its edges must have weight greater than $b$, so we have improved the matching. If no such matching can be found, then the previous matching was the best. We continue this process until the matching which maximizes the minimum weight of an edge is found. Can you formally write an algorithm that solves the bottleneck assignment problem?

We conclude this section with a study of some interesting mathematical properties of marriage. For the remainder of this section we shall use some notions about matrices to study marriages. We take the marriage point of view because of the interesting and unusual manner the statements of our results will take. We begin with some ideas on matrices.

A matrix $D = (d_{i,j})$ is *doubly stochastic* if each $d_{i,j} \geq 0$ and the sum of the entries in any row or column equals 1. A *permutation matrix* is any matrix obtained from the identity matrix $I$ by performing a permutation on the rows of $I$. A well-known result on doubly stochastic matrices from Birkhoff [4] and Von Neumann [15] states that any doubly stochastic $n \times n$ matrix $D$ can be written as a combination of suitable permutation matrices. That is, there exist constants $c_1, c_2, \ldots, c_n$ and permutation matrices $P_1, \ldots, P_n$ such that $D = \sum_{i=1}^{n} c_i P_i$.

We can use matchings to indicate an algorithm for finding the constants and the decomposition of $D$ into permutation matrices.

Suppose we model our doubly stochastic matrix $D$ with a bipartite graph. Let vertices $r_1, r_2, \ldots, r_n$ represent the rows of $D$ and let vertices $k_1, \ldots, k_n$ represent the columns. We draw an edge from $r_i$ to $k_j$ if, and only if, entry $d_{i,j}$ of $D$ is nonzero. Then the permutation matrix $P_1$ represents the edges of a matching in this bipartite graph and the constant $c_1$ is the minimum weight of an edge in this matching. We can now write $D$ as $D = c_1 P_1 + R$, where the matrix $R$ represents the remaining edges of our bipartite graph. The old edges were adjusted by subtracting $c_1$ from the weight of each edge of the matching and removing any edge with weight zero. We now repeat this process on $R$.

Suppose that at some stage we are unable to find a matching. Then by Hall's theorem there must exist some set $A$ of vertices representing rows of $D$ such that $|A| > |N(A)|$. That is, there are more rows than "neighboring" columns. Now, consider what this means in our matrix $D$. If each of these rows sums to 1 (counting the entries that were possibly removed prior to this), then the total value of the weights in these rows is $|A|$. But then this amount must also be distributed over $|N(A)|$ columns, which means some column must sum to more than 1, contradicting that $D$ was doubly stochastic. Thus, we will be able to find a matching at each stage.

We now formally state the algorithm for finding this convex sum of permutation matrices.

**Algorithm 7.2.2 Decomposing Doubly Stochastic Matrices.**
**Input:** A doubly stochastic matrix $D$.
**Output:** A convex sum of permutation matrices $c_1 P_1 + \cdots + c_n P_n$.

1. Set $t_1 \leftarrow 1$, $X \leftarrow D$ and $k \leftarrow 1$.
2. Having a doubly stochastic matrix $X$ and nonnegative numbers $t_1, t_2, \ldots, t_k$ and permutation matrices $P_1, \ldots, P_{k-1}$ such that

   $$D = t_1 P_1 + \cdots + t_{k-1} P_{k-1} + t_k X \text{ and } \sum_{i=1}^{k} t_i = 1.$$

   If $X$ is a permutation matrix,
   then set $P_k \leftarrow X$ and halt;
   else use bipartite graphs to find a permutation matrix $X^*$ such that $x_{ij}{}^* = 0$ whenever $x_{ij} = 0$.
3. The $n$ entries $x_{i,j}$ of $X$ for which $x_{ij}{}^* = 1$ are all positive entries. Let $c$ be the least of these entries (note $c < 1$).
   Set $t \leftarrow t_k$, $t_k \leftarrow ct$, $t_{k+1} \leftarrow (1-c)t$ and $P_k \leftarrow X^*$. Now, replace $X$ by $\frac{1}{1-c}(X - cP_k)$, set $k \leftarrow k+1$ and go to step 2.

Using doubly stochastic matrices, we find that another unusual theorem about marriage is now possible. Suppose we consider a suitability matrix describing the marriage problem. That is, given a set of $n$ men and another set of $n$ women, let the matrix $S = (s_{i,j})$ be defined so that $s_{i,j}$ is a measure of the suitability or "happiness" of a marriage between man $i$ and woman $j$. Our goal is to study the type of marriage that brings this collection of men and women the most "happiness." In particular, we will compare monogamy and polygamy. These relationships can be shown in a matrix $M = (m_{ij})$. Each row of the matrix $M$ represents a man in our set of men and each column a woman. The entry $m_{i,j}$ in our marriage matrix $M$ represents the fraction of time man $i$ spends with woman $j$. Thus, monogamy would be a permutation matrix and polygamy a general doubly stochastic matrix. Our measure of the happiness of the present marriage relationship $M$ will be $h(M) = \sum_{i,j} s_{i,j} m_{i,j}$. Our solution is then to find $\max_M h(M)$, where the maximum is taken over all doubly stochastic matrices $M$. But we note that

$$\begin{aligned}\max_M h(M) &= \max_{c_1, \ldots, c_n} h(c_1 P_1 + \cdots + c_n P_n) \\ &= \max_{c_1, \ldots, c_n} c_1 h(P_1) + \cdots + c_n h(P_n) \\ &= h(P_i) \quad \text{for some } i.\end{aligned}$$

(The above follows easily for the maximum $h(P_i)$). That is, the maximum corresponds to the matching represented by some permutation matrix. In other words, monogamy is the preferred mathematical state of marriage. We have just proven the following interesting marriage theorem.

**Theorem 7.2.4** Among all forms of marriage, monogamy is optimal.

We conclude our study of marriage by considering its stability. Suppose we have a set of $n$ men $m_1, \ldots, m_n$ and $n$ women $w_1, \ldots, w_n$. Suppose, too, that man $m_1$ is married to woman $w_1$ and man $m_2$ to woman $w_2$. Further suppose that in reality $m_2$ prefers $w_1$ to his own wife and $w_1$ prefers $m_2$ to her own husband. It is easy to believe this is not a "stable" situation, in fact, we call such a pair of marriages *unstable.*

Let's construct two preference tables. In each table, the rows represent the men and the columns the women. The entries in any row of the first preference table are the integers 1 to n. This represents the order of preference of the women by the man corresponding to this row (with 1 being first choice). A similar description applies to the columns of the women's preference table.

Our problem, then, is given the two preference tables, can we find a stable set of marriages. That is, can we find a matching in which no pair of independent edges is unstable. We now describe an algorithm to produce our stable matching.

**Algorithm 7.2.3 Stable Matching Algorithm.**
**Input:** Given preference tables for the men and the women.
**Output:** A set of stable marriages

1. Each man proposes to his first choice.
2. The women with two or more proposals respond by rejecting all but the most favorable offer. However, no woman accepts a proposal.
3. The men that were rejected propose to their next choice. Those that were not rejected continue their offers.
4. We repeat step 3 until we reach a stage where no proposal is rejected.

Clearly, each woman can only reject a finite number (namely $n - 1$) of proposals, and so this process must eventually stop. We illustrate our algorithm on the following preference tables.

| men | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| $m_1$ | 1 | 2 | 3 | 4 |
| $m_2$ | 1 | 4 | 3 | 2 |
| $m_3$ | 2 | 1 | 3 | 4 |
| $m_4$ | 4 | 2 | 3 | 1 |

| women | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
|---|---|---|---|---|
| $m_1$ | 3 | 3 | 2 | 3 |
| $m_2$ | 4 | 1 | 3 | 2 |
| $m_3$ | 2 | 4 | 4 | 1 |
| $m_4$ | 1 | 2 | 1 | 4 |

The set of proposals $P_i$ appears below; starred proposals were rejected.

| proposals | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ |
|---|---|---|---|---|---|---|
| $m_1$ | 1 | 1 | 1 | $1^*$ | $2^*$ | 3 |
| $m_2$ | $1^*$ | 4 | 4 | 4 | 4 | 4 |
| $m_3$ | 2 | 2 | $2^*$ | 1 | 1 | 1 |
| $m_4$ | 4 | $4^*$ | 2 | 2 | 2 | 2 |

From this table we see that the final set of marriages is:

man $m_1$ with woman $w_3$
man $m_2$ with woman $w_4$
man $m_3$ with woman $w_1$
man $m_4$ with woman $w_2$.

It is easy to verify that this set of marriages is stable.

We now wish to prove we actually reach a stable matching. Suppose this were not the case; that is, suppose there was an unstable pair of marriages. Without loss of generality, let this pair be $(m_1, w_1)$ and $(m_2, w_2)$.

But if $m_2$ prefers $w_1$, he would have proposed to $w_1$ before he proposed to his present wife. Then $w_1$ would not have rejected $m_2$ if she actually preferred him over $m_1$. Hence, we could not have reached this unstable situation. Thus, the marriages cannot be unstable. We have now shown the following result, due originally to Gale and Shapely [10].

**Theorem 7.2.5** Given $n$ men and $n$ women, there always exists a set of stable marriages.

## Section 7.3 Factoring

We now wish to study matchings in a generalized setting. In addition, we want to consider relaxations of the concept of matchings. A perfect matching is often called a 1-factor, since the matching is a 1-regular spanning subgraph of the original graph. It is not a difficult leap to the idea of an $r$-factor, that is, an $r$-regular spanning subgraph of the original graph. We begin with a natural result.

**Theorem 7.3.1** If $G$ is a graph of order $2n$ and $\delta(G) \geq n$, then $G$ contains a 1-factor.

**Proof.** This result follows as a consequence of Dirac's theorem (Corollary 5.2.1). □

We see that an algorithm for finding such a matching is apparent. Having obtained an $r$-matching, scan the remaining $(2n - 2r)$ vertices to see if any pair is joined by an edge. If this fails to be the case, choose any two of these vertices, say $a$ and $b$, and scan the edges $xy$ of the matching until one is found such that $a$ is adjacent to $x$ and $b$ is adjacent to $y$. Then, replace $xy$ by the edges $ax$ and $by$ and repeat this process.

The fundamental result on 1-factors is from Tutte [14]. The proof presented here is that of Anderson [1]. We denote the number of components of odd order in a graph $G$ by $k_o(G)$.

**Theorem 7.3.2** (Tutte [14]) A nontrivial graph $G$ has a 1-factor if, and only if,

$k_o(G - S) \leq |S|$ for every proper subset $S$ of $V(G)$.

**Proof.** Let $F$ be a 1-factor of $G$ and suppose there exists a proper subset $S$ of $V(G)$ such that $k_o(G - S) > |S|$. For each of the odd components $C$ of $G - S$, there must exist an edge in $F$ that goes from $C$ to $S$. But this implies that there is a vertex in $S$ incident with at least two edges in $F$, which contradicts the definition of matching.

Conversely, note that $k_o(G - \phi) \leq |\emptyset| = 0$. Thus, $G$ has only even components, and the order $n$ of $G$ must, then, be even. Also, observe that for every proper subset $S$ of $V(G)$, the numbers $|S|$ and $k_o(G - S)$ are of the same parity.

Now, we proceed by induction on the order of $G$. If $n = 2$, then $G$ must be $K_2$, and clearly $G$ has a 1-factor. Next, assume that for all graphs $H$ of even order less than $n$, the condition $k_o(H - S) \leq |S|$ for every proper nonempty subset $S$ of vertices implies $H$ has a 1-factor. Let $G$ be a graph of even order $n$ and assume that $k_o(G - S) \leq |S|$ for every proper subset $S$ of $V(G)$. We now consider two cases.

Case 1. Suppose that $k_o(G - S) < |S|$ for all subsets $S$ of $V(G)$ with $2 \leq |S| < n$. Since $k_o(G - S)$ and $|S|$ have the same parity, $k_o(G - S) \leq |S| - 2$. Let $uv$ be an edge of $G$ and consider $G - u - v$. Let $S_1$ be a proper subset of $V(G - u - v)$. Thus,

$$k_o(G - u - v - S_1) \leq |S_1|,$$

or else

$$k_o(G - u - v - S_1) > |S_1| = |S_1 \cup \{u, v\}| - 2,$$

and, hence,

$$k_o(G - (S_1 \cup \{u, v\})) \geq |S_1 \cup \{u, v\}|,$$

which contradicts our assumptions. Then the matching obtained by applying the induction hypothesis, along with the edge $uv$, provides a 1-factor of $G$.

Case 2. Suppose that there exists some set $S_2$ such that $k_o(G - S_2) = |S_2|$. Among all such sets, let $S$ be one of maximum cardinality. Further, let $k_o(G - S) = |S| = t$ and let $C_1, \ldots, C_t$ be the odd components of $G - S$. If $E$ is an even component of $G - S$ and $x \in V(E)$, then $k_0(G - S - x)$ would equal $|S \cup \{x\}|$ contradicting the fact that $S$ was a set of maximum cardinality having this property. Thus, $G - S$ has no even components.

Let $S_i$ $(i = 1, \ldots, t)$ denote those vertices of $S$ with adjacencies in $C_i$. Each $S_i$ is nonempty, or else some $C_i$ would be an odd component of $G$. The union of any $k$ of the sets $S_1, \ldots, S_t$ contains at least $k$ vertices, or there exists an integer $k$ such that the union $U$ of some $k$ of these sets contains less than $k$ vertices. Thus, $k_o(G - U) > |U|$, which is a contradiction. Hence, by Hall's theorem (SDR), there exists a system of distinct representatives for the sets $S_1, \ldots, S_t$. This implies that in $S$ there are distinct vertices $v_1, \ldots, v_t$ and that in each $C_i$ there is a vertex $u_i$ such that $v_i u_i$ is an edge of $G$.

Let $W$ be a proper subset of $C_i - u_i$. Since $C_i - u_i$ has even order, $k_o(C_i - u_i - W)$ and $|W|$ have the same parity. If

$$k_o(C_i - u_i - W) > |W|,$$

then it must be that

$$k_o(C_i - u_i - W) \geq |W| + 2.$$

Thus,

$$\begin{aligned} k_o(G - (S \cup W \cup \{u_i\})) &= k_o(C_i - u_i - W) + k_o(G - S) - 1 \\ &\geq |S| + |W| + 1 \\ &= |S \cup W \cup \{u_i\}|. \end{aligned}$$

But this contradicts the maximality of $S$. Hence,

$$k_o(C_i - u_i - W) \leq |W|,$$

and so by induction, each $C_i - u_i$ has a 1-factor. These 1-factors, together with the edges $u_i v_i$, then form the desired 1-factor in $G$. □

Berge [3] noticed a useful related fact stemming from the proof of Tutte's theorem. This observation is often called the *Berge defect form* of Tutte's theorem. From Tutte's theorem, we see that a graph $G$ of even order $p$ contains a perfect matching unless there exists some set of $r$ vertices whose removal leaves a graph with more than $r$ odd components. However, because $G$ has even order, this forces the existence of at least $r + 2$ odd components (see the proof). Further, the defect form also states that if $G$ is a graph of odd order $p$, then $G$ contains a maximum matching of size $\frac{1}{2}(p - 1)$ unless there is some set of $r$ vertices whose removal leaves a graph with at least $r + 3$ odd components. This observation can be useful in dealing with graphs of odd order.

It is clear that every 1-regular graph contains (in fact, is) a 1-factor and that every 2-regular graph contains a 1-factor (in fact, is *1-factorable*, that is, its edge set can be decomposed into 1-factors) if, and only if, every component is an even cycle. The situation is not as simple for 3-regular graphs, however. Petersen [13] investigated 1-factors in 3-regular graphs and showed that they need not contain a 1-factor (see Figure 7.3.1). However, he was also able to show a situation in which such a graph would contain a 1-factor.

**Theorem 7.3.3** (Petersen [13]). Every bridgeless 3-regular graph $G$ can be expressed as the edge sum of a 1-factor and a 2-factor.

**Proof.** It suffices to show that such a graph contains a 1-factor since the remaining edges form a 2-factor. Suppose the graph $G$ fails to contain a 1-factor. Then by Tutte's theorem, there exists in $G$ some proper nonempty set $S$ of $k$ vertices such that $n = k_o(G - S) > |S| = k$. Suppose that $C_1, \ldots, C_n$ are the odd components of $G - S$. There must exist an edge from each $C_i$ to $S$ $(1 \le i \le n)$, or else some $C_i$ would be a 3-regular graph of odd order, which is impossible. Further, since $G$ is bridgeless, there cannot be a single edge joining $S$ to any $C_i$. If there were exactly two edges joining $S$ to some $C_i$, then again $C_i$ would contain an odd number of vertices of odd degree. Thus, at least three edges join any $C_i$ to $S$. Thus, there are at least $3n$ edges joining $S$ and the $C_i$ $(1 \le i \le n)$. However, since each vertex of $S$ has degree 3, there can be at most $3k$ edges into $S$. But since $3n > 3k$, a contradiction arises. Thus, no such set $S$ can exist, and by Tutte's theorem, we see that $G$ must contain a 1-factor. □

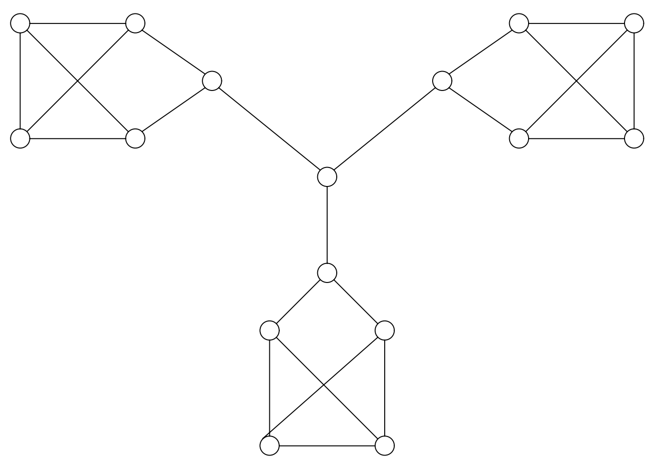


**Figure 7.3.1.** A 3-regular graph with no 1-factor.

Now that we know that every bridgeless 3-regular graph can be factored into a 1-factor and a 2-factor, it is natural to wonder if it can actually be 1-factored. Petersen also showed that this is not the case. His example, which has become perhaps the most famous of all graphs, is shown in Figure 7.3.2.

Petersen also characterized those graphs which are 2-factorable. It turns out that the obvious necessary condition that the graph be $2r$-regular for some $r \ge 1$ also suffices. The proof makes use of the fact that such graphs are eulerian.

**Theorem 7.3.4** A nonempty graph $G$ is 2-factorable if, and only if, $G$ is $2r$-regular $(r \ge 1)$ for some integer $r$.

**Proof.** Clearly, if $G$ is 2-factorable, then $G$ is $2r$-regular for some $r \ge 1$.

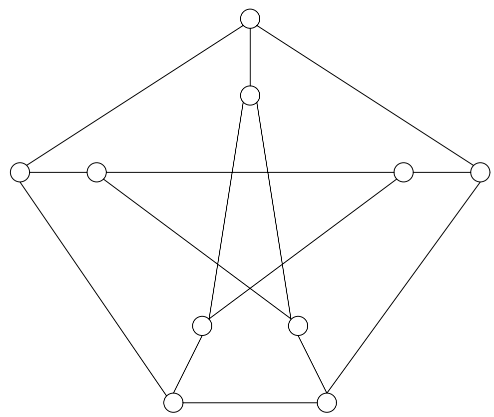


**Figure 7.3.2.** The Petersen Graph.

Conversely, let $G$ be a $2r$-regular graph ($r \geq 1$). Without loss of generality we may assume $G$ is connected, for otherwise we would simply consider each component separately. Thus, we see that $G$ is eulerian with circuit $C$. Let $V(G) = \{ v_1, v_2, \ldots, v_p \}$ and define a bipartite graph

$$B = (V_1 \cup V_2, E)$$

from $G$ as follows: Let

$$V_1 = \{ u_1, u_2, \ldots, u_p \}, \; V_2 = \{ w_1, w_2, \ldots, w_p \} \text{ and}$$
$$E(B) = \{ u_i w_j \mid v_j \text{ immediately follows } v_i \text{ on } C \}.$$

The graph $B$ is $r$-regular, and so by Corollary 7.1.1, $B$ contains a perfect matching $M_1$. Then the graph $B - M_1$ is $r - 1$-regular and again by Corollary 7.1.1, $B - M$ contains a perfect matching $M_2$. Continuing in this manner, we see that $E(B)$ can be partitioned into matchings $M_1, M_2, \ldots, M_r$.

Corresponding to each matching $M_k$ of $B$ is a permutation $\pi_k$ on the set of vertices defined by $\pi_k(v_i) = v_j$ if $u_i w_j \in E(M_k)$. We know that we can express $\pi_k$ as the product of disjoint permutation cycles. Note that in this product, no permutation cycle is of length 1, for this would imply that $\pi_k(v_i) = v_i$. But this implies that $u_i w_i \in E(B)$, and, hence, that $v_i v_i$ is an edge of $C$, contradicting the fact $G$ is a graph. Further note that there is no permutation cycle of length 2 in the product since this would imply that $\pi_k(v_i) = v_j$ and $\pi_k(v_j) = v_i$. But this means that $u_i w_j$ and $u_j w_i$ are edges of $B$ and that $v_j$ both precedes and follows $v_i$ on $C$. But this contradicts the fact that $C$ is a circuit and, hence, has no repeated edges. Thus, we are able to conclude that each permutation cycle in the product of disjoint permutation cycles representing $\pi_k$ has length at least 3.

Each permutation cycle in $\pi_k$ then gives rise to a cycle in $G$, and since the product of the permutation cycles is disjoint, the corresponding cycles span $V(G)$. But, these spanning cycles form a 2-factor of $G$. Further, since the matchings $M_1, M_2, \ldots, M_r$

partition the edges of $G$, the 2-factors that correspond to $\pi_1, \ldots, \pi_r$ are mutually edge disjoint. Thus, $G$ is 2-factorable. □

We conclude this section by considering some special classes of graphs. The obvious starting point is the complete graphs. It turns out that we can produce very special 2-factors in $K_{2p+1}$. A 2-factorization of $K_7$ is shown in Figure 7.3.3.

**Theorem 7.3.5** For every positive integer $p$, the graph $K_{2p+1}$ can be 2-factored into $p$ hamiltonian cycles.

**Proof.** The result is trivial when $p = 1$, so we can assume that $p \geq 2$. Let the vertices of $K_{2p+1}$ be $v_0, \ldots, v_{2p}$. We arrange the vertices $v_1, \cdots, v_{2p}$ cyclically in a regular $2p$-gon and place $v_0$ in the center of the arrangement. We define the edges of the 2-factor $F_i$ to consist of the edges $v_0 v_i$, $v_0 v_{p+i}$ along with $v_i v_{i+1}$ and all edges parallel to this edge, and $v_{i-1} v_{i+1}$ and all edges parallel to this edge. (All subscripts are expressed modulo $2p$). Then each $F_i$ is a hamiltonian cycle, and $K_{2p+1}$ is the edge sum of these 2-factors. □

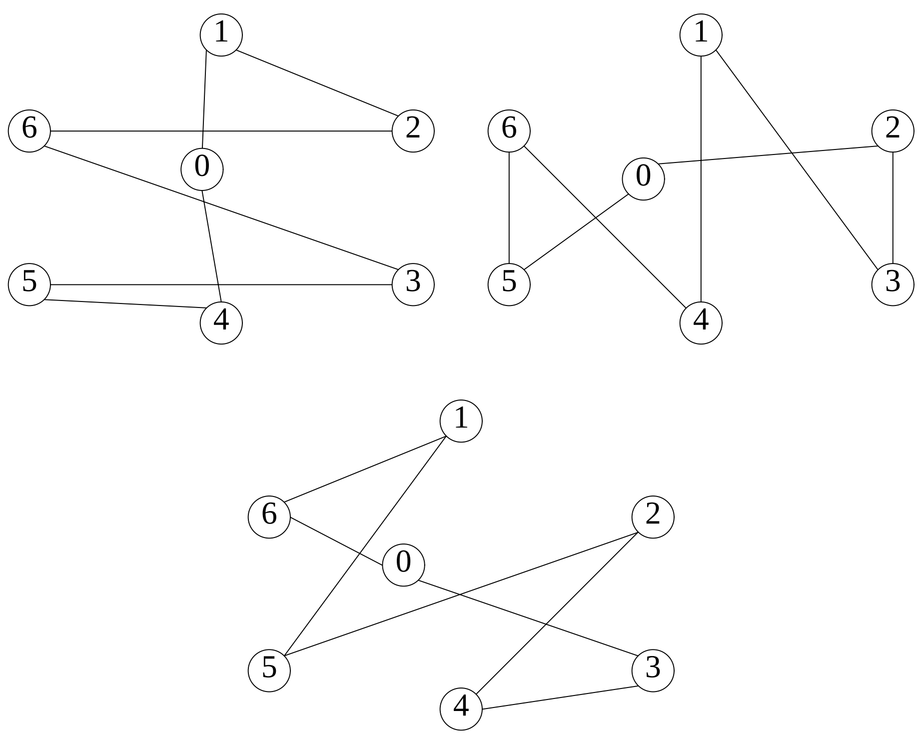



**Figure 7.3.3.** A 2-factorization of $K_7$.

**Corollary 7.3.1** For every positive integer $p$, the graph $K_{2p}$ can be factored into $p$

hamiltonian paths.

We conclude this section with another result on complete graphs. Its proof is left to the exercises.

**Theorem 7.3.6** For every positive integer $p$, the graph $K_{2p}$ is 1-factorable.

## Section 7.4 Degrees and 2-Factors

In this section we wish to consider several results that appear similar to some of the theorems we saw earlier dealing with hamiltonian graphs. Since a hamiltonian cycle is a 2-factor, it is not surprising that there is a relationship between these hamiltonian results and theorems dealing with 2-factors. We begin with a very nice result due to El-Zahar [7].

**Theorem 7.4.1** Let $G$ be a graph of order $n$ and let $n_1 \geq 3$ and $n_2 \geq 3$ be two integers such that $n = n_1 + n_2$. If $\delta(G) \geq \left\lceil \frac{n_1}{2} \right\rceil + \left\lceil \frac{n_2}{2} \right\rceil$, then $G$ contains two disjoint cycles $C_1$ and $C_2$ of length $n_1$ and $n_2$, respectively.

El-Zahar's Theorem can be viewed as a generalization of Dirac's Theorem on hamiltonian graphs. Dirac's Theorem provides for a 2-factor that is one cycle while El-Zahar's Theorem uses a slightly stronger degree condition to provide for a 2-factor that is two cycles. A stronger look at Dirac's condition allows us to actually say much more. We begin with a lemma.

**Lemma 7.4.1** Let $G$ be a graph of order $n$ with minimum degree $\delta(G) \geq n/2$. If $G$ contains $k \geq 1$ vertex disjoint cycles $C_1, C_2, \ldots, C_k$ such that

$$\left| V(G) - \bigcup_{i=1}^{k} V(C_i) \right| \leq 2,$$

then $G$ has a 2-factor with exactly $k$ vertex disjoint cycles.

**Proof.** If $V(G) - \bigcup_{i=1}^{k} V(C_i) = \{ w \}$, then $G$ contains the desired 2-factor since $deg\ w \geq n/2$ and hence $w$ is adjacent to two consecutive vertices of a least one of the cycles.

Thus we may assume $V(G) - \bigcup_{i=1}^{k} V(C_i) = \{ u, v \}$. If one of $u$ and $v$, say $u$, is adjacent to two consecutive vertices of one of the cycles, then, as before, we obtain the desired 2-factor. Thus we may assume that $deg\ u = deg\ v = n/2$ and that each of $u$ and $v$ is adjacent to alternate vertices of each of the cycles and necessarily to each other. Let

$$C_1 : u_1, u_2, \ldots, u_t, u_1$$

be one such cycle. If $u$ and $v$ are adjacent to the same set of vertices of $C_1$, say $\{ u_1, u_3, \ldots, u_{t-1} \}$. Then $C_1$ can be replaced by

$$u_1, u, v, u_3, u_4, \ldots, u_t, u_1$$

to obtain $k$ vertex disjoint cycles containing all but one vertex of $G$. In this case, as we have seen, $G$ has the desired 2-factor. On the other hand, if $u$ is adjacent to $u_1, u_3, \ldots, u_{t-1}$ and $v$ is adjacent to $u_2, u_4, \ldots, u_t$. Then we may replace $C_1$ with the cycle

$$u_1, u, u_3, u_2, v, u_4, u_5, \ldots, u_t, u_1$$

to complete the proof. □

Now, with the aid of the lemma, we can take the stronger look at Dirac's condition promised earlier. The result is from [5].

**Theorem 7.4.2** Let $k$ be a positive integer and let $G$ be a graph of order $n \geq 4k$ with minimum degree $\delta(G) \geq n/2$. Then $G$ has a 2-factor with exactly $k$ vertex disjoint cycles.

**Proof.** The cases $k = 1, 2$ follow from Dirac's Theorem and El-Zahar's Theorem, respectively. Thus we may assume that $k > 2$. Since $\delta(G) \geq n/2 \geq 2k$ and $n \geq 4k$, $G$ contains $k$ vertex disjoint cycles $C_1, C_2, \ldots, C_k$ by Theorem 5.8.4. Let $X = V(G) - \bigcup_{i=1}^{k} V(C_i)$ and assume $X \neq \emptyset$.

If $\delta(\langle X \rangle) < |X|/2$, let $w \in X$ with $deg_{\langle X \rangle}(w) < |X|/2$. Then, since $\delta(G) \geq n/2$, it follows that $w$ is adjacent to more than half of the vertices of some $C_i$, $1 \leq i \leq k$ and therefore adjacent to consecutive vertices of $C_i$. Therefore $w$ can be added to $C_i$. Continue this process to obtain $k$ vertex disjoint cycles $C'_1, C'_2, \ldots, C'_k$ such that either

$$V(G) = V(C'_1) \cup V(C'_2) \cup \cdots \cup V(C'_k) \quad \text{or}$$
$$X' = V(G) - \bigcup_{i=1}^{k} V(C'_i) \neq \emptyset \quad \text{and}$$
$$\delta(\langle X' \rangle) \geq |X'|/2.$$

In the first case we have the desired 2-factor. In the second case, either $\langle X' \rangle = K_2$ or $\langle X' \rangle$ is hamiltonian. If $\langle X' \rangle = K_2$, then by applying Lemma 7.4.1 we obtain the desired 2-factor. Thus we may assume that $C'_{k+1}$ is a hamiltonian cycle of $X'$. Without loss of generality, assume that $|V(C'_1)| \leq |V(C'_i)|$ for $i = 2, 3, \ldots, k+1$, so that

$|V(C'_1)| \le \dfrac{n}{k+1} \le n/4.$

Since $\delta(G) \ge n/2$, the number of edges between $V(C'_1)$ and $V(G)-V(C'_1)$ is at least

$$|V(C'_1)|\,(n/2 - |V(C'_1)| + 1).$$

If between every three consecutive vertices of $C'_1$ and of $C'_i$ $(2 \le i \le k+1)$ there are at most three edges, then the number of edges between $V(C'_1)$ and $V(G) - V(C'_1)$ is at most

$$(\tfrac{1}{3})\,(n - |V(C'_1)|)\,|V(C'_1)|.$$

This, however, implies that

$$(\tfrac{1}{3})\,(n - |V(C'_1)|)\,|V(C'_1)| \ge |V(C'_1)|\,(n/2 - |V(C'_1)| + 1),$$

so that

$$|V(C'_1)| \ge n/4 + 3/2,$$

contradicting the fact $|V(C'_1)| \le n/4$. Thus, for some $i$ with $2 \le i \le k+1$, three consecutive vertices of $C'_1$ have at least four adjacencies to three consecutive vertices of $C'_i$. In this case it is straightforward to verify that $C'_1$ and $C'_i$ can be combined to form a cycle containing all but at most two of the vertices of $C'_1$ and $C'_i$. Then an application of Lemma 7.4.1 completes the proof. □

With slightly more effort it is possible to extend our generalizations to an Ore-like result concerning degree sums of nonadjacent vertices. The following is also from [5].

**Theorem 7.4.3** Let $G$ be a graph of order $n \ge 4k$ such that $deg\ x + deg\ y \ge n$ for each pair of nonadjacent vertices $x, y$ in $V(G)$, then $G$ has a 2-factor with exactly $k$ vertex disjoint cycles.

## Exercises

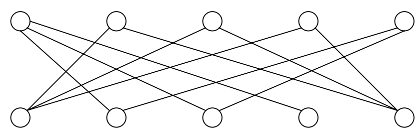
1. Show that the $n$-cube $Q_n$ $(n \ge 2)$ has a perfect matching.
2. Show that $Q_n$ is $r$-factorable if, and only if, $r \mid n$.
3. Characterize when the graph $K_{p_1, p_2, \ldots, p_n}$ has a perfect matching.
4. Determine the number of perfect matchings in the graphs $K_{p,p}$ and $K_{2p}$.

5. How many perfect matchings can exist in a tree?

6. Find a maximum matching and a minimum cover in the graph below using each of the indicated methods.

    a. Algorithm 7.2.1 and Theorem 7.2.1.

    b. A network model.



7. Use Dirac's theorem (Corollary 5.2.1) to show that if $G$ has even order $p$ and $\delta(G) \geq \frac{p}{2} + 1$, then $G$ has a 3-factor.

8. Show that every doubly stochastic matrix is a square matrix.

9. Show that if $G = (X \cup Y, E)$ is a bipartite graph, then

$$\beta_1(G) = |X| - \max_{S \subseteq X}\{|S| - |N(S)|\}.$$

10. Use the previous exercise to show that if the $(p, q)$ graph $G = (X \cup Y, E)$ is bipartite and $|X| = |Y| = n$ and $q > (k - 1)n$, then $G$ has a matching of cardinality $k$.

11. [9] Suppose that $G$ is a graph of order $p$ with the property that for every pair of nonadjacent vertices $x$ and $y$, $|N(x) \cup N(y)| \geq s$.

    a. Use Berge's defect form of Tutte's theorem to show that if $s > 2\left\lfloor \frac{p}{3} \right\rfloor - 2$ and $p$ is odd and $p \geq 6$, then

$$\beta_1(G) = \frac{1}{2}(p - 1).$$

    b. Find a graph of order 5 for which the conditions of part (a) fail to ensure $\beta_1(G) = \frac{1}{2}(p - 1)$.

    c. Use Tutte's theorem to show that if $s > \frac{2}{3}(p - 1) - 1$ and $p$ is even and $G$ is connected, then $\beta_1(G) = \frac{p}{2}$.

12. Use Tutte's theorem to prove Hall's theorem.

13. Use König's theorem to prove Hall's theorem.

14. Prove Corollary 7.3.1.

15. Prove Theorem 7.3.6.

16. Can $K_{2n}$ can be factored into $n - 1$ hamiltonian paths and one 1-factor?

17. Let $G$ be a $(p, q)$ graph of even order $p$ with $\delta(G) < \frac{p}{2}$. Show that if

$$q > \binom{\delta(G)}{2} + \binom{p - 2\delta(G) - 1}{2} + \delta(G)(p - \delta(G)),$$

then $G$ has a perfect matching.

18. Four men and four women apply to a computer dating service. The computer evaluates the unsuitability of each man for each woman as a percentage (see the table below). Find the best possible dates for each woman for this Friday night.

| | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|
| $W_1$ | 60 | 35 | 30 | 65 |
| $W_2$ | 30 | 10 | 55 | 30 |
| $W_3$ | 40 | 60 | 15 | 35 |
| $W_4$ | 25 | 15 | 40 | 40 |

19. Consider the table used for the last exercise as representing the weights assigned to a bipartite graph and solve the bottleneck assignment problem for this graph.

20. The math department at your college has six professors that must be assigned to teach each of five different classes. The department did an examination of the suitability of each professor for each class and the unsuitability table is shown below. What is the optimal teaching assignment that can be made if no professor is assigned more than one class?

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ |
|---|---|---|---|---|---|---|
| $C_1$ | 75 | 25 | 55 | 25 | 50 | 35 |
| $C_2$ | 60 | 30 | 45 | 35 | 45 | 20 |
| $C_3$ | 55 | 25 | 50 | 15 | 50 | 30 |
| $C_4$ | 40 | 35 | 40 | 45 | 35 | 25 |
| $C_5$ | 50 | 20 | 45 | 30 | 40 | 45 |

(Hint: Add a dummy class that each professor is equally suited to teach.)

21. Does the previous problem make sense as a bottleneck assignment problem? If so, solve it.

22. Consider the doubly stochastic matrix below. Use Algorithm 7.2.2 to decompose this matrix into permutation matrices.

$$\begin{vmatrix} 0.3 & 0.3 & 0.0 & 0.3 & 0.1 \\ 0.1 & 0.5 & 0.2 & 0.1 & 0.1 \\ 0.2 & 0.0 & 0.3 & 0.5 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.0 & 0.3 \\ 0.4 & 0.0 & 0.0 & 0.1 & 0.5 \end{vmatrix}$$

23. Consider the table of the previous problem as the weights assigned to the edges of a bipartite graph. Interpret your solution in relation to the last problem on this graph.

24. Explain why the adjustment process allows us to complete the hungarian algorithm applied to an unsuitability matrix.

25. A *decomposition* of $G$ is a collection $\{ H_i \}$ of subgraphs of $G$ such that $H_i = < E_i >$ for some subset $E_i$ of $E(G)$ and where the sets $\{ E_i \}$ partition $E(G)$. Prove that the complete graph $K_p$ can be decomposed as a collection of 3-cycles if, and only if, $p \geq 3$, $p$ is odd and 3 divides $\binom{n}{2}$.

26. Find a decomposition of $K_5$ as 5-cycles.

27. Find a decomposition of $K_{10}$ as paths of length 5.

28. Prove that for each integer $n \geq 1$, the graph $K_{2n+1}$ can be decomposed as a collection of stars $K_{1,\,n}$ and that the graph $K_{2n}$ can be decomposed as a collection of stars $K_{1,\,n}$.

29. By an ascending subgraph decomposition of a graph $G$ of size $\binom{n+1}{2}$ we mean an edge decomposition into subgraphs $G_1, \ldots, G_n$ with the properties that $|E(G_i)| = i$ and $G_1 \leq G_2 \leq \cdots \leq G_n$, that is, each $G_i$ is isomorphic to a subgraph of $G_{i+1}$. Show that $K_m$ has an ascending subgraph decomposition as stars and also an ascending subgraph decomposition as paths.

30. Find an example that shows that the $n \geq 4k$ condition in Theorem 7.4.2 cannot be reduced.

31. Find an example to show that the degree condition in El-Zahar's Theorem is sharp.

32. Prove Theorem 7.4.1.

## References


1. Anderson, I., Perfect Matchings of a Graph. *J. Combinatorial Theory,* 10B(1971), 183 – 186.
2. Berge, C., Two Theorems in Graph Theory. *Proc. Nat. Acad. Sci. USA,* 43(1957), 842 – 844.
3. Berge, C., Sur le Couplage Maximum d'un Graphe. *C. R. Acad. Sci. Paris,* 247(1958), 258 – 259.
4. Birkhoff, G., Tres Observaciones Sobre el Algebra Lineal. *Universidad Nacionale Tucumán Revista,* 5(1946), 147 – 151.
5. Brandt, S., Chen, G., Faudree, R.J., Gould, R.J., and Lesniak, L., On the Number of Cycles in a 2-Factor, (preprint).
6. Egerváry, J., Matrixok Kombinatorikus Tulajdonságairól. *Mathematika és Fizikai Lápok*, 38(1931), 16 – 28.
7. El-Zahar, M.H., On Circuits in Graphs, *Discrete Math.* 50(1984d), 227-230.
8. Faudree, R. J., Gould, R. J., Jacobson, M. S., and Schelp R. H., Extremal Problems Involving Neighborhood Unions, *J. Graph Theory,* (in press).
9. Ford, L. R., Jr. and Fulkerson, D. R., *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.
10. Gale, D. and Shapley, L.S., College Admissions and the Stability of Marriage, *The American Mathematical Monthly,* 69(1962), 9-14.
11. Hall, P., On Representatives of Subsets, *J. London Math. Soc.,* 10(1935), 26 – 30.
12. König, D., Graphs and Matrices (Hungarian). *Mat. Fig. Lapok,* 38(1931), 116 – 119.
13. Petersen, J., Die Theorie der Regulären Graphs. *Acta Math.*, 15(1891), 193 – 220.
14. Tutte, W. T., The Factorization of Linear Graphs. *J. London Math. Soc.,* 22(1947), 107 – 111.
15. Von Neumann, J., A Certain Zero-Sum Game Equivalent to the Optimal Assignment Problem. Contributions to the Theory of Games, Kuhn and Tucker, eds., *Annals of Mathematics Studies,* 28(1953), 206 – 212.