Introduction to the Vehicle Routing Problem

In the classical formulation of the Vehicle Routing Problem (VRP), we would like to find the most efficient way to visit a set of nodes with a given number of vehicles that must start and end at a specific location, called the depot. Specifically, we are given an original graph $\mathcal{G} = \{\mathcal{V}_G, \mathcal{E}_G\}$ which is weighted, direct and not necessarily complete. Each edge $e = (u, v) \in \mathcal{E}$ has an associated edge weight $c_e = c(u, v)$ for each edge in this graph. We can transform this original graph into a complete graph $\mathcal{G}' = \{\mathcal{V}_{G'}, \mathcal{E}_{G'}\}$. The nodes $\mathcal{V}_{G'} = \mathcal{V}_G$, and the weight of each edge $e' = (u, v) \in \mathcal{E}_{G'}\}$ is equal to the length of the shortest path $P(u, v) = (v_1, v_2, \ldots, v_n)$ in $\mathcal{G}$, where $v_1 = u$ and $v_n = v$. This weight $C_{e'} = C(u, v)$ for an edge $e' = (u, v) \in \mathcal{E}_{G'}$ is the sum of the weights $\sum_{i=1}^{n-1} c(v_i, v_{i+1})$ for $v_i \in P(u, v)$.

The formulation of the VRP extends Dantzig, Fulkerson, and Johnson's formulation of the Traveling Salesman Problem, in which there is only one vehicle. This requires finding a set of paths/routes in G', S, that minimizes an objective function, Q, subject to a set of constraints. In the case of the classical VRP, we want to minimize the sum of the total distance traveled by all vehicles. We define x_{ij} to be a binary variable that is 1 if the solution includes the edge $(i, j) \in \mathcal{E}_{G'}$ and 0 otherwise. Therefore, our objective function Q is:

$$Q = \min \sum_{i \in \mathcal{V}_{G'}} \sum_{j \in \mathcal{V}_{G'}} C(i, j) \cdot x_{ij}$$

A set of constraints dictates vehicle behavior and solution structure as well. In the classical VRP, they are:

$$\sum_{i \in \mathcal{V}_{G'}} x_{ij} = 1 \quad \forall j \in \mathcal{V}_{G'} \setminus \{0\} \qquad (1)$$

$$\sum_{j \in \mathcal{V}_{G'}} x_{ij} = 1 \quad \forall i \in \mathcal{V}_{G'} \setminus \{0\} \qquad (2)$$

Constraint (1) states that exactly one edge enters each non-depot node, and similarly, Constraint (2) states that exactly one edge leaves each non-depot node.

$$\sum_{j \in \mathcal{V}_{G'}} x_{0j} = V \qquad (3)$$

$$\sum_{i \in \mathcal{V}_{G'}} x_{i0} = V \qquad (4)$$

Constraints (3) and (4) deal with the behavior of the vehicles at the depot, which must be both the start and end location of the V available vehicles in the classical VRP. The depot in this case is node 0. Constraint (3) states that there must be V vehicles leaving this depot node, and Constraint (4) states that there must be V vehicles returning to the same depot node.

$$\sum_{i \in \mathcal{V}_{G'} \setminus S} \sum_{j \in \mathcal{S}} x_{ij} \geq r(S), \quad \forall S \subseteq \mathcal{V}_{G'} \setminus \{0\}, S \neq \emptyset \qquad (5)$$

Constraint (5) is known as the capacity-cut constraint, and ensures that solutions are connected. The quantity $r(S)$ is the minimum number of vehicles needed to visit the nodes in a nonempty subset S of non-depot nodes. The number of nodes entering the subset S from its complement, $V \setminus S$, must be greater than or equal to the number of vehicles needed to service the nodes of S. We see that this constraint, when combined with constraints (1) and (2) that specify that the in-degree and out-degree of all non-depot nodes must be 1, gives:

$$\sum_{i \in \mathcal{V}_{G'} \setminus S} \sum_{j \in S} x_{ij} = \sum_{i \in S} \sum_{j \in \mathcal{V}_{G'} \setminus S} x_{ij}, \quad \forall S \subseteq \mathcal{V}_{G'} \setminus \{0\}, S \neq \emptyset \qquad (5)$$

This states that the number of incoming and outgoing edges for any nonempty subset S of non-depot nodes must be equal. The validity of this becomes clear noting that for every such subset S, at least $r(s)$ vehicles must enter and leave S since the vehicles must arrive from and return to the depot. Another way of stating Constraint (5) is as a subtour elimination constraint:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S), \quad \forall S \subseteq \mathcal{V}_{G'} \setminus \{0\}$$

The subtour elimination constraint states that for each subset S of non-depot nodes that the number of internal edges between the nodes that are elements of S must be less than or equal to the number of nodes in S minus the minimum number of vehicles needed to service the set S. The subtour elimination constraint explicitly prevents the route from forming cycles, whereas the capacity-cut constraint acts in conjunction with Constraints (1) and (2) to enforce the same behavior.

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{V}_{G'} \qquad (6)$$

Constraint (6) states that the variables x_{ij} are binary variables.

Minimizing the objective function Q while abiding by the above set of constraints solves the classical VRP. However, since Constraint (5) gives an exponentially increasing number of constraints as the number of nodes under consideration increases, finding the best solution to the VRP becomes increasingly difficult. The VRP is an NP-hard problem.

Problem Scenario

The variant of the VRP we are investigating can be shown by the following scenario. We are given a set of waypoints in a forest that must be visited with drones. There is a road through the forest, and we have a truck with multiple drones. Drones can be launched from and recovered to the truck, where they can be recharged, but have a limited flight endurance. What is the most efficient way to map the forest? Our definition of an "efficient" solution can vary depending on which factors we deem to be important to optimize. For example, we can find an optimal solution for total travel time, number of drones used, or regularity of truck path, among other factors. In this scenario, it is necessary to visit every waypoint, but we have a limited number of drones with a limited battery life, and the truck can only move along the road. Therefore, required constraints to consider here are first, that every waypoint is visited, second, that drones may not exceed a maximum distance D in a given flight, third, that truck motion is constrained to the road's path, and fourth, that we use less than or equal to V drones. When finding the most efficient routes, it is necessary to consider the competing needs of minimizing the maximum distance traveled by a single vehicle and minimizing the number of vehicles used. After finding the most efficient routes, we must then assign them to our V vehicles to complete the entire process in the most efficient manner.

Route Finding: First Solution Proposal

One approach to solving the problem is to use a strategy where the truck travels sequentially along the road, traversing the road from start to finish while stopping at M depots that lie on the road. We make the assumption that drones are recharged with battery swaps, so that they may be instantly recharged upon returning to the truck. In this approach, the truck deploys and collects the drones at each depot before redeploying at the next depot, meaning that the truck only visits each of the M depots once. Solving the vehicle routing problem locally and eliminating visited nodes at each of the M depots could lead to inefficiencies in the overall solution, so instead, we take a global approach and consider the solution to the vehicle routing problem with all M depots simultaneously. This then transforms into an instance of the Multi-Depot Vehicle Routing Problem, or MDVRP.

The problem formulation of the MDVRP shares many similarities with the classical VRP. We again consider a complete, directed graph G' with edges formed from the shortest paths of an original graph G. There are N non-depot nodes and M depot nodes, giving $N + M$ total nodes. There are also K total vehicles available across all depots. We define x_{ijk} to be a binary variable that is 1 if the route taken by vehicle k includes the edge $(i,j) \in \mathcal{E}_{G'}$ and 0 otherwise. Assuming that we want to limit the total distance traveled by all drones, our objective function Q is:

$$Q = \min \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^{K} C(i,j) \cdot x_{ijk}$$

Constraints for MDVRP are:

$$\sum_{i=1}^{N+M} \sum_{k=1}^{K} x_{ijk} = 1 \quad \forall j \in \{1, \dots, N\} \qquad (1)$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^{K} x_{ijk} = 1 \quad \forall i \in \{1, \dots, N\} \qquad (2)$$

Constraint (1) states that each non-depot node has exactly one incoming vehicle, and similarly, Constraint (2) states that each non-depot node has exactly one outgoing vehicle. These two constraints also prevent the same vehicle from arriving or departing the same node more than once.

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 \quad \forall k \in \{1, \dots, N\}, \forall h \in \{1, \dots, N+M\} \qquad (3)$$

Constraint (3) states that the number of incoming edges to a particular node h on the route of vehicle k is equal to the number of outgoing edges from that same node. This ensures that if a vehicle k does not have an incoming edge to a node h on its route, then it also does not have a departing edge from node h, and that if a vehicle k has an incoming edge to a node h, then it also departs that node h. Therefore, combining this with Constraints (1) and (2) allows valid routes to only travel to nodes that have not been visited before, to visit nodes only once, and to make sure that routes are continuous.

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^{N+M} x_{ijk} \leq 1 \quad \forall k \in \{1, \dots, K\} \qquad (4)$$

$$\sum_{i=1}^{N+M} \sum_{j=N+1}^{N+M} x_{ijk} = 1 \quad \forall k \in \{1, \dots, K\} \qquad (5)$$

Constraints (4) and (5) enforce vehicle availability for routes. Constraint (4) states that a particular vehicle k cannot have a route with more than 1 outgoing edge from the set of depot nodes to the set of non-depot nodes. Constraint (5) similarly states that a particular vehicle k cannot have a route with more than 1 incoming edge from the set of non-depot nodes to the set of depot nodes.

$$\sum_{i \in S} \sum_{j \in S} \sum_{k=1}^{K} x_{ijk} \leq |S| - r(S), \quad \forall S \subseteq \mathcal{V}_{G'} \setminus \{0\} \qquad (6)$$

Constraint (6) is nearly identical to the subtour elimination constraint from the classical VRP, and ensures that vehicle routes are devoid of cycles.

$$x_{ijk} \in \{0,1\} \qquad (7)$$

Constraint (7) states that the variables x_{ijk} are binary variables.

With this problem formulation, if we have a maximum of V vehicles and M depots, and we decide to use the previously mentioned strategy, finding the optimal routes is an instance of the MDVRP with M depots and $V \cdot M$ vehicles. However, we must also limit the maximum number of vehicles that can leave and return from a single depot to V, leading to two more constraints:

$$\sum_{j=1}^{N} \sum_{k=1}^{K} x_{hjk} \leq V \quad \forall h \in \{N+1, \dots, N+M\} \qquad (8)$$

$$\sum_{i=1}^{N}\sum_{k=1}^{K} x_{ihk} \leq V \quad \forall h \in \{N+1, \ldots, N+M\} \qquad (9)$$

Constraint (8) states that the maximum number of outgoing edges from a particular depot h to the set of non-depot nodes cannot exceed V, and Constraint (9) similarly states that the maximum number of incoming edges from the set of non-depot nodes to a particular depot h cannot exceed V.

Modification A: We may also have an additional requirement that the vehicles can only travel a maximum distance of D, then we add another constraint:

$$\sum_{i=1}^{N+M}\sum_{j=1}^{N+M} C(i,j) \cdot x_{ijk} \leq D \quad \forall k \in \{1, \ldots, K\} \qquad (10)$$

Constraint (10) ensures that the cost of the route of a particular vehicle k is less than or equal to the maximum distance D.

Modification B: We may also want to remove the requirement that the solution must cover every node. Instead, if we fail to visit some node, we should incur some penalty. Therefore we remove Constraints (1) and (2), and instead modify the objective function, Q, to reflect any penalties.

$$Q = \min \sum_{i=1}^{N+M}\sum_{j=1}^{N+M}\sum_{k=1}^{K} C(i,j) \cdot x_{ijk} + p \cdot V^*$$

In the modified objective function, p is the penalty per missed node, and V^* is the number of nodes missed by the solution. This penalty becomes useful in applications where we would like solutions that prioritize visiting as many nodes as possible, or where specific nodes should be prioritized over others. In the first case, we set the penalty for all nodes to be very high, so that nodes are only dropped from the solution when absolutely necessary. In the second case, we the penalty for nodes should increase as their importance of inclusion in the solution increases.

Modification C: We can also prioritize minimizing the maximum distance that any vehicle travels. To do this, we again modify the objective function:

$$Q = \min \sum_{i=1}^{N+M}\sum_{j=1}^{N+M}\sum_{k=1}^{K} C(i,j) \cdot x_{ijk} + t \cdot \max(R)$$

In the modified objective function, R is the set of all vehicle route costs, and t is a chosen coefficient. The value of t is set to be very large when minimizing the maximum distance is very important, since the weight of the second term in the objective function is increased.

The competing needs of minimizing the maximum vehicle distance and minimizing the number of vehicles result in solutions that experience a tradeoff between these two important factors. Optimal solutions can be thought of as a Pareto frontier, giving many options to choose from. One method for finding an optimal solution set of routes is described below.

We will find the solution with the least distance traveled by any one vehicle, D and minimize the number of vehicles used while constraining them by this distance.
Step 1: In an MDVRP, limit the number of vehicles per depot to $V = 1$ (enforce constraints 8 and 9). Relax the maximum distance constraint (modification A), and prioritize minimizing the maximum distance (modification C). Increment V by 1 until the maximum distance any vehicle travels is minimized while still visiting all waypoints. Let this distance be D.
Step 2: Again in an MDVRP with the same nodes, limit the number of vehicles per depot to $V = 1$. Relax the constraints governing visiting all nodes, and set a high penalty per dropped node instead (modification B). Set the maximum distance of any vehicle to be D and increment V by 1 until all nodes are included in the solution.

The solution found in Step 2 should ideally be the same as the solution from Step 1, but because VRP is NP-hard, in practice this may not be the case. The solution found in Step 2 can also find a solution with the same maximum distance D, but using less vehicles, since the objective function for Step 1 prioritized minimizing the maximum distance instead of visiting the highest possible number of nodes with the least number of vehicles.

Alternate methods for finding solutions that lie on the optimal Pareto frontier may introduce other criteria such as total distance traveled by all vehicles or total time to complete the solution to determine which solution out of the family of optimal solutions is the best for a given application. The underlying similarity between all MDVRP-based solutions, however, is that because all drones start and end at the same depots, the planned truck path is very simple: start at one end of the road, stop at the first depot, deploy and collect drones, and repeat this process for all M depots until the entire road has been traversed.

Route Finding: Second Solution Proposal

Another approach to solving the overarching problem is to alter the classical VRP by modifying the graphs G and G' and modifying the objective function Q. We again have a problem with M depots placed on the road, have N non-depot nodes, have V vehicles available, and have some maximum distance per vehicle D. Our desired behavior in this approach is to allow the vehicles to travel freely along the road as if they are being transported by the truck, leave the road to visit waypoints, and return to the road to be picked up by the truck. Once the vehicles leave the road, they should not be able to use the road to "cheat"; that is, they should not exploit the free cost of travel between depots on the road to reduce the travel time between non-depot nodes. However, since we are using the classical VRP, we still only have 1 primary depot. We must model the existence of the $M-1$ other total depots.

The defining property of a depot is that vehicles leave and enter the depot. In the classical VRP and in the MDVRP, vehicles were required to start and end at the same depot. In this variant, vehicles may start and end at any of the M depots, since they all lie on the road. We want to transform previously non-depot nodes into $M-1$ pseudo-depots. When we examine the classical VRP constraints, we see that Constraints (1) and (2) require non-depot nodes to have exactly one incoming and one outgoing edge in the solution. However, any number of the V vehicles could leave or enter the pseudo-depots in our desired approach. In fact, a particular pseudo-depot could have in-degree and out-degree between 0 and V. To model this fact, we split the nodes at the locations of the $M-1$ pseudo-depots into $2V$ nodes that share the same pseudo-depot location. Each of the $2V$ nodes behaves exactly like a normal non-depot node, with maximum in-degree of 1 and maximum out-degree of 1. Therefore, the M depots are modeled by $1+2V(M-1)$ nodes, with 1 true depot node and $2V(M-1)$ pseudo-depot nodes.

The rest of the problem can be modeled through modifying edges in the graph G' and altering the objective function Q. In the core formulation, our objective function Q is similar to the classical VRP, but we add a term for penalties:

$$Q = \min \sum_{i \in \mathcal{V}_{G'}} \sum_{j \in \mathcal{V}_{G'}} C(i,j) \cdot x_{ij} + p \cdot V^*$$

However, the penalty for dropped nodes, p, only applies to non-depot nodes. Dropped depot or pseudo-depot nodes incur no penalty for being excluded by a solution. This makes sure that solutions prioritize visiting non-depot nodes and are unaffected when they exclude nodes that model the M depots, effectively removing the requirement to visit all of the pseudo-depot nodes, making them optional to visit. Furthermore, because of the presence of a penalty, Constraints (1) and (2) from the classical VRP, which enforce that every node must be visited, are removed. We keep Constraints (3) and (4) from the classical VRP, however:

$$\sum_{j \in \mathcal{V}_{G'}} x_{0j} = V \qquad (3)$$

$$\sum_{i \in \mathcal{V}_{G'}} x_{i0} = V \qquad (4)$$

These constraints state that there must be V vehicles leaving and returning to the true depot, which is node 0. This constraint still applies since all vehicles still leave and return to node 0, the true depot. However, we also want to enforce free movement between the true depot and pseudo-depot locations, modeling the ability of the truck to collect and redeploy drones at different locations along the road, but while a drone is on its route, we want to prohibit it from using the road to "cheat" on its route. To do so, we introduce three modifications to the edges of the graph, G':

$$C(0, j) = 0 \quad \forall j \in \{1, \dots, 2V(M-1)\} \qquad (a)$$

$$C(i, 0) = 0 \quad \forall i \in \{1, \dots, 2V(M-1)\} \qquad (b)$$

$$C(i, j) = \infty \quad \forall i, j \in \{1, \dots, 2V(M-1)\} \text{ where } i \neq j \qquad (c)$$

The first two edge weight modifications, (a) and (b), state that the cost of moving from the true depot, node 0, to any pseudo-depot node is 0, and that the cost of moving from any pseudo-depot node to the true depot node is also 0. This allows for free travel at no cost along the road from the true depot to any pseudo-depot node. The third constraint, (c), makes direct travel between pseudo-depot nodes impossible. Vehicles may only travel between pseudo-depot nodes through non-depot nodes. This makes it impossible for vehicles to use the road to take shortcuts in their routes. In fact, a vehicle may only take a shortcut through node 0, but since node 0 is the true depot, and the in-degree and out-degree of the true depot node are both V by Constraints (3) and (4), the vehicle's route must end when it reenters node 0. Therefore, a vehicle that starts at the true depot, travels to pseudo-depot node u, visits some waypoints, returns to the road at pseudo-depot node u', and ends back at the true depot has route cost only equal to the section of the route from u to u'. Because edge modification (c) prohibits cheating along the road, we have enforced the desired behavior along the M depots that model the road.

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S), \quad \forall S \subseteq \mathcal{V}_{G'} \setminus \{0\} \qquad (5)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{V}_{G'} \qquad (6)$$

Lastly, we still retain Constraints (5) and (6) from the classical VRP, which are the subtour elimination constraint and the definition of the binary variable x_{ij}, respectively.

Similar to the previous MDVRP problem formulation, we can implement Modifications A and C in this modified VRP approach to arrive at problem formulations that enforce a maximum distance for any drone or minimize the maximum distance traveled by any drone, respectively. We have already used a form of Modification B in our objective function Q, since we applied a penalty when dropping non-depot nodes. Similar optimal route-finding methods would also apply to this modified VRP approach as in the MDVRP, but a significant difference in the overall solution to the problem would be the process for assigning vehicle routes and constructing the optimal truck path. Unlike in the MDVRP-based formulation, the vehicle routes may not start and end at predetermined depots, making optimization of the truck route for timely collection and deployment of drones a significant task. More attention should be provided here once the process for finding optimal routes has been more thoroughly refined in this approach.