Problem Set 10 Solutions

May 14, 2005

5.5.2

The problem is

$$\text{minimize } 10x_1 + 3x_2$$

$$\text{subject to } 5x_1 + x_2 \geq 4, \quad x_1, x_2 \in \{0, 1\}.$$

In Exercise 5.1.2, we found that the dual optimal value is $q^* = 8$. Now, consider the constraint-relaxed problem

$$\text{minimize } 10x_1 + 3x_2$$

$$\text{subject to } 5x_1 + x_2 \geq 4, \quad 0 \leq x_i \leq 1, \ i \in \{0, 1\}.$$

Since $10x_1 + 3x_2 = 2(5x_1 + x_2) + x_2 \geq 8$, we have that the optimal value of the above LP problem is $\tilde{f} = 8$ attained at the point $\tilde{x} = (\frac{4}{5}, 0)$. Hence Lagrangian relaxation and constraint relaxation give the same lower bound $LB = 8$.

By rounding the solution $\tilde{x}$, we obtain a feasible point $(1, 0)$ for the original problem, so that we can set the upper bound to be $UB = 10$.

We now branch to the following two solutions/nodes of the branch-and-bound tree:

(a) $x_1 = 0$: The relaxed problem is

$$\text{minimize } 3x_2$$

$$\text{subject to } x_2 \geq 4, \quad 0 \leq x_2 \leq 1.$$

The problem is infeasible, so the node is fathomed.

(b) $x_1 = 1$: The relaxed problem is

$$\text{minimize } 10 + 3x_2$$

$$\text{subject to } 5 + x_2 \geq 4, \quad 0 \leq x_2 \leq 1.$$

The solution is $(1, 0)$, and because this is integer and the only remaining feasible solution, it is optimal.

5.5.4

(a) First, by induction, we show that $x^k \leq x^*$ for all k. Note that the initial point x^0 is chosen such that $x^0 \leq x^*$, and assume that $x^k \leq x^*$ for some k. We need to show that $x^{k+1} \leq x^*$, where x^{k+1} is given by

$$x^{k+1} = x^k + (\xi - x_i^k)e_i,$$

with $g_i(x^k) > 0$ and ξ being the smallest element of X_i such that $g_i(x^k + (\xi - x_i^k)e_i) \leq 0$. Consider the point
$overlinex = (x_1^k, \ldots, x_{i-1}^k, x_i^*, x_{i+1}^k, \ldots, x_n^k)$. If we prove that

$$g_i(\overline{x}) \leq 0, \tag{1}$$

then by the definition of ξ, we will have $\xi \leq x_i^*$, which by inductive hypothesis implies that $x^{k+1} \leq x^*$. To prove the relation (1), consider points

$$\tilde{x}^1 = (x_1^k, x_2^*, x_3^*, \ldots, x_n^*),$$

$$\tilde{x}^2 = (x_1^k, x_2^k, x_3^*, \ldots, x_n^*),$$

$$\tilde{x}^{i-1} = (x_1^k, x_2^k, \ldots, x_{i-1}^k, x_i^*, \ldots, x_n^*),$$

$$\tilde{x}^{i+1} = (x_1^k, x_2^k, \ldots, x_{i-1}^k, x_i^*, x_{i+1}^k, x_{i+2}^*, \ldots, x_n^*),$$

$$\tilde{x}^n = \overline{x} = (x_1^k, \ldots, x_{i-1}^k, x_i^*, x_{i+1}^k, \ldots, x_n^k).$$

Evidently,

$$x^* \geq \tilde{x}^1 \geq \tilde{x}^2 \geq \cdots \tilde{x}^{i-1} \geq \tilde{x}^{i+1} \geq \cdots \geq \tilde{x}^n = x.$$

By using the fact $x^* \geq \tilde{x}^1$, the feasibility of x^*, and the monotonicity property of g_j's, we have

$$0 \geq g_j(x^*) \geq g_j(\tilde{x}^1), \qquad 2 \leq j \leq n.$$

Then, similarly, we obtain

$$0 \geq g_j(\tilde{x}^1) \geq g_j(\tilde{x}^2), \qquad 3 \leq j \leq n.$$

By repeating this, we can see that

$$0 \geq g_j(\tilde{x}^{i-1}), \qquad i \leq j \leq n.$$

Then by using $\tilde{x}^{i-1} \geq \tilde{x}^{i+1}$, we similarly obtain

$$0 \geq g_j(\tilde{x}^{i+1}), \qquad i+2 \leq j \leq n \text{ and } j = i.$$

Proceeding in this way, we finally have

$$0 \geq g_i(\tilde{x}^n) = g_i(\overline{x}),$$

thus showing relation (1).

Since at every iteration we increase some component x_i and each x_i can be adjusted at most $|X_i|$ times ($|\cdot|$ denotes the cardinality of a set), the total number of iterations is at most $\sum_{i=1}^n |X_i|$, which is finite. If the problem is feasible, then by construction, the algorithm terminates with a feasible point, say $\tilde{x}$. Furthermore, by the preceding analysis, we have $\tilde{x} \leq x^*$, which by monotonicity of f implies that $f(x^*) \geq f(\tilde{x})$, so that $\tilde{x}$ is an optimal solution. This optimal solution is the same as x^* if and only if all points $x \in X$ with $x < x^*$ are infeasible, where $x < x^*$ means that $x_j \leq x_j^*$ for all j and $x_i < x_i^*$ for some i.

(b) The method stops either yielding an optimal solution or not being able to find the new iterate. In any case, since the number of increments cannot be larger then $\sum_{i=1}^n |X_i|$, the algorithm must terminate in a finite number of steps.

In order to detect infeasibility, the starting point should be x^0 with $x_i^0 = \min_{\xi \in X_i} \xi$ for all i. With this choice, if the problem is feasible, the algorithm will yield an optimal solution [follows from part (a)].

Suppose now that the problem is infeasible and the algorithm terminates at some iteration x^k with $g_i(x^k) > 0$ for some i such that

$$g_i(x^k + (\xi - x_i^k)e_i) > 0, \qquad \forall\, \xi \in X_i. \tag{2}$$

Note that all $x \in X$ with $x_j \leq x_j^k$ for some j are infeasible, for otherwise as seen in part (a) the method will terminate with an optimal solution before reaching the point x^k. Therefore, if there is a feasible point y, then it must satisfy $x^k \leq y$. Applying the same reasoning as in the proof of part (a), we can show that

$$g_i(x^k + (y_i - x_i^k)e_i) \leq 0,$$

which contradicts (2). Therefore, whenever the method terminates because it can not generate a new iterate [see Eq. (2)], we know that the problem is infeasible.

Here is an example showing that if x^0 is not the lowest point of the grid $X_1 \times \cdots \times X_n$, then the algorithm can fail. Let $g_1(x) = -x_1 + 3x_2$, $g_2(x) = x_1 - 3x_2$, $X_1 = X_2 = \{0, 1, 2\}$. It is easy to see that the feasible set is singleton $\{(0, 0)\}$. However, unless $x^0 = 0$, the algorithm will stop not being able to find a new iterate, which can be misinterpreted as infeasibility of the problem.

(c) For all $i \notin I$, define

$$g_i(x) = \min_{\xi \in X_i} \xi - e_i' x,$$

so that the monotonicity property is satisfied for all g_i. Note that $g_i(x) \leq 0$ for all $i \notin I$ and all $x \in X$, and therefore, at any iteration k, the algorithm never updates the i-th coordinate of x^k for any $i \notin I$.

5.5.6

Statement: Let E be a matrix with entries -1, 0, or 1, and at most two nonzero entries in each of its columns. Show that E is totally unimodular if and only if the rows of E can be divided into two subsets such that for each column with two nonzero entries, the following hold: if the two nonzero entries in the column have the same sign, their rows are in different subsets, and if they have the opposite sign, their rows are in the same subset.

Solution: Note that E is totally unimodular if and only if its transpose E' is totally unimodular. Hence according to Exercise 5.5.5, an $m \times n$ matrix E is totally unimodular if and only if every $I \i \{1, \ldots, m\}$ can be partitioned into two subsets I_1 and I_2 such that

$$\Big|\sum_{i \in I_1} e_{ij} - \sum_{i \in I_2} e_{ij}\Big| \leq 1, \quad \forall\, j = 1, \ldots, n.$$

Let E be an $m \times n$ matrix with entries $e_{ij} \in \{-1, 0, 1\}$, and such that each of its columns contains at most two nonzero entries. By assumption, the set $\{1, \ldots, m\}$ can be partitioned into two subsets M_1 and M_2 so that if a column has two nonzero entries, the following hold:

(1) If both nonzero entries have the same sign, then one is in a row contained in M_1 and the other is in a row contained in M_2. (2) If the two nonzero entries have opposite sign, then both are in rows contained in the same subset. It follows that

$$|\sum_{i \in M_1} e_{ij} - \sum_{i \in M_2} e_{ij}| \leq 1, \quad \forall\, j = 1, \ldots, n. \tag{1}$$

Let I be any subset of $\{1, \ldots, m\}$. Then $I_1 = I \cap M_1$ and $I_2 = I \cap M_2$ constitute a partition of I, which in view of Eq. (1) satisfies

$$|\sum_{i \in I_1} e_{ij} - \sum_{i \in I_2} e_{ij}| \leq 1, \quad \forall\, j = 1, \ldots, n.$$

Hence E is totally unimodular.

5.5.9

Let $S = \{x \in P \mid x \in^n\}$. Let X_1, X_2, and X_3 be the sets described by conditions 1, 2, and 3, respectively. Consider $x \in S$. Since $Ax \leq b$ and $u \geq 0$, it follows that $x \in X_1$. Furthermore, $x \geq 0$, so rounding down the coefficients on the left hand side can only make the LHS smaller. Hence, $x \in X_2$. Finally, since x consists of integer components and so does $\lfloor u^T a_j \rfloor$, it follows that $\lfloor u^T a_j \rfloor x_j$ is integer and we can round down the right hand side and still satisfy the inequality. Thus $x \in X_3$.

Note that it is possible we could find $x \in P$ such that $x \notin X_3$ if x is not integer. Thus, we potentially can "cut off" points in P that are noninteger while retaining all integer points, as argued above. This commonly used practice in integer programming is known as *generating cutting planes.*