Optimal routing for fleets of multi-package delivery drones

MAST90014 Group #5

Brendan Hill, Kushwant Singh, Yuci Jiang, Nicholas Liu, Vladimir Borzykh

August 21, 2018

Abstract

Delivery drones are an emerging technology with potential for widespread applications. Route planning is a critical part of managing a fleet of drones. In this report we explore multiple ILP models for optimising route planning for a fleet of multi-package, multi-trip delivery drones, with respect to minimizing travel distance and overall waiting time. Various sub-tour elimination and symmetry elimination procedures are considered, and the performance and scalability of the solutions are evaluated with a focus on the variability in solving time. We find that optimal solutions are attainable up to 30-50 packages within 10 minute time limits, but that solution times overall are highly variable.

1 Introduction

Delivery drones are an emerging technology, with applications ranging from parcel delivery, food delivery, emergency relief, surveillance, and more. Given the potential for autonomous flight, these vehicles present an opportunity for business and organisations to deliver products and packages faster and more cost effectively then ever before, to locations which may be remote or hard to access, and without the path planning constraints of ground based vehicles.

In recent years, some major companies have announced the trialling or deployment of delivery drones:

- Amazon is currently testing Amazon Prime Air [1] to make deliveries in less than 30 minutes in rural areas. The drones can carry up to 5 pounds per delivery trip, and the customer must be within 16km of an Amazon distribution center. Along with Amazon, Ali Baba, Swiss Post and Google are all operating or planning to operate delivery drone services too.
- In 2013, DHL introduced its 'Parcelcopter' [2] which delivered medical supplies, highlighting the importance of operating drones during disaster relief scenarios. Over recent years, this has become an important focus, especially in isolated regions in African countries. For example, the company Zipline is actively using drones to transport medical supplies, such as blood, in Rwanda and Tanzania.
- Another area in which the use of drones has grown rapidly over the last 5 years is surveying. Compared to conventional types of surverying drones offer much broader and denser sampling in a fraction of time. Companies such as 3DR are now offering site scans autonomously done by a drone as a service. In the process of surveying the drone is required to visit a set of locations to collect the required data. Even though no deliveries are to be made by the drone, the problem of routing closely correlates with a typical VRP.
- Bell and Boeing have developed a prototype delivery drone called Autonomous Pod Transport (APT) [4], a five-foot-tall, four-rotor drone that can deliver 10 pound packages up to 50 miles. Their plans include an 8-rotor version that can carry 200 pounds of payload up to 300 miles.

As the range of applications widens and the sizes of fleets grow, the ability to optimise the route planning becomes more critical in maximising the efficiency and cost effectiveness of the technology. From an organisation's perspective, management of the financial costs includes consideration of hardware usage, battery pack size, flight time and power consumption. From a recipient's perspective, deliver time is likely to take priority, particularly in an emergency relief situation.

Thus, the different uses of drone services suggest the need to focus on multiple objectives when optimising drone routes. For companies that use drones for delivering parcels as part of their business operations, the objective is to minimise the operational costs of deploying the drones. In addition, in emergency situations, the objective is to minimise the waiting time for customers or patients that urgently need medical supplies. In this report, we attempt to accommodate each of these concerns in our set of models.

The drone routing problem is a special case of the the more general vehicle routing problem (VRP), which models the movement of multiple vehicles between various destinations, generally with the objective of minimising time or distance traveled. Much of the existing literature on this topic focuses on manned vehicles, such as delivery trucks. With the advent of drone technology, more importance is now placed on formulating VRPs for unmanned aerial vehicles (UAVs). One difference between UAVs and traditional vehicles is the ability to make multiple trips from a common location, usually a depot, leading to an emphasis on multiple-trips vehicle routing problems (MTVRP).

MTVRP's can be solved using various methods including a large neighborhood search algorithm [7] and a hybrid genetic algorithm [5], and other heuristic methods. Mixed Integer Linear Programming (MILP) on the other hand is capable of attain optimal solutions, or at least guaranteed convergence to optimality with a proven bound on the optimality. However, it has been showed that solving MILP problems are in general NP-hard, and even with the best commercial solvers in existence today, the attainment and proving of global optimality can be extremely slow for large problem instances [6].

In our project, we focus on the use of MILP to optimise the delivery routes of multi-package, multi-trip delivery drones, with respect to the minimization of distance traveled, and separately, the minimization of total waiting time for all trips to complete. For experimentation we adopt some real world drone specifications, compare multiple ILP models, and evaluate the scalability and variability in solving time of the best models with respect to both objective functions.

2 Problem formulation

2.1 Assumptions and simplifications

To model the problem using MILP techniques, we adopted the following assumptions:

- The velocity of drones is approximately constant, and drones travel in direct lines, where the effects of payload weight, weather, etc are excluded. This assumption means that minimising distance is equivalent to minimising time for any particular drone; a fact relied on for one of our models.
- The time taken for drones to drop off parcels at each location, as well as recharge their batteries, is either negligible or constant (hence ignorable).
- Each drop-off point has a single package; multiple packages would simply be modeled as multiple drop-off points at the same location.
- The total demand of each customer is less than the carrying capacity of a drone. Hence, a drop off point only needs to be visited once to fulfil the demand.
- All spatial elements are modeled with 2D Euclidean space, without consideration for geographic or aerospatial restrictions on flight paths.

The following table provides some real world drone specifications:

Drone Specifications				
	Quadracopter 3D Robotics IRIS		Octocopter Turbo Ace Infinity 9	
	Standard	Modified	Standard	Modified
Weight(kg)	0.5	1	8.1	10
Range(km)	1.3	3.5	1.8	4.2
Speed(km/h)	5-7	5-7	9-13	9-13

In our experimentation, we adopted the Modified Octocopter Turbo Ace Infinity 9 specifications.

2.2 Notation

A warehouse/depot h is located at the origin in 2D Euclidean space. A set of packages $\mathcal{P}$ is stored at h. Each package $p \in \mathcal{P}$ has weight w_p and must be delivered to the customer at position x_p, y_p. The fleet is comprised of the set of drones $\mathcal{D}$ where each drone $d \in \mathcal{D} = \{1, \ldots, |\mathcal{D}|\}$ has a set of possible trips taken $t \in \mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$, a maximum weight capacity of C, a maximum flying distance for each trip of F.

The graph representing all possible movements is the complete digraph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ where $\mathcal{N} = \mathcal{P} \cup \{h\}$. The precomputed Euclidean distance (i.e. cost) between i, j is denoted c_{ij}. The binary decision variable $X_{ij}^{dt} = 1$ exactly when drone d on its t'th trip travels from node i to node j, where $\langle i, j \rangle \in \mathcal{E}$.

2.3 Objectives

In this analysis we consider two objectives:

1. ("DISTANCE") Minimize the total distance traveled by all drones:

$$\min \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} \sum_{ij \in \mathcal{E}} c_{ij} X_{ij}^{dt}$$

   Note that the allocation of trips between the drones does not affect the objective value; a fact relied on in our first model.

2. ("WAITINGTIME") Minimize the total waiting time for all trips to complete:

$$\min \left(\max \left\{ \sum_{t \in \mathcal{T}} \sum_{ij \in \mathcal{E}} c_{ij} X_{ij}^{dt} : d \in \mathcal{D} \right\} \right)$$

   In this objective, the allocation of trips between drones can affect the total flying time of each drone, hence the maximum waiting time.

2.4 Solutions

Solutions to the problem are defined as list of (d, t, i, j) tuples which define the set of movements for all drones for all trips - where the d'th drone on the t'th trip travelled from i to j. Example of solutions are shown in Figure (1)

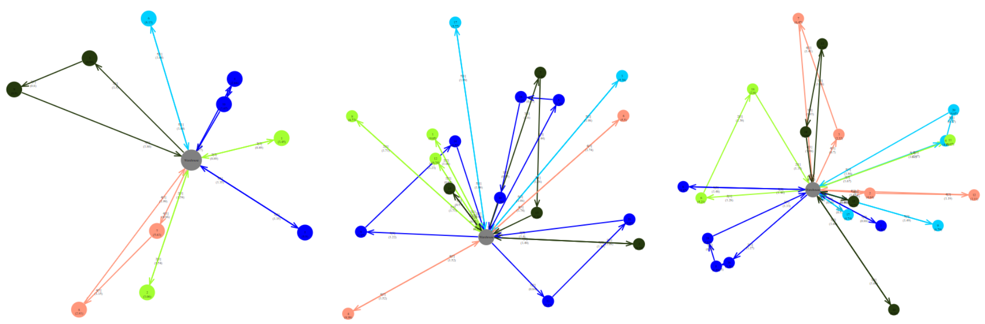

Figure 1: Optimal solutions for randomly generated scenarios

It can be seen in these sample solutions that some routes appear sub-optimal with respect to distance traveled, as some obvious swaps of packages between routes would reduce the total distance (as per the triangle inequality). However in these cases, it can be shown that such package swaps would result in the package weight restrictions being violated (eg. a trip would exceed the drone carrying capacity); and indeed these solutions are optimal for the complete problem.

3 Models implemented

We implement and evaluate three separate ILP models of the problem. All models are able to solve the DISTANCE objective, while only the third model is sufficiently expressive to solve WAITINGTIME objective.

- The simplest model ("SINGLE") models the trips made by a single drone, without concern for how the trips might subsequently be distributed among multiple drones. Given the DISTANCE objective, this problem is equivalent to the full problem.
- The next model ("TWOSTAGE") is a two stage process which first minimizes the total distance traveled using the MTZ constraints for subtour elimination; then in the second stage, the trips are distributed among the individual drones. This model excludes consideration of individual package weights and assumes a mean value.
- The final model ("MULTIOBJ") incorporates all elements of the problem including multiple drones, and individual package weights. The objective function is comprised of a parameterized weighted combination of total distance traveled, and maximum distance traveled by any drone. With appropriate weighting, this model is able to solve both the DISTANCE and WAITINGTIME objectives.

  Within MULTIOBJ we experiment with three different approaches to sub-tour elimination including a reduced version of the Dantzig approach, which leverages aspects of the problem domain to avoid the exponential increase in the number of constraints. Furthermore, we explicitly address the high degree of symmetry given the possible allocations of trips to drones, and directions of travel.

3.1 SINGLE model

In this model we ignore the different drones and different trips and essentially model the path of a single drone which visits all nodes, returning to the warehouse h as required to satisfy the weight (C) and distance (F) limitations. This formulation models the cumulative distance traveled, strictly increasing, at each node (starting with zero at the warehouse), and the weight still carried at each node, strictly decreasing (ending with zero at the warehouse). Any subtour which lacked a visit to the warehouse h would contain some point of non-increasing cumulative distance, or non-decreasing weight carried, which would violate these constraints and be infeasible - hence, subtour elimination occurs automatically in a manner analogous to the Commodity Flow formulations, without the need for further constraints.

Variables

- $X_{ij} \in \{0, 1\}, X_{ij} = 1$ when the drone travels along arc $\langle i, j \rangle$
- $D_i \geq 0$ as the cumulative distance traveled up to and including node i, since visiting h
- $W_i \geq 0$ as the total weight carried upon arriving at node i (i.e. before release)

Objective function

$$\min \sum_{ij \in \mathcal{E}} c_{ij} X_{ij} \tag{1}$$

Constraints

Each package must have one incoming and one outgoing arc utilized:

$$\sum_{i \in \mathcal{N} \setminus j} X_{ij} = 1 \qquad \forall j \in \mathcal{P} \tag{2}$$

$$\sum_{j \in \mathcal{N} \setminus i} X_{ij} = 1 \qquad \forall i \in \mathcal{P} \tag{3}$$

The cumulative distance from the warehouse must increase along arcs traveled:

$$D_h = 0 \tag{4}$$

$$-F(1 - X_{ij}) + D_i + c_{ij} \le D_j \le D_i + c_{ij} + F(1 - X_{ij}) \qquad \forall ij \in \mathcal{E}, j \ne h \tag{5}$$

The weight carried from the warehouse must decrease along arcs traveled:

$$-C(1 - X_{ij}) + W_i - w_i \le W_j \le W_i - w_i + C(1 - X_{ij}) \qquad \forall ij \in \mathcal{E}, i \ne h \tag{6}$$

$$W_h = 0 \tag{7}$$

Distance and capacity limits must be respected at each node:

$$D_i + c_{ih} \le F \qquad \forall i \in \mathcal{P} \tag{8}$$

$$W_i \le C \qquad \forall i \in \mathcal{P} \tag{9}$$

Constraint (8) reflects the fact that at each node, the cumulative distance traveled, plus the distance remaining if returning to the warehouse immediately, must not exceed the maximum distance.

3.2 TWOSTAGE model

This model breaks the problem up into two stages: 1) find the optimal set of trips to minimize total distance traveled; 2) allocate the trips to drones. The first stage uses a *multi traveling salesman problem* (mTSP); the second solves a *makespan scheduling problem.* This model ignores the individual weights of packages and assumed the mean deliverable package weight given the drone specifications. This potentially weakens the optimality and feasibility of the solution with respect to the original problem, but may still serve as an approximation.

3.2.1 Stage 1: mTSP model

The ILP model is as follows:

Variables

- $X_{ij}^t \in \{0, 1\}$, $X_{ij}^d = 1$ when trip t involves moving from i to j
- $U_i \ge 0$, the order of nodes visited, as per the MTZ formulation

Objective

$$\min \sum_{t \in \mathcal{T}} \sum_{ij \in \mathcal{E}} c_{ij} X_{ij}^t$$

Constraints

Every trip must start and end at the warehouse:

$$\sum_{j \in \mathcal{P}} X_{hj}^t = 1 \qquad \forall t \in \mathcal{T} \tag{10}$$

$$\sum_{i \in \mathcal{P}} X_{ih}^t = 1 \qquad \forall t \in \mathcal{T} \tag{11}$$

Each package must be visited, and left, within the same trip

$$\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} X_{ij}^t = 1 \qquad \forall j \in N \tag{12}$$

$$\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}} X_{ij}^t = 1 \qquad \forall i \in N \tag{13}$$

$$\sum_{i \in \mathcal{N}} X_{ip}^t - \sum_{j \in \mathcal{N}} X_{pj}^t = 0 \qquad \forall t \in \mathcal{T}, p \in \mathcal{P} \tag{14}$$

Each trip must not exceed the weight and distance capacities

$$\sum_{ij \in \mathcal{E}} X_{ij}^t \leq C \qquad \forall t \in \mathcal{T} \tag{15}$$

$$\sum_{ij \in \mathcal{E}} c_{ij} X_{ij}^t \leq F \qquad \forall t \in \mathcal{T} \tag{16}$$

plus the standard MTZ subtour elimination constraints.

3.2.2 Makespan Schedule

This is a standard formulation of the makespan schedule, given a fixed solution from stage 1:

Objective

$$\min \max_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} c_i x_{ij}^{dt} \tag{17}$$

Constraints

$$\sum_{d \in \mathcal{D}} x_{ij}^{dt} = 1 \qquad \forall j \in \mathcal{N} \tag{18}$$

Where the $x_{ij} \in \{0, 1\}$ denotes the ith drone in jth box, and the c_i is the cost of ith tour.

3.3 Full model, multi-objective (MULTIOBJ)

This model includes variables for individual drones and trips made in a single model, allowing for the notion of "total distance each drone travels" to be expressed, and the maximum such value to be minimized.

Several subtour elimination methods are considered, along with symmetry elimination constraints.

Variables

- $T^{dt} \in \{0, 1\} = 1$ when drone d actually makes a t'th trip
- $X_{ij}^{dt} \in \{0, 1\} = 1$ when drone d on trip t goes from node i to node j
- $V_i^{dt} \in \{0, 1\} = 1$ when drone d on trip t visits node i
- $W^{dt} \geq 0$ be the total weight carried by drone d on trip t
- $D^{dt} \geq 0$ be the total distance traveled by drone d on trip t
- $D^* \geq 0$ be the maximum distance traveled by any drone in total summed across all its trips

Objective

$$\min \phi \left[\sum_{d \in D} \sum_{t \in \mathcal{T}} \sum_{ij \in E} c_{ij} X_{ij}^{dt} \right] + \psi D^* \tag{19}$$

The ϕ, ψ values dictate whether the model minimizes total distance traveled, or minimizes the maximum waiting time for any drone to complete all its trips, or some linear combination of the two elements.

Constraints

Constrain visits and movements to trips actually taken

$$V_i^{dt} \leq T^{dt} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T}, i \in \mathcal{N} \tag{20}$$

$$X_{ij}^{dt} \leq T^{dt} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T}, ij \in \mathcal{E} \tag{21}$$

Every trip taken must start and end at the warehouse

$$\sum_{j\in\mathcal{P}} X_{hj}^{dt} = T^{dt} \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{22}$$

$$\sum_{i\in\mathcal{P}} X_{ih}^{dt} = T^{dt} \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{23}$$

Every package must be delivered

$$\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}} V_i^{dt} = 1 \qquad \forall i\in\mathcal{P} \tag{24}$$

Inflow, outflow for visited nodes

$$V_i^{dt} = \sum_{j\in\mathcal{N}\setminus\{i\}} X_{ij}^{dt} \qquad \forall d\in\mathcal{D}, t\in\mathcal{T}, i\in\mathcal{P} \tag{25}$$

$$V_i^{dt} = \sum_{i\in\mathcal{N}\setminus\{j\}} X_{ij}^{dt} \qquad \forall d\in\mathcal{D}, t\in\mathcal{T}, j\in\mathcal{P} \tag{26}$$

Calculate and cap the distance on each trip; calculate maximum distance

$$D^{dt} = \sum_{ij\in\mathcal{E}} c_{ij} X_{ij}^{dt} \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{27}$$

$$D^{dt} \leq F \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{28}$$

$$\sum_{t\in\mathcal{T}} D^{dt} \leq D^* \qquad \forall d\in\mathcal{D} \tag{29}$$

Calculate and cap the weight of each trip

$$W^{dt} = \sum_{i\in\mathcal{P}} w_i V_i^{dt} \qquad \forall d\in D, t\in\mathcal{T} \tag{30}$$

$$W^{dt} \leq C \qquad \forall d\in D, t\in\mathcal{T} \tag{31}$$

3.3.1 Variations on sub-tour elimination

We evaluated three versions of sub-tour elimination in this model:

Reduced Dantzig method ("RD")

The Dantzig approach to sub-tour elimination is to explicitly eliminate each possible sub-tour S with the following constraint:

$$\sum_{ij\in S} X_{ij} \leq |S| - 1$$

In practice, the number of possible sub-tours is exponential in the number of nodes and this approach is impractical for non-trivial problems.

However in the MULTIOBJ model, we find that most theoretical sub-tours are automatically eliminated by the weight and/or distance constraints (28, 31). For example, a sub-tour which spans every node will surely exceed both the package and distance constraints and hence is not a part of any feasible solution.

Hence, it is unnecessary to explicitly eliminate every such sub-tour - instead we can scan for sub-tours which are within the weight and distance limits and eliminate them only. This can be done relatively efficiently by identifying such cycles of length 2, then considering each possible extension of each cycle to length 3, then each possible extension of each cycle to length 4, and so on. By the triangle inequality, cycles of length k which already exceed the distance/weight limits need not be extended further. When no possible cycles of length k are found, then the process terminates, and all feasible cycles have been identified. Each such cycle S is then eliminated for each drone, for each trip:

$$\sum_{ij \in S} X_{ij}^{dt} \leq |S| - 1 \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \tag{32}$$

Thus, this reduced version of Dantzig approach to sub-tour elimination can be used without an exponential increase of constraints in the number of nodes. In practice for the scenarios considered, we found that 100's or 1000's of subtours required elimination, compared to the theoretical billions/trillions necessary with the full Dantzig model.

However there are potential downsides with this method. As the weight and distance capacities of the drones increase, the number of feasible sub-tours could increase significantly. Also, if the packages are more concentrated around the warehouse (eg. as may be typical around a city center), then the number of sub-tours within the distance constraints will increase. Hence the Reduced Dantzig model is only practical when each drone is expected to be able deliver a relatively small number of packages (eg. 1-5) on each trip.

MTZ method

This approach implements a version of the MTZ constraints, customized for the multi-drone, multi-trip problem.

Additional variables

- $U_i^{dt} \geq 0$ - reflects the order drone d on trip t visits the nodes, as per the MTZ formulation. Set to 0 for nodes which are not visited by the drone on this trip.

Additional constraints

$$U_i^{dt} - U_j^{dt} + (|\mathcal{N}| - 1)X_{ij}^{dt} \leq (|\mathcal{N}| - 2) \qquad \forall d \in \mathcal{D}, t \in \mathcal{T}, i \in \mathcal{N} s.t. i \neq j, i \neq h, j \neq h \tag{33}$$

Distinguish the warehouse node

$$U_h^{dt} = 0 \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \tag{34}$$

Should be 0 for non-visited nodes

$$U_i^{dt} \leq |N| * V_i^{dt} \qquad \forall d \in D, t \in \mathcal{T} \tag{35}$$

GG method

The GG (Gavish and Graves) method for subtour elimination models a single commodity flow across the path taken. We customized this for the multi-trip version of the problem as follows.

Additional variables

- $G_{ij} \geq 0$, the volume of "commodity" still slowing at i

Additional constraints

Outflow must be inflow minus one unit

$$\sum_{j \in \mathcal{N}} G_{pj} = \sum_{i \in \mathcal{N}} G_{ip} - 1 \qquad \forall p \in \mathcal{P} \tag{36}$$

The flow is zero on arcs not utilized

$$G_{ij} \leq |N| \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} X_{ij}^{dt} \qquad \forall ij \in \mathcal{D}, j \neq h \tag{37}$$

Combinations

For some MIP problems, adding further constraints (including logically redundant ones) can improve the time taken to find the solution by creating a tighter formulation. However, extra constraints can also increase the overhead. In this approach we consider the combination of MTZ+GG, as well as RD+MTZ+GG, to see whether their combined constraints can improve the solving time.

3.3.2 Variations on symmetry elimination

The solution space naturally contains a high degree of symmetry in the allocation of trips to drones, the order of trips taken by a drone, and the orientation of each trip.

The following constraints ensure each drone has a least as many trips as the next; each trip is at least a long at the next; each first node visited is at least a far away from the warehouse as the final node visited; and so on, forcing a unique solution within these elements of symmetry.

$$D^{dt} \geq D^{d(t+1)} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \setminus \{T\} \tag{38}$$

$$D^{d1} \geq D^{(d+1)1} \qquad \forall d \in \mathcal{D} \setminus \{D\} \tag{39}$$

$$T^{dt} \geq T^{d(t+1)} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \setminus \{T\} \tag{40}$$

$$T^{dt} \geq T^{(d+1)t} \qquad \forall d \in \mathcal{D} \setminus \{D\}, t \in \mathcal{T} \tag{41}$$

$$\sum_{i \in \mathcal{P}} i X_{ih}^{dt} \geq \sum_{j \in \mathcal{P}} j X_{hj}^{dt} \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \tag{42}$$

$$\tag{43}$$

Note that not all of these constraints were used in the final solution, since some delivered a performance improvement while others worsened the solution time.

4 Experimental evaluation

Scenarios were randomly generated by placing packages uniformly in the disk around the warehouse within a maximum radius of 2.0 km, and assigning weights uniformly from 0.0 to 10.0 kg. This guaranteed that each scenario was feasible within the drone specifications selected.

The framework was coded in Julia/JuMP. All solutions were generated using CPLEX 12.7 in the Spartan High Performance Computing cluster [3] provided by the University of Melbourne. Each job was allocated 8 cores and 62GB RAM, with a time limit of 10 minutes.

4.1 Comparison of subtour elimination methods (MULTIOBJ)

To identify the most efficient subtour elimination method for MULTIOBJ, we considered 20 randomly generated scenarios with 5 drones and 16 packages each with the WAITINGTIME objective. Each method was evaluated, along with a "None" approach where no subtour elimination was performed - this generates infeasible solutions, but may serve as a rough baseline for performance. The distributions of each method (and certain combinations) are shown in Figure (2).

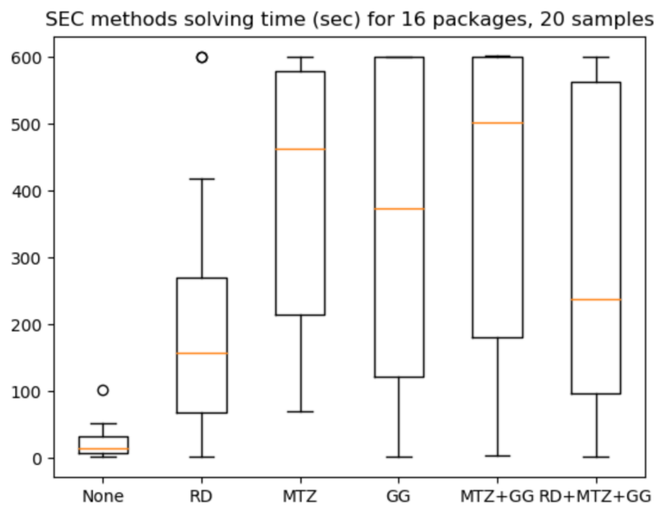

Figure 2: Distribution of runtimes (sec) over 20 samples, 16 packages

Among the methods considered, the Reduced Dantzig (RD) method is the fastest on average. We note that a significant amount of the computation time is attributed to subtour elimination as the “None” method was an order of magnitude faster, on average. We also note the significant variation in solving time for all methods.

We conclude that RD is the most efficient method tested for subtour elimination, and this was fixed for MULITOBJ during the remaining experimentation.

4.2 Evaluation of symmetry elimination constraints (MULTIOBJ)

To establish the benefit of the symmetry elimination constraints in reducing the run-time, we considered the following combinations of symmetry elimination constraints:

Identifier	Constraints applied
None	
A	38
B	39
C	40
D	41
E	42
F	38, 41
G	38, 42
All	38,39,40,41,42

In total, 50 scenarios with 5 drones and 15 packages were randomly generated, and all combinations were tested against the same set of scenarios. The distributions of solving times for each combination are shown in Figure (3). The distribution of solving times relative to “None” for each scenario is shown in Figure (4), where values below 1.0 indicate a performance improvement. The average improvement, relative to “None”, given by each configuration is given in Figure (5).

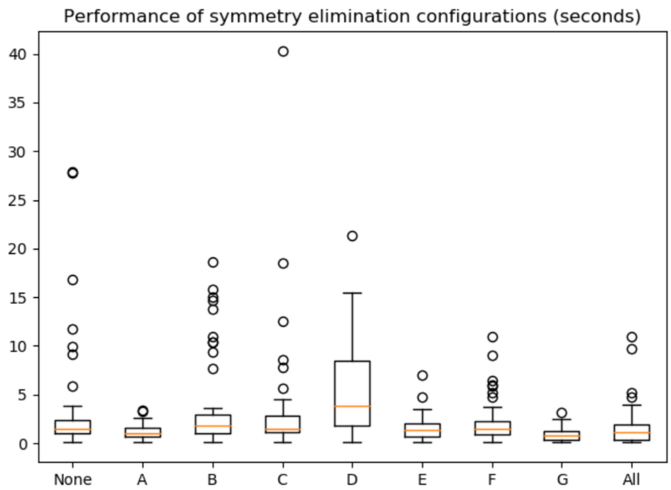

Figure 3: Absolute solution time across 20 scenarios

We note that combinations A and G were faster on average, with significantly lower variability. Some methods such as B and D actually worsened the average performance.

The average relative performance for each method is given in the following table:

Configuration	**Relative performance**
A	0.6546
B	1.4448
C	1.061
D	2.8512
E	0.7608
F	0.9184
G	0.5214
All	0.7646

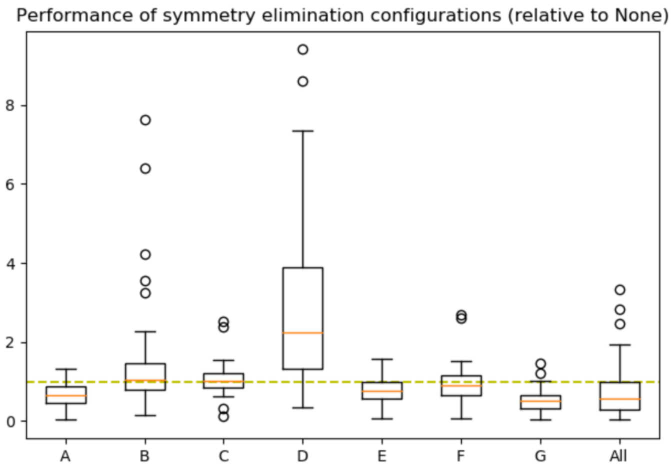

Figure 4: Relative solution time (to None) across 20 scenarios

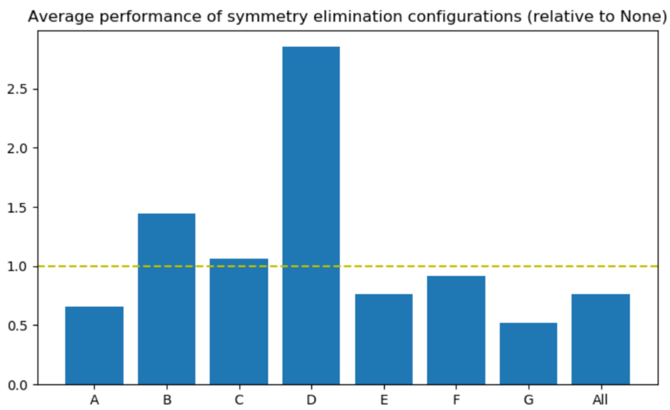

Figure 5: Average relative time (to None) across 20 scenarios

It is clear from these results that certain combinations of the symmetry elimination constraints deliver a performance benefit on average. We conclude that combination G is the best tested, and this configuration was fixed for MULTIOBJ during the remaining experimentation.

4.3 Models comparison for minimising DISTANCE objective

Each of the models, SINGLE, TWOSTAGE and MULTIOBJ are able to model the minimization of total distance traveled. To identify the most efficient, we generated 20 random scenarios with 5 drones and 15 packages, and tested each model against this set. The distribution of solving times is given in Figure (6), and the average solving times given in the table below.

Model	Average solving time (sec)
SINGLE	0.53
TWOSTAGE	81.09
MULTIOBJ	90.16

Clearly, the SINGLE model is on average over 100x faster than the other methods, and this model was used for the scalability test.

This performance gain may be attributable to the significantly reduced number of integer variables, given the exclusion of concerns around allocation of trips to drones.

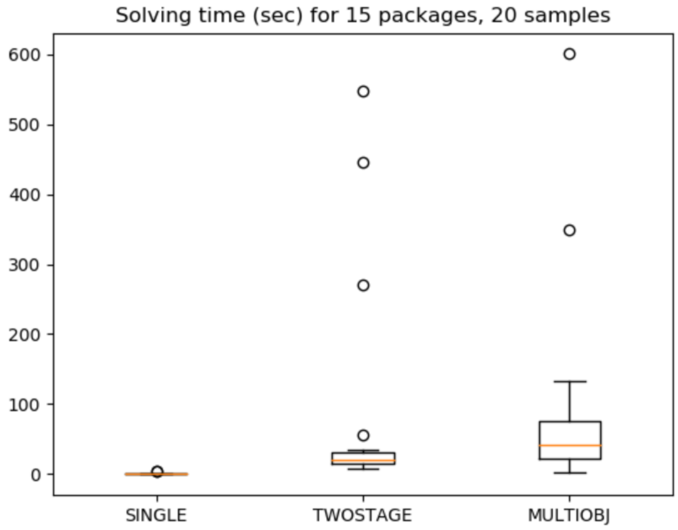

Figure 6: Comparison of all models for DISTANCE objective

4.4 Scalability of SINGLE for DISTANCE objective

Since SINGLE was by far the most efficient model for the DISTANCE objective, we next considered its scalability. Package counts from 14 to 48 were considered and 20 random scenarios were generated for each count. The results shown in Figure (7) depict the average solving time, along with ± 1 standard deviation to give some indication of the variability.

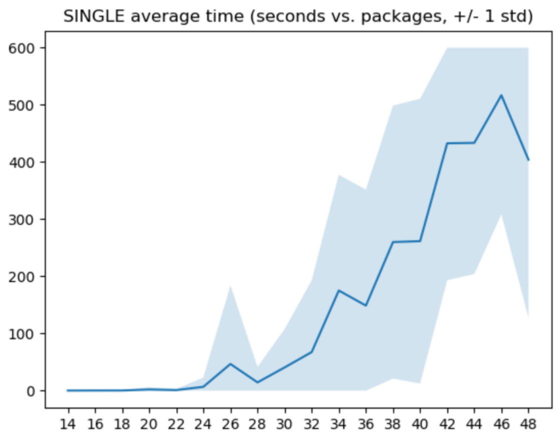

Figure 7: Solving time of SINGLE as packages increase

We find that the solution times increase significantly from 24 packages onwards, although optimal solutions are usually found for scenarios with up to 48 packages. However the variation in solving times increases dramatically for 32 packages onwards.

4.5 Scalability of MULTIOBJ, minimising waiting time

Since MULTIOBJ is the only model capable of minimising waiting time, we consider the scalability of this model with the best subtour and symmetry elimination configurations identified earlier. Package counts from 8 to 30 were considered, and 20 random scenarios were generated for each count. The results are shown in Figure (8) depicting the average solving time, along with ± 1 standard deviation to given some indication of the variability.

We observe that the performance degrades severely as the number of packages increases beyond 16. From around 24 packages onwards, the time limit is consistently hit. However, in all instances considered, a feasible solution had been attained by this point.

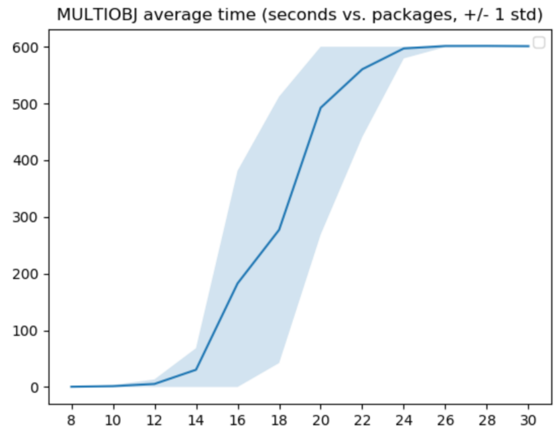

Figure 8: Performance of MULTIOBJ as packages increased

This suggests that the exact model implemented is not scalable to large problems, and once again, a significant amount of variation in the solving times is observed.

5 Summary

The emergence of autonomous delivery drone technology may revolutionize practices in logistics, emergency relief and other areas. This creates challenges in the generation of optimal routing and maintenance plans.

In this analysis, we modeled a simplified version of the routing problem with a view to minimising total distance traveled, or minimising the total waiting time. Three models were considered in total. For our most complex model, we identified a novel adaptation of the Dantzig approach to subtour elimination which took advantage of other constraints in the problem to form the most efficient method. We improved the performance further through the selective application of symmetry elimination constraints.

For the purpose of minimising total distance traveled, our simplest model, representing a single drone, was by far the most efficient and was able to find optimal solutions up to around 48 packages within the 10 minute time limit. For the purposes of minimising total waiting time, a more complex model was required, which scaled poorly beyond 16 packages, and was unable to solve instances above 24 packages to optimality within the time limit.

In most non-trivial cases we found significant variation in the solving times for randomly generated scenarios. This suggests that minor variations in the random selection of package weights and locations can have a significant effect on the difficulty of the problem. In a real world practical context, a time limit or optimality gap may be necessitated - this may make the MILP approach unreliable for delivering consistent results in a wide range of scenarios.

However, for large scale problems, eg. $P > 50$ packages and larger, solving to optimality using standard MILP approaches is likely to be impractical. We conclude that to apply MILP successfully to the problem, it may be necessary to consider decomposition techniques: for example, it may be possible to apply column generation if individual columns represent individual trips made. Even so, realistically, it is probably inevitable that larger problems would require some heuristic approach to fall back on. Considering that commercial operations in general desire improvement in efficiency, rather than theoretical notions of global optimality, this seems like a sensible next step.

References

[1] Amazon Prime Air. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011, 2018. [Online; accessed 25-May-2018].

[2] DHL Parcelcopter 3.0. http://www.dpdhl.com/en/media_relations/specials/parcelcopter.html, 2018. [Online; accessed 25-May-2018].

[3] Spartan HPC-Cloud Hybrid: Delivering Performance and Flexibility. https://doi.org/10.4225/49/58ead90dceaaa, 2018. [Online; accessed 25-May-2018].

[4] The Dream of Drone Delivery Just Became Much More Real. https://www.popularmechanics.com/technology/infrastructure/a15061374/drone-delivery-bell-boeing/, 2018. [Online; accessed 25-May-2018].

[5] D. F. D. Cattaruzza, N. Absi and T. Vidal. A memetic algorithm for the multi trip vehicle routing problem. *Eur. J. Oper. Res.*, 236(3):833–848, 2014.

[6] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, Jan 2017.

[7] M. G. N. Azi and J. Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Comput. operations Res.*, 41:167–173, 2014.