10 Maximum Flows and Perfect Matchings

10.1 The Maximum Flow Problem

Consider a flow network (V, E) with a single source 1, a single sink n, and finite capacities $\overline{m}_{ij} = C_{ij}$ for all $(i, j) \in E$. We will also assume for convenience that $\underline{m}_{ij} = 0$ for all $(i, j) \in E$. The *maximum flow problem* then asks for the maximum amount of flow that can be sent from vertex 1 to vertex n, i.e., the goal is to

$$\begin{aligned} &\text{maximize} \quad \delta \\ &\text{subject to} \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} \delta & \text{if } i = 1 \\ -\delta & \text{if } i = n \\ 0 & \text{otherwise} \end{cases} \\ &\qquad 0 \leqslant x_{ij} \leqslant C_{ij} \quad \text{for all } (i,j) \in E. \end{aligned} \tag{10.1}$$

To see that this is again a special case of the minimum cost flow problem, set $c_{ij} = 0$ for all $(i, j) \in E$, and add an additional edge $(n, 1)$ with infinite capacity and cost $c_{n1} = -1$. Since the new edge $(n, 1)$ has infinite capacity, any feasible flow of the original network is also feasible for the new network. Cost is clearly minimized by maximizing the flow across the edge $(n, 1)$, which by the flow conservation constraints for vertices 1 and n maximizes flow through the original network. This kind of problem is known as a *circulation problem*, because there are no sources or sinks but flow merely circulates in the network.

10.2 The Max-Flow Min-Cut Theorem

Consider a flow network $G = (V, E)$ with capacities C_{ij} for all $(i, j) \in E$. A *cut* of G is a partition of V into two sets, and the capacity of a cut is defined as the sum of capacities of all edges across the partition. Formally, for $S \subseteq V$, the capacity of the cut $(S, V \setminus S)$ is given by

$$C(S) = \sum_{(i,j)\in E\cap(S\times(V\setminus S))} C_{ij}.$$

Assume that x is a feasible flow vector that sends δ units of flow from vertex 1 to vertex n. It is easy to see that δ is bounded from above by the capacity of any cut S with $1 \in S$ and $n \in V \setminus S$. Indeed, for $X, Y \subseteq V$, let

$$f(X, Y) = \sum_{(i,j)\in E\cap(X\times Y)} x_{ij}.$$

Then, for any $S \subseteq V$ with $1 \in S$ and $n \in V \setminus S$,

$$\begin{aligned}
\delta &= \sum_{i \in S} \left(\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} \right) \\
&= f(S,V) - f(V,S) \qquad (10.2) \\
&= f(S,S) + f(S, V \setminus S) - f(V \setminus S, S) - f(S,S) \\
&= f(S, V \setminus S) - f(V \setminus S, S) \leqslant f(S, V \setminus S) \leqslant C(S).
\end{aligned}$$

The following result states that this upper bound in fact tight, i.e., that there exists a flow of size equal to the minimum capacity of a cut that separates vertex 1 from vertex n.

THEOREM 10.1 (Max-flow min-cut theorem). *Let* δ *be the optimal solution of* (10.1) *for a network* (V, E) *with capacities* C_{ij} *for all* $(i,j) \in E$. *Then,*

$$\delta = \min\{ C(S) : S \subseteq V, 1 \in S, n \in V \setminus S \}.$$

Proof. It remains to be shown that there exists a cut that separates vertex 1 from vertex n and has capacity equal to δ. Consider a feasible flow vector x. A path $P = v_0, v_1, \dots, v_k$ is called an *augmenting path* for x if $x_{v_{i-1}v_i} < C_{v_{i-1}v_i}$ or $x_{v_i v_{i-1}} > 0$ for every $i = 1, \dots, k$. If there exists an augmenting path from vertex 1 to vertex n, then we can push flow along the path, by increasing the flow on every forward edge and decreasing the flow on every backward edge along the path by the same amount, such that all constraints remain satisfied and the amount of flow from 1 to n increases.

Now assume that x is optimal, and let

$$S = \{1\} \cup \{ i \in V : \text{there exists an augmenting path for } x \text{ from } 1 \text{ to } i \}.$$

By optimality of x, $n \in V \setminus S$. Moreover,

$$\delta = f(S, V \setminus S) - f(V \setminus S, S) = f(S, V \setminus S) = C(S).$$

The first equality holds by (10.2). The second equality holds because $x_{ij} = 0$ for every $(i,j) \in E \cap ((V \setminus S) \times S)$. The third equality holds because $x_{ij} = C_{ij}$ for every $(i,j) \in E \cap (S \times (N \setminus S))$. □

10.3 The Ford-Fulkerson Algorithm

The *Ford-Fulkerson algorithm* attempts to find a maximum flow by repeatedly pushing flow along an augmenting path, until such a path can no longer be found:

1. Start with a feasible flow vector x.
2. If there is no augmenting path from 1 to n, then stop.
3. Otherwise pick some augmenting path from 1 to n, and push a maximum amount of flow along this path without violating any constraints. Then go to Step 2.

Assume that all capacities are integral and that we start with an integral flow vector, e.g., the flow vector x such that $x_{ij} = 0$ for all $(i, j) \in E$. It is then not hard to see that the flow vector always remains integral and overall flow increases by at least one unit in each iteration. The algorithm is therefore guaranteed to find a maximum flow after a finite number of iterations. It can in fact be shown that $O(|E| \cdot |V|)$ iterations suffice if only augmenting paths with a minimum number of edges are used. Such an augmenting path can for example be found using breadth-first search, which requires $O(|E|)$ steps and leads to an overall running time of $O(|E|^2 \cdot |V|)$.

10.4 Max-Flow Min-Cut from Strong Duality

Consider the following formulation of the maximum flow problem as a minimum cost flow problem, which we have already discussed above:

$$\begin{aligned} \text{minimize} \quad & -x_{n1} \\ \text{subject to} \quad & \sum_{j:(i,j)\in E'} x_{ij} - \sum_{j:(j,i)\in E'} x_{ji} = 0 \quad \text{for all } i \in V \\ & 0 \leqslant x_{ij} \leqslant C_{ij} \quad \text{for all } (i,j) \in E \\ & x_{n1} \geqslant 0, \end{aligned}$$

where $E' = E \cup \{(n, 1)\}$. The Lagrangian (8.1) becomes

$$L(x, \lambda) = (-1 - \lambda_n + \lambda_1)x_{n1} - \sum_{(i,j)\in E} (\lambda_i - \lambda_j)x_{ij},$$

which has a bounded minimum where $x_{n1} > 0$ only if $\lambda_1 - \lambda_n = 1$. We know from the general case that one of the dual variables can be set arbitrarily, so we let $\lambda_1 = 1$ and obtain $\lambda_n = 0$. For a fixed λ, $L(x, \lambda)$ is minimized by setting $x_{ij} = 0$ whenever $\lambda_i - \lambda_j < 0$ and $x_{ij} = C_{ij}$ whenever $\lambda_i - \lambda_j > 0$, and thus

$$g(\lambda) = \inf_x L(x, \lambda) = -\sum_{(i,j)\in E} \max(\lambda_i - \lambda_j, 0)C_{ij}.$$

By introducing new variables $d_{ij} \geqslant \max(\lambda_i - \lambda_j, 0)$ for $(i, j) \in E$, we obtain

$$g(\lambda) \geqslant -\sum_{(i,j)\in E} d_{ij}C_{ij},$$

with equality if $d_{ij} = \max(\lambda_i - \lambda_j, 0)$. We can thus maximize $g(\lambda)$ by minimizing $\sum_{(i,j)\in E} d_{ij}C_{ij}$ subject to $d_{ij} \geqslant \lambda_i - \lambda_j$ and $d_{ij} \geqslant 0$ for all $(i, j) \in E$, and obtain the following dual of (10.1):

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} d_{ij}C_{ij} \\ \text{subject to} \quad & d_{ij} - \lambda_i + \lambda_j \geqslant 0 \quad \text{for all } (i,j) \in E \\ & d_{ij} \geqslant 0 \quad \text{for all } (i,j) \in E \\ & \lambda_1 = 1, \quad \lambda_n = 0. \end{aligned}$$

It can be shown that this dual has an optimal solution in which $\lambda_i \in \{0, 1\}$ for all $i \in V$. By the complementary slackness conditions, the set $S = \{i \in V : \lambda_i = 1\}$ must then be a minimum cut, and the max-flow min-cut theorem follows from strong duality.

10.5 The Bipartite Matching Problem

A *matching* of a graph (V, E) is a set of edges that do not share any vertices, i.e., a set $M \subseteq E$ such for all $(s, t), (u, v) \in M$, $u \neq s \neq v$ and $u \neq t \neq v$. Matching M is called perfect if it covers every vertex, i.e., if $|M| = |V|/2$.

A graph is k-regular if every vertex has degree k. Using maximum flows it is easy to show that every k-regular bipartite graph, for $k \geqslant 1$, has a perfect matching. For this, consider a k-regular bipartite graph $(L \uplus R, E)$, orient all edges from L to R, and add two new vertices s and t and new edges (s, i) and (j, t) for every $i \in L$ and $j \in R$. Finally set the capacity of every new edge to 1, and that of every original edge to infinity. We can now send $|L|$ units of flow from s to t by setting the flow to 1 for every new edge and to $1/k$ for every original edge. By Theorem 8.2, there must exist an integral solution with the same value, and it is easy to see that such a solution corresponds to a perfect matching.

This result is a special case of a well-known characterization of the bipartite graphs that have a perfect matching. It should not come as a surprise that this characterization can be obtained from the max-flow min-cut theorem as well.

THEOREM 10.2 (Hall's Theorem). *A bipartite graph* $G = (L \uplus R, E)$ *with* $|L| = |R|$ *has a perfect matching if and only if* $|N(X)| \geqslant |X|$ *for every* $X \subseteq L$, *where* $N(X) = \{j \in R : i \in X, (i, j) \in E\}$.

Proof. The direction from left to right is obvious: in a perfect matching, every vertex in X is matched to a different vertex in $N(X)$.

For the direction from right to left, assume that G does not have a perfect matching and again consider the graph with additional vertices s and t described above. The maximum flow from s to t is this graph must be smaller than $|L|$, so by the max-flow min-cut theorem there has to exist a cut $S \subseteq L \uplus R \cup \{s\}$ with $s \in S$ and $C(S) < |L|$. Let $L_S = L \cap S$, $R_S = R \cap S$, and $L_T = L \setminus S$. Since $C(S)$ is finite, $i \in S$ implies that $j \in S$ for every $(i, j) \in E$. On the one hand, this means that $N(L_S) \subseteq R_S$. On the other, the capacity of the cut must thus come precisely from the edges in $\{s\} \times L_T$ and $R_S \times \{t\}$. Each of these edges has capacity 1, so $C(S) = |L_T| + |R_S|$, and we obtain

$$|N(L_S)| \leqslant |R_S| = C(S) - |L_T| < |L| - |L_T| = |L_S|.$$

□