IOE 691: Approximation Algorithms | Date: 4/3/2017

Lecture Notes: k-Minimum Spanning Tree (Lagrangian Relaxation)

Instructor: Viswanath Nagarajan | Scribe: Cupjin Huang

1 LP Relaxation and Lagrangian Relaxation

Let's consider a variant of minimum spanning tree problem. We are given a graph $G = (V, E)$, where $r \in V$ is a special root node, a cost function $c : E \to \mathbb{R}^+$ on edges and a target $k \in [|V| - 1]$. Our goal is to find a Steiner tree (i.e. a tree with minimum weight) connecting r to at least k other nodes. For an IP formulation, denote $x_e \in \{0, 1\}$ as the indicator of whether an edge $e \in E$ is chosen, and $z_v \in \{0, 1\}$ as the indicator of whether node v is disconnected from r. We have the following LP relaxation of the problem, where the constraints on x and z are relaxed to $x_e, z_v \geq 0$:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
s.t. \quad & x(\delta S) + z_v \geq 1, \qquad \forall v, S : v \in S \subseteq V \setminus \{r\}, \\
& \sum_{v \in V \setminus \{r\}} z_v \leq n - k, \\
& \mathbf{x}, \mathbf{z} \geq 0.
\end{aligned}
$$

Note that in this problem, we have a constraint $\sum_v z_v \leq n-k$ that we are not sure how to deal with. We can introduce Lagrangian multiplier to get rid of this extra constraint. Namely, define

$$
\begin{aligned}
L(\lambda) := \min \quad & \sum_{e \in E} c_e x_e + \lambda \cdot \Big(\sum_{v \in V \setminus \{r\}} z_v - (n - k) \Big) \\
s.t. \quad & x(\delta S) + z_v \geq 1, \qquad \forall v, S : v \in S \subseteq V \setminus \{r\}, \\
& \mathbf{x}, \mathbf{z} \geq 0.
\end{aligned}
$$

for all $\lambda \geq 0$. We have the following two observations:

- Intuitively, λ represents the penalty one pays for the violation of the constraint $\sum_v z_v \leq n-k$; the bigger λ is set, the more "serious" it will be considered by the LP. Let's look at the two extreme cases first. When $\lambda = 0$, there is no penalty at all, so the optimal solution will be the trivial one, i.e. no edge will be chosen; when $\lambda \to \infty$, the first term becomes insignificant, and the fewer vertices are disconnected, the less the value of the objective function would be, thus in the limit case we recover the ordinary minimum spanning tree.

- The Lagrangian dual $\max_{\lambda \geq 0} L(\lambda) \leq LP$, where LP is the optimal value for the original LP.

Let's take a closer look at $L(\lambda)$ for a fixed λ. The constraints look exactly the same as the ones in prize-collecting Steiner tree (PCST), and the objective function can be written as $\sum_e c_e x_e + \lambda \cdot \sum_v z_v - \lambda(n-k)$, where the first two terms are edge costs and and node penalties respectively (where penalty for each node is λ), and the last term is just a constant shift. Recall the PCST algorithm from last lecture:

Theorem 1.1. *Given any instance of PCST, the algorithm outputs tree F such that*

$$C(F) + 2\Pi(F) \leq 2LP,$$

where $C(F)$ and $\Pi(F)$ are the sum of edge costs and sum of node penalties respectively, and LP is the optimal value of the LP relaxation.

So, given an arbitrary λ, there is an efficient algorithm which outputs a Steiner tree rooted at r, with edge cost $C(\lambda)$ and connects $K(\lambda)$ nodes, and

$$C(\lambda) + 2\lambda(n - K(\lambda)) \leq 2(L(\lambda) + (n-k)\lambda),$$

i.e. $C(\lambda) \leq 2L(\lambda) + 2\lambda(K(\lambda) - k) \leq 2OPT + 2\lambda(K(\lambda) - k)$. Ideally, if we can find λ such that $K(\lambda) = k$, then we can directly get a 2-approximation; however we don't have the guarantee that there exists such a λ, or even if we do, it is unclear how to find one in polynomial time. Therefore, whichever λ we try, it is possible that we either get an infeasible solution $K(\lambda) < k$ if λ is chosen too small, or a terrible approximation ratio when $K(\lambda) > k$ and λ is chosen too large. How can we balance between these two scenarios?

2 Algorithm

Note that we know $K(0) = 0$ and $\lim_{\lambda \to \infty} K(\lambda) = n$. This means that, there exists a particular choice $\lambda = \theta$ such that $K(\lambda)$ transits from less than or equal to k to greater than equal to k in some small neighborhood of θ. Such a point can be determined efficiently by a binary search on feasible candidates of λ.

Assume that we have found θ now. Denote $c_1 = C(\theta_-), k_1 = K(\theta_-) < k, c_2 = C(\theta_+), k_2 = K(\theta_+) > k$ where θ_-, θ_+ are θ perturbed in both directions respectively. We know that

$$c_i + 2\theta(k - k_i) \leq 2L(\theta),\ i = 1, 2.$$

Thus a convex combination μ_1, μ_2 such that $\mu_1 + \mu_2 = 1, \mu_1 k_1 + \mu_2 k_2 = k$ would give us

$$\mu_1 c_1 + \mu_2 c_2 \leq 2LP.$$

Note that this crucially uses the "Lagrangian multiplier preserving property" in Theorem 1.1: that the multiplier of the penalty term is equal to the approximation ratio.

Still, this is not quite enough; what we have got is a fractional solution which is guaranteed to be a 2-approximation. More work is needed for rounding it to an integral solution.

Let the two trees be T_1 and T_2 respectively. Consider the following rounding scheme:

- Take T_2 with cost $A = c_2$. Note that T_2 is already a feasible solution;

- Take T_1. This is yet to be a feasible solution, so we need to add more vertices to it. Note that $|T_2| - |T_1| = k_2 - k_1$, thus $|T_2 \setminus T_1| \geq k_2 - k_1$. Take T_2 and shortcut it to a cycle on $T_2 \setminus T_1$. Note that the weight of the cycle is bounded by twice the cost of the tree, hence at most $2c_2$. Take the path P on this cycle with $k - k_1$ nodes with the minimum cost, we know that the cost is lower than the average, i.e. $2c_2 \cdot \frac{k-k_1}{k_2-k_1} = 2c_2 \cdot \mu_2$. Attach this path P to C_1 by connecting an arbitrary node of P to r via the shortest path. The total cost would be bounded by $B = \Delta + C_1 + 2C_2 \cdot \mu_2$, where $\Delta = \max_{v \in V \setminus \{r\}} d_{rv}$ is the diameter of the graph.

3 Analysis

We prove the following lemma, which gives a 5-approximation given the promise that $\Delta \leq OPT$:
Lemma 3.1. $\min\{A, B\} \leq \Delta + 4LP$.

Proof. Prove by cases.

- If $\mu_2 \geq 1/2$, then $A = c_2 \leq 2\mu_2 \cdot c_2 \leq 2(\mu_1 c_1 + \mu_2 c_2) \leq 4LP$;
- If $\mu_2 \leq 1/2$, then $\mu_1 \geq 1/2$, and

$$B = C_1 + 2\mu_2 c_2 + \Delta \leq 2\mu_1 c_1 + 2\mu_2 c_2 + \Delta \leq 4LP + \Delta,$$

  finishing the proof.

□

Our algorithm is proven to give a 5-approximation on all graphs where $\Delta \leq OPT$, but are we safe to assume this, or does that matter at all? Actually it matters for the reason below.

3.1 Getting Rid of Δ

For most of the LP-based approximation algorithm we encountered, we actually proved an approximation ratio of α with respect to the LP value LP, since the approximation ratio follows from the fact $LP \leq IP$. One limitation of this general approach is that our approximation ratio can never beat the *integrality gap* of the LP relaxation, which is defined to be $\sup \frac{IP}{LP}$ where the supremum is taken over all instances of the problem. In this k-MST problem, we cannot get rid of this Δ by simply arguing with respect to the LP value (i.e. without mentioning Δ) since it actually has an integrality gap of $\Omega(n)$, $n := |V|$.

Consider one simple example $G = (\{v_i\}_{i=1}^n \cup \{r, u\}, E)$, where all v_i's are Steiner nodes. There is an edge from r to u with weight M, and from u to each v_i with weight 1. Set $k = 1$, it can be observed that the min-cost spanning tree would be of cost $M + 1$. However, there is a feasible fractional solution where $z_{v_i} = 1 - 1/n, x_{(u,v_i)} = 1/n$ and $x_{(u,r)} = 1/n$, giving a total cost of $M/n + 1$. It is thus impossible or this example and more generally, for k-MST problem, to prove a bound of the form $ALG \leq \alpha LP$ for constant α.

However, we are not completely doomed. Notice that our algorithm will still output a solution close to the optimal one; the only thing that went wrong is the LP value itself. Also the lemma $ALG \leq 4LP + \Delta$ always holds, and we can use the argument of *guess and verify* to get rid of the $\Delta \leq OPT$ assumption.

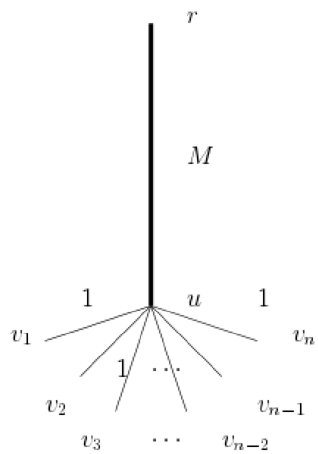

Figure 1: Illustration of the counterexample

Consider a variant of the problem where we know OPT beforehand as a promise. What can we do better? It turns out we can preprocess the input graph by first eliminate all nodes v where $d_{rv} \geq OPT$, since these nodes are surely not in the optimal solution. Therefore the optimal solution in the reduced graph is the same as the optimal solution in the original graph, and in the reduced graph we have $ALG \leq 4LP + OPT$ since the diameter is bounded by OPT already, giving us a 5-approximation.

To generalize this idea, we can intuitively pretend that we know OPT beforehand by guessing it roughly. To be more specific, for each $B \in [0, \Delta]$, define G_B the subgraph of G restricted to all nodes within distance B from the root r. Our approximation algorithm will give a value of ALG_B with the promise that

$$ALG_B \leq 4LP_B + B,$$

where $LP_B \leq OPT_B$ is the LP value of the problem restricted on G_B. We have the promise that $LP_B \leq OPT$ given $B \geq OPT$ by the argument above, so the following algorithm would give a $5+\epsilon$ approximation (assuming every weight is integer):

Algorithm 1 Guess and Verify

```
O ← ∞
for 0 ≤ i ≤ (log n·c_max)/(log 1+ε) do
    B ← (1 + ε)^i
    if ALG_B ≤ O then
        O ← ALG_B
    end if
end for
return O
```

To see the correctness of the algorithm, note that at least one value in $B \in [OPT, (1+\epsilon)OPT]$ will be chosen in some iteration, and in that iteration, we have found a solution with $LP_B \leq OPT$ and $B \leq (1+\epsilon)OPT$, resulting in an approximation ratio of $5+\epsilon$ due to Lemma 3.1.

The number of iterations of the outer loop is $\frac{\log n \cdot c_{\max}}{\log 1+\epsilon}$, where $c_{\max} = \max_e c_e$. This is within $poly(n, \log D, 1/\epsilon)$ hence providing an efficient approximation algorithm.