3.15. Suppose that the augmenting path algorithm always chooses an augmenting path having as few reverse arcs as possible. Prove that the number of augmentations will be $O(mn)$. Give an $O(m)$ algorithm for finding each augmentation.

3.3 APPLICATIONS OF MAXIMUM FLOW AND MINIMUM CUT

In this section we discuss a number of applications of maximum flow and minimum cut. Several more applications appear in later chapters.

Bipartite Matchings and Covers

We are given disjoint sets P of men and Q of women, and the pairs (p,q) that like each other. The ***marriage problem*** is to arrange as many (monogamous) marriages as possible with the restriction that married people should (at least initially!) like each other. We can associate with the input a graph $G = (V, E)$ such that $V = P \cup Q$ and $E = \{pq : \ p \in P, \ q \in Q\}$. Such graphs, that is, ones in which there is a partition of the nodes into two parts such that every edge has its ends in different parts, are called ***bipartite***. (Sometimes $\{P, Q\}$ is called a ***bipartition*** of G.) The marriage problem asks for a ***matching*** of G of maximum size, that is, a subset M of E such that no two edges in M share an end. A bipartite graph appears in Figure 3.5, and the thick edges constitute a matching of size 6; we shall see that there is no larger one. Although the problem of finding a maximum matching in a general graph is more difficult (and is treated in Chapter 5), that for bipartite graphs is an easy application of maximum flows.

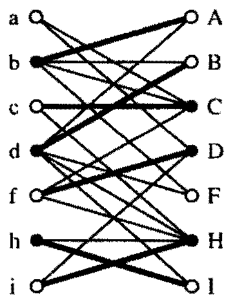

Figure 3.5. A bipartite graph and a matching

In fact, the bipartite matching problem was solved by Kőnig [1931] long before the development of network flow theory. He discovered a characterization of the maximum size of a matching that can be thought of as a prototype of the Max-Flow Min-Cut Theorem. (He also introduced, in the restricted setting of bipartite matching, the notion of flow-augmenting path.) Again, the basic idea is a general method to provide bounds, in this case, an upper bound for the maximum size of a matching. A *cover* of a graph G is a set C of nodes such that every edge of G has at least one end in C. For any matching M and any cover C, each edge $vw \in M$ has an end in C, but because matching edges cannot have an end in common, the corresponding nodes of C are all distinct. Therefore, $|M| \leq |C|$. It follows that, if we can find a matching M and a cover C with $|M| = |C|$, then we know that M is maximum. For the graph of Figure 3.5, the black nodes form a cover of size 6 and so the displayed matching is indeed of maximum size. Kőnig proved that, for bipartite graphs, it is always possible to make this kind of argument.

Theorem 3.14 *(Kőnig's Theorem) For a bipartite graph G,*

$$\max\{|M| : M \textit{ a matching}\} = \min\{|C| : C \textit{ a cover}\}.$$

We shall show how the Max-Flow Min-Cut Theorem implies Kőnig's Theorem, and how a maximum flow algorithm provides an efficient algorithm for constructing a maximum matching and a minimum cover. Given G with bipartition $\{P, Q\}$, we form a digraph G' with capacity vector u as follows. $V' = V \cup \{r, s\}$, where r, s are new nodes. For each edge pq of G with $p \in P$, $q \in Q$ there is a (directed) arc pq of G' with capacity ∞. For each $p \in P$ there is an arc rp of capacity 1. For each $q \in Q$ there is an arc qs of capacity 1. For the graph of Figure 3.5 we show the corresponding flow network in Figure 3.6.

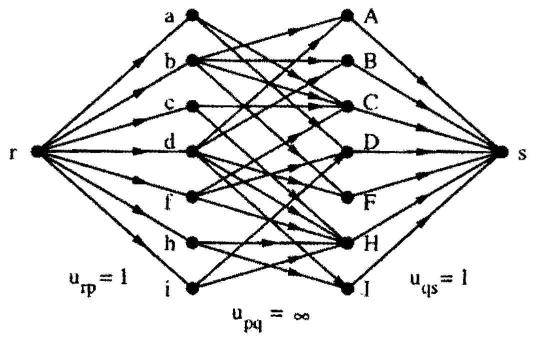

Figure 3.6. Flow network for bipartite matching

Let x be an integral feasible flow in G' of value k. In fact, this implies that x is $\{0,1\}$-valued. (Why?) Define $M \subseteq E$ by: $pq \in M$ if $x_{pq} = 1$ and $pq \notin M$ if $x_{pq} = 0$. Then M is a matching of G, and $|M| = k$. Now suppose that we are given a matching M of G. Define $(x_{vw} : vw \in E')$ by: If $v \in P$, $w \in Q$ then $x_{vw} = 1$ if $vw \in M$ and 0 if $vw \notin M$; if $v = r$, $w \in P$, then $x_{vw} = 1$ if there is an edge of M incident with w and $x_{vw} = 0$ otherwise; if $v \in P$, $w = s$, then $x_{vw} = 1$ if there is an edge of M incident with v and $x_{vw} = 0$ otherwise. Then x is an integral feasible flow of value $|M|$ in G'. Hence we can find a maximum cardinality matching in G by solving the maximum flow problem on G'. There will be at most $|P| \leq n$ augmentations, by Theorem 3.9, since the maximum matching size is at most $|P|$. So we get an algorithm for maximum bipartite matching having running time $O(mn)$.

Now consider a minimum cut $\delta'(\{r\} \cup A)$ where $A \subseteq V$. Since it has finite capacity, there can be no edge of G from $A \cap P$ to $Q \backslash A$. Therefore, every edge of G is incident with an element of $C = (P \backslash A) \cup (Q \cap A)$. That is, C is a cover. Moreover, the capacity of the cut is $|P \backslash A| + |Q \cap A| = |C|$, so C is a cover of cardinality equal to the maximum size of a matching. This proves Kőnig's Theorem, and shows that the algorithm also finds a minimum-cardinality cover.

There are many other related applications. Some of them are investigated in the exercises.

Optimal Closure in a Digraph

There are many applications in which we want to choose an optimal subset of "projects," where each project has a benefit. This benefit may be positive, negative, or zero. There is no restriction on the number of projects to be chosen, but there are restrictions of the form: If project v is to be chosen, then project w must be chosen also. If we model the projects as the nodes of a digraph G and the restrictions (v, w) as its arcs, then we must choose a maximum benefit set $A \subseteq V$ such that $\delta(A) = \emptyset$. We call such a subset A a *closure* of G. We remark that the problem is trivial if either all the benefits are nonnegative (V will be optimal) or all are nonpositive ($\emptyset$ will be optimal).

A classical application of this form is in the design of an open-pit mine. Here the region under consideration is divided into 3-dimensional blocks. For each block v there is a known estimated net profit b_v associated with excavating block v. The constraints come from the fact that it is not possible to excavate a block without also excavating those above it. The definition of "above" will depend on restrictions on the steepness of the sides of the pit.

It turns out that the optimal closure problem can be reduced to a minimum cut problem, an observation due to Picard [1976]; in earlier work, Rhys [1970] solved an important special case. Given G and b, define a digraph G' and capacity vector u as follows. Put $V' = V \cup \{r, s\}$ for new nodes r, s. For each $v \in V$ with $b_v > 0$, G' has an arc rv with $u_{rv} = b_v$. For each $v \in V$ with $b_v < 0$, G' has an arc vs with $u_{vs} = -b_v$. The remaining arcs of G' are just the arcs of G, each with capacity ∞. Figure 3.7 summarizes this construction. It is easy to see that any finite-capacity (r, s)-cut $\delta'(R)$ in G' will be such that $R = \{r\} \cup A$, where A is a closure of G. (In particular, a minimum-capacity cut will have this property.) Any closure $A \subseteq V$ determines an (r,s)-cut $\delta'(A \cup \{r\})$ having capacity $\Sigma(b_v : v \notin A,\ b_v > 0) - \Sigma(b_v : v \in A,\ b_v < 0)$. By adding $\Sigma(b_v : v \in A,\ b_v \geq 0)$ to both terms, this can be rewritten as

$$\Sigma(b_v : v \in V,\ b_v \geq 0) - b(A).$$

Since the first term of the latter expression is constant, that is, does not depend on A, this expression is minimized when $b(A)$ is maximized. In summary, we simply find a minimum cut $\delta'(A \cup \{r\})$ of G', and A is a maximum-weight closure.

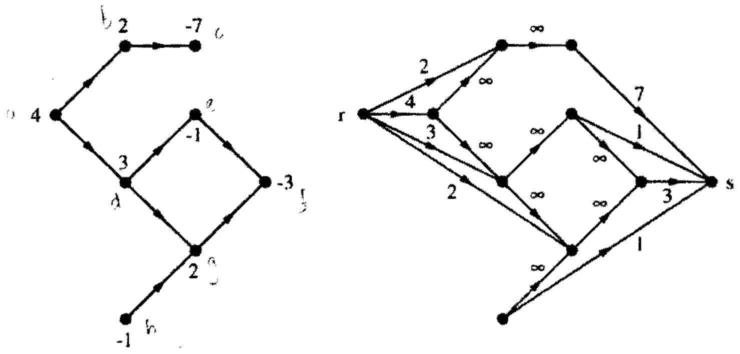

Figure 3.7. Flow network for the optimal closure problem

Elimination of Sports Teams

Sports writers are fond of using the term "mathematically eliminated" to refer to a team that cannot possibly finish the season in first place. More formally, let us say that the Buzzards are *eliminated* if, no matter what the outcome of the remaining games, they cannot finish with the most wins (even in a tie). For convenience, we assume that there are no tie games.

The simplest situation in which the Buzzards are eliminated (and the only one of which sportswriters seem to be aware!) is illustrated in Table 3.1. In this case even if the Buzzards win all their remaining games, they will have fewer wins than the Anteaters already have. Notice that in this case we can see that the Buzzards are eliminated, no matter what pairs of teams are involved in the remaining games. A more interesting situation involves the data of Table 3.2. (The additional columns of the table indicate the number of remaining games against various opponents. Notice that there may be other teams that we have not included in the table.) Here it is possible for the Buzzards to finish with as many wins as the Anteaters, if the Anteaters lose all of their remaining games and the Buzzards win all of theirs. However, in this case the Banana Slugs must finish with more wins than the Buzzards, since they win their remaining games against the Anteaters. Thus, although we cannot be sure which team will finish first, we can see that the Buzzards are eliminated.

Team	Wins	To Play
Anteaters	33	8
Buzzards	28	4

Table 3.1. A simple example of elimination

Team	Wins	To Play	A.	B.	B.S.	R.
Anteaters	33	8	–	1	6	1
Buzzards	29	4	1	–	0	3
Banana Slugs	28	8	6	0	–	1
Fighting Ducks	27	5	1	3	1	–

Table 3.2. A second example of elimination

We will show that in any situation in which a team is eliminated, there is a simple reason, as in the previous examples. Let T denote the set of teams other than the Buzzards. For each $i \in T$, let w_i denote the number of wins for team i, and for $i, j \in T$ with $i \neq j$, let r_{ij} denote the number of remaining games between teams i and j. We will need also a notation for the set $\{\{i,j\} : \{i,j\} \subseteq T,\ i \neq j,\ r_{ij} > 0\}$; we denote it by P. Finally, let M denote the number of wins for the Buzzards at the end of the season if they win all their remaining games.

Let A be a subset of T. Since every game between two teams in A is won by one of them, the total number of wins for teams in A at the end of the season is at least $w(A) + \sum(r_{ij} : \ \{i,j\} \subseteq A,\ \{i,j\} \in P)$. If this number is bigger than $M|A|$, then the average number of wins of teams in A at the end of the season is more than M. But M is the most wins that the Buzzards can hope to have, so at least one team in A will finish with more wins than the Buzzards. In summary, the Buzzards are eliminated if there exists $A \subseteq T$ such that

$$w(A) + \sum(r_{ij} : \ \{i,j\} \subseteq A,\ \{i,j\} \in P) > M|A|. \tag{3.5}$$

This criterion for elimination is general enough to include the arguments used n the examples above. In the first example, A consists of the Anteaters alone, n the second, of the Anteaters and the Banana Slugs.

We will show that if the Buzzards have been eliminated there is a set A with roperty (3.5). To prove this we make use of the fact that, if the Buzzards are *not* eliminated, there is a set of possible outcomes of the remaining games so that the Buzzards finish with the most wins. Let y_{ij} denote the (unknown) number of wins for team i over team j in the remaining games between them. Then if the Buzzards are not eliminated, there must exist values for the y_{ij} satisfying

$$\begin{gathered} y_{ij} + y_{ji} = r_{ij}, \text{ for all } \{i,j\} \in P \\ w_i + \sum(y_{ij} : j \in T,\ j \neq i) \leq M, \text{ for all } i \in T \\ y_{ij} \geq 0, \text{ for all } \{i,j\} \in P \\ y_{ij} \text{ integral, for all } \{i,j\} \in P. \end{gathered} \tag{3.6}$$

We create a flow network $G = (V, E)$ as follows. $V = T \cup P \cup \{r, s\}$. For each $i \in T$, there is an arc (r, i) having capacity $M - w_i$. For each $i \in T$ and $j \in T$ with $\{i,j\} \in P$, there are arcs $(i, \{i,j\})$ and $(j, \{i,j\})$ with capacity ∞, and there is an arc $(\{i,j\}, s)$ with capacity r_{ij}. The network arising from the data of Table 3.2 is illustrated in Figure 3.8. Now suppose that in this network there is an integral feasible (r,s)-flow of value $\sum(r_{ij} : \{i,j\} \in P)$. Then if we put y_{ij} equal to the flow on arc $(i, \{i,j\})$, we get a solution to (3.6). Conversely, a solution to (3.6) yields such an integral feasible flow, by assigning flow y_{ij} to arc $(i, \{i,j\})$ and then defining the flows on arcs incident to the source and sink to satisfy conservation of flow.

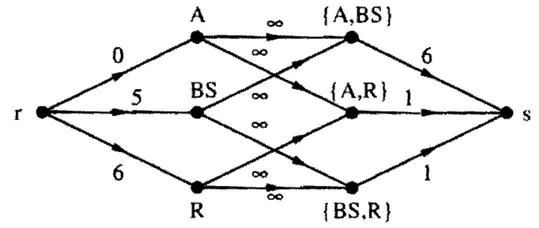

Figure 3.8. Flow network for the elimination problem

It follows that we can determine whether the Buzzards are eliminated by solving a maximum (integral) flow problem. If they are not eliminated, a maximum flow will determine a set of outcomes for the remaining games in which the Buzzards finish first. Now we show that if the Buzzards are eliminated, then a minimum cut will determine a set A satisfying (3.5). Let $\delta(S)$ be a minimum (r,s)-cut. By the Max-Flow Min-Cut Theorem its capacity is less than $\sum(r_{ij} : \{i,j\} \in P)$. Let $A = T \backslash S$. We claim that $S = \{r\} \cup (T \backslash A) \cup \{\{i,j\} \in P : i \text{ or } j \notin A\}$. First, if i or j is not in A but $\{i,j\} \notin S$, then $\delta(S)$ has capacity ∞. Second, if $\{i,j\} \in S$ and $i, j \in A$, then deleting $\{i,j\}$ from S decreases the capacity of $\delta(S)$ by r_{ij}. In either case $\delta(S)$ is not a minimum cut, a contradiction. Now it is easy to compute th capacity of $\delta(S)$. It is

$$M|A| - w(A) + \sum(r_{ij} : \ \{i,j\} \in P, \ \{i,j\} \not\subseteq A).$$

This is less than $\sum(r_{ij} : \ \{i,j\} \in P)$ if and only if A satisfies (3.5), and we are done.

Flow Feasibility Problems

An example of what might be called a flow feasibility problem is that of deciding, given (G, u, r, s, k), whether there exists a feasible flow from r to s of value at least k. Of course, this is essentially a restatement of the maximum flow problem. We have solved this problem in two senses. First, we have given a good algorithm that will construct such a flow if one exists. Second, we have given a good characterization for its existence. (Such a flow exists if and only if every (r,s)-cut has capacity at least k.) A number of useful and interesting flow feasibility problems will be solved here. Although some of them appear to be significantly more general, all of them can be reduced to the above problem, and solved (in the two senses) by the maximum flow algorithm and the Max-Flow Min-Cut Theorem.

As a first example, consider the problem of deciding whether the transportation model mentioned at the beginning of this chapter has a feasible solution. This problem can be restated as: Given a bipartite graph $G = (V, E)$ with bipartition $\{P, Q\}$ and vectors $a \in \mathbf{Z}_+^P$, $b \in \mathbf{Z}_+^Q$, to find $x \in \mathbf{R}^E$ satisfying

$$\begin{aligned} \sum(x_{pq} : q \in Q, \ pq \in E) &\leq a_p, \ \text{for all } p \in P \\ \sum(x_{pq} : p \in P, \ pq \in E) &= b_q, \ \text{for all } q \in Q \\ x_{pq} &\geq 0, \ \text{for all } pq \in E \\ x_{pq} & \text{ integral, for all } pq \in E. \end{aligned} \tag{3.7}$$

The method for converting this to a flow problem is similar to the one used for bipartite matching. Form digraph G' with $V' = V \cup \{r, s\}$. Each $pq \in E$ gives rise to an arc pq of G' with $u_{pq} = \infty$. For each $p \in P$ there is an arc rp with $u_{rp} = a_p$. For each $q \in Q$, there is an arc qs with $u_{qs} = b_q$. This construction is illustrated in Figure 3.9. It is easy to see that there exists $x \in \mathbf{Z}^E$ satisfying (3.7) if and only if there is an integral feasible flow in G' from r to s of value $\Sigma(b_q : q \in Q)$. Thus we can find such a solution, if one exists, with a maximum flow algorithm. The Max-Flow Min-Cut Theorem tells us that (3.7) has a solution if and only if every (r,s)-cut of G' has capacity at least $\Sigma(b_q : q \in Q)$. The capacity of a cut $\delta'(A \cup B \cup \{r\})$ where $A \subseteq P$, $B \subseteq Q$, is $\Sigma(a_i : i \in P \backslash A) + \Sigma(b_j : j \in B)$, assuming there is no arc pq for $p \in A$, $q \in Q \backslash B$. (Otherwise the cut capacity is ∞.) This capacity is at least $\Sigma(b_j : j \in Q)$ if and only if $\Sigma(a_i : i \in P \backslash A) \geq \Sigma(b_j : j \in Q \backslash B)$. It is clear that it is enough to check this condition for the sets A such that