Operations Research, Spring 2021 (109-2)
Homework 2

Ling-Chieh Kung[*]

April 6, 2021

1 Rules

- This homework is due at ***23:59 am, April 18***. Those who submit their works late but are late by less than one hour gets 10 points off. Works that are late more than one hour get no point.
- For this homework, students should work individually. While discussions are encouraged, copying is prohibited.
- Please submit a ***PDF file*** through NTU COOL and make sure that the submitted work contains the student ID and name. Those who fail to do these will get 10 points off.
- You are required to ***type*** your work with LaTeX (strongly suggested) or a text processor with a formula editor. Hand-written works are not accepted. You are responsible to make your work professional in mathematical writing by following at least the following rules:[1]
  1. When there is a symbol denoted by an English letter, make it italic. For example, write $a + b = 3$ rather than $\mathrm{a} + \mathrm{b} = 3$.
  2. An operator (e.g., $+$) should not be italic. A function with a well-known name (e.g., log, max and sin) is considered as an operator.
  3. A number should not be italic. For example, it should be $a + b = 3$ rather than $a + b = \mathit{3}$.
  4. Superscripts or subscripts should be put in the right positions. For example, a_1 and $a1$ are completely different: The former is a variable called a_1 while the latter is actually $a \times 1$.
  5. When there is a subtraction, write $-$ rather than -. For example, write $a - b = 3$ rather than a - $b = 3$. The same thing applies to the negation operator. For example, write $a = -3$ rather than $a =$ -3.
  6. If you want to write down the multiplication operator, write $\times$ rather than *.
  7. For an exponent, write it as a superscript rather than using ^. For example, write 10^2 rather than 10^2.
  8. There should be proper space between a binary operator. For example, it should be $a + b = 3$ rather than a+b=3.

  Those who fail to follow these rules may get at most 10 points off.
- As we may see, there are many students, many problems, but only a few TAs. Therefore, when the TAs grade this homework, it is possible for only some problems to be randomly selected and graded. For all problems, detailed suggested solutions will be provided.

[*]Department of Information Management, National Taiwan University. E-mail: lckung@ntu.edu.tw.

[1]A more complete list of formatting rules is on NTU COOL.

2 Problems

1. (25 points) Consider the following LP

$$\begin{aligned}\max \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 - x_2 \leq 4 \\ & x_1 + x_2 + x_3 \leq 9 \\ & x_3 \geq 3 \\ & x_i \geq 0 \quad \forall i = 1, ..., 3.\end{aligned}$$

   (a) (5 points) Find the standard form of this LP.

   (b) (5 points) List all the basic solutions and basic feasible solutions for the standard form of this LP.

   (c) (5 points) List all the extreme points of the feasible region of the original LP. DO NOT prove that they are extreme points; just list them.

   (d) (10 points) Use the simplex method (with the two-phase implementation, if needed) and the smallest index rule to solve the LP. Is there any iteration that has no improvement?

2. (15 points) Consider the following LP.

$$\begin{aligned}\max \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 - x_2 \leq 4 \\ & x_1 + x_2 + x_3 \leq 7 \\ & x_3 \geq 3 \\ & x_i \geq 0 \quad \forall i = 1, ..., 3.\end{aligned}$$

   (a) (5 points) An LP is *degenerate* if there are two bases correspond to the same basic solution (i.e., the values of all basic variables are identical). Use this definition to determine whether the standard form of the given LP is degenerate.

   (b) (10 points) Use the simplex method (with the two-phase implementation, if needed) and the smallest index rule to solve the LP. Is there any iteration that has no improvement? If so, highlight it.

3. (15 points) Consider the following integer program, which represents an instance of the "two-copy knapsack problem:"

$$\begin{aligned}\max \quad & 3x_1 + 2x_2 + 5x_3 + 9x_4 + 10x_5 \\ \text{s.t.} \quad & x_1 + 4x_2 + 3x_3 + 5x_4 + 3x_5 \leq 20 \\ & x_i \in \{0, 1, 2\} \quad \forall i = 1, ..., 5.\end{aligned}$$

   (a) (5 points) Use the greedy algorithm introduced in class to solve the linear relaxation of this integer program.

   (b) (10 points) Use the branch-and-bound algorithm to solve the original integer program. Depict the full branch-and-bound tree. Do not write down the solution process of each node; write down just an optimal solution and its objective value of each node.

4. (30 points) A city is divided into n districts. The time (in minutes) it takes an ambulance to travel from District i to District j is denoted as d_{ij}. The population of District i (in thousands) is p_i. An example is shown in Table 1. In this instance, we have $n = 8$ districts. We may see that, e.g., it takes 5 minutes to travel from District 2 to District 3, and there are 40,000 citizens living in District 1.

   The city has m ambulances and wants to locate them to m of the districts. For each district, the *population-weighted firefighting time* is defined as the product of the district population times

District (from)	District (to)								Population
	1	2	3	4	5	6	7	8	
1	0	3	4	6	8	9	8	10	40
2	3	0	5	4	8	6	12	9	30
3	4	5	0	2	2	3	5	7	35
4	6	4	2	0	3	2	5	4	20
5	8	8	2	3	0	2	2	4	15
6	9	6	3	2	2	0	3	2	50
7	8	12	5	5	2	3	0	2	45
8	10	9	7	4	4	2	2	0	60

Table 1: Data for Problem 2

the amount of time it takes for the closest ambulance to travel to it. The decision maker aims to locate the m ambulances to minimize the *maximum* population-weighted firefighting time among all districts.

As an example, suppose that $m = 2$, $n = 8$, d_{ij} and p_i are provided in Table 1, and the two ambulances are located in District 1 and 8. We then know that for Districts 1, 2, and 3 the closest ambulance is in District 1 and for the remaining five districts the closet ambulance is in District 8. The firefighting time for the eight districts are thus 0, 3, 4, 4, 4, 2, 2, and 0 minutes, respectively. The population-weighted firefighting times may then be calculated as 0, 90, 140, 80, 60, 100, 90, and 0. The maximum among the eight districts is therefore 140.

(a) (5 points) Formulate an integer program that can minimize the maximum population-weighted firefighting time among all districts.

(b) (5 points) Write a program to invoke a solver (e.g., write a Python program to invoke Gurobi Optimizer) to solve the above instance with $m = 3$ and find an optimal solution. Write down your optimal solution in a way that a practitioner may understand.

(c) (10 points) For any value of m, consider the following heuristic algorithm which runs m iterations. In each iteration, we locate an ambulance in a district that (1) currently does not have an ambulance, and (2) may minimize the maximum population-weighted firefighting times among all districts. If there are multiple districts satisfying these two conditions, pick the one with the smallest district ID among them. We then proceed to the next iteration to look for the next district to locate an ambulance.

Consider a tiny example with $n = 4$, $m = 2$, and d_{ij} and p_j are provided in Table 2. To locate the first ambulance, we examine the maximum population-weighted firefighting times of locating an ambulance in Districts 1, 2, 3, and 4 as 140, 175, 160, and 240, respectively. We will choose District 1. To locate the second ambulance, we take the ambulance in District 1 as given and examine the maximum population-weighted firefighting times of locating an ambulance in Districts 2, 3, and 4 as 140, 90, and 90, respectively. We will choose District 3. The final objective value of locating two ambulances in Districts 1 and 3 is 90 (which is 30×3 for District 2).

District (from)	District (to)				Population
	1	2	3	4	
1	0	3	4	1	40
2	3	0	5	8	30
3	4	5	0	1	35
4	1	8	1	0	5

Table 2: Data for Problem 3

Coming back to the instance provided in Table 1 with $n = 8$. Now let $m = 3$. Implement the heuristic algorithm introduced above to generate a feasible solution. Copy and paste your program (in C++, Python, Java, or whatever programming language you like) onto your report. Write down your optimal solution in a way that a practitioner may understand.

(d) (5 points) Use your programs developed in Parts (b) and (c) for $m = 1, 2, ...,$ and 8. For each value of m, compare the objective values obtained by the two methods and calculate the optimality gap with respect to an optimal solution. Use a table to present the objective values and optimality gaps.

5. (20 points; 10 points each) Consider the EOQ problem of finding an order quantity q to minimize

$$f(q) = \frac{KD}{q} + \frac{hq}{2}$$

with $K = 80$, $D = 8000$, $h = 2$.

(a) Start from $q^0 = 2000$ to run two iterations of the gradient descent method to solve this instance. In iteration k, use $\frac{1}{k}$ as the step size (more precisely, move from x^{k-1} to $x^{k-1} - \frac{1}{k}\nabla f(x^{k-1})$ in iteration k). Write down the detailed process of the two iterations.

(b) Start from $q^0 = 2000$ to run two iterations of the Newton's method to solve this instance. Write down the detailed process of the two iterations.