CS602 Combinatorial Optimization via Linear and Convex Programming Jan-Apr 2022

Homework: Primal-Dual

Lecture 9 (Feb 3)

- Prove that the optimal value of the LP we wrote is equal to the length of the shortest path.
- Start with the LP for maximum matching in a bipartite graph. Write the dual LP for it. Observe that integral solutions of the dual LP correspond to vertex covers. Using LP duality, can you argue that maximum matching size is equal to the minimum vertex cover size in a bipartite graph?

Lecture 10 (Feb 7)

- Prove that a pair of primal and dual feasible solutions which satisfy approximate complementary slackness are approximately optimal solutions for their respective linear programs.

Lecture 11 (Feb 11)

- Primal dual approach can be used for a general LP. Let the primal LP be

  $$\min w^T x \text{ subject to } Ax = b, x \geq 0,$$

  and its dual LP be

  $$\max b^T y \text{ subject to } A^T y \leq w.$$

  Let β be a dual feasible solution. Suppose there is no primal feasible solution α for which α and β satisfy complementary slackness conditions. We want to argue that in this case there exists a way to improve the dual feasible solution.

  Let the matrix A have entries $(a_{i,j})$. Complementary slackness condition would say that

  $$\text{for } 1 \leq j \leq n, \sum_i a_{i,j} y_i < w_j \implies x_j = 0.$$

  Without loss of generality let us say that

  $$\text{for } 1 \leq j \leq n_1, \sum_i a_{i,j} \beta_i = w_j,$$

  $$\text{and for } n_1 < j \leq n, \sum_i a_{i,j} \beta_i < w_j.$$

  That means the primal feasible solution α we are looking for must satisfy

  $$\text{for } n_1 < j \leq n, \alpha_j = 0$$

  together with $A\alpha = b, \alpha \geq 0$. Suppose there is no such α. Use one of the forms of Farkas' Lemma and show that there is a dual feasible solution β' with $b^T \beta' > b^T \beta$.

Lecture 12 (Feb 14)

- Prove that the greedy algorithm described in the class for minimum size vertex cover problem is a 2-approximation algorithm.

Lecture 13 (Feb 17)

- Let Δ_n be the n-dimensional probability simplex, i.e.,

$$\Delta_n = \{\lambda \in \mathbb{R}^n : \sum_{i=1}^{n} \lambda_i = 1 \text{ and } \lambda_i \geq 0 \text{ for each } 1 \leq i \leq n\}.$$

  Let R be a $k \times n$ real matrix. Let $R_{i,j}$ be the (i,j)-th entry of R. Show that the quantity

$$\max_{x \in \Delta_k} \min_{y \in \Delta_n} x^T R y$$

  is same as the optimal value of the below LP.

$$\begin{aligned} &\max z \\ &\text{subject to} \\ &\sum_{i=1}^{k} x_i R_{i,1} \geq z \\ &\sum_{i=1}^{k} x_i R_{i,2} \geq z \\ &\vdots \\ &\sum_{i=1}^{k} x_i R_{i,n} \geq z \\ &\sum_{i=1}^{k} x_i = 1 \\ &x_i \geq 0 \text{ for each } 1 \leq i \leq k. \end{aligned}$$

- Using strong duality show that the optimal value of the above LP is same as the optimal value of the below LP.

$$\begin{aligned} &\min z \\ &\text{subject to} \\ &\sum_{j=1}^{n} R_{1,j} y_j \leq z \\ &\sum_{j=1}^{n} R_{2,j} y_j \leq z \\ &\vdots \\ &\sum_{j=1}^{n} R_{k,j} y_j \leq z \\ &\sum_{j=1}^{n} y_j = 1 \\ &y_j \geq 0 \text{ for each } 1 \leq i \leq n. \end{aligned}$$

- Finally observe that the optimal value of the last LP is same as

$$\min_{y \in \Delta_n} \max_{x \in \Delta_k} x^T R y$$

Lecture 14 (Feb 28)

- Recall the LP we wrote for the Steiner forest problem. Argue that every optimal integral solution to the LP corresponds to a optimal Steiner forest.
- Recall the LP we wrote for the Steiner forest problem. Find an example of a graph with edge weights and a set of pairs of terminals, where the the weight of the optimal Steiner forest is almost twice of the LP optimal value.

Lecture 15 (Mar 3)

- **Survivable network design.** Given a graph G with edge weights, a set of terminal pairs, say $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, and numbers $n_1, n_2, \ldots, n_k$. We want to find the minimum weight subset of edges F such that for each i, there should be n_i edge-disjoint paths between s_i and t_i. Write an appropriate LP and design a primal-dual algorithm. What is the best approximation factor you can get?
- Consider an instance of the Steiner tree problem where every vertex is a terminal and we want to connect each vertex with every other vertex. This is the minimum spanning tree problem. If you run the primal dual algorithm (discussed in class) on this instance, will you always get the minimum spanning tree? Actually, there are examples where the tree you will get will have weight almost twice of the MST. Can you construct such an example?
- Recall that in the Steiner tree problem we are given a graph G with a set of terminals, say $t_1, t_2, \ldots, t_k \in V$ and we want to connect each terminal with every other terminal. Consider the following algorithms.

  Algorithm 1. Construct a minimum spanning tree for the graph (while ignoring which vertices are terminals). Then prune out any useless edges from the spanning tree, i.e., any edges which do not lie on any terminal-to-terminal path. Show via some examples that the Steiner tree given by this algorithm can be arbitrarily worse than the optimal Steiner tree.

  Algorithm 2. Find the shortest path between every pair of terminals. Take the union of these shortest paths and call this subgraph H. Find an MST in H. Prune out any edges that are not used in connecting any two terminals. Argue that this is a 2-approximation algorithm.

  Hint for algorithm 2: consider the optimal Steiner tree. Make an arbitrary terminal as the root of this tree. Start a tree traversal from the root and come back after visiting every vertex. Cost of this traversal is exactly twice of the weight of the optimal Steiner tree. You need to show that the tree computed by your algorithm is at most the cost of this traversal.

  Note: Algorithm 2 gives 2-approximation in case of Steiner tree (when we want all terminals connected with each other). But, for Steiner forest problem (when we are given an arbitrary set of terminal pairs), such an algorithm will perform poorly. You can try to find examples.