Last Mile Delivery Optimization

Prototype Evaluation

- **Yashwanth Kumar Yelamanchili**

INTRODUCTION

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 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.

The project is aimed at finding the most cost-efficient route for the trucks and the drones. The number of drones were varied between iterations to include 1, 2 and 3 drones as viable options to test how this changes the cost and routes. A linear mixed integer programming approach was taken to achieve the same, with the code tested and results obtained through FICO Xpress IVE software. This report elucidates the model that was built with the sets, decisions and constraints, the iterations on a dataset with varying truck and drones’ combination and the summary of iterations that test the model algorithm on different datasets. A simple cost sensitivity analysis was also performed to check how the routes and decisions change based on changing the cost to serve a customer.

MODELING

The problem can be modeled as a linear mixed integer solution. The main datasets we have are the total number of customer nodes, the set of drones as we need to formulate a decision arc for each of them. For the purpose of breaking loops, we have used the MTZ formulation, because of which, we have mimicked the depot as 2 more nodes as a virtual customer node – a start depot node from which the truck starts, and an end depot node at which the truck finishes it journey.

The decisions we would have to take involve choosing whether a node has to be delivered by the truck or by a drone, based on which, we would have to decide from where does the delivery medium arrive at the node and where it goes to next (incoming and outgoing arc). We also need to know the order in which the customers are being delivered.

The key information that is necessary for this model includes the distance between each node (including the customer and the depot) and the cost to serve the customer by the truck or the drone. The objective of the solution is to come up with the most optimal set of travel path for the truck and drone that serves every customer at the least cost possible. The constraints for the same are elucidated in the problem description.

The formulation can be found below -

SETS

- Set of total nodes - Nodes = {1, totalCust} index i
- Set of customer nodes - customerNodes = {} index j
- Set of drones – drones = {} index k

DECISION VARIABLES

- $truckArc_{i,j}$: $\begin{cases} 1\ if\ decision\ is\ to\ travel\ between\ i,j \in \text{Nodes}\ by\ truck \\ 0\ otherwise \end{cases}$
- $droneArc_{i,j,k}$: $\begin{cases} 1\ if\ decision\ is\ to\ travel\ between\ i,j\ \in \text{Nodes using k} \in \text{drones} \\ 0\ otherwise \end{cases}$
- $truckNode_i$: $\begin{cases} 1\ if\ i \in \text{Nodes is visited by the truck} \\ 0\ otherwise \end{cases}$
- $droneNode_i$: $\begin{cases} 1\ if\ i \in \text{Nodes is visited by a drone} \\ 0\ otherwise \end{cases}$
- $position_i$: The position of the customer i ∈ Nodes to be delivered by truck or drone

DATA

- $distance_{i,j}$: Distance between i,j ∈ Nodes
- costTruck: Cost for each distance unit travelled by Truck
- costDrone: Cost for each distance unit travelled by Drone
- totalCustomers: The number of customers in the delivery iteration

OBJECTIVE

min (($\sum_{i\ in\ Nodes} \sum_{j\ in\ Nodes} truckArc(i,j)\ *\ distance(i,j) * costTruck$) +
($\sum_{i\ in\ Nodes} \sum_{j\ in\ Nodes} \sum_{k\ in\ drones} droneArc(i,j,k)\ *\ distance(i,j) * 2 * costDrone$)

CONSTRAINTS

- Truck leaves the start depot exactly once to some other customer node
$$\sum_{j\ in\ Nodes} truckArc(1,j) = 1 \qquad 1$$
- No travel can end at the start depot node
$$\sum_{i\ in\ Nodes} truckArc(i,1) = 0 \qquad 2$$
- Truck reaches the end depot from exactly one other customer node
$$\sum_{i\ in\ Nodes} truckArc(i, totalCustomers + 2) = 1 \qquad 3$$
- No travel can start at the end depot node
$$\sum_{j\ in\ Nodes} truckArc(totalcustomers + 2, j) = 0 \qquad 4$$
- There can be only 1 outgoing truck movement for each customer node
$$\sum_{j\ in\ Nodes} truckArc(i,j) = 1 * truckNode(i)\ \forall i \in customerNodes \qquad 5$$
- There can be only 1 incoming truck movement for each customer node
$$\sum_{i\ in\ Nodes} truckArc(i,j) = 1 * truckNode(j)\ \forall j \in customerNodes \qquad 6$$
- No drone can be launched from the start depot
$$\sum_{j\ in\ Nodes} droneArc(1,j,k) = 0\ \forall k \in drones \qquad 7$$
- No drone can land at the start depot node
$$\sum_{i\ in\ Nodes} droneArc(i,1,k) = 0\ \ \forall k \in drones \qquad 8$$

- No drone can land at the end depot node

$$\sum_{i\ in\ Nodes} droneArc(i, totalCustomer + 2, k) = 0 \ \ \forall k \in drones \tag{9}$$

- No drone can start from the end depot node

$$\sum_{j\ in\ Nodes} droneArc(totalCustomer + 2, j, k) = 0 \ \ \forall k \in drones \tag{10}$$

- A customer node can either be delivered by a drone or the truck

$$truckNode(i) + droneNode(i) = 1 \ \ \forall i \ \in Nodes \tag{11}$$

- A drone is allowed to start from a customer node only if it is served by the truck

$$\sum_{j\ in\ Nodes} droneArc(i, j, k) \leq truckNode(i) \ \ \forall i \in Nodes, \forall k \in drones \tag{12}$$

- Only one drone can visit a customer from a truck node

$$\sum_{k\ in\ drones} droneArc(i, j, k) \leq 1 \ \ \forall i \in Nodes, \forall j \in Nodes \tag{13}$$

- If a node is selected to be delivered by a drone, it can have only one incoming travel

$$\sum_{i\ in\ Nodes} \sum_{k\ in\ drones} droneArc(i, j, k) = 1 * droneNode(j) \ \ \forall j \in Nodes \tag{14}$$

- The start depot and the end depot are the 1st and the last customers to be delivered

$$position(1) = 1 \tag{15}$$

$$position(totalCustomer + 2) = totalCustomer + 2 \tag{16}$$

- MTZ customer position constraint

$$position(j) \geq position(i) + 1 + \sum_{j\ in\ nodes} \sum_{k\ in\ drones} droneArc(i, j, k) - M * \big(1 - truckArc(i, j)\big) \tag{17}$$

- No customer should be selected as the origin and destination at the same time for truck or drone travel

$$truckArc(i, i) = 0 \ \ \forall i \in nodes \tag{18}$$

$$droneArc(i, i, k) = 0 \ \ \forall i \in nodes, \forall k \in drones \tag{19}$$

- The decision variables for truck and drone travel arcs are binary variables

$$truckArc(i, i) \in [0,1] \ \ \forall i \in nodes \tag{20}$$

$$droneArc(i, i, k) \in [0,1] \ \ \forall i \in nodes, \forall k \in drones \tag{21}$$

- The decision variables for choosing if a node is to be visited by a truck or a drone are binary variables

$$truckNode(i) \in [0,1] \forall i \in Nodes \tag{22}$$

$$droneNode(i) \in [0,1] \forall i \in Nodes \tag{23}$$

- The decision variable that holds the position of the customer is an integer variable

$$position(i) = Z^{+} \forall i \in Nodes \tag{24}$$

COMPLETE MODEL

min (($\sum_{i\ in\ Nodes} \sum_{j\ in\ Nodes} truckArc(i,j)\ *\ distance(i,j) * costTruck$) +
($\sum_{i\ in\ Nodes} \sum_{j\ in\ Nodes} \sum_{k\ in\ drones} droneArc(i,j,k)\ *\ distance(i,j) * 2 * costDrone$)
s.t

$$\sum_{j\ in\ Nodes} truckArc(1,j) = 1 \tag{1}$$

$$\sum_{i\ in\ Nodes} truckArc(i,1) = 0 \tag{2}$$

$$\sum_{i\ in\ Nodes} truckArc(i, totalCustomers + 2) = 1 \tag{3}$$

$$\sum_{j\ in\ Nodes} truckArc(totalcustomers + 2, j) = 0 \tag{4}$$

$$\sum_{j\ in\ Nodes} truckArc(i,j) = 1 * truckNode(i)\ \forall i \in customerNodes \tag{5}$$

$$\sum_{i\ in\ Nodes} truckArc(i,j) = 1 * truckNode(j)\ \forall j \in customerNodes \tag{6}$$

$$\sum_{j\ in\ Nodes} droneArc(1,j,k) = 0\ \forall k \in drones \tag{7}$$

$$\sum_{i\ in\ Nodes} droneArc(i,1,k) = 0\ \ \forall k \in drones \tag{8}$$

$$\sum_{i\ in\ Nodes} droneArc(i, totalCustomer + 2, k) = 0\ \ \forall k \in drones \tag{9}$$

$$\sum_{j\ in\ Nodes} droneArc(totalCustomer + 2, j, k) = 0\ \ \forall k \in drones \tag{10}$$

$$truckNode(i) + droneNode(i) = 1\ \ \forall i\ \in Nodes \tag{11}$$

$$\sum_{j\ in\ Nodes} droneArc(i,j,k) \leq truckNode(i)\ \ \forall i \in Nodes, \forall k \in drones \tag{12}$$

$$\sum_{k\ in\ drones} droneArc(i,j,k) \leq 1\ \ \forall i \in Nodes, \forall j \in Nodes \tag{13}$$

$$\sum_{i\ in\ Nodes} \sum_{k\ in\ drones} droneArc(i,j,k) = 1 * droneNode(j)\ \ \forall j \in Nodes \tag{14}$$

$$position(1) = 1 \tag{15}$$

$$position(totalCustomer + 2) = totalCustomer + 2 \tag{16}$$

$$position(j) \geq position(i) + 1 + \sum_{j\ in\ nodes} \sum_{k\ in\ drones} droneArc(i,j,k)) - M * \big(1 - truckArc(i,j)\big) \tag{17}$$

$$truckArc(i,i) = 0\ \ \forall i \in nodes \tag{18}$$

$$droneArc(i,i,k) = 0\ \ \forall i \in nodes, \forall k \in drones \tag{19}$$

$$truckArc(i,i) \in [0,1]\ \ \forall i \in nodes \tag{20}$$

$$droneArc(i,i,k) \in [0,1]\ \ \forall i \in nodes, \forall k \in drones \tag{21}$$

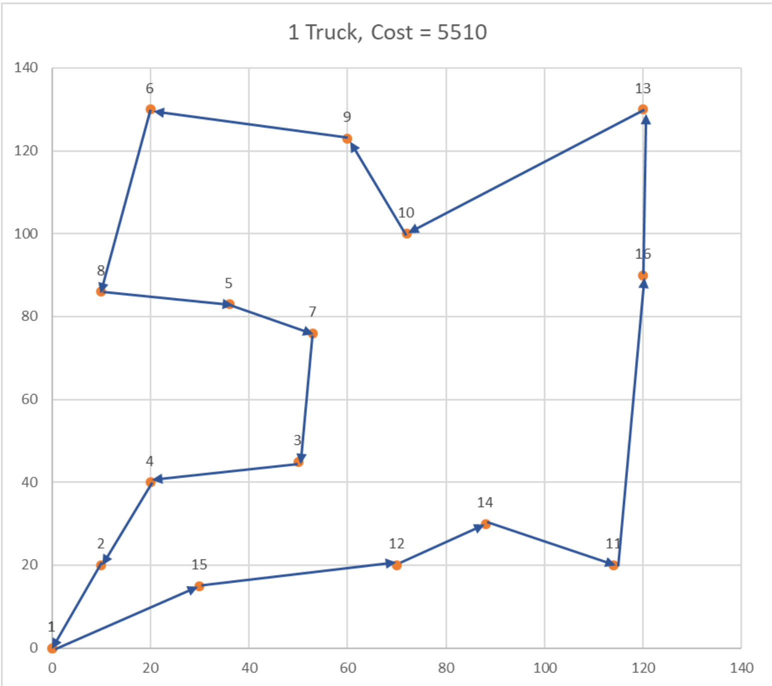
$$truckNode(i) \in [0,1] \forall i \in Nodes \tag{22}$$

$$droneNode(i) \in [0,1] \forall i \in Nodes \quad (23)$$
$$position(i) = Z^{+} \forall i \in Nodes \quad (24)$$

SOLUTION

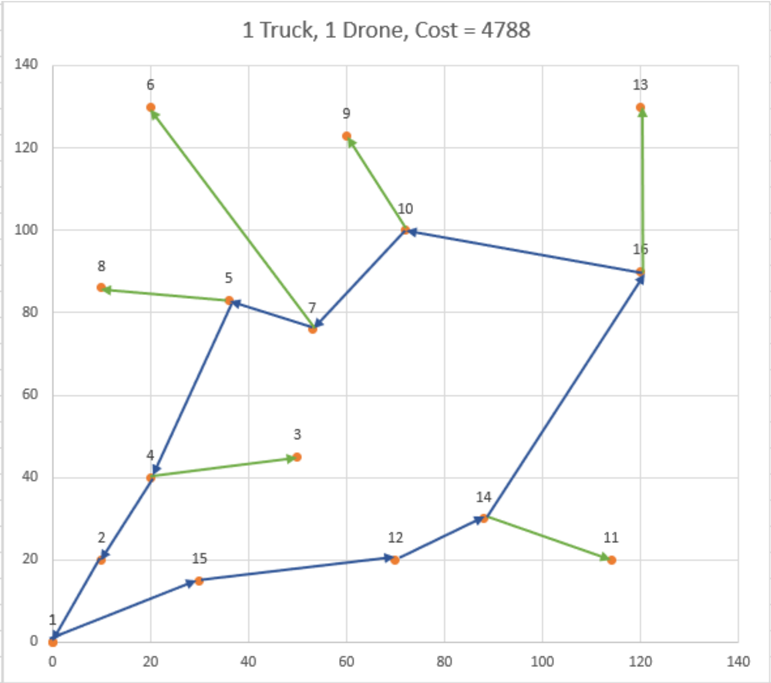
In the visualizations below, the truck path and the drone path are shown in different colours. The cost of the route is mentioned and the travel arcs are displayed as a sequence of nodes as {Depot->Node1->Node2->….->Depot}, while each drone path is shown as a set of triples such as {(StartNode->DroneNode->StartNode,…)}. The cost of the truck is $10/unit and the cost for a drone is $3/unit

Simulation 1
This solution includes 15 customer nodes that were generated randomly, with the cost to serve for truck as 10/unit distance and for the drone as 3/unit distance. The simulated data and a simple delivery route solution including only the single truck is shown below.

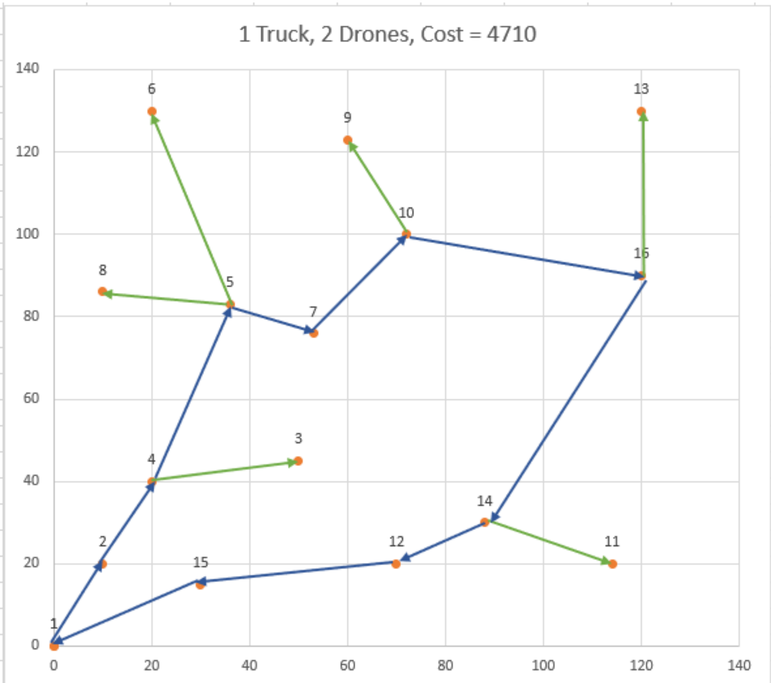


Cost = $5510
Truck Route = {Depot->15->12->14->11->16->13->10->9->6->8->5->7->3->4->2->Depot}
The solution for the same set of customers for different drone iterations are elaborated below.

Case 1 – 1 Truck, 1 Drones
Cost = $4788
Truck Route = {Depot->15->12->14->16->10->7->5->4->2->Depot}
Drone Route = {(14->11->14), (16->13->16), (10->9->10), (7->6->7), (5->8->5), (4->3->4)}

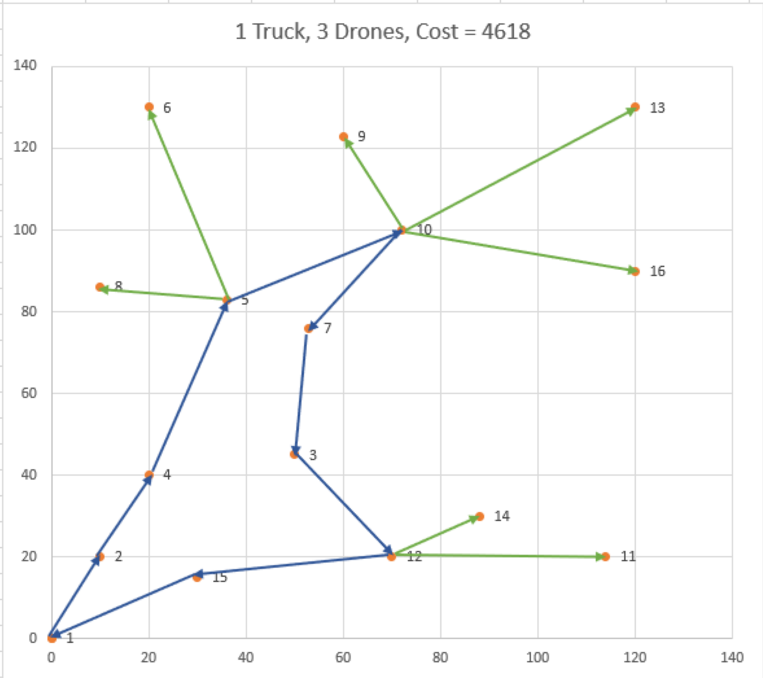


Case 2 – 1 Truck, 2 Drones
Cost = $4710
Truck Route = {Depot->2->4->5->7->10->16->14->12->15->Depot}
Drone 1 Routes = {(5->6->5), (10->9->10), (16->13->16), (14->11->11)}
Drone 2 Routes = {(5->8->5)}

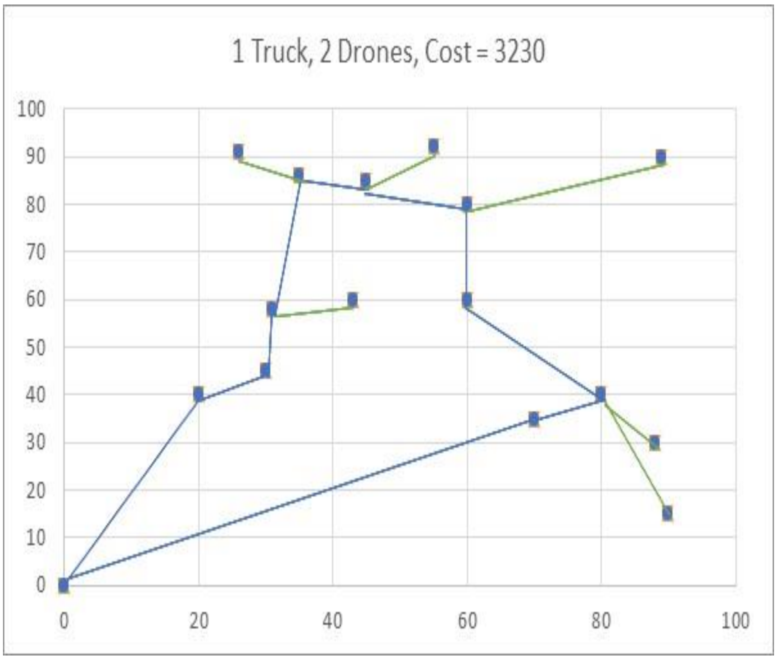


Case 3 – 1 Truck, 3 Drones
Cost = $4618
Truck Route = {Depot->2->4->5->10->7->3->12->15->Depot}
Drone 1 Routes = {(5->8->5), (10->9->10), (12->14->12)

Drone 2 Routes = {(5->6->5), (10->13->10), (12->11->12)}
Drone 3 Routes = {(10,16,10)}



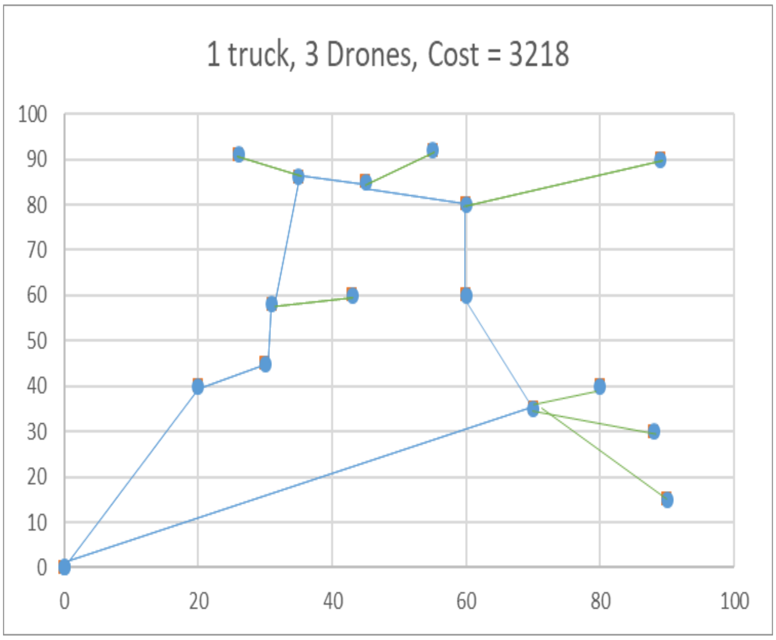
Simulation 2
The data present as example in the problem statement was used to see how the solution comes out based on the suggested methodology. This contains again of 15 customer nodes, and the cost remains the same as before.



Case 1 – 1 Truck, 2 Drones
Cost = $3230
Truck Route = {Depot->2->3->4->8->9->11->7->13->12->Depot}

Drone 1 Routes = {(4->5->4), (8->6->8), (9->10->9), (11->16->11), (13->14->13)}
Drone 2 Routes = {(13->15->13)}

Case 2 – 1 Truck, 3 Drones
Cost = $3218
Truck Route = {Depot->12->7->11->9->8->4->3->2->Depot}
Drone 1 Routes = {(4->5->4), (8->6->8), (9->10->9), (11->16->11), (12->13->12)}
Drone 2 Routes = {(12->14->12)}
Drone 3 Routes = {(12->15->12)}



Simulation 3
An iteration was also run using 20 customer nodes, the result of which is summarized below.

Number of Customers	Number of Drones	Objective Achieved	Best Bound	Truck Nodes	Drone	Path
20	2	3432	2999	13	7	Truck(Depot-8-5-13-15-10-21-14-6-4-17-7-9-12-Depot) Drone 1 = ((4,3,4),(7,2,7),(12,11,12),(14,16,14),(15,19,15)) Drone 2 = ((7,18,7),(14,20,14))
20	3	3462	2914	10	10	Truck = (Depot-8-5-13-15-21-14-4-17-9-12-Depot) Drone 1 = ((4,3,4),(12,11,12),(14,6,14),(15,10,15),(17,2,17),(21,20,21)) Drone 2 = ((14,16,14),(15,19,15),(17,7,17))) Drone 3 = ((17,18,17))

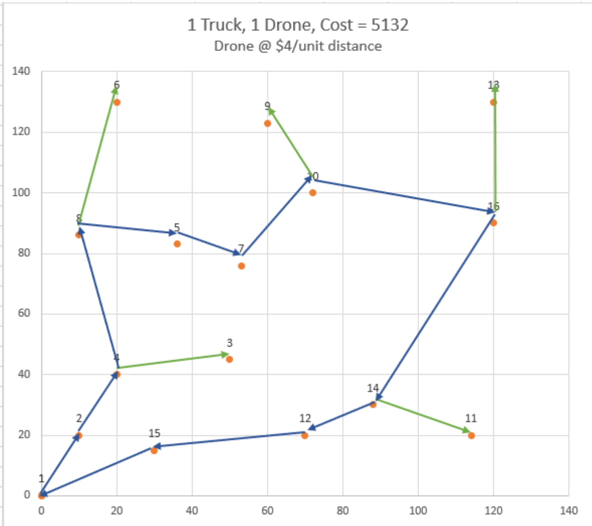
COST SENSITIVITY ANALYSIS

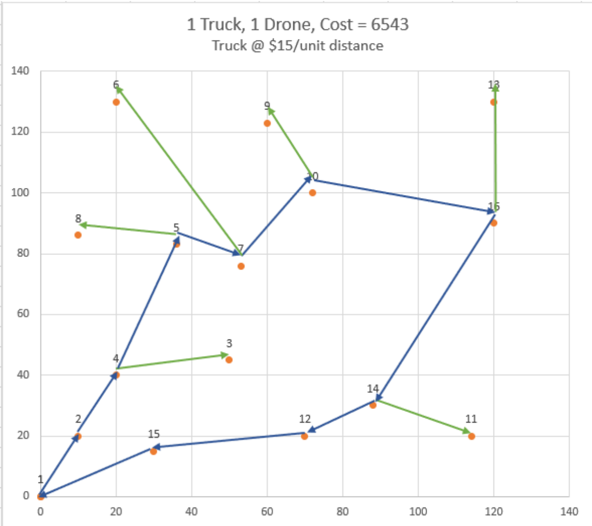
Using the same dataset used in the simulation above, 2 variations were run with respect to the cost to serve. In the first iteration, the cost to serve the customer using the drone was changed to $4/unit distance, keeping the truck cost the same. In the second iteration, the cost to serve the customer using the truck was changed to $15/unit distance, keeping the drone cost the same. The results are shown below.

Simulation 1 – 1 Truck 1 Drone

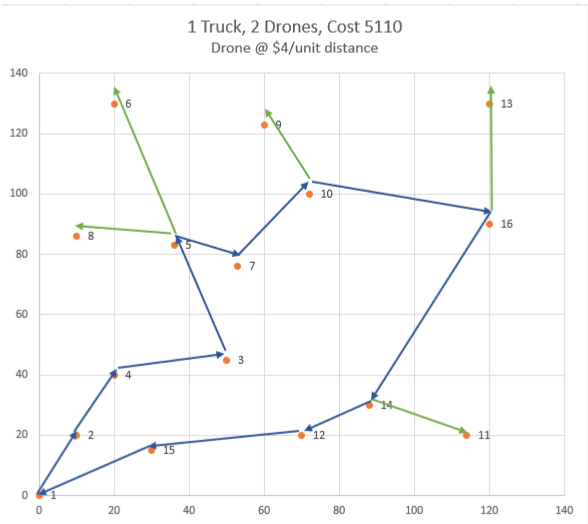
Case 1 – Truck @ $10/unit distance, Drone @ 4/unit distance
Truck Route = {Depot->2->4->->8->->5->7->10->16->14->12->15->Depot}
Drone Routes = {(4->3->4), (8->6->8), (10->9->10), (16->13->16), (14->11->14)}

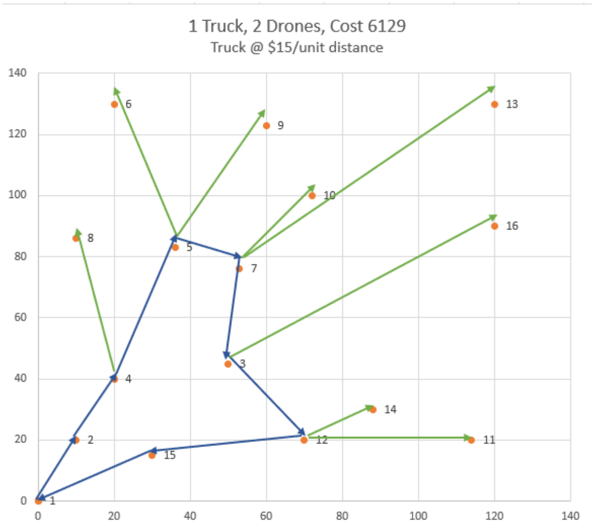
Case 2 – Truck @ $15/unit distance, Drone @ 3/unit distance
Truck Route = {Depot->2->4->5->7->10->16->14->12->15->Depot}
Drone Routes = {(4->3->4), (5->8->5), (7->6->7), (10->9->10), (16->13>16), (14->11->14)}





Simulation 2 – 1 Truck, 2 Drones





Case 1 – Truck @ $10/unit distance, Drone @ 4/unit distance

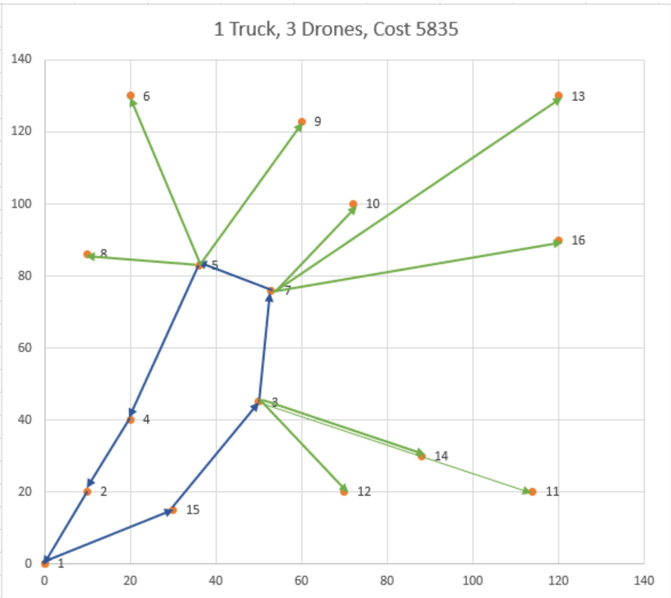
Truck Route = {Depot->2->4->3->5->7->10->16->14->12->15->Depot}
Drone 1 Routes = {(5->8->5), (10->9->10), (16->13->16), (14->11->14)}
Drone 2 Routes = {(5,6,5)}

Case 2 – Truck @ $15/unit distance, Drone @ 3/unit distance
Truck Route = {Depot->2->4->5->7->10->3->12->15->Depot}
Drone 1 Routes = {(4->8->4), (5->6->5), (7->10->7), (3->16->3), (12->14->12)}

Simulation 3 – 1 Truck, 3 Drones

Case 1 – Truck @ $10/unit distance, Drone @ 4/unit distance, the solution is same as above

Case 2 – Truck @ $15/unit distance, Drone @ 3/unit distance
Truck Route = {Depot->15->3->7->5->4->2->Depot}
Drone 1 Routes = {(3->12->3), (7->16->7), (5->9->5)}
Drone 2 Routes = {(3->11->3), (7->13->7), (5->6->5)}
Drone 3 Routes = {(3->14->3), (7->10->7), (5->8->5)}



CONCLUSION

Running the algorithm through different datasets and different cost structures, we can safely conclude that –

- We notice that as the number of drones being used increase, the overall cost is further optimized, but this comes with some computational overhead.
- The algorithm performs well for spread out nodes, effectively deploying drones from the appropriate customer node, but minimizes the drones when the spread is small.
- When we increase the truck cost, the algorithm corrects itself to launch more drones to further away customers, forming a close truck loop.
- When the cost of drone is increased, there is no significant difference in the model results, although the number of drones were lesser than the previous iteration.

APPENDIX

Code -

```
!@encoding CP1252
model ModelName
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

setparam("XPRS_MAXTIME", 900)

declarations
	nodes = 1..17			!All nodes including start and end
	customerNodes = 2..16		!Customer nodes without start and end
	drones = 1..3

! Binary variable for checking truck takes arc
	truckArc : array(nodes, nodes) of mpvar
	droneArc : array(nodes, nodes, drones) of mpvar  !MN

! Distance arc
	distance: array(nodes, nodes) of integer

!Position
	position : array(nodes) of mpvar

!Inserted by MN
!Truck node or drone node
	truckNode : array(nodes) of mpvar
	droneNode : array(nodes) of mpvar

!Cost for truck
	costTruck : real			!Cost per unit for truck travel
	bigMNumber : real		!Just a big big number
	costDrone : real			!Cost per unit for drone travel
	totalCustomer : integer
end-declarations

!Data Declarations
costTruck := 10
costDrone := 3

initializations from 'D:\Academia\Grad - UC MSBANA\Fall 19\BANA 7020 Optimization\Project\Codes\DataInPDF.txt'
	distance
end-initializations
```

```
!Initializing a big big number
bigMNumber := 10000
totalCustomer := 15

!Making the decision variables binary - for Truck arcs
forall(startNode in nodes, endNode in nodes) truckArc(startNode, endNode) is_binary

!Making the decision variables binary - for Drone arcs - MN
forall(startNode in nodes, endNode in nodes, drone in drones) droneArc(startNode,
endNode, drone) is_binary

!Making truckNode and droneNode decision variables - MN
forall (customerNode in nodes) truckNode(customerNode) is_binary
forall (customerNode in nodes) droneNode(customerNode) is_binary

!Making the position an integer array
forall(node in nodes) position(node) is_integer

!Objective Function
obj :=
(sum(startNode in nodes, endNode in nodes)
	(truckArc(startNode, endNode) * distance(startNode, endNode) * costTruck) +
sum(startNode in nodes, endNode in nodes, drone in drones)
	(droneArc(startNode, endNode, drone) * distance(startNode, endNode) *
2*costDrone))

!Constraints

!Truck related constraints
	!Truck leaves depot exactly once to some other node
	sum(endNode in nodes) truckArc(1, endNode) = 1
	!No node can end at the Depot node
	sum(startNode in nodes) truckArc(startNode, 1) = 0

	!Truck reaches the depot from exactly one other node
	sum(startNode in nodes) truckArc(startNode, totalCustomer + 2) = 1
	!No node can start at the End Depot node
	sum(endNode in nodes) truckArc(totalCustomer + 2,endNode) = 0

	!Only one arc can be travelled from some arc | outgoing for all nodes is 1
	forall(customerNode in customerNodes)
		sum(endNode in nodes) truckArc(customerNode, endNode) = 1 *
truckNode(customerNode)
```

```
	!Only one arc can be travelled from some arc | incoming for all nodes is 1
	forall(customerNode in customerNodes)
		sum(startNode in nodes) truckArc(startNode, customerNode) = 1 *
truckNode(customerNode)

	! No node should be selected as both origin and destination at same time
	forall(node in nodes) truckArc(node,node) = 0
	forall(node in nodes, drone in drones) droneArc(node,node, drone) = 0

truckArc(1,totalCustomer+2) = 0

!Drone related constraints -MN
	!Drone cannot be launched from the Depot node
	forall(drone in drones)
		sum(endNode in nodes) droneArc(1, endNode, drone) = 0
	!Drone node cannot end at the Depot Node
	forall(drone in drones)
		sum(startNode in nodes) droneArc(startNode, 1, drone) = 0

	!Drone cannot land at the Depot node
	forall(drone in drones)
		sum(startNode in nodes) droneArc(startNode, totalCustomer + 2, drone) = 0
	!No node can start at the End Depot node
	forall(drone in drones)
		sum(endNode in nodes) droneArc(totalCustomer + 2,endNode, drone) = 0

!A customer node can be either drone node or truck node
forall(node in nodes)
	truckNode(node) + droneNode(node) = 1

!An arc can be either truck arc or drone arc or neither
!forall(startNode in nodes)
!	sum(endNode in nodes, drone in drones) droneArc(startNode, endNode, drone) <=
2*truckNode(startNode)

forall(startNode in nodes, drone in drones)
	sum(endNode in nodes) droneArc(startNode, endNode, drone) <=
truckNode(startNode)

forall(startNode in nodes, endNode in nodes)
	sum(drone in drones) droneArc(startNode, endNode, drone) <= 1

forall(endNode in nodes)
	sum(startNode in nodes, drone in drones) droneArc(startNode, endNode, drone) =
1*droneNode(endNode)
```

```
!Breaking Loops

position(1) = 1
position(totalCustomer + 2) = totalCustomer+2

!MTZ position constraint
forall(startNode in nodes) forall(endNode in nodes)
	(position(endNode) >= ((position(startNode) + 1) + (sum(node in nodes, drone in drones)droneArc(startNode, node, drone)) - (bigMNumber * (1 - truckArc(startNode, endNode)))))

minimize(obj)

writeln("Begin running model")
	writeln("\n Total Cost : ", getobjval)
	forall(startNode in nodes, endNode in nodes|getsol(truckArc(startNode, endNode))>0)
		writeln("Truck","Start Node: ", startNode, ", EndNode: ", endNode, " - Travel : ", getsol(truckArc(startNode, endNode)))
	forall(startNode in nodes, endNode in nodes, drone in drones|getsol(droneArc(startNode, endNode, drone))>0)
		writeln("Drone", drone ," Start Node: ", startNode, ", EndNode: ", endNode, " - Travel : ", getsol(droneArc(startNode, endNode, drone)))
	forall(node in nodes) writeln("Customer Node ", node," ", getsol(truckNode(node))," ",getsol(droneNode(node)))

	!forall(node in nodes) writeln(getsol(position(node)))
writeln("End running model")

end-model
```