Math 482: Linear Programming, Fall 2020

Due Friday, December 11, 6PM CST

# Homework 10

1. *In chess, a queen is a piece that can move any number of spaces horizontally, vertically, or diagonally.*

   *The "eight queens" puzzle is to place eight queens on an $8 \times 8$ chessboard so that they do not attack each other. That is, no two queens can be placed in the same rank, the same file, or the same diagonal. (To be clear, diagonals are lines with slope $1$ or $-1$.)*

   *Write an integer linear program that generalizes this puzzle. For an arbitrary $n$, your program should find the maximum number of queens that can be placed on an $n \times n$ chessboard so that they do not attack each other.*

   *Since the number of variables and constraints will depend on $n$, it's fine if you just include one constraint of each type, and explain how it generalizes.*

   Number the rows and the columns from 1 to $n$. For $1 \leq i, j \leq n$, let $x_{ij}$ be an integer variable which we interpret as $x_{ij} = 1$ if there is a queen in row $i$, column $j$, and $x_{ij} = 0$ if that square is empty.

   To maximize the number of queens, we

   $$\text{maximize } \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}.$$

   No two queens can be placed in the same row or column, giving us the constraints:

   $$\sum_{i=1}^{n} x_{ij} \leq 1 \qquad \text{for } 1 \leq j \leq n$$

   $$\sum_{j=1}^{n} x_{ij} \leq 1 \qquad \text{for } 1 \leq i \leq n$$

   We write similar constraints for the two types of diagonals. These are annoying to write down in summation notation, but one way to do so formally is to write them as:

   $$\sum_{\substack{(i,j) \\ i+j=t}} x_{ij} \leq 1 \qquad \text{for } 1+1 \leq t \leq n+n$$

   $$\sum_{\substack{(i,j) \\ i-j=t}} x_{ij} \leq 1 \qquad \text{for } 1-n \leq t \leq n-1$$

   That is, in the first set of inequalities, we take all pairs $(i, j)$ with a fixed sum; in the second set, we take all pairs $(i, j)$ with a fixed difference. (It's fine if you only give a single example of constraint for each type of diagonal.)

   Finally, in addition to being integer variables, we want the $x_{ij}$ to be binary variables, and so we ask that

   $$0 \leq x_{ij} \leq 1 \qquad \text{for } 1 \leq i \leq n, 1 \leq j \leq n.$$

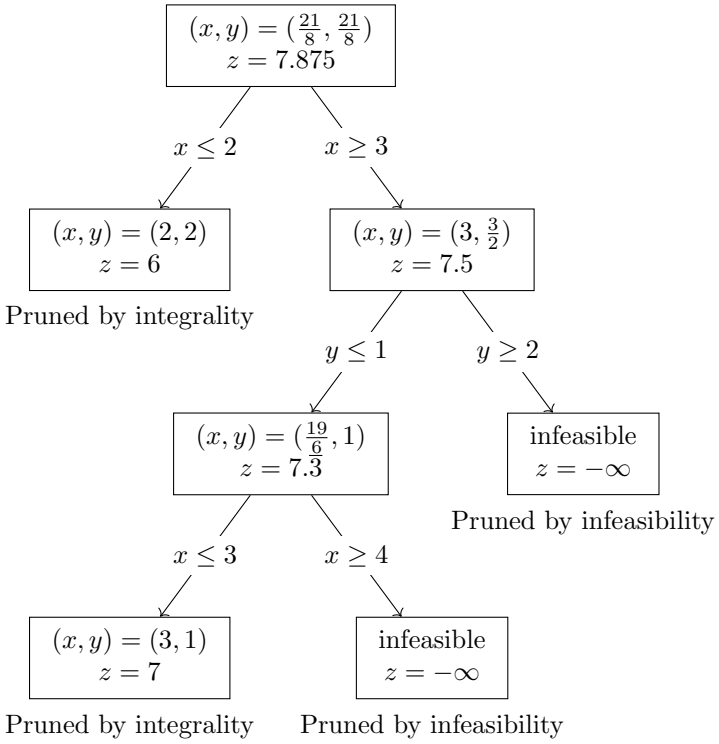
   The $\leq 1$ constraint can be dropped, because we can infer it from any of the other constraints where $x_{ij}$ shows up, but it's important not to forget about nonnegativity.

2. *Solve the integer linear program below using the branch-and-bound method.*

$$\begin{array}{lll} \underset{x,y\in\mathbb{Z}}{\textit{maximize}} & 2x + & y \\ \textit{subject to} & -x + & y \leq 0 \\ & 6x + & 2y \leq 21 \\ & x, y \geq 0 & \end{array}$$

*(Note: in some subproblems, you may be able to solve an LP by looking at it, without using the simplex algorithm. If you do, that's fine. But I do want you to write down, at each step of the branch-and-bound method, which LP you solve, what the optimal solution is, and what its objective value is.)*

Here is one possible tree obtained from the branch-and-bound method. Other solutions are possible, depending on the choice of branching variable, and the order in which the nodes are considered.




The optimal solution is $(x, y) = (3, 1)$, with objective value 7.

3. *Solve the integer linear program from problem 2 again, this time using fractional cuts.*

We begin by solving the LP relaxation:

| | $x$ | $y$ | $s_1$ | $s_2$ | |
|---|---|---|---|---|---|
| $s_1$ | $-1$ | 1 | 1 | 0 | 0 |
| $s_2$ | 6 | 2 | 0 | 1 | 21 |
| $-z$ | 2 | 1 | 0 | 0 | 0 |

$\rightsquigarrow$

| | $x$ | $y$ | $s_1$ | $s_2$ | |
|---|---|---|---|---|---|
| $x$ | 1 | 0 | $-1/4$ | $1/8$ | $21/8$ |
| $y$ | 0 | 1 | $3/4$ | $1/8$ | $21/8$ |
| $-z$ | 0 | 0 | $-1/4$ | $-3/8$ | $-63/8$ |

By rounding down the coefficients in the equation for $x$, $x - \frac{1}{4}s_1 + \frac{1}{8}s_2 = \frac{21}{8}$, we get the inequality $x - s_1 \leq 2$, which we can rewrite as $\frac{3}{4}s_1 + \frac{1}{8}s_2 \geq \frac{5}{8}$ by subtracting it from the original. Adding this

into the tableau as $-\frac{3}{4}s_1 - \frac{1}{8}s_2 + s_3 \leq -\frac{5}{8}$, and solving using the dual simplex method, we get

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | |
|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | $-1/4$ | $1/8$ | 0 | $21/8$ |
| $y$ | 0 | 1 | $3/4$ | $1/8$ | 0 | $21/8$ |
| $s_3$ | 0 | 0 | $-3/4$ | $-1/8$ | 1 | $-5/8$ |
| $-z$ | 0 | 0 | $-1/4$ | $-3/8$ | 0 | $-63/8$ |

$\rightsquigarrow$

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | |
|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | 0 | $1/6$ | $-1/3$ | $17/6$ |
| $y$ | 0 | 1 | 0 | 0 | 1 | 2 |
| $s_1$ | 0 | 0 | 1 | $1/6$ | $-4/3$ | $5/6$ |
| $-z$ | 0 | 0 | 0 | $-1/3$ | $-1/3$ | $-23/3$ |

From the equation $x + \frac{1}{6}s_2 - \frac{1}{3}s_3 = \frac{17}{6}$, we deduce the cut $x - s_3 \leq 2$ or $\frac{1}{6}s_2 + \frac{2}{3}s_3 \geq \frac{5}{6}$, which we can add into the tableau as $-\frac{1}{6}s_2 - \frac{2}{3}s_3 + s_4 = -\frac{5}{6}$:

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | |
|---|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | 0 | $1/6$ | $-1/3$ | 0 | $17/6$ |
| $y$ | 0 | 1 | 0 | 0 | 1 | 0 | 2 |
| $s_1$ | 0 | 0 | 1 | $1/6$ | $-4/3$ | 0 | $5/6$ |
| $s_4$ | 0 | 0 | 0 | $-1/6$ | $-2/3$ | 1 | $-5/6$ |
| $-z$ | 0 | 0 | 0 | $-1/3$ | $-1/3$ | 0 | $-23/3$ |

After solving this tableau using the dual simplex method, we get

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | |
|---|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | 0 | $1/4$ | 0 | $-1/2$ | $13/4$ |
| $y$ | 0 | 1 | 0 | $-1/4$ | 0 | $3/2$ | $3/4$ |
| $s_1$ | 0 | 0 | 1 | $1/2$ | 0 | $-2$ | $5/2$ |
| $s_3$ | 0 | 0 | 0 | $1/4$ | 1 | $-3/2$ | $5/4$ |
| $-z$ | 0 | 0 | 0 | $-1/4$ | 0 | $-1/2$ | $-29/4$ |

From the equation $x + \frac{1}{4}s_2 - \frac{1}{2}s_4 = \frac{13}{4}$, we deduce the cut $x - s_4 \leq 3$ or $\frac{1}{4}s_2 + \frac{1}{2}s_4 \geq \frac{1}{4}$, which we can add into the tableau as $-\frac{1}{4}s_2 - \frac{1}{2}s_4 + s_5 = -\frac{1}{4}$:

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | |
|---|---|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | 0 | $1/4$ | 0 | $-1/2$ | 0 | $13/4$ |
| $y$ | 0 | 1 | 0 | $-1/4$ | 0 | $3/2$ | 0 | $3/4$ |
| $s_1$ | 0 | 0 | 1 | $1/2$ | 0 | $-2$ | 0 | $5/2$ |
| $s_3$ | 0 | 0 | 0 | $1/4$ | 1 | $-3/2$ | 0 | $5/4$ |
| $s_5$ | 0 | 0 | 0 | $-1/4$ | 0 | $-1/2$ | 1 | $-1/4$ |
| $-z$ | 0 | 0 | 0 | $-1/4$ | 0 | $-1/2$ | 0 | $-29/4$ |

Finally, after we solve *this* tableau to optimality, we get a tableau with an integer optimal solution:

| | $x$ | $y$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | |
|---|---|---|---|---|---|---|---|---|
| $x$ | 1 | 0 | 0 | 0 | 0 | $-1$ | 1 | 3 |
| $y$ | 0 | 1 | 0 | 0 | 0 | 2 | $-1$ | 1 |
| $s_1$ | 0 | 0 | 1 | 0 | 0 | $-3$ | 2 | 2 |
| $s_3$ | 0 | 0 | 0 | 0 | 1 | $-2$ | 1 | 1 |
| $s_2$ | 0 | 0 | 0 | 1 | 0 | 2 | $-4$ | 1 |
| $-z$ | 0 | 0 | 0 | 0 | 0 | 0 | $-1$ | $-7$ |

As in the previous problem, we get the optimal solution $(x, y) = (3, 1)$ with objective value $z = 7$.

4. *Suppose that, in the setting of the traveling salesman problem, we have $n = 3k$ cities, and instead of trying to find a single tour of all of them, we want to find $k$ "triangle tours", each visiting 3 different cities, with the minimum total cost.*

*Starting from the incomplete formulation of TSP, add constraints ensuring that the solution* $\mathbf{x}$ *represents* $k$ *triangle tours.*

There are many ways to do it. One way is to ask that for every ordered triple $(h, i, j)$ of distinct cities, if we go from $h$ to $i$ and from $i$ to $j$, we must go from $j$ to $h$, completing a "triangle tour".

Let's write this as an inequality. We can write it as a OR of three statements: "we don't go from $h$ to $i$, OR we don't go from $i$ to $j$, OR we do go from $j$ to $h$." As long as any of the three statements hold, the condition is satisfied. Putting the OR statement in inequality form, we have

$$(1 - x_{hi}) + (1 - x_{ij}) + x_{jh} \geq 1$$

or equivalently

$$x_{hi} + x_{ij} - x_{jh} \leq 1.$$

(We can double-check this: the only way for $x_{hi} + x_{ij} - x_{jh}$ to be bigger than 1 is for it to be $1 + 1 - 0$, which is exactly the case we want to rule out.)

Adding this inequality for every ordered triple of distinct cities, we guarantee that the solution $\mathbf{x}$ is made up only of triangle tours.

5. *(Only 4-credit students need to do this problem.)*

*Suppose we are solving an integer linear program*

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{Z}^n}{\textit{maximize}} \quad & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \textit{subject to} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

*for which the branch-and-bound algorithm is going to take too long to find the optimal answer (call this unknown optimal answer* $\mathbf{x}^*$*). However, we're willing to settle for an approximate solution. Specifically, we want a* 2*-approximation: a feasible (integer) solution* $\mathbf{x} \in \mathbb{Z}^n$ *such that* $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \frac{1}{2}\mathbf{c}^\mathsf{T}\mathbf{x}^*$*. (Assume that* $\mathbf{c} \geq \mathbf{0}$*, so that none of these objective values are negative.)*

*Describe a modification of the branch-and-bound algorithm that finds a* 2*-approximation (and branches less often as a result).*

When looking for a 2-approximation, we can prune branches with non-integer solutions more aggressively. Specifically:

- As before, when encountering a node with an integer solution, we remember it as a lower bound on the optimal solution.
- As before, when encountering an infeasible node, we prune it because there's nothing else to do.
- However, when we encounter a node with a fractional solution $\mathbf{x}$, we prune it whenever $\mathbf{c}^\mathsf{T}\mathbf{x}$ is *at most twice* the lower bound from integer solutions we have already found.

This is guaranteed to get a 2-approximation. If we miss any integer points, they must have been solutions to one of the linear programs with fractional solutions that we pruned. But then, their objective values were at most twice our lower bounds; conversely, our lower bounds (which are integer solutions we found) achieve at least half the objective value of such points.