CS 598CSC: Combinatorial Optimization	**Lecture date: 02 March, 2010**
Instructor: Chandra Chekuri	**Scribe: Ben Raichel**

1 T-joins and Applications

This material is based on [1] (Chapter 5), and also [2] (Chapter 29).

Edmonds was motivated to study T-joins by the Chinese postman problem which is the following.

Problem 1 *Let $G = (V, E)$ be an undirected graph and $c : E \to \mathbb{R}^+$ be non-negative edge weights on the edges. A Chinese postman tour is a walk that starts at some arbitrary vertex and returns to it after traversing each edge of E. Note that an edge may be traversed more than once. The goal is to find a postman tour of minimum total edge cost.*

Proposition 1 *If G is Eulerian then the optimal postman tour is an Eulerian tour of G and has cost equal to $\sum_{e \in E} c(e)$.*

Thus the interesting case is when G is not Eulerian. Let $T \subseteq V$ be the nodes with odd degree in G.

Fact 1 *$|T|$ is even.*

Consider a postman tour and say it visits an edge $x(e)$ times, where $x(e) \geq 1$ is an integer. Then, it is easy to see that the multigraph induced by placing $x(e)$ copies of e is in fact Eulerian. Conversely if $x(e) \geq 1$ and $x(e) \in \mathbb{Z}^+$ such that the graph is Eulerian, then it induces a postman tour of cost $\sum_{e \in E} c(e)x(e)$.

We observe that if $x(e) > 2$ then reducing $x(e)$ by 2 maintains feasibility. Thus $x(e) \in \{1, 2\}$ for each e in any minimal solution. If we consider the graph induced by $x(e)' = x(e) - 1$ we see that each node in T has odd degree and every other node has even degree. This motivates the definition of T-joins.

Definition 2 *Given a graph, $G = (V, E)$, and a set, $T \subseteq V$, a T-join is a subset $J \subseteq E$ such that in the graph (V, J), T is the set of nodes with odd degree.*

Proposition 3 *There is a T-join in G iff $|K \cap T|$ is even for each connected component K of G. In particular, if G is connected then there exists a T-join iff $|T|$ is even.*

Proof: Necessity is clear. For sufficiency, assume G is connected, otherwise we can work with each connected component separately. Let $T = \{v_1, v_2, \ldots, v_{2k}\}$. Let P_i be an arbitrary path joining v_i and v_{i+k}. Then the union of the paths $P_1, P_2, \ldots, P_k$ induces a multigraph in which the nodes in T are the only ones with odd degree. Let $x(e)$ be the number of copies of e in the above union. Then $x'(e) = x(e) \bmod 2$, is the desired T-join. (Note that the pairing of the vertices was arbitrary and hence any pairing would work.) □

Proposition 4 *J is a T-join iff J is the union of edge disjoint cycles and $\frac{1}{2}|T|$ paths connecting disjoint pairs of nodes in T.*

Proof: This is left as an exercise. □

1.1 Algorithm for Min-cost T-joins

Given $G = (V, E)$, $c : E \to \mathbb{R}$ and $T \subseteq V$, where $|T|$ even, we want to find the min-cost T-join. If all edge costs are non-negative then one can easily reduce the problem to a matching problem as follows. Assume without loss of generality that G is connected.

1. For each pair $u, v \in T$ let $w(uv)$ be the shortest path distance between u and v in G, with edge length given by c. Let P_{uv} be the shortest path between u and v.
2. Let H be the complete graph on T with edge weights $w(uv)$.
3. Compute a minimum weight perfect matching M in H.
4. Let $J = \{e \mid e \text{ occurs in an odd number of paths } P_{uv},\ uv \in M\}$. Output J.

Theorem 5 *There is a strongly polynomial time algorithm to compute a min-cost T-join in a graph, $G = (V, E)$ with $c \geq 0$.*

Proof Sketch. To see the correctness of this algorithm first note that it creates a T-join since it will return a collection of $\frac{1}{2}|T|$ disjoint paths, which by Proposition 4 is a T-join (Note the fourth step in the algorithm is required to handle zero cost edges, and is not necessary if $c > 0$). It can be seen that this T-join is of min-cost since the matching is of min-cost (and since, ignoring zero cost edges, the matching returned must correspond to disjoint paths in G). □

The interesting thing is that min-cost T-joins can be computed even when edge lengths can be negative. This has several non-trivial applications. We reduce the general case to the non-negative cost case by making the following observations.

Fact 2 *If A,B are two subsets of U then $|A \Delta B|$ is even iff $|A|$ and $|B|$ have the same parity, where we define $X \Delta Y$ as the symmetric difference of X and Y.*

Proposition 6 *Let J be a T-join and J' be a T'-join then $J \Delta J'$ is a $(T \Delta T')$-join.*

Proof: Verify using the above fact that each $v \in T \Delta T'$ has odd degree and every other node has even degree in $J \Delta J'$. □

Corollary 7 *If J' is a T'-join and $J \Delta J'$ is a $(T \Delta T')$-join then J is a T-join.*

Proof: Note that $(T \Delta T') \Delta T' = T$ and similarly $(J \Delta J') \Delta J' = J$. Hence the corollary is implied by application of the above proposition.

□

Given $G = (V, E)$ with $c : E \to \mathbb{R}$, let $N = \{e \in E \mid c(e) < 0\}$. Let T' be the set of nodes with odd degree in $G[N]$. Clearly N is a T'-join by definition. Let J'' be a $(T \Delta T')$-join in G with the costs on edges in N negated (i.e. $c(e) = |c(e)|, \forall e \in E$).

Claim 8 *$J = J'' \Delta N$ is a T-join, where $N = \{e \in E \mid c(e) < 0\}$, $T' = \{v \in G[N] \mid \delta_{G[N]}(v) \text{ is odd}\}$, and J'' is a $(T \Delta T')$-join.*

Proof: By the above corollary, since J'' is a $(T \Delta T')$-join and N is a T'-join, $J'' \Delta N$ is a $(T \Delta T') \Delta T' =$ T-join. □

Claim 9 *$c(J) = |c|(J'') + c(N)$, where $|c|(X) = \sum_{x \in X} |c(x)|$ and J, J'', and N are as defined above.*

Proof:

$$\begin{aligned} c(J) = c(J''\Delta N) &= c(J'' \setminus N) + c(N \setminus J'') \\ &= c(J'' \setminus N) - c(J'' \cap N) + c(J'' \cap N) + c(N \setminus J'') \\ &= c(J'' \setminus N) + |c|(J'' \cap N) + c(N) = |c|(J'') + c(N). \end{aligned}$$

□

Corollary 10 *$J = J''\Delta N$ is a min cost T-join in G iff J'' is a min cost $(T\Delta T')$-join in G with edge costs $|c|$, where $N = \{e \in E \mid c(e) < 0\}$, $T' = \{v \in G[N] \mid \delta_{G[N]}(v)$ is odd$\}$, and J'' is a $(T\Delta T')$-join.*

Proof Sketch. By using the last claim, necessity is clear since $c(N)$ is a constant and hence when $c(J)$ is minimized so is $|c|(J'')$. To use the same argument for sufficiency, one must argue that for any T-join, J, we have that $J = J''\Delta N$ for some $(T\Delta T')$-join, J''. □

The above corollary gives a natural algorithm to solve the general case by first reducing it to the non-negative case. In the algorithm below, let $c : E \to \mathbb{R}$, $|c| : E \to \mathbb{R}^+$ such that $|c|(e) = |c(e)|$, $G_{|c|}$ be the graph with the weight function $|c|$, $N = \{e \in E \mid c(e) < 0\}$, and $T' = \{v \in G[N] \mid \delta_{G[N]}(v)$ is odd$\}$.

1. Compute a $(T\Delta T')$-join , J'', on $G_{|c|}$ using the algorithm above for $c \geq 0$
2. Output $J = J''\Delta N$.

Theorem 11 *There is a strongly polynomial time algorithm for computing a min-cost T-join in a graph, even with negative costs on the edges.*

Proof: We know the above algorithm outputs a T-join by Claim 8. Since J'' was computed on $G_{|c|}$, which has non-negative edge weights, by the proof of Theorem 5, J'' is a min-cost T-join. Hence by Corollary 10 J is a min-cost T-join. □

1.2 Applications

1.2.1 Chinese Postman

We saw earlier that a min-cost postman tour in G is the union of E and a T-join where T is the set of odd degree nodes in G. Hence we can compute a min-cost postman tour.

1.2.2 Shortest Paths and Negative lengths

In directed graphs the well known Bellman-Ford and Floyd-Warshall algorithms can be used to check whether a given directed graph, $D = (V, A)$, has negative length cycles or not in $O(mn)$ and $O(n^3)$ time respectively. Moreover, if there is no negative length cycle then the shortest s-t path can be found in the same time. However, one cannot use directed graph algorithms for undirected graphs when there are negative lengths, since bi-directing an undirected edge creates a negative length cycle. However, we can use T-join techniques.

Proposition 12 *An undirected graph, $G = (V, E)$, with $c : E \to \mathbb{R}$ has a negative length cycle iff an $\emptyset$-join has negative cost.*

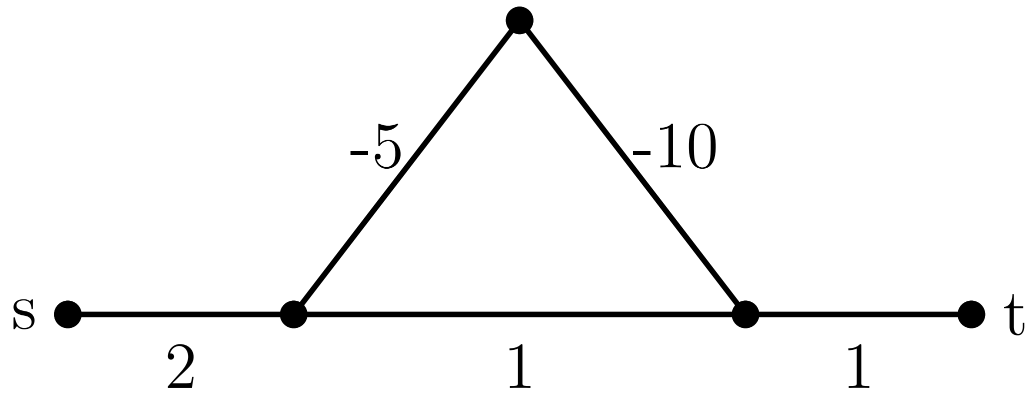

Figure 1: An example of a graph with a negative cost $\emptyset$-join

Proposition 13 *If G has no negative length cycle then the min-cost $\{s, t\}$-join gives an s-t shortest path.*

Remark 14 *It is important to first check for negative length cycles before finding an $\{s, t\}$-join.*

Theorem 15 *There is a strongly polynomial time algorithm that given an undirected graph, $G(V, E)$, with $c : E \to \mathbb{R}$, either outputs a negative length cycle or an s-t shortest path.*

Proof Sketch. We first compute a min-cost $\emptyset$-join. By Proposition 12 we know that if this $\emptyset$-join has negative cost then we can produce a negative length cycle. Otherwise, we know there is no negative length cycle and hence by Proposition 13 we can compute a min-cost $\{s, t\}$-join in order to find an s-t shortest path. (In each case the T-join can be computed using the algorithm from the previous section.) □

1.2.3 Max-cut in planar graphs

Since one can compute min-cost T-joins with negative costs, one can compute max-cost T-joins as well. The max-cut problem is the following.

Problem 2 *Given an undirected graph with non-negative edge weights, find a partition of V into $(S, S \setminus V)$ so as to maximize $w(\delta(S))$.*

Max-cut is NP-hard in general graphs, but Hadlock showed how T-joins can be used to solve it in polynomial time for planar graphs. A basic fact is that in planar graphs, cuts in G correspond to collections of edge disjoint cycles in the dual graph G^*. Thus to find a max-cut in G we compute a max $\emptyset$-join in G^* where the weight of an edge in G^* is the same as its corresponding edge in the primal.

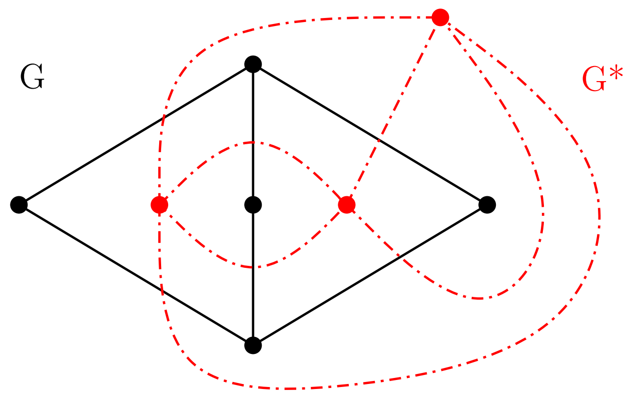

Figure 2: A planar graph, G, in black, and its dual, G^*, in dashed red.

1.2.4 Polyhedral aspects

The following set of inequalities can be shown to determine the characteristic vectors of the set of T-joins in a graph G.

$$0 \leq x(e) \leq 1$$
$$x(\delta(U) \setminus F) - x(F) \geq 1 - |F| \qquad U \subseteq V,\ F \subseteq \delta(U),\ |U \cap T| + |F| \text{ is odd}$$

References

[1] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1998.

[2] A. Schrijver. *Theory of Linear and Integer Programming (Paperback)*. Wiley, 1998.