BANA7020 - 002

Optimization Final Project

Group Members:

Shukla, Anupam M13469377

Kumar, Ankit M13436962

Sankhe, Nikita M13437885

Sonavani, Apoorva M13456525

Abstract:

The solution is designed to help a transportation company in last-mile delivery services utilizing trucks and drones. A delivery truck starts at the depot and visits launch sites corresponding to the customer 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 the orders are delivered.

The report contains the following sections:

1. Model Formulation, Variables, Constraints, Parameters and objective function
2. Model Solution for K = 0, 1, 2, 3, 4 and 5
3. Sensitivity Analysis –
   a. Vary cost per unit distance for drone to see if more drone paths to obtain a better solution
   b. Decrease the cost from $6 per unit distance to $2 per unit distance, and see if more number of drones are being used in the current solution
4. Outputs for the 10 test beds for k = 2, 3, 4
5. Conclusions

Model Formulation:

Decision Variables:

$$X(i,j) = \begin{cases} 1, & \textit{if truck goes from node i to j} \\ 0, & \textit{otherwise} \end{cases}$$

$$Y(i,j) = \begin{cases} 1, & \textit{if drone goes from node i to j} \\ 0, & \textit{otherwise} \end{cases}$$

$\lambda(i) \in I^+$, position of node i, I^+ = Set of positive integers

Parameters:

First Node: (0,0) !Depot

Last Node: (0,0) !Copy Of Depot

A : {1,..,22} !All nodes(including depot and copy of depot)

L : {1,..21} !Launching nodes for truck and drone(for drones,L starts from 2)

R : {2..22} !Returning nodes for truck and drone(for drones, R ends at 21)

N : {2..21} !Customer nodes

E(i,j) = euclidean matrix between nodes i and j

Objective function:

Minimize total cost of transportation:

min. (i $\epsilon L, j\epsilon R) \sum 10 * X(i,j) * E(i,j)$ + (i, $j\epsilon N) \sum 6 * Y(i,j) * E(i,j)$

Constraints:

1. Truck must start and end at a depot

$$\sum_{j\epsilon N} X(1,j) = 1$$

$$\sum_{j\epsilon N} X(j,12) = 1$$

2. Position of node constraint

$$\lambda(1) = 1$$

$$\lambda(12) = 22$$

3. Subtour Elimination:

$$\lambda(j) \geq \lambda(i) + 1 - M * \left(1 - X(i,j)\right) \quad \forall i\epsilon L, j\epsilon R, i \neq j$$

4. Each customer location must be visited once

$$\sum_{i\epsilon L, i\neq j} X(i,j) + \sum_{i\epsilon N, i\neq j} Y(i,j) = 1$$

5. If truck comes from i to j it must depart from j

$$\sum_{i\epsilon L, i\neq j} X(i,j) = \sum_{l\epsilon R, l\neq j} X(j,l) \quad \forall j\epsilon N$$

6. Drone route is dependent on truck route(drone can only fly from nodes where truck arrives)

$$Y(i,j) \leq \sum_{l\epsilon L, l\neq i} X(l,i) \quad \forall i\epsilon N, j\epsilon N, i \neq j$$

7. Upto k drones can be used at a location

$$\sum_{l\epsilon N, l\neq j} Y(j,l) \leq k \quad \forall i\epsilon L, j\epsilon N, i \neq j$$

Model Solutions:

A. Data Used:

There are 20 order nodes, 1 depot node where the truck begins its route and Depot Copy node which is a copy of the Depot node – where truck ends its route.

ID	OrderID	x coordinate	y coordinate
1	Depot	0	0
2	Order1	17	73
3	Order2	75	41
4	Order3	93	90
5	Order4	11	17
6	Order5	87	33
7	Order6	40	27
8	Order7	26	75
9	Order8	72	65
10	Order9	85	95
11	Order10	48	70

12	Order11	6	30
13	Order12	10	58
14	Order13	13	98
15	Order14	76	5
16	Order15	33	100
17	Order16	62	44
18	Order17	66	60
19	Order18	90	99
20	Order19	66	79
21	Order20	81	75
22	Depot_Copy	0	0

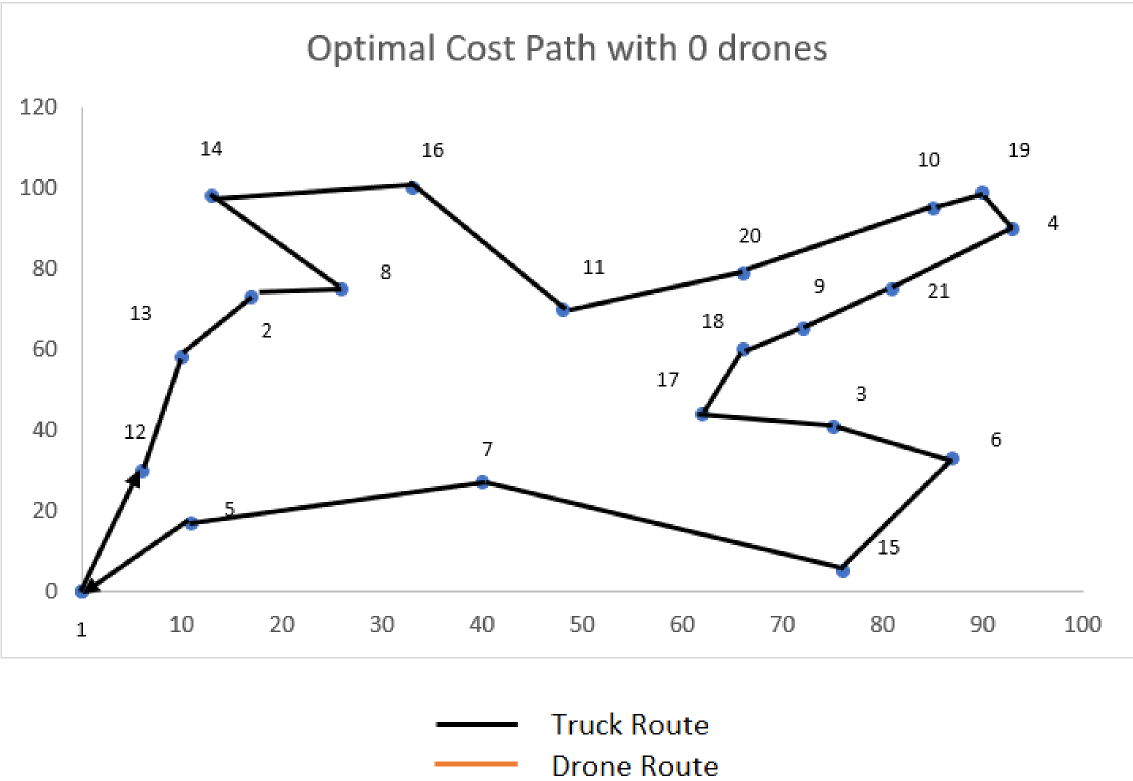
B. Euclidean Matrix:

Below is the Euclidian Matrix obtained – the distances between each of the nodes including the depot and depot copy.

```
0 75 85 129 20 93 48 79 97 127 85 31 59 99 76 105 76 89 134 103 110 0
75 0 66 78 56 81 51 9 56 71 31 44 17 25 90 31 54 51 77 49 64 75
85 66 0 52 68 14 38 60 24 55 40 70 67 84 36 72 13 21 60 39 35 85
129 78 52 0 110 57 82 69 33 9 49 106 89 80 87 61 55 40 9 29 19 129
20 56 68 110 0 78 31 60 78 108 65 14 41 81 66 86 58 70 114 83 91 20
93 81 14 57 78 0 47 74 35 62 54 81 81 98 30 86 27 34 66 51 42 93
48 51 38 82 31 47 0 50 50 82 44 34 43 76 42 73 28 42 88 58 63 48
79 9 60 69 60 74 50 0 47 62 23 49 23 26 86 26 48 43 68 40 55 79
97 56 24 33 78 35 50 47 0 33 25 75 62 68 60 52 23 8 38 15 13 97
127 71 55 9 108 62 82 62 33 0 45 102 84 72 90 52 56 40 6 25 20 127
85 31 40 49 65 54 44 23 25 45 0 58 40 45 71 34 30 21 51 20 33 85
31 44 70 106 14 81 34 49 75 102 58 0 28 68 74 75 58 67 109 77 87 31
59 17 67 89 41 81 43 23 62 84 40 28 0 40 85 48 54 56 90 60 73 59
99 25 84 80 81 98 76 26 68 72 45 68 40 0 112 20 73 65 77 56 72 99
76 90 36 87 66 30 42 86 60 90 71 74 85 112 0 104 41 56 95 75 70 76
105 31 72 61 86 86 73 26 52 52 34 75 48 20 104 0 63 52 57 39 54 105
76 54 13 55 58 27 28 48 23 56 30 58 54 73 41 63 0 16 62 35 36 76
89 51 21 40 70 34 42 43 8 40 21 67 56 65 56 52 16 0 46 19 21 89
134 77 60 9 114 66 88 68 38 6 51 109 90 77 95 57 62 46 0 31 26 134
103 49 39 29 83 51 58 40 15 25 20 77 60 56 75 39 35 19 31 0 16 103
110 64 35 19 91 42 63 55 13 20 33 87 73 72 70 54 36 21 26 16 0 110
0 75 85 129 20 93 48 79 97 127 85 31 59 99 76 105 76 89 134 103 110 0
```

20 nodes and the same x and y coordinates/Euclidian matrix has been used in each of the solutions below –

C. Solution with 0 Drones:

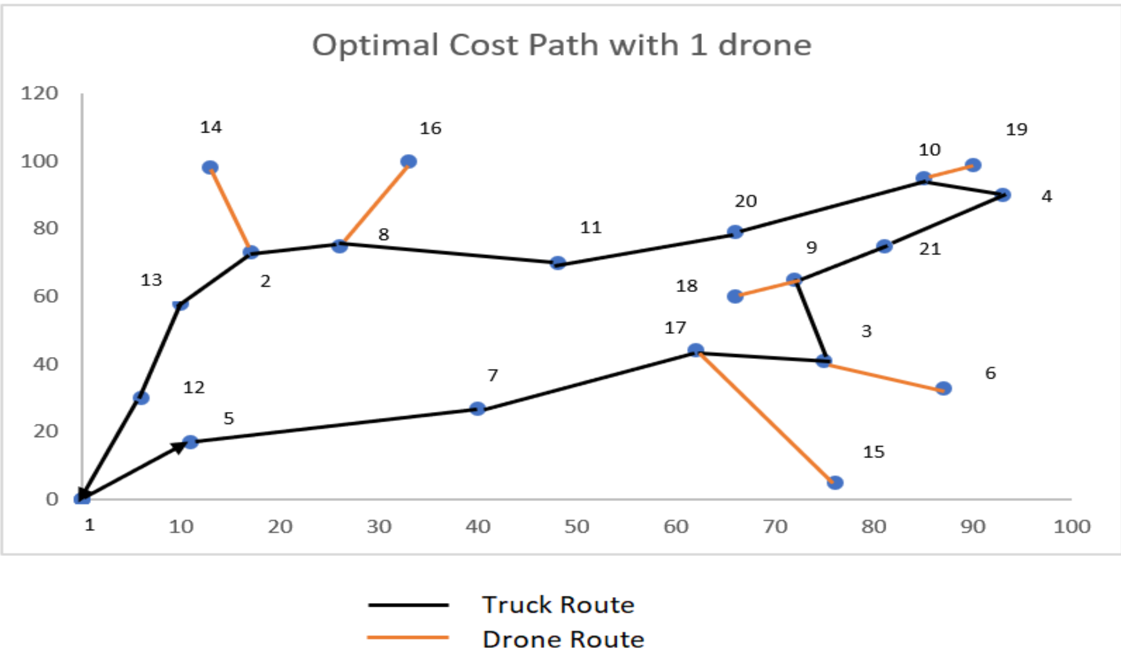


The optimal cost obtained was $4,310

Insight:

The cost would ideally be the highest amongst 0 - 5 drones, since drones (cheaper option) are not been considered here.

D. Solution with 1 Drone:

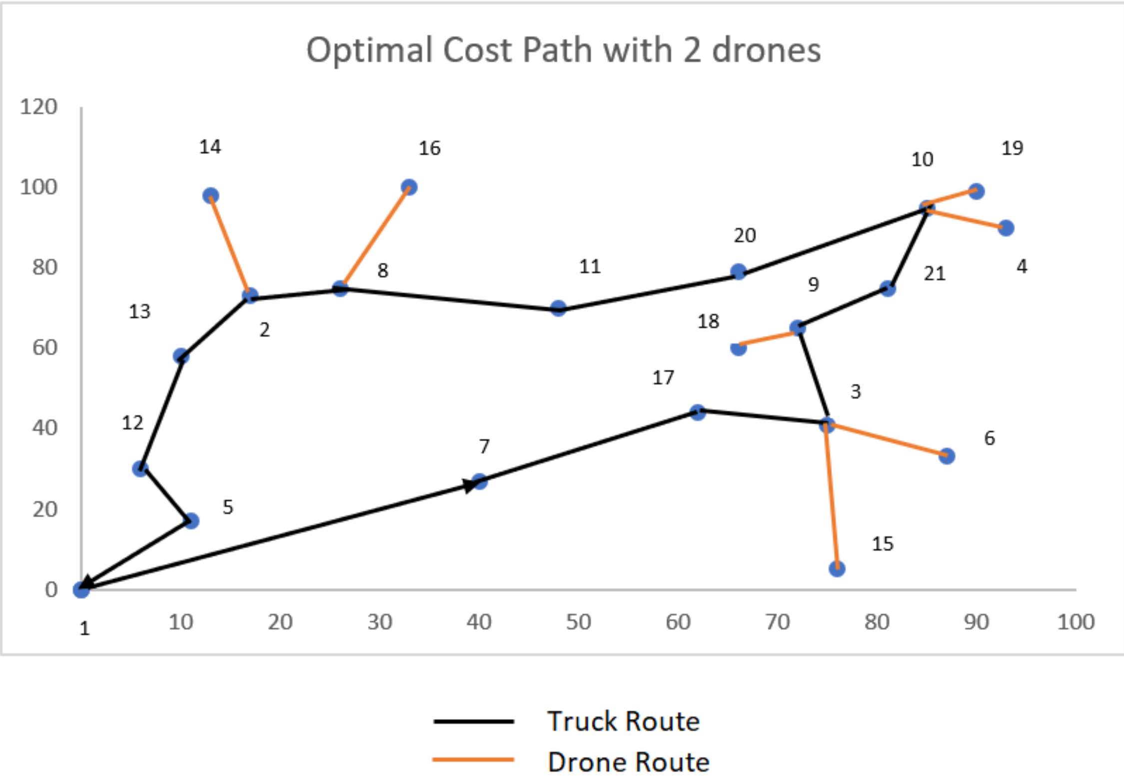


The optimal cost obtained was $3,820

Insight:

As we have introduced <= 1 drone constraint, we observe that the optimal cost is lower than with 0 drones. We would expect the cost to further reduce as we increase the number of drones.

E. Solution with 2 Drones:



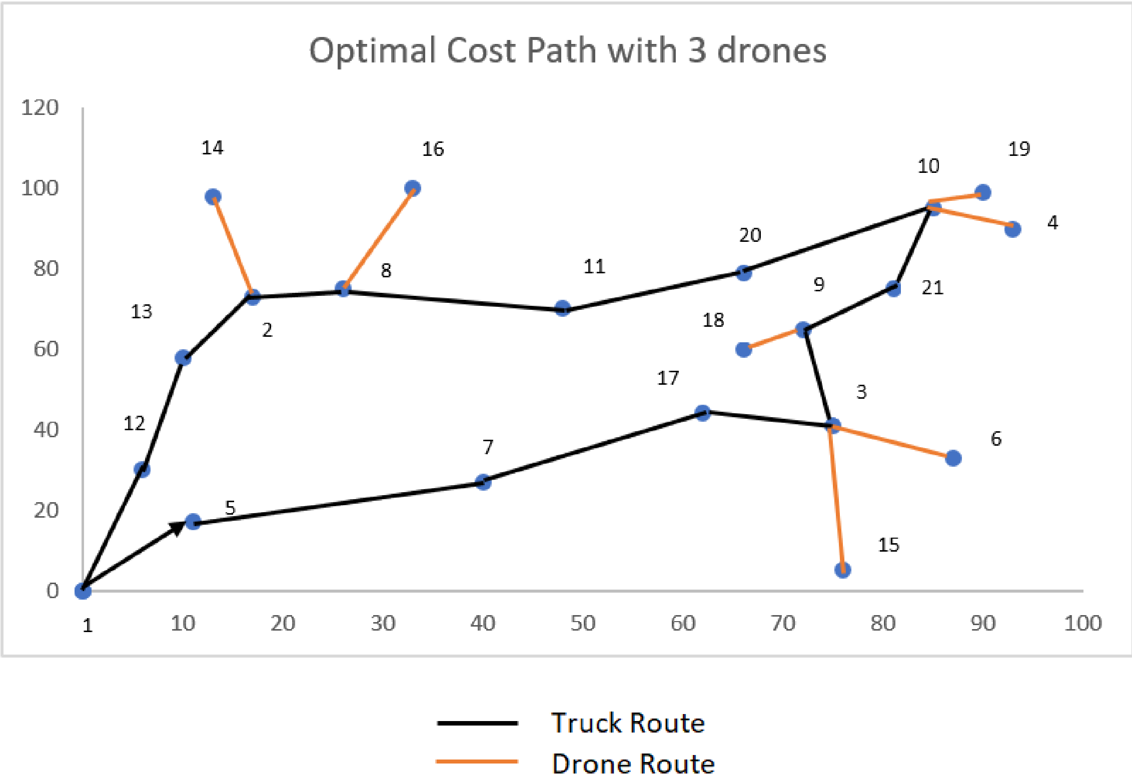
The optimal cost was $3,764.

Insight:

As expected we notice that the introduction of 2 drones has further reduced the optimal cost to $3,764.

Let us see if this continues for 3 drones

F. Solution with 3 Drones:



Insight:

The optimal cost remains same - $3,764.

Here in spite of the constraint <=3 drones, only 2 drones are being utilized. As we run the solution for 4, 5, 6 nodes we notice that the cost remains same at $3,764, hence not reported here.

This means at this cost of truck and drone this is the best solution possible.

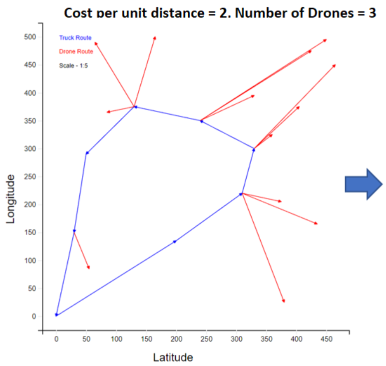
May be if we reduce the cost of the drone more drones can be utilized and even lesser optimal cost can be reached. We would be checking this below in the cost-sensitivity analysis

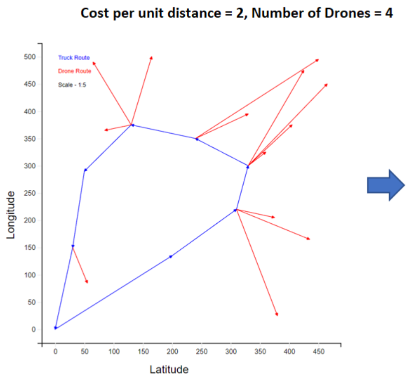
<u>Cost Sensitivity Analysis –</u>

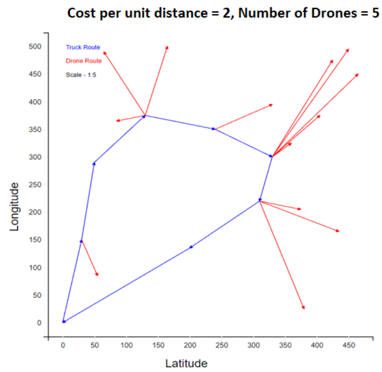
a. Comparing the number of drones utilized and optimal cost by varying cost per unit distance and number of drones used:

$ 3 per unit distance for 3 drones	$ 2 per unit distance for 4 drones	$ 2 per unit distance for 5 drones
X(1,7) = 1	X(1,7) = 1	X(1,12) = 1
X(7,17) = 1	X(7,17) = 1	X(7,22) = 1
X(8,13) = 1	X(8,13) = 1	X(8,11) = 1
X(11,8) = 1	X(11,8) = 1	X(11,18) = 1
X(12,22) = 1	X(12,22) = 1	X(12,13) = 1
X(13,12) = 1	X(13,12) = 1	X(13,8) = 1
X(17,18) = 1	X(17,18) = 1	X(17,7) = 1
X(18,11) = 1	X(18,11) = 1	X(18,17) = 1
Y(8,2) = 1	Y(8,2) = 1	Y(8,2) = 1
Y(8,14) = 1	Y(8,14) = 1	Y(8,14) = 1
Y(8,16) = 1	Y(8,16) = 1	Y(8,16) = 1
Y(11,10) = 1	Y(11,10) = 1	Y(11,20) = 1
Y(11,19) = 1	Y(11,20) = 1	Y(12,5) = 1
Y(11,20) = 1	Y(12,5) = 1	Y(17,3) = 1
Y(12,5) = 1	Y(17,3) = 1	Y(17,6) = 1
Y(17,3) = 1	Y(17,6) = 1	Y(17,15) = 1
Y(17,6) = 1	Y(17,15) = 1	Y(18,4) = 1
Y(17,15) = 1	Y(18,4) = 1	Y(18,9) = 1
Y(18,4) = 1	Y(18,9) = 1	Y(18,10) = 1
Y(18,9) = 1	Y(18,19) = 1	Y(18,19) = 1
Y(18,21) = 1	Y(18,21) = 1	Y(18,21) = 1

Optimal Cost: $2,862 **Optimal Cost: $2,852** **Optimal Cost: $2,842**







Insight:

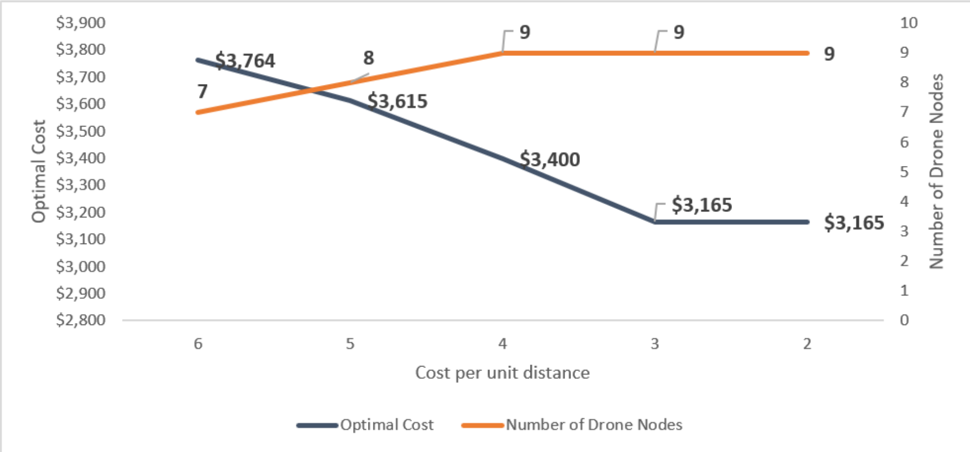
As we bring down the drone cost to $3 per unit distance we observer all 3 drone paths are now being utilized.

Further reducing the cost to $2 per unit distance we notice that 4 drones path can be utilized leading to a little lower optimal cost.

Further we checked if 5 drones could give us an even smaller optimal cost. We obtained an even reduced optimal cost $2,842

b. For 20 nodes, keep number of drones to be 2 and vary the cost per unit distance from $6 to $2 – ideally the number of nodes covered by the drones should increase

Upon conducting this analysis, we obtained the following results:



Insight:

As expected, We observe that as we keep on decreasing the cost per unit distance for the drones the number of drone paths increase and the optimal cost decreases as using more drone paths is a cheaper option here.

For k = 3, 2 the optimal cost and number of drone paths are not changing, because the best solution has been reached at that particular cost of the truck and drones

4. Test Beds for 10 instances -

We ran the test beds for 10 randomly generated datasets with x and y coordinates between 1 and 100 for Order IDs from 1-20. Number of drones = 5 could not be completed since IVE and workbench took a lot of time to run/didn't run.

Truck Cost per unit distance = 10
Drone Cost per unit distance = 6

Dataset Number	Optimal Cost		
	k = 2	k = 3	k=4
1	3898	3898	3898
2	3358	3358	3358
3	3668	3668	3668
4	3796	3796	3796
5	3466	3400	3400
6	3396	3396	3396
7	2954	2954	2954
8	3834	3810	3810
9	3508	3508	3508
10	3554	3554	3554

a. For the test beds marked in blue, we notice the same thing that we noticed before, inspite of more drones being used, the additional drones are not getting utilized due to the current cost of the truck and the drone for the given Euclidian distance Matrix. An optimal cost is being reached by using less number of drones Itself. If we reduce the cost of the drones, maybe we would be able to see a different/lower cost with more number of drones being utilized

b. For the best beds marked in orange, for the given Euclidian distance matrix, we notice that as the number of drones are being increased, we are able to arrive at a better optimal cost due to lower cost of the drones. (from k = 2 to k = 3)

Conclusions:

Some key observations we have made in this project journey which would help the transportation company make a better decision in reaching a better optimal cost / design the delivery routes more efficiently –

1. As the number of drones are increased, the overall optimal cost reduces due to more number of drones being utilized but becomes constant after a certain number of drones because for the given distance matrix, that is the best solution that could be reached at the given truck and drone cost.

2. Reducing the cost per unit distance of the drones leads to more drones being utilized for the deliveries and a better optimal cost. We notice that the cost becomes constant after a certain number of nodes and decreasing the cost does not have an impact on the optimal cost, because the best solution had been reached at the given truck and drone cost.

Appendix:

```
model TruckDrone
uses "mmsheet","mmxprs","mmsvg" !gain access to the Xpress-Optimizer solver
parameters
Excel_InputFile = "OrdersData_20_01.xlsx"
 end-parameters
!sample declarations section
declarations
L = 1..21           !launching nodes for trucks and drones
R = 2..22           !returning node for trucks and drones
N = 2..21           !Customer nodes/locations
A = 1..22           !all nodes
k = 2               !number of drones
ORDERID:array(A) of string
La:array(A) of integer
Lo:array(A) of integer
EUC_DIST: array(A,A) of integer  ! Distance between customer locations
X:array(A,A) of mpvar
Y:array(A,A) of mpvar
lambda:array(A) of mpvar
end-declarations
forall(i in A, j in A)
	X(i,j) is_binary
forall(i in A, j in A)
	Y(i,j) is_binary
forall(i in A)
	lambda(i) is_integer
!Read data from excel
initializations from "mmsheet.xlsx:"+Excel_InputFile
	[ORDERID,La,Lo] as "skiph;[Sheet1$A:D]"
end-initializations
```

```
!Build Euclidean matrix
forall(i in A,j in A) do
	EUC_DIST(i,j):= round(sqrt((La(i)-La(j))^2 + (Lo(i)-Lo(j))^2))
end-do
!print Euclidean matrix
forall(i in A) do
	forall(j in A) do
		write(EUC_DIST(i,j),' ')
	end-do
	writeln('')
end-do
!objective
obj:= sum(i in L,j in R)10*X(i,j)*EUC_DIST(i,j) + sum(i in N,j in N)2*Y(i,j)*EUC_DIST(i,j)
!truck must start and end at a depot
sum(j in N)X(1,j) = 1
sum(j in N)X(j,getsize(A)) = 1

!each customer location must be visited once
forall(j in N)(sum(i in L|i<>j)X(i,j) + sum(i in N|i<>j)Y(i,j) = 1)
!position of node constraint
lambda(1) = 1
lambda(getsize(A)) = getsize(A)
!subtour elimination
forall(i in L,j in R|i<>j) lambda(j) >= lambda(i) + 1 - (getsize(A)+1)*(1-X(i,j))
!if truck comes to j it must depart from j
forall(j in N)sum(i in L|i<>j)X(i,j) = sum(l in R|l<>j)X(j,l)
!associate drone delivery to truck route
forall(i in N,j in N|i<>j)Y(i,j) <= sum(l in L|l<>i)X(l,i)
!upto K drones can be used at a location
forall(j in N) do
	sum(l in N|l<>j)Y(j,l) <= k
```

```
end-do
minimize(obj)
forall(i in A , j in A | getsol(X(i,j)) > 0) do
	writeln('X(',i,',',j,') = ',getsol(X(i,j)))
end-do
forall(i in A , j in A | getsol(Y(i,j)) > 0) do
	writeln('Y(',i,',',j,') = ',getsol(Y(i,j)))
end-do
!visualization of graph
svgerase
svgaddgroup("g1", "", SVG_BLUE)
svgaddtext(90,99, "Truck Route")
svgaddgroup("g2", "", SVG_RED)
svgaddtext(90,94, "Drone Route")
svgaddgroup("g3", "", SVG_BLACK)
svgaddtext(10,99, "Scale - 1:5")
svgshowgraphaxes(true)
svgsetgraphlabels("Latitude","Longitude")
forall(i in L) do
	forall(j in R) do
		if(getsol(X(i,j)) > 0) then
			svgaddarrow("g1",La(i), Lo(i), La(j), Lo(j))
		forall(l in N|l<>j)
			if(getsol(Y(j,l)) > 0) then
			svgaddarrow("g2",La(j), Lo(j), La(l), Lo(l))
			end-if
		end-if
	end-do
end-do
svgsetgraphscale(5)
svgrefresh
```

```
svgwaitclose("Close browser window to terminate model execution.", 1)

writeln("Begin running model")

writeln("Optimal cost is ",getobjval)

end-model
```