18.438 Advanced Combinatorial Optimization

Lecture 22

Lecturer: Michel X. Goemans *Scribe: Alantha Newman (2004), Ankur Moitra (2009)*

1 MultiFlows and Disjoint Paths

Here we will survey a number of variants of disjoint path problems, and give reductions between these different variants. We will also consider fractional relaxations of these problems, and in the special case of two source-sink pairs, we will prove a max-flow min-cut relation and in fact in this case, this flow can even be realized half-integrally.

1.1 Notation

We will consider disjoint path problems in both directed, and undirected graphs. Additionally, we will consider both edge-disjoint and vertex-disjoint paths problems. In the case of a digraph, we will let $D = (V, A)$ be the digraph and we will denote the set of arcs by A. In the case of an undirected graph, we will use $G = (V, E)$ to denote the set of edges by E. Let $s_1, t_1, s_2, t_2, \ldots s_k, t_k \in V$ be terminals. Our goal is to find disjoint paths between s_i and t_i for each $i, 1 \leq i \leq k$. Again, the notion of path and the notion of disjoint depends on the variant of the disjoint path problem that we are considering.

We can in fact consider a more general problem, in which each pair of terminals s_i, t_i is also given an integer demand d_i, and our goal is to connect *every* source-sink pair s_i, t_i by d_i paths, and all paths connecting any source-sink pair must be disjoint. We can also generalize this question further, by allowing capacities on edges in the case of edge disjoint path problems. Here we are also given a capacity function $c : A \to \mathbb{Z}_+$ or $c : E \to \mathbb{Z}_+$ and our goal is again to connect each pair of terminals s_i, t_i by d_i paths, subject to the constraints that each (arc or) edge e is used at most $c(e)$ times. So we can interpret the capacity on an (arc or) edge as a bound on the number of paths that are allowed to traverse this edge. Or we can even relax this further and consider a collection of (possibly non-integral) flows, which collectively satisfy the capacity constraints and such that the ith flow has d_i units between s_i and t_i.

In the case of just a single source-sink pair s, t and integer demand d, this problem is well-understood: Let f be the maximum number of paths connecting s, t (subject to capacity constraints) in $G = (V, E)$ (or in a digraph $D = (V, A)$). Then f is equal to the capacity of the minimum cut separating s from t.

But for many source-sink pairs, the minimum cut is no longer equal to the maximum number of paths (subject to capacity constraints). In fact, each of the variants of the disjoint paths problem mentioned above is NP-hard, but the difficulty of these problems is quite different.

1.2 Directed and Undirected Edge Disjoint Paths

The undirected and directed edge disjoint path problems are drastically different for a fixed number of source-sink pairs:

Undirected Edge Disjoint Paths. In the case of fixed k (i.e. there are k source-sink pairs), there is a polynomial time algorithm to decide if there are edge disjoint paths connecting each s_i, t_i pair. This result follows from the Graph Minor Project of Robertson and Seymour (1995), and is very involved.

Yet for super-constant k, the undirected edge disjoint paths problem is NP-complete.

Directed Arc Disjoint Paths. Even for $k = 2$ it is NP-complete to determine if there are arc-disjoint paths P_1, P_2 connecting s_1, t_1 and s_2, t_2. In fact, we can even choose $s_1 = t_2$ and $s_2 = t_1$ and this problem still remains NP-complete.

This drastic difference in the range for which these problems are hard, leads to very different inapproximability results. For the problems of maximizing the number of arc-disjoint paths, in the case of a directed graph, it is hard to determine the maximum number of arc-disjoint paths within any $\Omega(m^{\frac{1}{2}-\epsilon})$ for m arcs, and for any $\epsilon > 0$. Yet for the case of maximizing the number of edge disjoint paths in an undirected graph, the best known hardness results are only poly-logarithmic. And in either case, the best known approximation ratio is $O(\sqrt{m})$ (or actually $O(\sqrt{n})$ for undirected graphs on n vertices).

1.3 Reductions Between Variants

Here we consider all four problems, (undirected or directed) (edge or vertex) disjoint paths. We first note that the arc-disjoint path problem, and the vertex-disjoint path problem in a directed graph are equivalent - i.e. we can give a polynomial time reduction in either direction. Given a vertex-disjoint path problem in a directed graph, we can perform the reduction in Figure 1 and get a are-disjoint path problem that is equivalent.

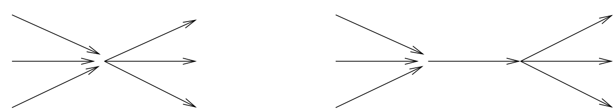

Figure 1: Each vertex undergoes the illustrated transformation.

In the case of an arc-disjoint path problem, we can perform a similar transformation given in Figure 2.

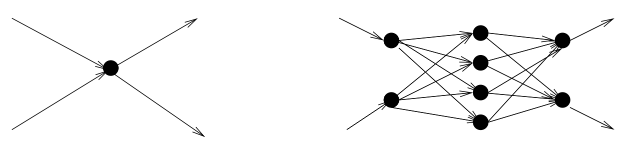

Figure 2: Each vertex undergoes the illustrated transformation.

Next we consider reductions from undirected variants of the disjoint path problem, and prove that both the edge disjoint path problem and the vertex-disjoint path problem (in undirected graphs) can be reduced to either directed path problem above.

We can also reduce the edge disjoint path problem in undirected graphs to the arc-disjoint path problem, and this reduction is given in Figure 3. This also implies that we can reduce the edge disjoint path problem in undirected graphs to the directed vertex-disjoint path problem.

Figure 3: Each edge in the original undirected graph is replaced by the above gadget.

For the case of the vertex disjoint path problem in undirected graphs, we can just replace any edge (u, v) by two arcs (u, v) and (v, u). So this problem can also be reduced to either the arc-disjoint path problem or the vertex-disjoint path problem in directed graphs.

Lastly, the edge disjoint path problem in undirected graphs can be reduced to the vertex-disjoint path problem in undirected graphs by taking the line-graph of G. A path in the line-graph of G will also correspond to a path in the original graph G, but vertices in the line-graph correspond to edges in the original graph, so paths will be edge-disjoint in G iff the corresponding paths are vertex-disjoint in the line graph of G.

1.4 Fractional Relaxations

We focus on edge disjoint paths in undirected graphs. When $k = 1$, flow is easy. We can find integer flow using the max-flow min-cut theorem. In general, we can give a fractional relaxation (that can be solved because it can be written as a linear program):

Let $\mathcal{P}_i$ be set of all paths between s_i and t_i. We have a variable x_p for every such path $p \in \mathcal{P}_i$. We have the following primal LP:

$$\begin{array}{rcl} \max\ 0 \cdot x & & \\ \sum_{p \in \mathcal{P}_i} x_p & = & d_i \\ \sum_{i: p \in \mathcal{P}_i, e \in p} x_p & \leq & c(e) \\ x_p & \geq & 0. \end{array}$$

What does dual mean in this case? We use variables $\ell(e)$ for each edge $e \in E$, and variables b_i for $i = 1, \dots, k$.

$$\begin{array}{rcl} \min\ \sum_{e \in E} c(e)\ell(e) - \sum_{i=1}^{k} b_i d_i & & \\ \sum_{e \in p} \ell(e) - b_i & \geq & 0 \quad \forall p \in \mathcal{P}_i\ i = 1, \dots k \\ \ell(e) & \geq & 0. \end{array} \tag{1}$$

To make the term $(-\sum_{i=1}^{k} b_i d_i)$ small, we should make b_i as large as possible. Fix the edge function $\ell : E \to \mathcal{Q}$. Then b_i is the (minimum) $\text{dist}_\ell(s_i, t_i)$. The objective function of the dual (1) can be rewritten:

$$\sum_{e \in E} c(e)\ell(e) - \sum_{i=1}^{k} d_i\ dist_\ell(s_i, t_i).$$

If the primal LP is feasible, then there is no solution for the dual LP with a negative objective value. So there exists a fractional multiflow if and only if $\forall \ell(e) \geq 0, e \in E$, the following holds:

$$\sum_{e \in E} c(e)\ell(e) \geq \sum_{i=1}^{k} d_i\ dist_\ell(s_i, t_i). \tag{2}$$

Duality shows that this is a necessary and sufficient for the existence of a fractional multiflow.

2 Integer Multiflows

In general, the problem of determining when there is an integer multiflow is NP-complete. However, there are special conditions that imply the existence of an integer multiflow in certain classes of graphs.

Let R be a set of edges:

$$R = \{(s_i, t_i) : i = 1, \dots k\}. \tag{3}$$

The set of edges in E outgoing from vertex set U is denoted by $\delta_E(U)$ and the set of edges in R outgoing from vertex set U is denoted by $\delta_R(U)$. A necessary condition for the existence of a multiflow (and thus of an integer multiflow) is the cut condition:

$$c(\delta_E(U)) \geq d(\delta_R(U)), \quad \forall U \subset V.$$

In general, the cut condition is not sufficient to guarantee the existence of an integer multiflow (or fractional multiflow) in a graph. However, in some cases of the multiflow problem, the cut condition is sufficient for the existence of a fractional multiflow. Furthermore, there are several cases known where the cut condition implies the existence of an integer multiflow when the *Euler condition* is satisfied:

$$c(\delta_E(v)) + d(\delta_R(v)) \text{ is even, for each vertex } v.$$

For example, when $k = 2$, we have the following implications:

(i) Cut condition $\Rightarrow$ fractional multiflow.

(ii) Cut condition and integer capacities $\Rightarrow$ half-integral multiflow.

(iii) Cut condition, integer capacities, and Euler condition $\Rightarrow$ integral multiflow.

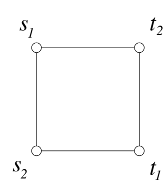

Figure 4: When the capacity of each edge in this graph is 2 and $d_1, d_2 = 1$, the Euler condition is not satisfied. There exists a half-integral multiflow, but no integral multiflow.

The first proof of (i) and (ii) for the case when $k = 2$ was is due to Hu. The proof of (iii) is due to Rothschild and Winston. Note that (iii) implies (i) and (ii). For example, Consider the graph in Figure 4, let $d_1 = 1, d_2 = 1$. Let the capacity of each edge be 2. Note that the cut condition is satisfied but the Euler condition is not. However, suppose we double every capacity and demand, then the Euler condition is satisfied. We can convert an integer solution for this latter problem to a half-integral solution for the original problem.

Some "good" cases in which conditions (i), (ii) and (iii) are satisfied are:

1. If there are two commodities, i.e. $k = 2$, then cut condition and Euler condition are sufficient for integer multiflow.

2. $G + D$ has no K_5 minor, (a special case is when $G + D$ is planar) where D is the demand graph, $D = (V, R)$ (see (3)).

3. $|\{(s_1, t_1), \ldots, (s_k, t_k)\}| \leq 4$.

4. G is planar and all (s_i, t_i) are on boundary of outside face. (Note that this does not imply case 2 because the demand graph could be a clique)

5. If there are 2 faces F_1, F_2 and for each i, (s_i, t_i) are both on the boundary of the same face F_j for $j \in \{1, 2\}$.

3 Two-Commodity Flows

Theorem 1 (Rothschild and Whinston) *$G = (V, E)$ is an undirected graph such that $c(e) \in \mathbb{Z}_+$ for $e \in E$. Terminals s_1, t_1, s_2, t_2 are in V, and demands d_1, d_2 are positive integers. Additionally, the Euler condition is satisfied for G. Then G has an integer two-commodity flow if and only if the cut condition is satisfied.*

Proof: Our goal is to find flows from s_1 to t_1 and from s_2 to t_2 with values d_1 and d_2, respectively. We will show that if the cut condition is satisfied on G, then we can find such flows.

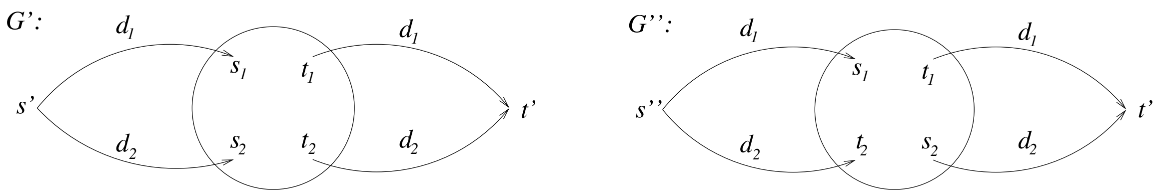

Figure 5: The graphs G' and G'' are constructed based on the given graph G.

First, based on the graph G, construct the graph G' as shown in Figure 5. Note that the edges out of s' are the only directed edges in the graph G', and all other edges are inherited from G and hence undirected. Let the edges (s', s_1) and (t_1, t') in G' have capacity d_1 and the edges (s', s_2) and (t_2, t') in G' have capacity d_2. By the max-flow min-cut theorem, we can find an integer s'-t' flow g with value $d_1 + d_2$, since the min-cut of G' has value $d_1 + d_2$. Note that this s'-t' flow does not necessarily give a two-commodity flow for the original problem (since some of the flow going through s_1 may end up in t_2).

Since the Euler condition is satisfied, we will prove that we can assume that $g(e) \equiv c(e) \mod 2$. To show this, first notice that the Euler condition implies that the total capacity incident to any vertex (except s' or t') of G' is even. Furthermore, any integral flow will use up an even amount of capacity incident to any vertex. Now consider all the edges $e \in E$ such that $g(e) \not\equiv c(e) \mod 2$. Since it is the case that $\sum_{e \in \delta(v)} (g(e) - c(e)) = 0 \mod 2$, it follows that an even number of edges adjacent to vertex v have $g(e) \not\equiv c(e) \mod 2$. Thus, the edges such that $g(e) \not\equiv c(e) \mod 2$ make up an Eulerian graph (and do not contain the arcs incident to s' and t' that we added to G to make up G', as these arcs are saturated). We can decompose this Eulerian graph into cycles, and push push one unit of flow across all these cycles (either increasing or decreasing the flow by one unit along it depending on the orientation), changing the parity of $g(e)$ for each such edge. Thus, for all edges $e \in E$, we have that $g(e) \equiv c(e) \mod 2$.

For G'', we have the same argument. Thus, we find an integer flow h in G'' with value $d_1 + d_2$ such that $h(e) = c(e) \mod 2, \forall e \in E$. Thus, for all edges $e \in E$, $h(e) = g(e) \mod 2$. We arbitrarily orient the edges of E to obtain A. So for all $a \in A, h(a) \equiv g(a) \mod 2$.

Now we define two flows on the graph G:

$$f_1(a) = \frac{1}{2}[g(a) + h(a)]$$
$$f_2(a) = \frac{1}{2}[g(a) - h(a)].$$

The following properties are true for the flows f_1 and f_2:

1. $f_1(a), f_2(a)$ are integer flows (since $g(a)$ and $h(a)$ have the same parity).
2. $|f_1(a)| + |f_2(a)| = \frac{1}{2}|g(a) + h(a)| + \frac{1}{2}|g(a) - h(a)| \leq \max(|g(a)|, |h(a)|) \leq c(a)$.
3. f_1 is d_1 units of flow from s_1 to t_1 and f_2 is d_2 units of flow from s_2 to t_2.

The last property holds because we can show that $f_1(\delta^+(s_1)) - f_1(\delta^-(s_1)) = d_1$ and $f_1(\delta^-(t_1)) - f_1(\delta^+(t_1)) = d_1$. By conservation of flow, if we consider the vertex s_1 in G, we have:

$$g(\delta^+(s_1)) - g(\delta^-(s_1)) = d_1 \tag{4}$$
$$h(\delta^+(s_1)) - h(\delta^-(s_1)) = d_1 \tag{5}$$

Equations (4) and (5) imply $f_1(\delta^+(s_1)) - f_1(\delta^-(s_1)) = d_1$.

$$g(\delta^-(t_1)) - g(\delta^+(t_1)) = d_1 \tag{6}$$
$$h(\delta^-(t_1)) - h(\delta^+(t_1)) = d_1. \tag{7}$$

Equations (6) and (7) imply $f_1(\delta^-(t_1)) - f_1(\delta^+(t_1)) = d_1$. Similarly, we can show that the last property holds for flow f_2. If we consider vertices s_2 and t_2 in G, we have:

$$g(\delta^+(s_2)) - g(\delta^-(s_2)) = d_2 \tag{8}$$
$$h(\delta^-(s_2)) - h(\delta^+(s_2)) = d_2 \tag{9}$$
$$g(\delta^-(t_2)) - g(\delta^+(t_2)) = d_2 \tag{10}$$
$$h(\delta^+(t_2)) - h(\delta^-(t_2)) = d_2. \tag{11}$$

Equations (8) and (9) imply $f_2(\delta^+(s_2)) - f_2(\delta^-(s_2)) = d_2$ and equations (10) and (11) imply $f_2(\delta^-(t_2)) - f_2(\delta^+(s_2)) = d_2$. □

As a final note, consider the problem of maximizing the sum of the flow between s_1 and t_1 and between s_2 and t_2. This is the *max biflow problem*. A *bicut* is a cut separating s_1 from t_1 and s_2 from t_2, thus it is either a cut separating s_1, s_2 from t_1, t_2 or a cut separating s_1, t_2 from s_2, t_1. One can show that the following theorem follows from Theorem 1.

Theorem 2 *The maximum biflow equals the minimum bicut.*