BANA 7020
OPTIMIZATION
Section - 2

Final Project

Submitted by

Himaja Gaddam

Srujana Guduru

Praveen Guntaka

Vamsi Chand Emani

Nikhila Nayana Bobba

INTRODUCTION

In daily life problems several times there is a need to transport the product from various sources to different destinations. To find a way to transport the product in such a manner so that the total transportation cost is minimum is called the optimal way.

Transportation problem is type of linear programming problem which is focused at transporting the goods and services at a minimum cost or time.

It involves the details of various sending and receiving locations aimed at designing a formula which will give the best possible results.

It is very useful in business and industry for maximization of profit, reducing the transportation time and ensuring safe delivery of goods.

PROBLEM STATEMENT

A transportation company is evaluating a prototype system that combines trucks and drones for last-mile delivery services. The test run considers a set of orders that must be delivered to known locations. A delivery truck starts from a depot and visits "launch sites" corresponding to the customers locations. From each launch site, the truck deploys a series of drones that deliver the orders and return back to meet the truck at the launch site. Once all the drones are recovered, the truck moves to the next launch site and repeats the process until all orders are delivered.

CONSTRAINTS

Constraints for selecting the entry are:

a. Generate random x and y coordinates between 0 and 100 for each order. Repeat this process 10 times to generate a testbed of problem instances.
b. Assume that the depot is located at the origin (0,0).
c. Build a Euclidean distance matrix between all the orders using the coordinates generated. Round the distances to the nearest integer.
d. The truck must start and end at the depot.
e. The truck can deploy up to K drones at each stop in a launch site.
f. The drones have limited cargo capacity and as a result they can only visit 1 customer (not counting the starting point) before returning to the truck.
g. Each customer should be visited at least once by a drone. The customer locations used as launch sites will be visited by the truck and also will serve as the starting and ending point of a drone tour.
h. For each unit of distance traveled by the truck there is a cost of $10, while for each unit of distance traveled by a drone there is a cost of $3.
i. Consider the objective of minimizing the total travel cost (both by the trucks and the drones).

MATHEMATICAL FORMULATION

This problem is modeled as a mixed - integer linear program as follows:

Variables:

nodes=< no.of customer visiting points >

$$N = \{1,2, \ldots., nodes + 2\}$$

$$N_1 = \{1,2, \ldots, nodes + 1\}$$

$$P = \{1, \ldots, drones\}$$

Decision Variables:

$$x_i = 1, If\ the\ delivery\ is\ made\ to\ the\ customer\ by\ truck$$

$$y_i = 1, If\ the\ delivery\ is\ made\ to\ the\ customer\ by\ drone$$

$$A_{ij} = 1\ , if\ Arc\ of\ the\ truck\ \ from\ node\ i\ to\ node\ j\ is\ used$$

$$B_{ijk} = 1, if\ Arc\ of\ the\ drone\ k\ from\ node\ i\ to\ j\ is\ used\ by\ drone\ k$$

$$d_{ij} = Distance\ between\ node\ i\ and\ node\ j$$

$$u_i = position\ of\ nodes$$

Parameters:

A_{ij}= {0,1}

$$u_i \geq 1$$

B_{ijk} *= {0,1}*

Given fixed values: Distance matrix, d_{ij}

Cost per unit distance for truck, $C_t = 10$

Cost per unit distance for drone, $C_d = 3$

Objective function:

$$Z = \sum_{i \in N} \sum_{j \in N} d_{ij} * C_t * A_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in P} d_{ij} * 2 * C_d * B_{ijk}$$

Minimize(Z)

Constraints:

Variables are binary & position is integer

$$x_i \text{ is binary } \forall i \in N$$

$$y_i \text{ is binary } \forall i \in N$$

$$A_{ij} \text{ is binary } \forall i, j \in N$$

$$B_{ijk} \text{ is binary } \forall i, j \in N \;\&\; k \in P$$

$$u_i \text{ is integer } \forall i \in N$$

Every node should be visited exactly once by a drone or truck

$$x_i + y_i = 1 \; \forall i \in N$$

1st(depot) & last (duplicate depot) nodes should always be visited by truck

$$\boldsymbol{x_1} = 1$$

$$\boldsymbol{x_{nodes+2}} = 1$$

`

Truck must start and end at the depot

$$\sum_{j \in N} A_{1,j} = 1$$

$$\sum_{j \in N} A_{j,nodes+2} = 1$$

$$\sum_{i \in N} A_{i1} = 0$$

$$\sum_{i \in N} A_{nodes+2,i} = 0$$

No arc should be active between truck nodes

$$A_{ii} = 0 \; \forall i \in N$$

$$B_{iik} = 0 \; \forall i \in N \;\&\; \text{k} \in P$$

Incoming and Outgoing constraints for arcs

$$\sum_{j \in N} A_{ij} = 1 * x_i \; \forall i \in N1$$

$$\sum_{j \in N} A_{ji} = 1 * x_i \; \forall i \in N1$$

$$\sum_{i \in N} A_{ij} + \sum_{i \in N} \sum_{k \in P} B_{ijk} = 1 \; \forall \, j \in N1$$

Constraints to enable or disable drone arcs

$$\sum_{j\in N}\sum_{k\in P} B_{jik} <= drones * x_i \ \ \forall i \in N$$

$$\sum_{j\in N}\sum_{k\in P} B_{jik} = 1 * y_i \ \ \forall i \in N$$

$$\sum_{j\in N} B_{i,j,k} <= 1 \ \ \forall i \in N, k \in P$$

$$B_{1,i,k} = 0 \ \forall j \in N, k \in P$$

$$B_{nodes+2,j,k} = 0 \ \forall j \in N, k \in P$$

Constraints for ordering nodes

$$u_1 = 1$$

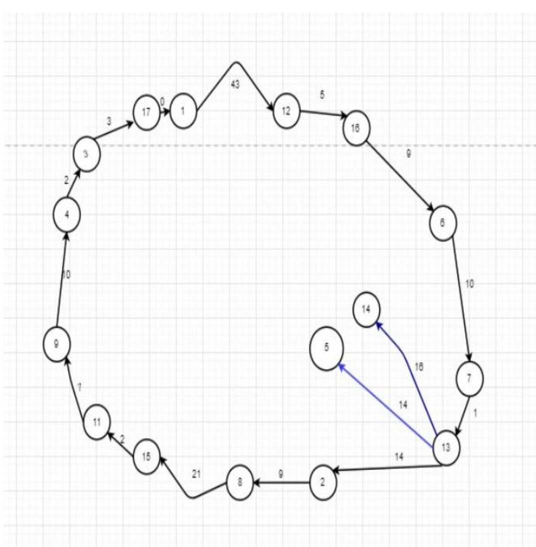
$$u_{nodes+2} = \sum_{i\in N} x_i$$

$$u_j >= u_i + 1 - 100 * \left(1 - A_{ij}\right) \ \forall i \ \in N$$

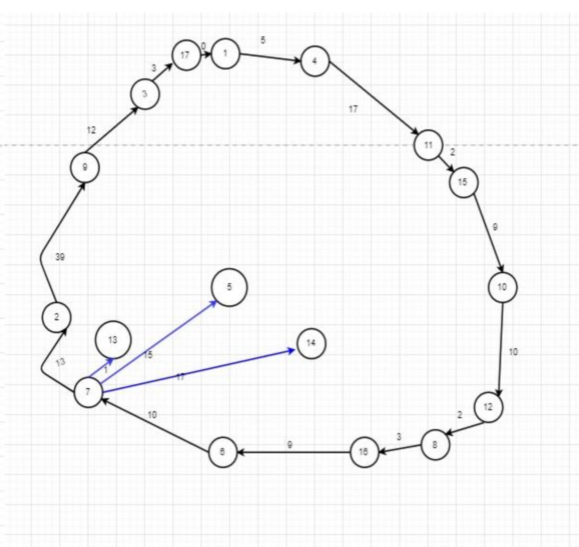
RESULTS

Executed above formulations in Xpress for 15 customer nodes and plotted the route map below:

Drones = 2



Drones = 3, 4 and 5



Inference: With this sample, there was no difference in objective for drones = 3,4 & 5.

We executed these formulations for 10 other test beds with 10 customer nodes with cost/unit distance for drone = 3 & cost/unit distance for truck = 10

Results for these test beds are presented below:

Test bed	No of drones	No.of nodes visited by drones	Total cost for truck	Total cost for drones	Optimal cost
1	2	2	500	30	530
1	3	3	480	48	528
1	4	3	480	48	528
1	5	3	480	48	528
2	2	3	240	132	372
2	3	3	240	126	366
2	4	3	240	126	366
2	5	3	240	126	366
3	2	2	400	30	430
3	3	3	380	48	428
3	4	3	380	48	428
3	5	3	380	48	428
4	2	2	380	72	452
4	3	3	360	90	450
4	4	3	360	90	450
4	5	3	360	90	450
5	2	2	380	126	506
5	3	3	340	162	502
5	4	3	340	162	502
5	5	3	340	162	502
6	2	3	380	162	542
6	3	3	380	156	536
6	4	3	380	156	536
6	5	3	380	156	536
7	2	2	400	102	502
7	3	3	380	120	500
7	4	3	380	120	500
7	5	3	380	120	500

Test bed	No of drones	No.of nodes visited by drones	Total cost for truck	Total cost for drones	Optimal cost
8	2	2	440	54	494
8	3	3	360	126	486
8	4	3	360	126	486
8	5	3	360	126	486
9	2	3	300	150	450
9	3	3	300	144	444
9	4	3	300	144	444
9	5	3	300	144	444
10	2	2	400	84	484
10	3	3	360	120	480
10	4	3	360	120	480
10	5	3	360	120	480

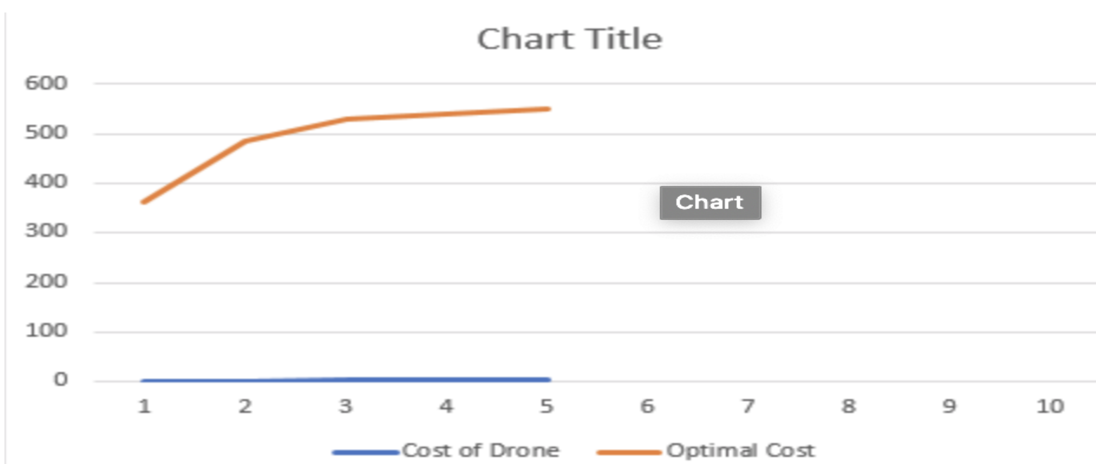
Cost sensitivity analysis:

Executed the above formulations to do cost sensitivity analysis once by keeping cost/unit distance for truck constant and changing cost/unit distance for drone and again by keeping cost/unit distance for drone constant and changing cost/unit distance for truck. Results from this analysis are presented below.

Results when cost/unit distance for drone is changing:

Here, cost/unit distance for truck is taken as 10 and number of drones used are 2

Cost of Drone	No.of nodes visited by drones	Cost of Truck	Cost of Drone	Optimal Cost
5	1	540	10	550
4	2	500	40	540
3	2	500	30	530
2	4	260	224	484
1	5	220	142	362

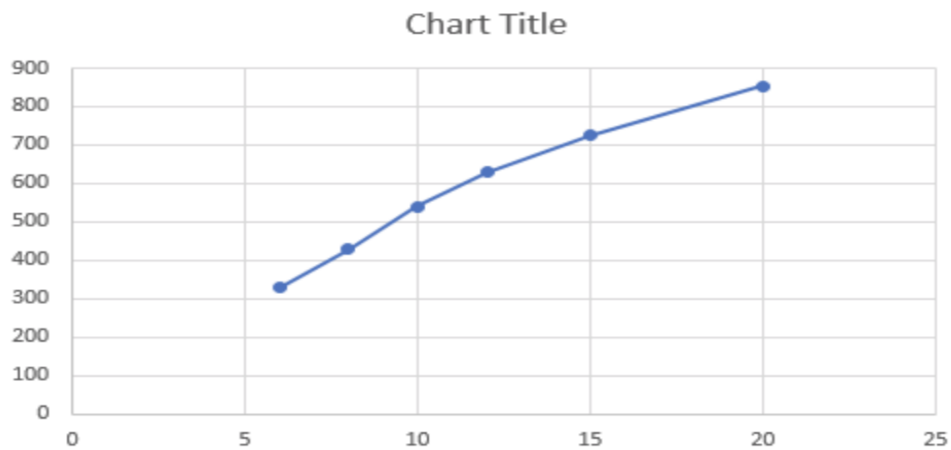


Inference: From the above table we can observe that the optimal cost is decreasing, and number of nodes visited by nodes is increasing as cost/unit distance for drone is decreasing.

Results when cost/unit distance for truck is changing:

Here, cost/unit distance for drone is taken as 3 and number of drones used are 2

Cost of Truck	No.of nodes visited by drones	Cost of Truck	Cost of Drone	Optimal Cost
6	1	324	6	330
8	2	400	30	430
12	2	600	30	630
15	4	390	336	726
20	4	520	336	856



Inference: From the above table we can observe that the optimal cost is increasing, and number of nodes visited by nodes is also increasing as cost/unit distance for truck is increasing.

Xpress code:

```
model Project
  uses "mmxprs"


  declarations
    nodes=10
    drones=2
    N=1..(nodes+2)       !all nodes including starting and ending points
    N1=2..(nodes+1)      !all customer nodes
    P=1..drones      !number of drones
    x: array(N) of mpvar   !truck nodes
    y: array(N) of mpvar !drone nodes
    A: array(N,N) of mpvar !truck route arcs
    B: array(N,N,P) of mpvar !drone node arcs
    d: array(N,N) of integer  ! distance betweenn nodes
    u: array(N) of mpvar      !ordering for nodes
  end-declarations
    forall(i in N,j in N) do
    A(i,j) is_binary
    end-do
    forall(i in N,j in N,k in P) do
    B(i,j,k) is_binary
  end-do
  !data from external file
  initializations from "distancematrix.txt"
   d
  end-initializations
  !objective function definition
  obj:=sum(i in N,j in N) d(i,j)*10*A(i,j) + sum(k in P) sum(i in N,j in N) d(i,j)*8*B(i,j,k)
!constraints
  forall(i in N) x(i) is_binary
  forall(i in N) y(i) is_binary
  forall(i in N) u(i) is_integer
  forall(i in N) x(i)+y(i)=1
  x(1)=1
  x(nodes+2)=1
```

```
  sum(j in N) A(1,j) = 1
  sum(j in N) A(j,nodes+2)=1
  forall(i in N) A(i,1)=0
  forall(j in N) A(nodes+2,j)=0
  forall(i in N1) sum(j in N) A(i,j)=1*x(i)
  forall(i in N1) sum(j in N) A(j,i)=1*x(i)
  !forall(i in N) (sum(j in N) A(i,j)) + (sum(j in N) sum(k in P) B(i,j,k))=1
  forall(j in N1) (sum(i in N) A(i,j)) + (sum(i in N, k in P) B(i,j,k))=1

  !A(1,12)=0
  !A(12,1)=0
  forall(i in N) A(i,i)=0
  forall(i in N) forall(k in P) B(i,i,k)=0

forall(i in N) sum(j in N) sum(k in P) B(i,j,k) <= drones*x(i)
 forall(j in N) sum(i in N) sum(k in P) B(i,j,k) = 1*y(j)
 !forall(i in N) sum(j in N,k in P) B(i,j,k) < 0
 forall(i in N,k in P) sum(j in N) B(i,j,k)<= 1
!forall(j in N) sum(i in N) sum(k in P) B(i,j,k)=x(j)
forall(j in N) forall(k in P) B(1,j,k)=0
forall(j in N) forall(k in P) B(nodes+2,j,k)=0
  u(1)=1
  u(nodes+2)=sum(i in N) x(i)
  forall(i in N) forall(j in N) u(j)>=u(i)+1-(100*(1-A(i,j)))

!obj function
  minimize(obj)

!output
  writeln("Optimal cost:",getobjval)
  forall(i in N | getsol(x(i))>0) do
   writeln("truck node ",i)
  end-do
  forall(j in N | getsol(y(j))>0) do
   writeln("drone node ",j)
```

```
  end-do
  forall(i in N,j in N| getsol(A(i,j))>0) do
   writeln("truck arc used ",i," to ",j)
  end-do
  forall(i in N,j in N,k in P| getsol(B(i,j,k))>0) do
   writeln("drone arc used ",i," to ",j," with drone ",k)
  end-do
end-model
```