**BUAD 5022 – Individual Problem Set 4**

Due: December 15 by 11:59pm via Blackboard submission

Grade: 20% of your overall class grade

Category A Assignment (i.e., do not discuss with anyone else except the instructor)

*Homework should be neat and organized. It is subject to grade penalties if it is not.*

**1.** Simplex City has been divided into 8 districts. The time (in minutes) it takes an ambulance to travel from one district to another is shown below**.** The population of each district (in thousands) is as follows:

District 1, 40; District 2, 30; District 3, 35; District 4, 20; District 5, 15; District 6, 50; District 7, 45; District 8, 60. The city has only two ambulances and wants to locate them **to maximize the number of people** who **live within 2 minutes** of an ambulance.

* Formulate the model as an IP.
* Solve the model using Python (submit your code).
* Summarize the solution in a succinct and presentable manner.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | To District | | | | | | | |
| From District | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 3 | 4 | 6 | 8 | 9 | 8 | 10 |
| 2 | 3 | 0 | 5 | 4 | 8 | 6 | 12 | 9 |
| 3 | 4 | 5 | 0 | **2** | **2** | 3 | 5 | 7 |
| 4 | 6 | 4 | **2** | 0 | 3 | **2** | 5 | 4 |
| 5 | 8 | 8 | **2** | 3 | 0 | **2** | **2** | 4 |
| 6 | 9 | 6 | 3 | **2** | **2** | 0 | 3 | **2** |
| 7 | 8 | 12 | 5 | 5 | **2** | 3 | 0 | **2** |
| 8 | 10 | 9 | 7 | 4 | 4 | **2** | **2** | 0 |

From the table, we can see that only 3,4,5,6,7,8 districts are the districts within 2 minutes.

Variables:

Let x\_ij be the binary variable for ambulance i in district j, where i=1,2; j=3,4,5,6,7,8

Let x\_i be the binary variable if ambulance is assigned to i.

Objective: Max 35\*x3+20\*x4+15\*x5+50\*x6+45\*x7+60\*x8

Subject To:

x\_13+ x\_14 x\_15+ x\_16+ x\_17+ x\_18 = 1 (assign ambulance 1 to an area)

x\_23+ x\_24 x\_25+ x\_26+ x\_27+ x\_28 = 1 (assign ambulance 2 to an area)

x\_13+x\_23 <=1 (district 3 ambulance)

x\_14+x\_24 <=1 (district 4 ambulance)

x\_15+x\_25 <=1 (district 5 ambulance)

x\_16+x\_26 <=1 (district 6 ambulance)

x\_17+x\_27 <=1 (district 7 ambulance)

x\_18+x\_28 <=1 (district 8 ambulance)

x3 <= x13+x23+x14+x24+x15+x25

x4 <= x13+x23+x16+x26+x14+x24

x5 <= x13+x16+x17+x23+x26+x27+x15+x25

x6 <= x14+x15+x16+x18+x24+x25+x26+x28

x7 <= x15+x17+x18+x25+x27+x28

x8 <= x16+x17+x18+x26+x27+x28

x\_ij >=0 for all i, j (non-negativity constraint)

status=Optimal

x13 = 1.0, x28 = 1.0

Objective = 215. Ambulance 1 located in district 3 and ambulance 2 located in district 8, the people lived within 2 mins of an ambulance is maximize at 225 thousands.

**2.** Joe Inc. sells air conditioners. **The annual demand for air conditioners in each region of the country is as follows: East, 100,000; South, 150,000; Midwest, 110,000; West, 90,000.** Joe Inc. is considering building the air conditioners in four different cities: New York, Atlanta, Chicago, and Los Angeles. The cost of producing an air conditioner in a city and then shipping it to a region is given in the first table below. **Any factory can produce as many as 150,000 air conditioners per year.**  The annual fixed cost of operating a factory in each city is given in the second table below. **At least 50,000 units of the Midwest demand for air conditioners must come from New York, or at least 50,000 units of the Midwest demand must come from Atlanta**. Joe Inc. would like to **minimize annual cost** of meeting demand for air conditioners.

* Formulate the model as an IP.
* Solve the model using Python (submit your code).
* Summarize the solution in a succinct and presentable manner.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Price by Region (Per Unit, $) | | | |
| City | East | South | Midwest | West |
| New York | 206 | 225 | 230 | 290 |
| Atlanta | 225 | 206 | 221 | 270 |
| Chicago | 230 | 221 | 208 | 262 |
| Los Angeles | 290 | 270 | 262 | 215 |

|  |  |
| --- | --- |
| City | Annual Fixed Cost ($ Million) |
| New York | 6 |
| Atlanta | 5.5 |
| Chicago | 5.8 |
| Los Angeles | 6.2 |

Variables:

Let x\_ij be the number of conditioner produced in city i and shipped to region j.

Let Y\_i be the binary variable of whether a factory is open or not.

Let b be the binary variable for controlling Midwest demand from New York or Atalanta.

Objective: Min.

Subject To:

(Demand constraint for East)

(Demand constraint for south)

(Demand constraint for Midwest)

(Demand constraint for west)

(50000 Midwest demand from New York)

(50000 Midwest demand from Atlanta)

(Upper limit for each factory if factory is open)

(non-negativity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | East | South | Midwest | West |
| New York | 100000 | 0 | 50000 | 0 |
| Atlanta | 0 | 150000 | 0 | 0 |
| Chicago | 0 | 0 | 0 | 0 |
| Los Angeles | 0 | 0 | 60000 | 90000 |

The optimal objective is minimized at 115770000.0.

**3.** Joe’s Fun Bus operates buses between Baltimore and Boston. A bus trip between these two cities takes 6 hours. Federal law requires that a driver rest for four or more hours between trips**. A driver’s workday consists of two trips: one from Baltimore to Boston and one from Boston to Baltimore (or vice versa).** The table below gives the departure times for the buses. Joe’s Fun Bus wants to create a schedule to **minimize the sum of downtime for all drivers**. How should Joe’s Fun Bus assign crews?

Note: It is permissible for a driver’s “day” to overlap midnight. For example, a Baltimore-based driver can be assigned to the Baltimore-Boston 3PM trip and the Boston-Baltimore 6AM trip.

* Formulate the model as an Assignment problem.
* Solve the model by-hand using the Hungarian Algorithm.
* Solve the Assignment problem formulation model using Python (submit your code).
* Summarize the solutions in a succinct and presentable manner.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trip | Departure Time |  | Trip | Departure Time |
| Boston 1 | 6:00 AM |  | Baltimore 1 | 5:30 AM |
| Boston 2 | 7:30 AM |  | Baltimore 2 | 9:00 AM |
| Boston 3 | 11:30 AM |  | Baltimore 3 | 3:00 PM |
| Boston 4 | 7:00 PM |  | Baltimore 4 | 6:30 PM |
| Boston 5 | 12:30 AM |  | Baltimore 5 | Midnight |

Variables:

Let xij be the binary for bus from Baltimore to Boston, i=1,2,3,4,5; j=1,2,3,4,5.

Let xji be the binary for bus from Boston to Baltimore.

Objective Minimize:

Subject To:

x11+x12+x13+x14+x15 = 1 (Baltimore 1)

x21+x22+x23+x24+x25 = 1 (Baltimore 2)

x31+x32+x33+x34+x35 = 1 (Baltimore 3)

x41+x42+x43+x44+x45 = 1 (Baltimore 4)

x51+x52+x53+x54+x55 = 1 (Baltimore 5)

x11+x21+x31+x41+x51 = 1 (Boston 1)

x12+x22+x32+x42+x52 = 1 (Boston 2)

x13+x23+x33+x43+x53 = 1 (Boston 3)

x14+x24+x34+x44+x54 = 1 (Boston 4)

x15+x25+x35+x45+x55 = 1 (Boston 5)

18.5 20 24 7.5 13 lambda =7.5

15 16.5 20.5 4 9.5 4

9 10.5 14.5 22 27.5 9

5.5 7 11 18.5 24 5.5

24 25.5 5.5 13 18.5 5.5

11 12.5 16.5 0 5.5

11 12.5 16.5 0 5.5

0 1.5 5.5 13 18.5

0 1.5 5.5 13 18.5

18.5 20 0 7.5 13

Mu=0 1.5 0 0 5.5

11 11 16.5 0 0

11 11 16.5 0 0

0 0 5.5 13 13

0 0 5.5 13 13

18.5 18.5 0 7.5 7.5

Min. 7.5+9.5+9+7+5.5 = 38.5

Problem

status=Optimal

x\_15 = 1.0

x\_24 = 1.0

x\_31 = 1.0

x\_42 = 1.0

x\_53 = 1.0

Objective = 38.5

**4.** A truck must **travel from New York to Los Angeles.** As shown in the figure below, a variety of routes are available. The number associated with each arc is the number of gallons of fuel required by the truck to traverse the arc. Your goal is **the minimize the amount of fuel used**.

* Formulate the model as a Shortest Path problem.
* Solve the model by-hand using Dijkstra’s Algorithm.
* Solve the Shortest Path formulation model using Python (submit your code).
* Summarize the solutions in a succinct and presentable manner.



Let (i,j) be an arc from node i to node j and let m denotes the terminal node.

Decision Variables: xij = 1 if (i,j) is in the shortest path; 0 otherwise.  
  
Data: cij = cost to traverse arc (i, j)

Think of shipping one unit from node 1 to node m.

Formulation:  
  
s.t.  
  


Objective: 2450.0

x\_1,3 1.0

x\_3,5 1.0

x\_5,8 1.0

1300

400

600

1000

900

600

1200

600

900

1100

400

950

800

1800

2

5

8

1

4

3

7

6

Node 8

(1,3,5,8)

2050+400=2450

(1,2,6,8)

1300+1300=2600

(1,4,7,8)

2000+600=2600

P0 = 1

Node 7

(1,2,6,7)

1300+1000=2300

(1,4,7)

800+1200=2000

Node 6

(1,2,6)

400+900=1300

(1,3,6)

950+600=1550

(1,4,6)

800+600=1400

(1,3,5,6)

2050+900=2950

Node 5

(1,2,5)

400+1800=2200

(1,3,5)

950+1100=2050

Node 2

P=400

Node 3

T = 950

Node 4

T = 800