CS602 Applied Algorithms **Jul-Nov 2023**

Homework (No submission)

Primal-dual approximation algorithms

Lecture 14-16 (Sep 26, 29, Oct 3)

1. Consider the following algorithms for minimum weight Steiner tree problem. For each algorithm, either prove that it gives a 2-approximation or give a counter-example.
   - Compute minimum spanning tree for all vertices in the graph. Delete every edge which is not on a path from a terminal to another terminal.
   - Compute shortest path for each pair of terminals. Construct a complete graph on the terminal vertices where edge weights are the computed shortest path lengths. Construct an MST for this graph and take the union of the paths corresponding to the MST edges.
   - Go over the terminals in some order $t_1, t_2, \dots, t_k$ and construct Steiner tree incrementally. Let tree(i) be the Steiner tree computed that connects $t_1, t_2, \dots, t_i$. For terminal t_{i+1}, find the shortest path from t_{i+1} to the tree(t_i). And so on.
2. Express the minimum Steiner forest problem as a covering problem.
3. Primal-dual algorithm for Steiner forest: show via an example that if the dual variables are increased in an arbitrary order then the obtained primal solution can be far from the optimal, i.e., more that a constant factor larger than the optimal.
4. Complete the analysis of the 2-factor approximation for the primal dual algorithm discussed in the class.
5. Prove that in the case when there is only one terminal pair, the primal dual algorithm will give an exactly optimal solution. Is the algorithm exactly same as Dijkstra's algorithm in that case?
6. 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?
7. **Survivable network design.** Given a graph G with edge weights, a set of terminal pairs, say $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$, and numbers $n_1, n_2, \dots, n_k$. 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?

LP rounding

Lecture 17-18 (Oct 6, 10)

8. Consider the following algorithm for the maximum weight satisfiability problem. Pick the highest weight clause. Set some variable in this clause so as to satisfy it. Remove all the satisfied clauses and repeat the same procedure. And continue so on.

   Show that this algorithm can give a very bad (arbitrarily low) approximation factor.

9. Consider a different algorithm for the maximum weight satisfiability problem. Find to total weight of clauses with x_1 and total weight of clauses with $\overline{x}_1$. Set x_1 depending on which weight is higher. Remove the already satisfied clauses. Then continue the process with x_2 and so on.

   Show that this algorithm always gives at least 1/2-approximation.

10. Recall the randomized algorithm for the maximum weight satisfiability problem, discussed in the class, where variables are set to True or False independently with probability 1/2. And then the deterministic version based on the method of conditional expectations. Show that the total weight of satisfied clauses in the deterministic algorithm will always be at least the expected total weight of satisfied clauses in the randomized algorithm.

11. Recall that W is the random variable denoting total weight of satisfied clauses. How can you efficiently compute the conditional expectation $\mathbb{E}[W \mid x_1 = True]$.

12. The second greedy algorithm above and the algorithm based on conditional expectations seem quite similar. Do you think one of these algorithms is always better than the other?

13. Write an appropriate linear program for MAXSAT problem.

14. Prove linearity of expectation.

15. Recall the LP we wrote for the maximum weight satisfiability problem with m clauses and n variables. Suppose w_i is the weight of the i-th clause. Let P_i and N_i be the sets of indices of positive and negative literals appearing in clause C_i.

$$\max \sum_{i=1}^{m} w_i y_i$$
$$\text{subject to}$$
$$t_j \geq 0 \text{ for } 1 \leq j \leq n$$
$$0 \leq y_i \leq 1 \text{ for } 1 \leq i \leq m$$
$$y_i \leq \sum_{j \in P_i} t_j + \sum_{j \in N_i} (1 - t_j) \text{ for } 1 \leq j \leq m.$$

   Suppose (t^*, y^*) is an optimal solution for this LP. Suppose we set our Boolean variables with the following rounding scheme. For each i, set $x_i = True$ if and only if $t^* \geq 1/2$.

   We had seen in the class an example, where there were multiple LP optimal solutions and some LP solutions gave a bad integral solution when we applied the above rounding procedure.

   Show that for the below example, there is only one LP optimal solution. And rounding that LP optimal solution with the above scheme does not give a good solution.

   $(x_1 \vee x_2), (x_2 \vee x_3), \ldots, (x_{2n} \vee x_{2n+1}), (x_{2n+1} \vee x_1), (\overline{x}_1 \vee \overline{x}_2), (\overline{x}_2 \vee \overline{x}_3), \ldots, (\overline{x}_{2n} \vee \overline{x}_{2n+1}), (\overline{x}_{2n+1} \vee \overline{x}_1)$

16. Consider any randomized algorithm for the maximum weight satisfiability problem, where variables are set to True or False with some randomized scheme. Suppose we deterministically implement this algorithm using the method of conditional expectations (as described in the class). Show that the total weight of the satisfied clauses obtained from the deterministic implementation is at least as large as the expectation of the total weight of the satisfied clauses in the randomized algorithm.

17. Prove that $g(z) = 1 - (1 - z/\ell)^\ell$ is a concave function in the region $[0, 1]$, for $\ell = 1, 2, 3, \ldots$.

18. Prove that $(1 - 1/2^\ell)/2 + (1 - (1 - 1/\ell)^\ell)/2 \geq 3/4$ for every $\ell = 1, 2, 3, 4, \ldots$.

19. Using the above, or otherwise, prove that a combination of the two algorithms – simple randomized and LP rounding based – gives an approximation factor of 3/4.

20. Write an LP for the Max-Cut problem and design a LP rounding based approximation algorithm with approximation factor 1/2.

Semidefinite programming and approximation algorithm for MAX cut

Lecture 19-21 (Oct 13, 17, 20)

21. Prove that sum of two convex functions is convex.
22. Prove that $f(x) = x^2$ is convex.
23. Prove that the following is a convex function

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{k} (\alpha_{i,1}x_1 + \alpha_{i,2}x_2 + \dots + \alpha_{i,n}x_n)^2 .$$

24. Consider a function $f(x, y) = ax^2 + bxy + cy^2$. Suppose $f(x, y)$ is convex then prove that $f(x, y)$ can be written as $(\alpha x + \beta y)^2 + (\gamma x + \delta y)^2$ for some $\alpha, \beta, \gamma, \delta \in \mathbb{R}$.

    Hint: First prove that the matrix $\begin{pmatrix} 2a & b \\ b & 2c \end{pmatrix}$ must be positive semidefinite. Then any 2 by 2 PSD matrix can be written as $uu^T + vv^T$ for some $u, v \in \mathbb{R}^2$.
25. Prove that if $f(x)$ is convex function then $K = \{x : f(x) \leq \alpha\}$ for any $\alpha \in \mathbb{R}$ is a convex set.
26. Consider $f(x, y) = -xy$ and show that the converse of the previous statement is not true.
27. Prove that the following are equivalent for an $n \times n$ symmetric matrix A
    - $v^T Av \geq 0$ for all $v \in \mathbb{R}^n$.
    - All eigenvalues of A are non-negative.
    - $A = U^T U$ for some $n' \times n$ matrix U and some $n' \leq n$.
    - For any non-empty subset $S \subseteq \{1, 2, \dots, n\}$, $\det(A_{S,S}) \geq 0$, where $A_{S,S}$ is the submatrix of A whose rows are indexed by S and the columns are indexed by S.
28. Prove that the set of all $n \times n$ positive semidefinite matrices is a convex set.
29. Prove that the following two standard forms of semidefinite programs can be converted into each other.

    Standard form 1

$$\min \sum_{1 \leq i \leq j \leq n} w_{i,j} x_{i,j}$$

subject to

$$\begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{1,2} & x_{2,2} & \cdots & x_{2,n} \\ & \vdots & & \\ x_{1,n} & x_{2,n} & \cdots & x_{n,n} \end{pmatrix} \succeq 0$$

$$\sum_{1 \leq i \leq j \leq n} a_{1,i,j} x_{i,j} \leq b_1$$

$$\sum_{1 \leq i \leq j \leq n} a_{2,i,j} x_{i,j} \leq b_2$$

$$\vdots$$

$$\sum_{1 \leq i \leq j \leq n} a_{k,i,j} x_{i,j} \leq b_k$$

Standard form 2

$$\min \sum_{h=1}^{n} w_h x_h$$

subject to

$$\begin{pmatrix} \sum_h a_{1,1,h}x_h & \sum_h a_{1,2,h}x_h & \cdots & \sum_h a_{1,n,h}x_h \\ \sum_h a_{1,2,h}x_h & \sum_h a_{2,2,h}x_h & \cdots & \sum_h a_{2,n,h}x_h \\ & & \vdots & \\ \sum_h a_{1,n,h}x_h & \sum_h a_{2,n,h}x_h & \cdots & \sum_h a_{n,n,h}x_h \end{pmatrix} \succeq 0$$

30. For each of the given below convex sets, show that they can be expressed using SDPs. You can use constraints in one of the above forms.

- $\{(x, y) : x^2 + y^2 \leq 1.\}$
- $\{(x_1, x_2, \ldots, x_n) : Cx \leq d\}$ for some $k \times n$ matrix C and $d \in \mathbb{R}^k$.
- $\{(x, y) : (x + 2y - 3)^2 + (3x - 2y + 1)^2 \leq 9.\}$
- $\{(x, y) : x^2 - y^2 \geq 4, x \geq 0.\}$
- $\{(x, y) : xy \geq 5, x \geq 0, y \geq 0.\}$

31. Suppose P is a symmetric matrix that is not PSD. Design an algorithm to find a separating hyperplane between P and the PSD cone. That is, an inequality that is satisfied by all the PSD matrices M but not for P. More precisely, $\{a_{i,j}\}_{i,j}$ and b such that

$$\sum_{i,j} a_{i,j} M_{i,j} \geq b, \text{for all PSD matrices } M, \text{ but}$$

$$\sum_{i,j} a_{i,j} P_{i,j} < b.$$

32. Recall the LP we wrote for the Max cut problem.

$$\begin{aligned} \max \sum_{i,j} w_{i,j} x_{i,j} & \text{ subject to} \\ y_i \in \{0, 1\} & \text{ for each } 1 \leq i \leq n \\ x_{i,j} \in \{0, 1\} & \text{ for each } 1 \leq i < j \leq n \\ x_{i,j} \leq y_i + y_j & \text{ for each } 1 \leq i < j \leq n \\ x_{i,j} \leq 2 - y_i - y_j & \text{ for each } 1 \leq i < j \leq n \end{aligned}$$

Construct an example where the integrality gap for this LP is 1/2. That is, construct a graph with edge weights, such that the max cut value is roughly 1/2 of the LP optimal value.

33. Let $z_i^*, z_j^* \in \mathbb{R}^n$ be two vectors with an angle of θ between them. Let H be a hyperplane chosen randomly uniformly. What is the probability that the two vectors lie on the opposite sides of the hyperplane.

34. Show that the optimal values of the below two programs is same.

Vector Program

$$\max \sum_{i<j} w_{i,j} x_{i,j} \text{ subject to}$$
$$z_i \in \mathbb{R}^n \text{ for each } 1 \le i \le n$$
$$z_i^T z_i = 1 \text{ for each } 1 \le i \le n$$
$$x_{i,j} = \frac{1 - z_i^T z_j}{2} \text{ for each } 1 \le i < j \le n$$

Semidefinite program

$$\max \sum_{i<j} w_{i,j} x_{i,j} \text{ subject to}$$
$$\begin{pmatrix} 1 & 1-2x_{1,2} & \cdots & 1-2x_{1,n} \\ 1-2x_{1,2} & 1 & \cdots & 1-2x_{2,n} \\ & \vdots & & \\ 1-2x_{1,n} & 1-2x_{2,n} & \cdots & 1 \end{pmatrix} \succeq 0.$$

35. Show that
$$\min_\theta \frac{2\theta}{\pi(1-\cos\theta)} \approx 0.878.$$

36. Prove that for a symmetric matrix A, all eigenvalues are at most λ if and only if $\lambda I - A \succeq 0$. Write an SDP to find the maximum eigenvalue of a given symmetric matrix.

37. We want to find an example where the gap between maxcut value and optimal value of the SDP (discussed in class) is roughly 0.878. What is the gap for the complete graph on 3 vertices or 4 vertices.

38. Design an SDP based approximation algorithm for MAX-2-SAT (each clause is OR of two literals).