ORIE 6300 Mathematical Programming I	November 12, 2008
Recitation 10	
Lecturer: Maurice Cheung	*Topic: Lagrangean Relaxation for Integer Program*

We have seen a constraint-generation procedure to aid in solving the LP relaxation of the traveling salesman problem. This week we will consider another method for producing a bound on the optimal IP value.[1]

In general, consider an integer program of the form:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax & \geq b \\ & Dx & \geq d \\ & x & \text{integer} \end{array}$$

Assume all of the input data is integer-valued and suppose we can "quickly" optimize over the set

$$X = \{x : Dx \geq d,\ x \text{ integer}\}.$$

Now, we will relax the "hard" constraints $Ax \geq b$ by removing them and inserting a penalty for violations. Let p be a non-negative vector in $\mathbf{R}^n$, and consider the new problem:

$$\begin{array}{ll} \min & cx + p(b - Ax) \\ \text{s.t.} & x \in X. \end{array}$$

Let $Z(p)$ be the optimal objective value of this LP. Clearly, $Z(p)$ is no larger than the optimal IP value since the optimal IP solution is feasible for this problem and it satisfies $Ax^*_{IP} \geq b$, so $p(b - Ax^*_{IP}) \leq 0$.

The Lagrangean Dual

For any $p \geq 0$, we have $Z(p) \leq Z_{IP}$, giving us a bound on the optimal IP value. to get the best possible bound, consider the problem:

$$Z_D = \max_{p \geq 0} Z(p) = \max_{p \geq 0} \min_{x \in X} cx + p(b - Ax)$$

If $X = \{x^1, \ldots, x^k\}$ is a finite set, then we can compute the value Z_D by the following LP:

$$\begin{array}{lll} \max & q & \\ \text{s.t.} & q & \leq cx^i + p(b - Ax^i) \quad i = 1 \ldots, k \\ & p & \geq 0 \end{array}$$

Note that when X is large, this is inefficient. However taking the dual of this LP, we get:

$$\begin{array}{ll} \min & \sum_j y_j(cx^j) \\ \text{s.t.}: & \sum_j y_j(A_i x^j - b_i) \geq 0 \quad \forall i = 1, \ldots, m \\ & \sum_j y_j = 1 \\ & y \geq 0 \end{array}$$

[1]Based on previous notes of Rajithkumar Rajagopalan and Dennis Leventhal

If we rearrange the equations, and use the fact that $\sum_j y_j = 1$, we get an equivalent representation:

$$\begin{array}{ll} \min & c\left(\sum_j y_j x^j\right) \\ \text{s.t. :} & A\left(\sum_j y_j x^j\right) \geq b \\ & \sum_j y_j = 1 \\ & y \geq 0 \end{array}$$

Letting conv(X) be the convex hull of X, note that $x \in conv(X)$ iff $x = \sum_j \alpha_j x^j,\ \sum_j \alpha_j = 1,\ \alpha \geq 0,\ x^i \in X$. Hence, this LP is exactly the same as optimizing over the convex hull of X. Hence, this can be written as:

$$\begin{array}{ll} \min & cx \\ \text{s.t.} & Ax \geq b \\ & x \in conv(X) \end{array}$$

Hence, Z_D can be computed by solving this LP. Note that an immediate corollary of this is that $Z_{LP} \leq Z_D$ since $conv(X) \subseteq \{x : Dx \geq d\}$. That means the bound provided by the Lagrangean dual is as least as strong as the LP-bound. Additionally, note that the analysis here can be extended to the case where X is not finite.

The Held-Karp Bound for Traveling Salesman Problem

Consider the TSP, that is, the problem of finding a minimum cost tour in a graph. In a previous recitation, we used the following characterization of tours: A subgraph forms a tour iff each vertex has degree 2 and each cut has at least 2 edges crossing it. This led to an LP relaxation that could be used to give a bound on the value of the optimal tour. Here, we will use a slightly different characterization. Number the nodes in the graph 1 through n for some arbitrary numbering.

Definition 1 *A subgraph is a 1-tree if it is a spanning tree on nodes $\{2, \ldots, n\}$ along with two edges incident to node 1.*

Claim 1 *A subgraph is a tour iff it is a 1-tree where each vertex $2, \ldots, n$ has degree 2.*

Thus, the following IP solves for the minimum cost tour, where V_{-1} is the set of vertices $\{2, \ldots, n\}$, and $E(S)$ is the set of edges with both endpoints in S:

$$\begin{array}{lll} \min & \sum_e c_e x_e & \\ \text{s.t.:} & \sum_{e \in E(V_{-1})} x_e = n - 2 & \\ & \sum_{e \in E(S)} x_e \leq |S| - 1 & \forall S \subset V_{-1} \\ & \sum_{e \in \delta(\{1\})} x_e = 2 & \\ & \sum_{e \in \delta(\{i\})} x_e = 2 & \forall i \in V_{-1} \\ & x \text{ integer} & \end{array}$$

Let X be the set of vectors that satisfy all of these constraints except for the vertex constraints for $i \in V_{-1}$. These constraints say that there must be $n - 2$ edges in $E(V_{-1})$ and any subgraph on k nodes in V_{-1} can have at most $k - 1$ edges. Note that this implies that there are no cycles on that subgraph, therefore it must be a tree. Since we also require that there are two edges incident to node 1, the set X is exactly the set of 1-trees.

Thus, we can consider dualizing the vertex constraints for vertices in V_{-1}. The bound Z_D obtained this way is known as the Held-Karp lower bound. This bound is actually fairly easy to calculate. Since the set X is the set of 1-trees, we can calculate the min cost 1-tree for a particular vector p without solving an LP at all.

In this case, it turns out that the polytope defined by the non-dualized constraints has integer extreme points. Using the notation from the previous section, this says that $\{x : Dx \geq d\} = conv(X)$. Thus, for this formulation, $Z_{LP} = Z_D$. So the value of the Held-Karp lower bound is not in the strength of the bound, but in the efficient computation.