A Convex Relaxation for the Asymmetric TSP

Santosh Vempala * Mihalis Yannakakis†

1 The problem

The Traveling Salesman Problem (TSP) has been widely studied. Given a graph with lengths on its edges, the problem is to find a tour of minimum length that visits every vertex once. One usually makes the assumption that the edge lengths satisfy the triangle inequality, in which case the problem is equivalent to requiring the tour to visit every vertex at least (instead of exactly) once. With the further assumption that the edge lengths are symmetric (in other words, the graph is undirected), it is well-known that the optimum can be approximated to within a factor of $3/2$. Without the latter assumption, i.e., when the graph is directed, the best-known approximation is a factor of $\log n$ [1]. Unfortunately, this algorithm can be nearly $\log n$ off from the optimum even when the edge lengths are all 1, and the underlying graph is known to be Hamiltonian. Here we study a linear programming relaxation for the problem. Our main result is that one can find a fractional solution to the relaxation that is *very sparse* (with $< 3n$ edges). We also show that in the special case when the underlying graph is Hamiltonian with edge lengths 1 and the (in- and out-) degrees of every vertex are each at most 2, a solution to the relaxation can be rounded to an integral solution (a tour) whose length is at most twice the optimum. Note that even this special case of the problem is NP-hard! (and the previous algorithm is $\Omega(\frac{\log n}{\log \log n})$ off).

2 The relaxation

$$\min \quad \sum_{ij \in E} C_{ij} x_{ij}$$

$$\sum_{i:ij \in E} x_{ij} = \sum_{k:jk \in E} x_{jk} \quad \forall j \in V \tag{2.1}$$

$$\sum_{ij \in E: i \in S, j \notin S} x_{ij} \geq 1 \quad \forall S \subset V \tag{2.2}$$

$$x_{ij} \geq 0 \quad \forall ij \in E$$

It is easy to verify that any (integral) traveling salesman tour satisfies the balance and cut constraints (2.1 and 2.2); hence this is a relaxation. Although the number of constraints is exponential, the relaxation can be solved in poly-time by designing a poly-time separation oracle [3] or by reformulating as a poly-size LP using auxiliary variables.

If G is Hamiltonian with unit edges lengths, then any optimal solution has the property that the total weight on in-edges = total weight on out-edges = 1. So if a vertex u has out-degree 1, and its out-edge is (u, v) then $x_{uv} = 1$, and hence v has in-degree 1 in the fractional solution.

3 The rounding algorithm

From the above observations we can derive an algorithm for the case when the original graph has degrees at most 2, and is (fractionally) Hamiltonian.

1. Find an optimal fractional solution z of the relaxation for G. Discard all zero weight edges. Now, if $d_{in}(u) = d_{out}(u)$ for every vertex u in the remaining graph, output an Euler tour and stop.
2. Otherwise, let u be a vertex of out-degree 1. Let the out-edge from u be (u, v). Contract u and v to a single vertex uv and collapse multiple edges to get a new graph G'. and an induced fractional solution z'. Recursively round z' to an integral tour C' that visits every vertex of G' at most twice (and at least once!).
3. Uncontract uv. The tour C' passes through the contracted vertex uv once or twice. Add as many copies of the edge (u, v) to C' to get a tour C of G. Output the tour C.

THEOREM 3.1. *The algorithm finds a tour of length at most $2n$.*

4 Structure of a basic solution

Consider a basic (vertex) fractional solution to the relaxation. A subset of vertices forms a *tight* set if the total weight of edges into the subset (equal to the weight of edges going out) is exactly 1. Two sets A and B *cross* if $A \cap B \neq \phi$. A family of sets is called *laminar* if no two of its sets cross.

*Department of Mathematics, M.I.T. vempala@math.mit.edu
†Bell Laboratories, Lucent Technologies. mihalis@bell-labs.com

A constraint in the relaxation can also be viewed as an incidence vector of edges involved in the constraint; that is, a vector in $\mathbf{R}^E$. With this viewpoint, we can define the rank of a set of contraints as the number of linearly independent constraints. The following lemma is very similar to one derived in [2] and more recently in [4], in different contexts.

LEMMA 4.1. *The rank of the set of tight constraints in any basic solution is at most the size of a maximal laminar family of tight sets plus* $n-1$.

Proof. (idea) The rank of the balance constraints (2.1) is at most $n-1$. To determine the additional rank of the tight cut constraints (2.2), we observe that any two tight sets that cross can be "uncrossed". That is, we can show that if A and B are tight sets and $A \cap B$ is non-empty, then either $A \cap B$ and $A \cup B$ are both tight, or $A - B$ and $B - A$ are both tight. In either case, the resulting set systems are laminar. By repeating this operation, we arrive at a laminar family that spans all the tight sets.

THEOREM 4.1. *The number of non-zero edges in a basic solution is at most 3n-2.*

Proof. The number of non-zero edges in a basic solution is at most the number of linearly independent balance and cut constraints that are tight. There are at most $n-1$ linearly independent balance constraints. The maximum possible size of a maximal laminar family of sets on n vertices is $2n-1$. Hence the maximum number of linearly independent cut contraints is $2n-1$. Therefore the total number of non-zero edges in a basic solution is at most $3n-2$.

References

[1] A. Frieze, G. Galbiati, and F. Maffioli, "On the worst-case performance of some algorithms for the asymmetric traveling salesman problem, " Networks 12, 23-39, 1982.

[2] G. Cornuejols, J. Fonlupt, and D. Naddef, "The travelling salesman problem on a graph and some related integer polyhedra," Mathematical Programming 33, 1-27, 1985.

[3] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer, 1988.

[4] K. Jain, "A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem," Proceedings of the IEEE Symposium on the Foundations of Computer Science, 1998.