Math 30210 — Introduction to Operations Research

Assignment 12 (55 points total)

Due before class, Wednesday December 5, 2007

Instructions: Same as always

Reading: Sections 5.3, 9.1.1, 9.1.2, 9.2.1 and 9.3.

1. (8 points) Taha 5.4A Q1 (parts a) and c) only, and for data set i) only). Note that your answers for parts a) and c) should be the same ... if not, check your work.

2. (6 points) Taha 9.1A Q2 (plus optional: Taha 9.1A Q4).

3. (6 points) Taha 9.1B Q3 (plus optional: Taha 9.1B Q4).

4. (8 points) Taha 9.2A Q2 (part b) only). Ignore Taha's comment about initially branching on x_1; instead, branch on the variable which is furthest from being an integer, and choose x_1 if there is a tie.

5. (0 points, but please spend a few minutes on this) Taha 9.2A Q7 (part a) only).

6. (6 points) Taha 9.3B Q1 (part c) only).

7. (7 points) Taha 9.3C Q3.

8. (14 points total) This question is rather long to state, because I'm including a lot of recap for those of you who were not in class on the day before Thanksgiving. To begin, here is a recap of the load-balancing problem:

   Given n objects, numbered 1 through n, with object i having weight a_i, the objective is to put the objects on a balance scale in such a way as to make the scale as balanced as possible. For example, if there are three objects weighing 2, 3 and 4 pounds, then the best balance would be achieved by putting the 2 and 3 pound weights together on one side of the scale, and the 4 pound weight on its own on the other side.

   The load-balancing problem can be mathematically formulated by assigning variable x_i to item item i, with

$$x_i = \begin{cases} 1 & \text{if } i \text{ goes on right-hand side of scale} \\ -1 & \text{if } i \text{ goes on left-hand side of scale} \end{cases}$$

and then solving the following program:

Minimize $|a_1x_1 + a_2x_2 + \ldots + a_nx_n|$ subject to each $x_i = \pm 1$. (By the definition of x_i, the quantity inside the absolute value sign is the sum of the weights on the right-hand side of the scale minus the sum of the weights on the left-hand side; we want to make this as close to zero as possible, so we minimize the absolute value).

To make the objective linear one, we could break up the feasible space into two pieces, one in which the right-hand side of the scale is heavier than the left-hand side, and the other in which the right-hand side is lighter, and solve the two problems:

P1: Minimize $a_1x_1 + a_2x_2 + \ldots + a_nx_n$ subject to $a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq 0$ (this constraint encodes the fact that we are only considering assignments with the right-hand side heavier than the left) and all $x_i = \pm 1$, and

P2: Maximize $a_1x_1 + a_2x_2 + \ldots + a_nx_n$ subject to $a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq 0$ (a maximization since all feasible assignments here give negative objective values, and we want to get as close to zero as possible) and all $x_i = \pm 1$.

We then take as our solution to the problem the closest value to zero from among the above two problems.

(a) (3 points) Show that if the optimum for the first program is C, then the optimum for the second is $-C$; since these two values are equally close to zero, this says that in solving the assignment problem we need only problem P1.

(b) (7 points) Problem P1 has a linear objective function and a linear constraint, but it not yet of the right form to be a solvable integer programming problem, since the integral constraint is the very specific $x_i = \pm 1$ for each i, rather than the more general $x_i \in \mathbb{Z}$. We cannot simply resolve this issue by adding the constraints $-1 \leq x_i \leq 1$ and $x_i \in \mathbb{Z}$, since that doesn't rule out $x_i = 0$.

Find a way to encode the fact that $x_i = \pm 1$ using only linear constraints, and the general constraint that all variables must be integers.

(c) Suppose n football players get together and want to break into two evenly matched teams to play a game. Each player i has strength a_i, and the strength of a team is the sum of the strengths of its players. By "evenly matched" it is meant that the two teams have strength as near equal as possible. Clearly, this is a load balancing problem.

  i. (2 points) The players decide that the division into teams should be subject to the constraint that the sizes of the teams be as near equal as possible (differing by only one if n is odd, and equal if n is even). What linear constraints should be added to the problem to encode this constraint?

  ii. (2 points) The players decide that 1 and 2 should be the two team captains, and so can't play on the same team. What linear constraints should be added to the problem to encode this constraint?