CSC373— Algorithm Design, Analysis, and Complexity — Spring 2016

Tutorial Exercise 6: Linear Programming and Approximation Algorithms

---

LP and Dual LP for Weighted Vertex Cover. Suppose we are given a graph $G = (V, E)$ and a set of non-negative, integer-valued weights $\{w(v) \mid v \in V\}$, where each $w(v)$ denotes the weight of vertex v. We seek a vertex cover $S \subseteq V$ such that the weight of S, i.e.,

$$w(S) = \sum_{v \in S} w(v), \tag{1}$$

is as small as possible. For a given graph, we denote a minimum weight vertex cover as S^* and its value as $W^* = w(S^*)$.

a) Show that this minimization problem can be expressed as a integer linear programming (ILP) problem. Moreover, write this ILP in the following form (i.e., as a dual LP problem in standard form):

$$\text{minimize } \vec{b}^T \vec{y} \tag{2}$$

subject to,

$$\vec{y}^T A \geq \vec{c}^T, \tag{3}$$

$$\vec{y} \geq \vec{0}. \tag{4}$$

For each vertex $v \in V$, there should be one LP variable, say $y(v)$, and this variable is associated with the v-row of A. If $y(v) = 1$ this represents the case in which v is selected for the vertex cover, and otherwise $y(v) = 0$. Also, each column of A is associated with an edge $e = (u, v) \in E$. (You do not need to impose the constraint that $y(v) \leq 1$, as this should be a consequence of the minimization and other constraints.) Explain what the appropriate values of A, b and c are.

b) Write out the primal linear program associated with the LP in part (a). The elements of the unknown vector x correspond to the $|E|$ variables associated with the columns of A. We refer to each of these entries as $x(e)$.

Define E_v to be the set of edges in E that end at v. Show that the feasibility constraints of the primal problem are equivalent to

$$\sum_{e \in E_v} x(e) \leq w(v), \text{ for all } v \in V, \text{ and} \tag{5}$$

$$x(e) \geq 0, \text{ for all } e \in E. \tag{6}$$

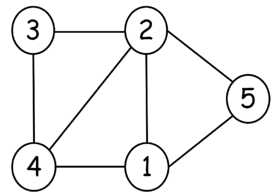
c) Explain how an integer-valued feasible solution to the problem in part (b) provides a lower bound on the minimum weight W^* for any vertex cover of the given graph. Specifically, show that

$$\sum_{e \in E} x(e) \leq W^*. \tag{7}$$

d) Consider the example graph shown below, where the weights $w(v)$ are displayed in each vertex. We define a "maximal" solution to the the primal problem in part (b) to be a feasible solution such that each $x(e)$ is individually increased as much as possible, assuming the other x's are all held fixed. Note this is not necessarily a solution which maximizes the profit of the LP. Rather, it is just a feasible solution for which simple, individual increases of the variables $x(e)$ cannot achieve a larger solution.

Find an integer-valued, maximal, feasible solution to the primal problem in part (b). You can find such a solution simply by selecting an ordering for all the edges in E, loop through these edges, and select the maximum possible value of $x(e)$ for each edge individually. The result will be a maximal feasible solution.

Represent your solution simply by redrawing the graph below and writing the value $x(e)$ beside each edge. On this drawing indicate the vertices at which the feasibility constraints (5) are equality constraints, along with the edges for which (6) are equality constraints. Perhaps do this simply by placing check marks near these vertices and edge weights in your drawing. We refer to these vertices and edges as "tight" (in the primal problem).



e) Try modifying your maximal solution in part (d) to reduce the number of edges $e \in E$ that: 1) connect two tight vertices; and 2) have $x(e) > 0$. One way to modify a solution is to pick a vertex $v \in V$, and "rotate" the weights on two edges that end at v. Such a rotation consists of adding an integer-valued constant to $x(e)$ on one edge that terminates at v, and subtracting this same constant from another edge that terminates at v, maintaining the feasibility of the resulting solution. After doing a few of these rotations it is sometimes possible to increase individual values of $x(e)$, and recover a new maximal feasible solution. Try this procedure on your example from part (d) and draw the resulting graph, indicating the new values of $x(e)$, and which vertices are tight.

f) For your maximal feasible solution in part (d) or (e), define S to be the set of vertices $v \in V$ which are tight, that is,

$$\sum_{e \in E_v} x(e) = w(v). \tag{8}$$

Show that S is a vertex cover. Do this first for your examples in parts (d) and (e), then try to show this is true in general.

g) Define $w(S) = \sum_{v \in S} w(v)$, where S is as in part (f). Show

$$w(S) = \sum_{v \in S} \sum_{e \in E_v} x(e). \tag{9}$$

h) For each edge $e \in E$, let $N(e)$ be the number of times the value $x(e)$ is counted on the right hand side of (9). Show

$$w(S) = \sum_{e \in E} N(e)x(e). \tag{10}$$

What's the maximum of $N(e)$ for all e? Note that equations (7) and (10) were the motivation for part (e), where we tried to minimize the number of edges e with $x(e) > 0$ and for which both endpoints were "tight".

i) Show $w(S) \le 2W^*$, where W^* is the minimum weight of any vertex cover. We can therefore conclude:

$$B = \sum_{e \in E} x(e) \le W^* \le W(S) = \sum_{v \in S} w(v) = \sum_{e \in E} N(e)x(e) \le 2W^*, \tag{11}$$

and it follows that this approach gives a 2-approximation for weighted Vertex Cover.

j) In the lecture notes on Linear Programming we described complementary slackness for constraints on optimal vertices of the primal and dual problems. We do not typically have optimal vertices here, for one because these optimal solutions could be non integer-valued and also because we did not explicitly try to find optimal solutions to either ILP problem (why not?). However, we can still look at the formalism of complementary slackness. That is, given the tight constraints in the primal problem in part (b), which constraints of the dual problem in part (a) would complementary slackness then require to be tight (and which can be loose)? How does this compare to the above choice of S?

k) Briefly compare this approach to the "Pricing Method for Vertex Cover", given at the end of the lecture notes on Approximation Algorithms (which were updated March 22, 2016).

Note that this approach gives more information about the optimum value W^* than simply the 2-approximation $w(S)$. Specifically, from equation (11) we see this approach also gives a poly-time computable lower bound B on W^*. It follows from (11) that this ratio of $w(S)$ and the lower bound B is always[1] less than or equal to 2. This ratio can often be significantly less that 2, and can sometimes be one (which would certify that S is actually an optimal solution). Moreover, $w(S)/B$ serves as a poly-time computable upper bound for the approximation ratio for the given input problem and, as we said, this ratio $w(S)/B$ is always[1] less than or equal to 2.

[1] Except in trivial examples for which the maximal solution $x(e)$ is identically zero.