Delft University of Technology

Operations Optimisation
AE1444

Verification and validation of a vehicle routing problem for drone delivery

Authors:

Guillermo Gonzalez (5036569)
Shawn Schröter (1234567)
Riccardo Torelli (5150256)

Group 16
January 10, 2020

Delft
University of
Technology

Contents

1 Introduction

In the field of Operations Research, a very well-known problem is the Vehicle Routing Problem, commonly abbreviated as VRP. Currently the name denotes a category of problems which involve a fleet of vehicles and a set of destinations [3]. A specific route is assigned to each vehicle depending on what particular parameter wants to be optimised: costs, energy, time, distance, etc.

Simple sets of VRP can be solved through combinatorial mathematics. However, as the problem gets more complex, more robust methods are required. In most of the cases, a VRP can be solved by using a MILP model, or through metaheuristics (simulated annealing) [1]. The most common VRP models take into account six basic constraints:

1. Each costumer is visited only once.
2. Each vehicle arrives at and leaves from the depot only once.
3. If a vehicle arrives to a customer, it must also leave.
4. Demand of all visited customers is less than the vehicle capacity.
5. Time at a node is equal or larger than time at the previous node.
6. Time of deliveries is within limits.

In this report, the VRP drone delivery proposed by [Kevin Doring et al.] is thoroughly analyzed[2]. A particular approach discussed in this MILP model is the inclusion of reusability (using a drone for more than one route) and energy cost in terms of battery specifications. As it will be seen, the reusability concept can produce a significant improvement in cost optimisation, while the battery specifications will be shown to give a deeper insight in capacity and distance constraints.

After the description of the model (Ch.2), it is tested and validated through a simple data set to understand the logic behind its algorithm (Ch.3). Then, the limitations of the model are defined by means of a sensitivity analysis which involves testing the model through different scenarios, and changing the values of the parameters around the boundaries of the feasibility region (Ch.4). Also, this final analysis will help to identify which parameters play a major role in the performance of the model, and thus which solutions can be expected from it.

2

Compiling the VRP

2.1. The mathematical problem

According to the article [Keving Doring et al.] the vehicle routing problem can be solved through a set of 23 different constraints, which can be classified in seven different groups. Each of them are described below [2].

1. **Every route is valid**

This group guarantees the most essential condition of the VRP: every destination *j* is reached only from one origin *i* and vice versa.

$$\sum_{\substack{j \in \mathcal{N} \\ i \neq j}} x_{ij} = 1 \qquad \forall i \in \mathcal{N}_0 \tag{2.1}$$

$$\sum_{\substack{j \in \mathcal{N} \\ i \neq j}} x_{ji} = 1 \qquad \forall i \in \mathcal{N}_0 \tag{2.2}$$

$$\sum_{\substack{j \in \mathcal{N} \\ i \neq j}} x_{ij} - \sum_{\substack{j \in \mathcal{N} \\ i \neq j}} x_{ji} = 0 \qquad \forall i \in \mathcal{N} \tag{2.3}$$

2. **Reusability constraints**

Reusability allows the same drone to cover more than one route. The slack variables of these constraints give valuable insight to understand the number of separated routes (see verification scenarios, Ch.3). The reuse from *i* to *j* is indicated through the binary variable sigma. Constraint 2.6 limits the number of reuses according to the total routes and the available drones M.
Reusability is a peculiar element for this kind of VRP.

$$\sum_{j \in \mathcal{N}_0} \sigma_{ij} \leq x_{i0} \qquad \forall i \in \mathcal{N}_0 \tag{2.4}$$

$$\sum_{j \in \mathcal{N}_0} \sigma_{ji} \leq x_{0i} \qquad \forall i \in \mathcal{N}_0 \tag{2.5}$$

$$\sum_{i \in \mathcal{N}_0} x_{0i} - \sum_{\substack{(i,j) \in \mathcal{N}_0 \times \mathcal{N}_0 \\ i \neq j}} \sigma_{ij} \leq M \tag{2.6}$$

3.Demand constraints

This group involves just two constraints defining the payload *y* a drone carries from *i* to *j* in such a way that it can satisfy the demand D of the destination *j*, as indicated in constraint 2.7. The other constraint 2.8 sets the demands of unactive links to 0kg by using a big constant value K, which is used also in several other constraints.

$$\sum_{\substack{j\in\mathcal{N}\\ i\neq j}} y_{ji} - \sum_{\substack{j\in\mathcal{N}\\ i\neq j}} y_{ij} = D_i \qquad \forall i \in \mathcal{N}_0 \tag{2.7}$$

$$y_{ij} \neq K x_{ij} \qquad \forall_{\substack{(i,j)\in\mathcal{N}\times\mathcal{N}\\ i\neq j}} \tag{2.8}$$

4. Time constraint

This set of constraints limits and stores in t_i at what time is an origin *i* leaved or a location *j* reached. An extra decision variable *a* keeps the time it takes to go back to the depot from *i*. By taking into account the distances *d*, the time of descend tau and the velocity of the drone *v*, constraints 2.9 2.10 and 2.11 defined the values for variables *a* and *t* in seconds. Finally the time of last delivery is registered in *l* and is kept below the Time limit T by means of constraint 2.12 and 2.13.
Notice that equation 2.11 is triggered by reusability and registers the time for previous route (before being reused).

$$t_i - t_j + \tau + d_{ij}/v \leq K(1 - x_{ij}) \qquad \forall_{\substack{(i,j)\in\mathcal{N}\times\mathcal{N}_0\\ i\neq j}} \tag{2.9}$$

$$t_i - a_i + \tau + d_{i0}/v \leq K(1 - x_{io}) \qquad \forall i \in \mathcal{N}_0 \tag{2.10}$$

$$a_i - t_j + \tau + d_{0j}/v \leq K(1 - \sigma_{ij}) \qquad \forall_{\substack{(i,j)\in\mathcal{N}_0\times\mathcal{N}_0\\ i\neq j}} \tag{2.11}$$

$$t_i \leq l \qquad \forall i \in \mathcal{N}_0 \tag{2.12}$$

$$l \leq T. \tag{2.13}$$

5. Carrying capacity constraints

This set of constraints introduces three more decision variables. First we have *q* which indicates the total weight a drone carries from origin *i* to location *j* in kg. The second decision variable ζ helps constraining the weight by registering the battery weight at each location *i*. A third variable, z, imposes the amount of energy a drone needs to get back to the depot from a destination *i*. z will then appear in constraints 2.20 and 2.21 to register the actual energy requirements that are been imposed here. Notice that in order to set these three variables these constraints include the total drone capacity Q and the energy density of the battery ξ in KJ/kg.

$$q_{ij} + y_{ij} \leq Q x_{ij} \qquad \forall_{\substack{(i,j)\in\mathcal{N}\times\mathcal{N}\\ i\neq j}} \tag{2.14}$$

$$z_i/\xi - \zeta_i \leq K(1 - x_{i0}) \qquad \forall i \in \mathcal{N}_0 \tag{2.15}$$

$$\zeta_i - \zeta_j \leq K(1 - x_{ji}) \qquad \forall_{\substack{(i,j)\in\mathcal{N}_0\times\mathcal{N}_0\\ i\neq j}} \tag{2.16}$$

$$q_{ij} \geq \zeta_j - K(1 - x_{ij}) \qquad \forall_{\substack{(i,j)\in\mathcal{N}\times\mathcal{N}_0\\ i\neq j}} \tag{2.17}$$

$$q_{i0} \geq \zeta_i - K(1 - x_{i0}) \qquad \forall i \in \mathcal{N}_0 \tag{2.18}$$

6. Energy restriction constraints

This final group of constraints introduces a new variable *f* to indicate the energy consumed when going from one place to another. In the constraint 2.19 there is a term *p(m)* which represents a linear approximation for the power, according to the load of the drone $m = q + y$. The load of each drone is entirely used for the cargo that has to be delivered and for the necessary battery. The optimal balance between the two elements is found as part of the solution of the Linear Problem. Constraint 2.20 forces the value of *z* to be equal to the total energy consumed in the whole route, see explaination in previous group of constraints.

$$f_i - f_j + p(m_{ij})(d_{ij}/v + \tau) \leq K(1 - x_{i0}) \qquad \forall_{\substack{(i,j)\in\mathcal{N}\times\mathcal{N}_0 \\ i\leq j}} \tag{2.19}$$

$$f_i - z_i + p(m_{i0})(d_{i0}/v + \tau) \neq K(1 - x_{i0}) \qquad \forall i \in \mathcal{N}_0 \tag{2.20}$$

$$z_i \leq K x_{i0} \qquad \forall i \in \mathcal{N}_0. \tag{2.21}$$

7. Cost constraints

The cost comes from two main sources. First the total cost of the drones used is obtained by subtracting the number of routes minus the amount of reuses. The second one comes from the total energy consumed for each route. The prices for each drone are set with parameter *F* and the cost of energy, represented by ϵ. This definition of the total cost is given by constraint 2.22 and limited to a certain budget through 2.23.

$$c = F \sum_{i\in\mathcal{N}_o} x_{0i} - F \sum_{\substack{\forall(i,j)\in\mathcal{N}\times\mathcal{N}_0 \\ i\neq j}} \sigma_{ij} + \epsilon \sum_{i\in\mathcal{N}_o} z_i \tag{2.22}$$

$$c \leq B. \tag{2.23}$$

8. Objective functions

From the several constraints defined, there are two different approaches for the model to be optimized: costs minimization, by using the definition from equation 2.22; or time minimization, through the definition of equations 2.12 and 2.13. These objective functions can be simply stated by the expressions shown in 2.24 and 2.25 respectively.

$$min \quad c \tag{2.24}$$

$$min \quad l \tag{2.25}$$

2.2. Methodology

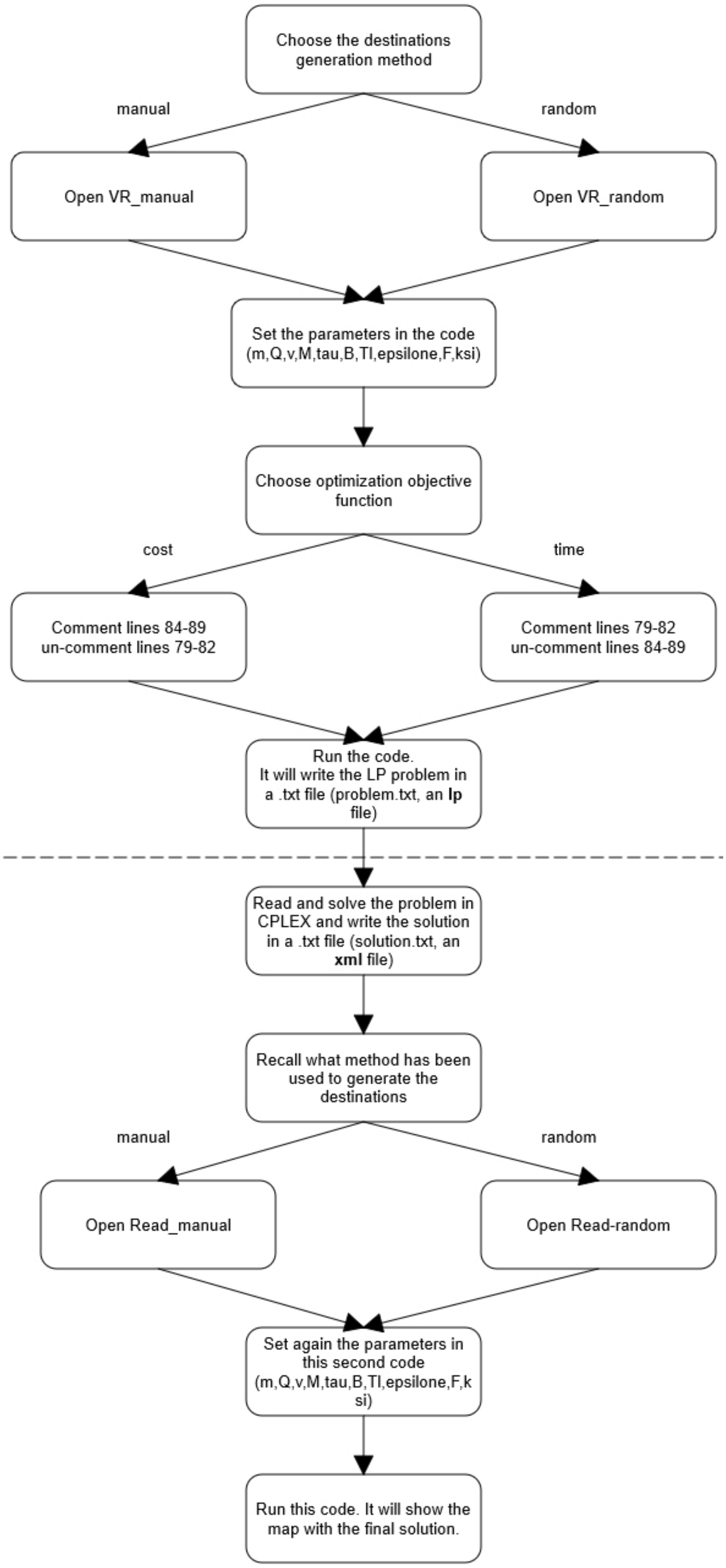

Figure 2.1: Flow chart explaining the integration of CPlex and Pyhton codes to generate a solution

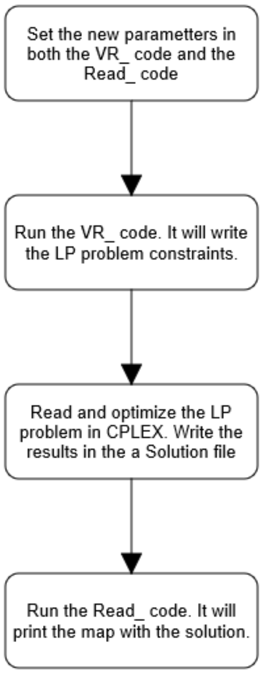

Figure 2.2: Flow chart explaining the integration of CPlex and Pyhton codes to generate a solution, when the kind of process has already been selected

In order to program the model, a Python code was designed so that it writes a text file containing the 23 constraints in such a way that CPlex can read it as a MILP model. Once CPlex creates the solution file, a second Python program reads it and plots the different routes on a graph.

Different versions of the Python codes were made in such a way that there are two ways of setting the data (by generating random destinations or by manually entering them through an Excel file), and two ways of optimizing the model (for cost or for time). Thus, in our implementation there are four different ways of obtaining a solution for the model.
The first flow chart, figure 2.1, gives a summarized insight into how the Python codes and Cplex interact to provide the final solution.
The second, smaller flow chart 2.2, is a scheme for when the kind of process (manual vs random destinations, cost vs time optimization) have already been selected.

3 Results

3.1. Verification of the model

We now want to explain how the LP problem can be used to optimize different cases of drone delivery and exploration. As will be highlighted in the next chapter (sensitivity analysis), the required time to solve optimization problems with 10 or more destinations is huge and above the possibilities of typical PC's.

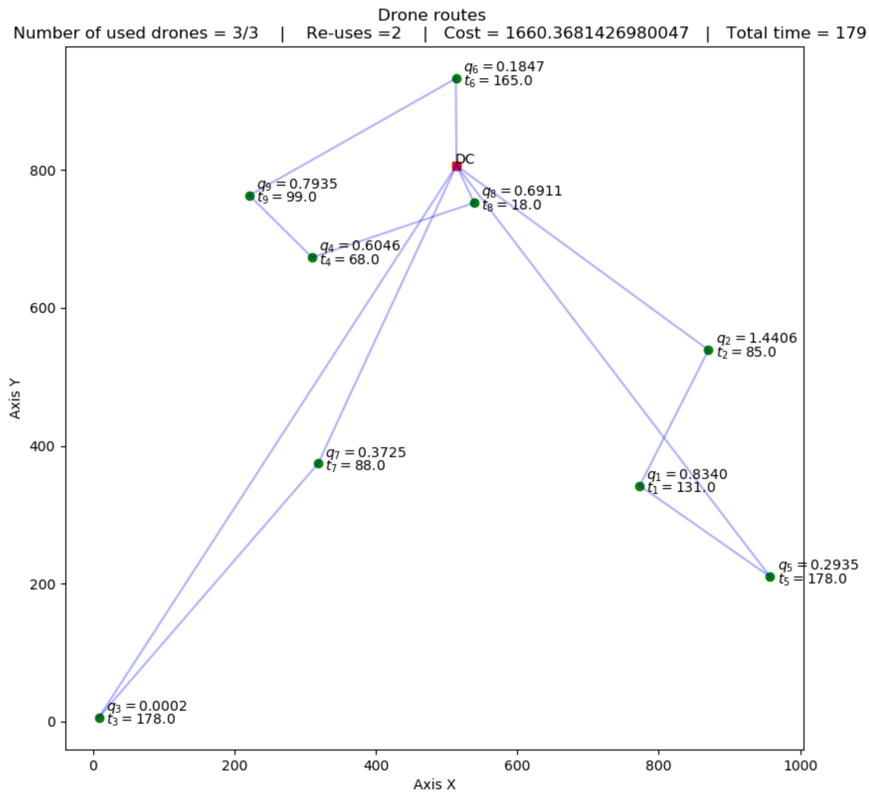

Figure 3.1: Example of a solution requiring high computational power.

Figure3.1 shows the solution obtained from the algorithm described in the previous chapter (section 2.2)) For each optimization procedure there is several data displayed in the images. As expressed

in the two axis labels, Y and X, the image shows a map with the coordinates of the destinations, the green spots. The distances, not showed in the image for simplicity but registered in the python code, are in meters.
For each destination there are two values: q_n gives the cargo requested by the single destination *n*, in kilograms; t is the time of the delivery after the drone has descended on the single location *n*, in seconds.
DC, the red square, is the Distribution Centre, the depot from which the drones depart and go back at the end of their trip or if they are re-used.
In “Number of used drones” it can be read how many drones have been used in the final solution out of the available drones (used/available).
The “Re-uses” counter says how many times a drone has been re-used. In the current status, the program cannot distinguish the routes taken by different drones, and they are all showed in the same colour (light blue). In this case, it is easy to understand that the three routes from figure 3.1 (0-4-9-6-0, 0-2-5-1-0, 0-7-3-0) are made by three different drones, since none of them has been re-used (see the ’Re-uses’ counter). The order of the locations can be deduced by checking the time for every delivery.
“Cost” and “Total time” show respectively the cost for the entire delivery process and the time of last delivery (so the time for the last drone to go back to the DC is not considered). In general, our algorithm can only optimize for one thing at a time, namely for cost or for time.
As it will be noted in the next examples, it is often the case that when optimizing for time all of the drones are used, while cost optimization always pushes the algorithm to process the last delivery at the last possible time (since no limits are imposed on the delivery other than the time limit defined in equation 2.13). Since the time limit in all tested scenarios is 2000, it is evident that in Figure 3.1, the optimization was done for time. Also, since the cost for a single drone is set to 500, we note that the total cost here is around the cost for 3 drones (1500) plus 160.39 due to energy expenses.
Energy expanses are calculated on the base of the linear approximation given in the constraints 2.19-2.21. The smaller the distance to cover, the smaller the battery and the lower the energy costs. Similarly, the smaller the cargo for the route, the lower the energy costs.

3.2. Validation of the model

Some elementary cases, almost solvable by hand, are shown in the following part of the report and the main characteristics of the algorithm are emphasized.

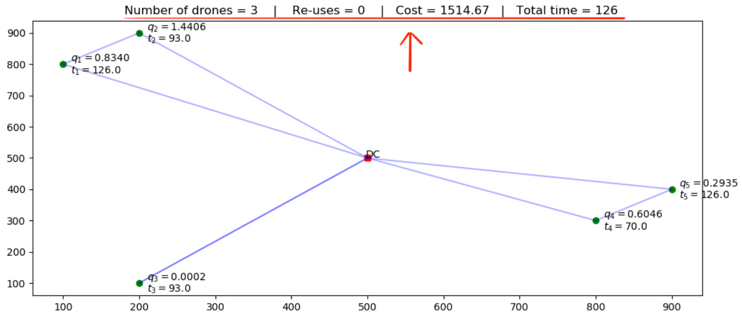

Figure 3.2: Time optimization, no re-usability allowed, scenario 1

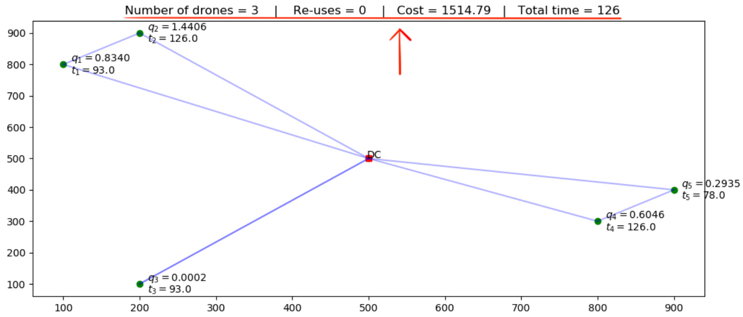

Figure 3.3: Time optimization, re-usability allowed, scenario 1

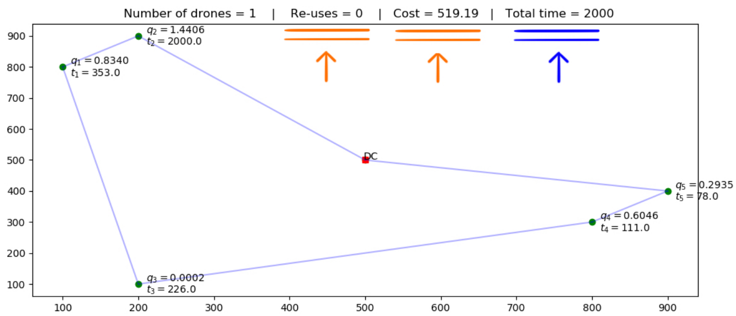

Figure 3.4: Cost optimization, no re-usability allowed, scenario 1

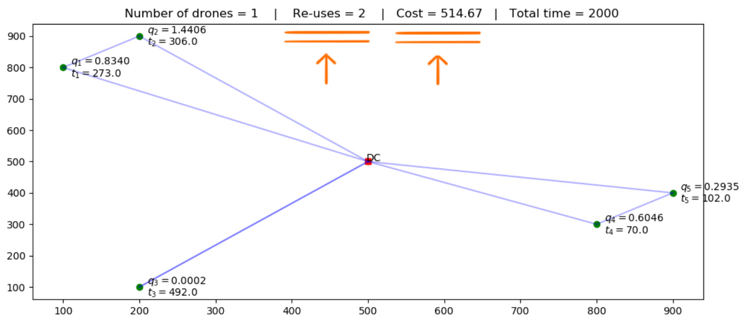

Figure 3.5: Cost optimization, re-usability allowed, scenario 1

First, the scenario involving five destinations and three available drones is solved, , in time optimization with no re-uses allowed (Figure 3.2), time optimization without limits on re-uses (Figure 3.3), cost optimization with no re-uses allowed (Figure 3.4)and cost optimization without limits on re-uses (Figure 3.5). Notice that reusability has no influence when working in time optimisation as three drones are used. Instead, reusability has a stronger influence when optimizing for cost. While in the no-reuse case the problem is optimal with just one drone going through all of the destinations, in the re-use scenario it is still optimal to use one single drone, but by splitting the delivery process in three routes. Despite being a slightly longer route, it allows the drone to carry a smaller cargo in each route, thus consuming less energy and using a smaller battery. This can be seen in the two difference of costs given in figures 3.4 and 3.5.

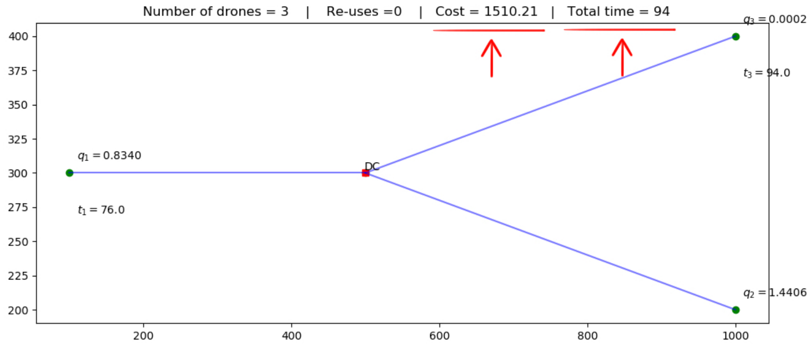

Figure 3.6: Time optimization, no re-usability allowed, scenario 2

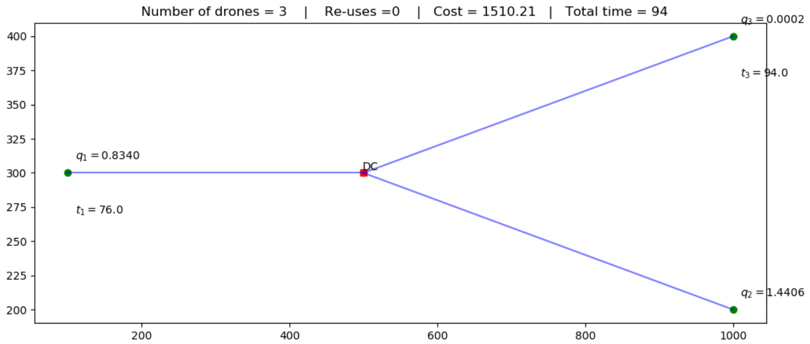

Figure 3.7: Time optimizaton, re-usability allowed, scenario 2

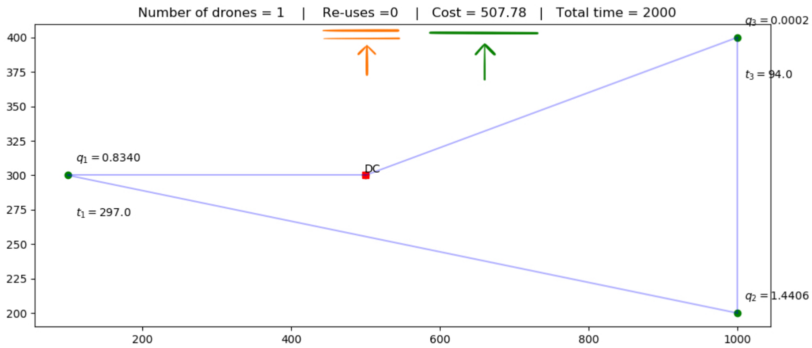

Figure 3.8: Cost optimizaton, no re-usability allowed, scenario 2

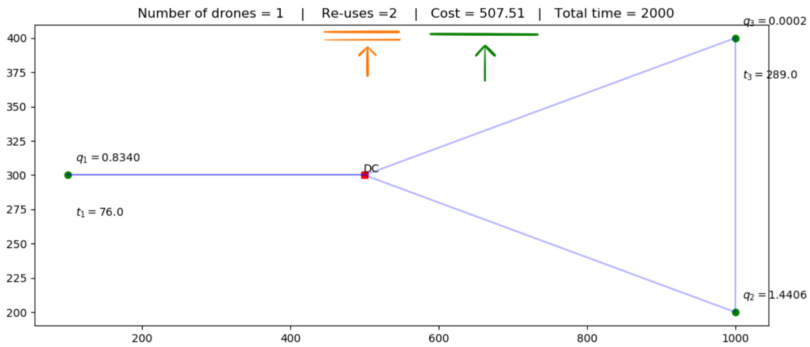

Figure 3.9: Cost optimizaton, re-usability allowed, scenario 2

In this second scenario, where the number and location of destinations have been changed, there is a similar behaviour for time optimization (Figure 3.6 and Figure 3.7): again, reusability plays no role. Although compared to first scenarios (Figure 3.2 and Figure 3.3), total time is now smaller due to smaller distances to cover. Similarly, in Figure 3.8 and Figure 3.9, cost optimization problems, we notice the same behaviour of the algorithm compared to scenario 1. Notice how, when allowed to re-use the drones (Figure 3.9), the solution is more convenient. The slight difference in cost (≈0.27) between Figure 3.8 and figure 3.9 is to be attributed not only to the shorter distance covered now by the drone, but also to the possibility, for the drone, to carry less cargo in each route by splitting the delivery in 2 separated routes, so that the required energy is less and consequently also the global cost.

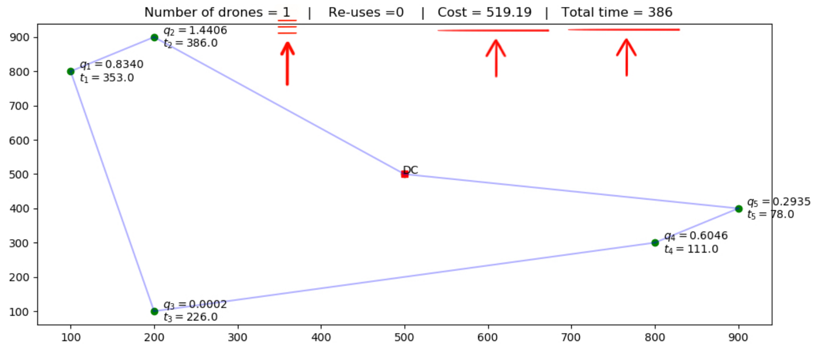

Figure 3.10: Time optimizaton with one drone, no re-usability allowed, scenario 1

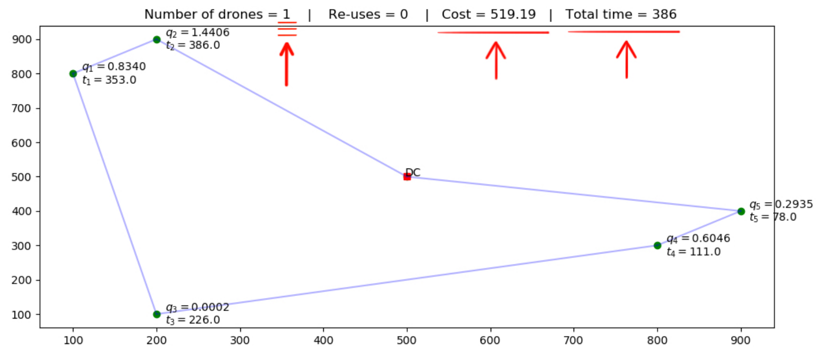

Figure 3.11: Time optimizaton with one drone, re-usability allowed, scenario 1

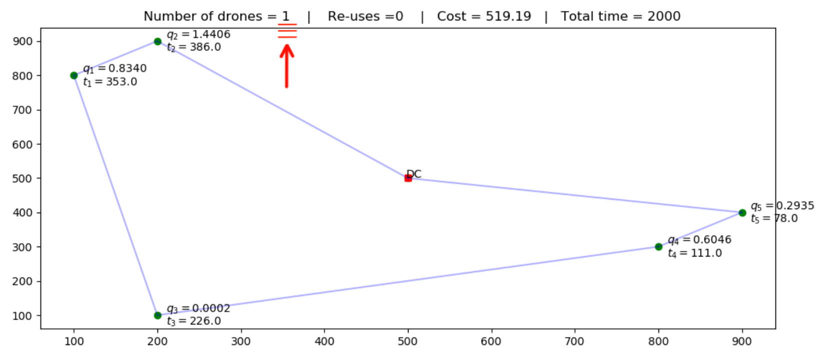

Figure 3.12: Cost optimizaton with one drone, no re-usability allowed, scenario 1

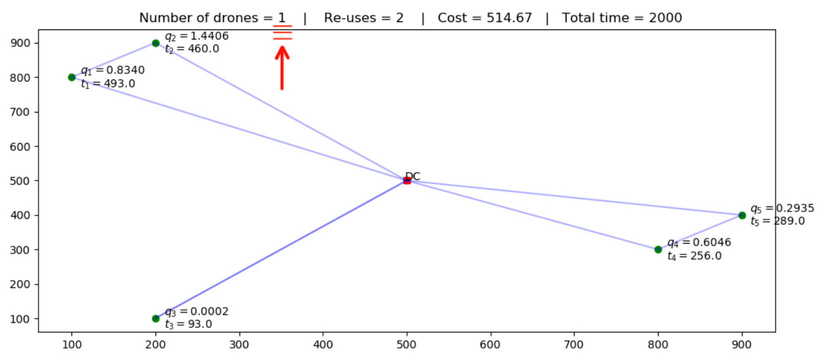

Figure 3.13: Cost optimizaton with one drone, re-usability allowed, scenario 1

The third scenario keeps the same destinations from the first one (figures 3.2 to 3.5), but now with only one drone available. Both simulations for time with and without reusability are equal. However in Figure 3.10 it is forbidden to go back to the depot DC, so that the single drone has to do all of the destinations in one run instead of the multiple routes of figure 3.2. On the contrary, in figure 3.11 the drone is actually allowed to go to the DC, but total time for this procedure is more than when doing a single route, so it is discarded by the algorithm. On the other side, cost optimisation (figure 3.13) does opt for splitting the deliveries so the drone can carry less cargo for each route. Then the required energy is reduced together with the total cost. Proof of this is that the total cost in Figure 3.11, time optimization with re-use (not used), is 519.19, while total cost in Figure 3.13, cost optimization with re-use (used), is 514.67.

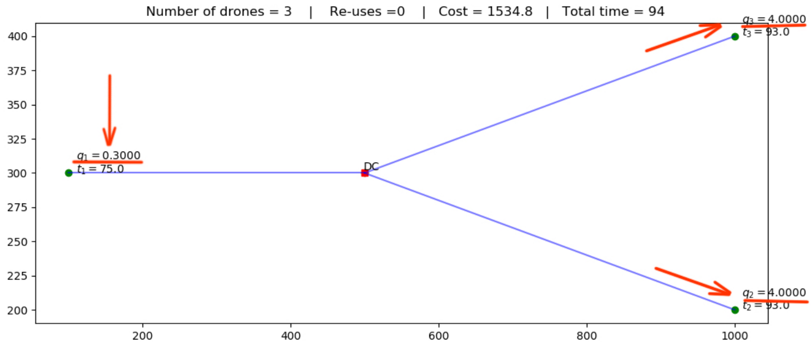

Figure 3.14: Time optimizaton with one drone, no re-usability allowed, scenario 2

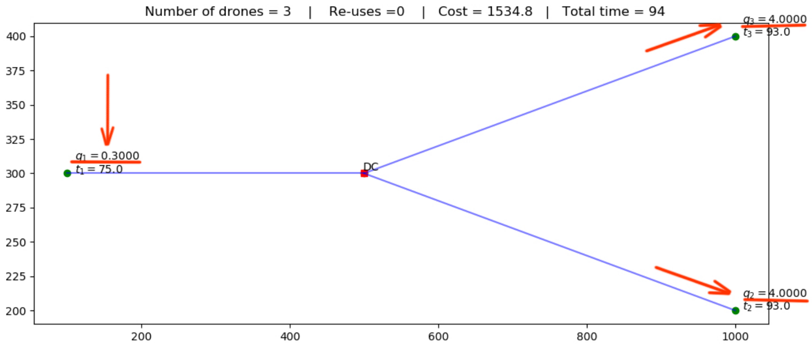

Figure 3.15: Time optimizaton with one drone, re-usability allowed, scenario 2

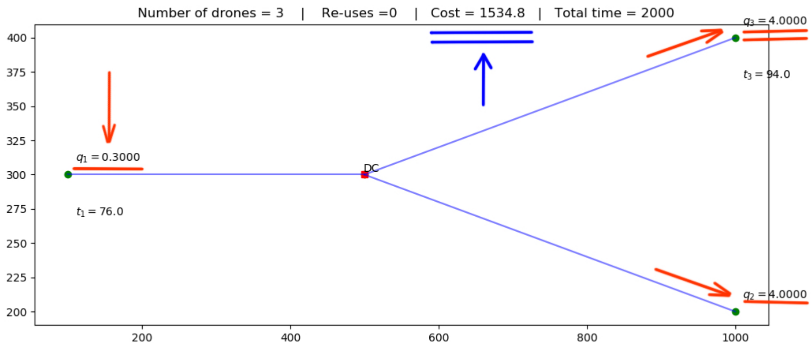

Figure 3.16: Cost optimizaton with one drone, no re-usability allowed, scenario 2

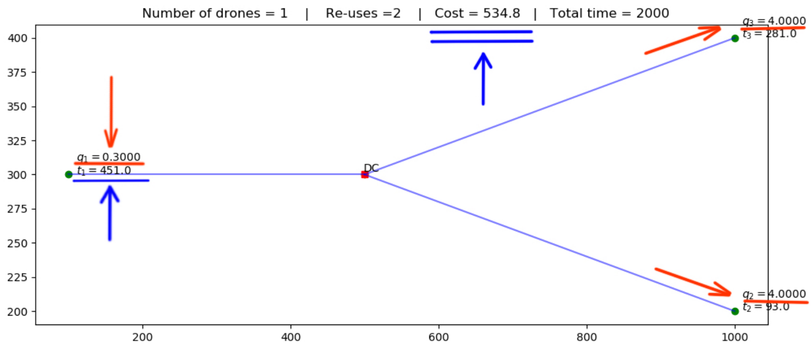

Figure 3.17: Cost optimizaton with one drone, re-usability allowed, scenario 2

What happens in the second scenario shown in figure 3.11 (no re-uses even when allowed) is not always possible. The fourth scenario shows what happens when changing the demand. Destination 1 now asks for a really small delivery, while destination 2 and 3 ask for 4 kilogram each. The maximum capacity for the drones is here set to 5 kg. This results in a time optimization (Figure 3.14) and Figure 3.15) that it is just the same as before, since the optimal procedure is still to distribute the deliveries among the three drones, without wasting time in re-use. While, cost optimization is evidently affected by the new cargo requests. It is not possible anymore to do all the deliveries with one single route, since it could not carry $4 + 4 + 0.3 = 8.3$ kg plus a battery that would have to be big enough for a single, long route. When re-use is forbidden (Figure 3.16), cost goes up to 1534.8 against 534.8 of the re-use case (Figure 3.17). The cost in Figure 3.17 is still higher than for the case of lower demand (Figure 3.9), due to both longer distance to cover (going back each time to the DC) and to the greater mass that requires more energy for the drones.

The LP files for the constraints and for the solution of the first case above (Figure 3.2) are in the Appendix B.

4

Sensitivity analysis

4.1. Computation times

Before performing a sensitivity analysis for each of the parameters, the model was tested by modifying the number of drones and destinations to see how the computation time changed. The following tables show the different computations times when optimizing for both cost and time.

	Destinations				
Drones available	2	4	**6**	7	8
1	0.06	0.09	4.38	56.72	>1000
2	0.08	0.11	4.00	62.48	>1000
3	0.11	0.27	5.73	66.53	>1000
4	0.13	0.42	5.03	57.16	>1000
5	0.19	0.37	5.86	56.63	>1000
10	0.22	0.36	5.20	59.75	>1000

Table 4.1: Computation time in seconds when optimizing for cost

	Destinations						
Drones available	2	4	6	7	8	9	10
1	0.06	0.14	0.68	6.97	159.94	>1000	>1000
2	0.05	0.19	0.22	0.59	0.86	8.77	432
3	0.03	0.11	0.20	0.38	0.25	1.59	4.3
4	0.03	0.05	0.13	0.3	0.14	0.19	0.26
5	0.02	0.13	0.11	0.16	0.11	0.16	0.25
10	0.06	0.09	0.08	0.11	0.06	0.06	0.06

Table 4.2: Computation time in seconds when optimizing for time

It's interesting to observe how the computation time increases differently for each of the two cases. As seen in table 4.1, the time increases dramatically when going from 7 to 8 destinations. This is due to the way the cost objective function shown in equation 2.23 works. The function will always tend to use reusability and thus the model will try to get a solution by analyzing all the possible routes using only one drone, regardless of the number of drones available. Therefore, the number of destinations is the one which really affects the computation time while the number of drones has almost no influence.

Now, when analyzing the values from table 4.2 the lowest computation times are achieved when the number of drones increases, because the model simply assigns each drone to one destination. On the other side, when the number of drone decreases and the destinations increase, the algorithm has to evaluate more possible routes causing the computation time to increase, growing exponentially when leaving just one drone with more than 8 destinations, since all possible routes must be checked and combined in order to find a feasible one.

4.2. Feasible ranges for parameters

From the computation times obtained in the previous section, an intermediate case was chosen to check the limits of the parameters involved in the model. Tables 4.3 and 4.4 show two values for each limit: the ones labeled with a letter *C* (for change) indicate the limit when a different solution is obtained, while the ones labeled with an *I* indicate when the value makes the problem infeasible.
Because some of the variables have a direct and obvious impact in the cost and/or time objective function, e.g. the cost of drone in cost optimization, a change in the solution was considered not when the final cost or time change, but, more interestingly, when the drone routes change. So it is important to keep this in mind when analyzing the values labeled with a *C*. Some remarks about the effects in time or cost are made for each variable along with some observations on their sensitivity in changing the solution.
Some other values are indicated with an *n/a*, which means that the value doesn't apply since increasing or decreasing that value will never cause the solution to become infeasible or to change.

Range of parameters for cost optimization

Variable	current	min	max	observations
M: available drones	3	C:1 I:1	C:∞ I:n/a	The amount of drones does not change the value of the final cost
Q: drone capacity [kg]	5	C:1.75 I:1.5	C:∞ I:n/a	Solution does not change, because heavy drone means more cost and the solver won't use the extra capacity. Capacity changes total cost when the path is changed due to cargo limits reached.
N: number of destinations	6	6	6	Changing number of destinations will automatically change the solution because a new route must be derived.
V: velocity [m/s]	6	C:2.9 I:2.9	C:58 I:n/a	Increasing velocity reduces the final cost. Setting it too low makes solution infeasible due to time constraints.
A: area within destinations [km^2]	1	C:0 I:0	C:17.43 I:17.43	Cost increases dramatically as area increases. Depending on velocity and time constraints, the area can make the solution infeasible.
ξ: Battery energy density [KJ/kg]	650	C:218 I:63	C:∞ I:n/a	Cost increases proportionally to energy density, and area has also an influence on it. Too low energy density makes feasibility dependant on drone capacity.
F: Cost of Drone [$]	500	C:$-\infty$ I:n/a	C:980 I:980	Upper limit is defined by budget: B. No lower limits apply since the objective is minimize costs.
τ: time of descend[s]	10	C:$-\infty$ I:n/a	C:193 I:623	Upper limit defined by time limit. When exceeded a second drone is used. τ does not change total cost if the path does not change.
ϵ: cost of energy[$]	0.1	C:0 I:n/a	C:7.8 I:7.8	When reaching lower limit, optimization is done with time limits instead. Upper limit bounded by budget causes infeasibility.
q: demand [kg]	0-2	C:0-0.96 I:n/a	C:0-4.13 I:0-6.8	Value is given as a range since demands are generated randomly. Sensitive because it changes the cost even if the paths are the same.
T: Time limit [s]	2000	C:557 I:557	C:∞ I:n/a	Time limit does not change total cost if the path does not change.
B: budget [$]	2000	C:519 I:519	C:∞ I:n/a	Lower limits depends on the cost of a drone, because at least one drone has to be bought. There is no influence on final cost.

Table 4.3: Range of parameter for cost optimization

Range of parameters for time optimization

Variable	current	min	max	observations
M: available drones	3	C:1 I:1	C:∞ I:n/a	Time decreases with more drones as long as they are less or equal to number of destinations.
Q: drone capacity [kg]	5	C:1.75 I:1.5	C:∞ I:n/a	Same as in cost optimization.
N: number of destinations	6	6	6	Same as in cost optimization.
V: velocity [m/s]	6	C:0.5 I:0.5	C:∞ I:∞	Too high values make the algorithm unstable. Very sensitive because velocity has a direct influence on delivery time.
A: area within destinations [km^2]	1	C:0 I:0	C:94.86 I:94.86	More distance than in cost optimization because always uses three drones. However time changes as much as distance does.
ξ: Battery energy density [KJ/kg]	650	C:63 I:63	C∞ I:n/a	Time is not influenced by this, so it directly goes infeasible without changing to a different solution.
F: Cost of Drone [$]	500	C:$-\infty$ I:n/a	C:661 I:661	It does not produce any change in the time limit, but it is bounded by the budget.
τ: time of descend[s]	10	C:$-\infty$ I:n/a	C:7.8 I:7.8	Sensitive as it changes the final value for time. Time limit defines value for τ. Changing limit is higher than for cost optimization, because the VRP already uses 3 drones from the beginning.
ϵ: cost of energy[$]	0.1	C:0 I:n/a	C:7.8 I:7.8	Upper limit causes infeasibility, due to budget limit. Same as in cost opt. Final time does not change.
q: demand [kg]	0-2	C:0 I:0	C:0-5.9 I:0-6.8	Value is given as a range since demands are generated randomly. No lower limit, since time-optimal route is not defined by *0 to 2* demand. Change is different from cost because the VRP is already using 3 drone here. Final time does not change when route does not.
T: Time limit [s]	2000	C:146 I:146	C:∞ I:n/a	Changing time limit will not change the solution but will define when the solution becomes infeasible
B: budget [$]	2000	C:1521 I:1521	C:∞ I:n/a	Lower limit prevents buying third drone. Only when limit is crossed final time changes.

Table 4.4: Range of parameter for time optimization

4.3. Different scenario analysis

After identifying the variables that have a stronger influence from the previous section, a scenario in which such variables change were made to see how they relate to each other.

For the first scenario, it is studied how to increase the distance a drone can cover. From the perspective of cost optimisation, it is necessary to define distances in terms of energy. As energy density is a parameter of the model, time can be defined in terms of the mass carried by the drone, thus deriving equation 4.1, which also accounts for the time of descend. Furthermore, using 4.1 and knowing the velocity of the drone, we get 4.2, which shows how the maximal reachable distance can be described in terms of capacity and energy density. Mind that the whole value is halved because the drone has to cover the distance to go and then come back.

$$t = \frac{(Q - D) \times \xi}{P} - \tau \tag{4.1}$$

$$d = \frac{v}{2}\left(\frac{(Q - D) \times \xi}{P} - \tau\right) \tag{4.2}$$

When plotting 4.2, the behavior of distance in function of battery density and drone capacity can be easily observed, as depicted in figure 4.1. Consequently, this scenario considers the simplest possible route: one drone going to one destination.

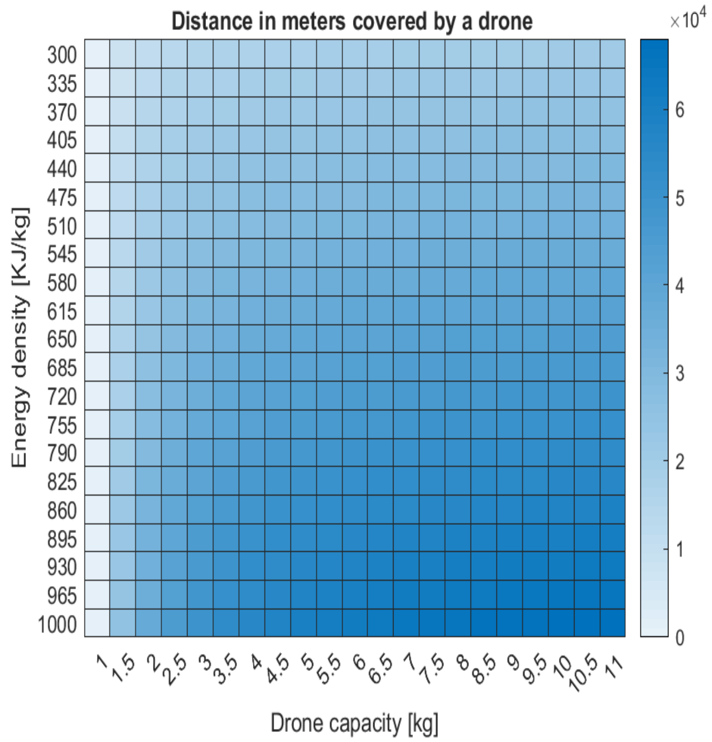

Figure 4.1: Distance as a function of battery energy density and drone capacity

The heatmap exhibits a linear fashion, so distance increases as both capacity and energy density increase. It's important to point out that, when capacity is at its lowest, the drone can't cover any distance because it cannot even carry the battery. In general, capacity has a stronger influence than energy density. Which makes sense because it allows for a heavier battery, giving a greater range of feasible values for energy density and thus a greater distance.

Secondly, time optimisation is applied to a new scenario. In particular, it focuses on the time at which the last delivery is done. As inferred from the previous parameter analysis, the time is mostly influenced by the velocity and the number of available drones. Figure 4.2 summarizes the behavior as a function of these elements:

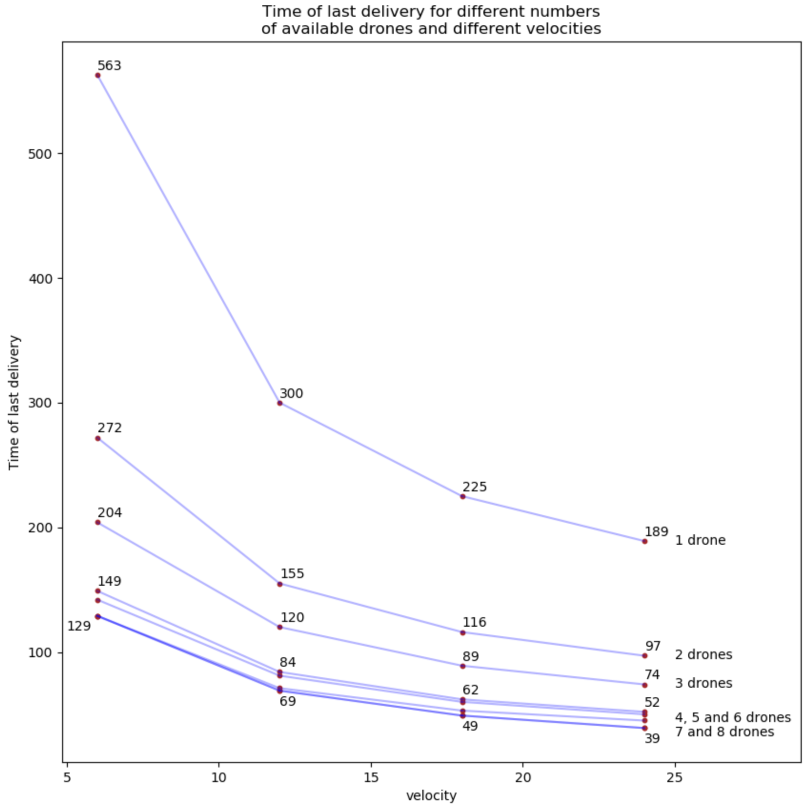

Figure 4.2: Time as a function of velocity and available drones

Intuitively, time improves as both velocity and available drones increase. An interesting observation is that the greatest time improvement occurs when velocity goes beyond 10 m/s, regardless of the number of drones. The point in which this maximum improvement occurs will vary depending on the distance to the destinations. Another remark deals with the improvement produced by number of drones. The maximum change occurs when using 2 drones because all delivery times are practically halved. Using more than 3 drones doesn't produce any significant improvements. This is related with the number of destinations, so if more destinations were been used, a more notorious improvement would be observed when using more than 3 drones.

About this, an apparently weird behavior becomes evident when using many drones. At lower velocities, when going from 7 to 8 drones, for instance, there is seemingly no difference. This in fact correct. This is evident when comparing the solution for 7 drones (Figure 4.3) and the one for 8 drones (Figure 4.4). The additional drone is used for the only new possible optimization, that is splitting the delivery to destinations 6 and 7 into two different routes, thus obtaining a reduced time for the two deliveries (24s and 38s with 7 drones, 24s and 28s with 8 drones). Despite this little change, though, the time of last delivery is not determined by this constraint, namely by the delivery to 6 and 7, but rather by the delivery to destination 4. That is reached at 39s when using both 7 and 8 drones. The only way to further optimize the solution is, then, by raising the air speed of the drones, as shown in the graph. The addition of the eighth drone does not further optimize the problem, since it cannot act on the constraint related to destination 4.

For the same reasons, adding a ninth drone would change nothing in the solution: there is not even a way for the solver to use the additional drone.

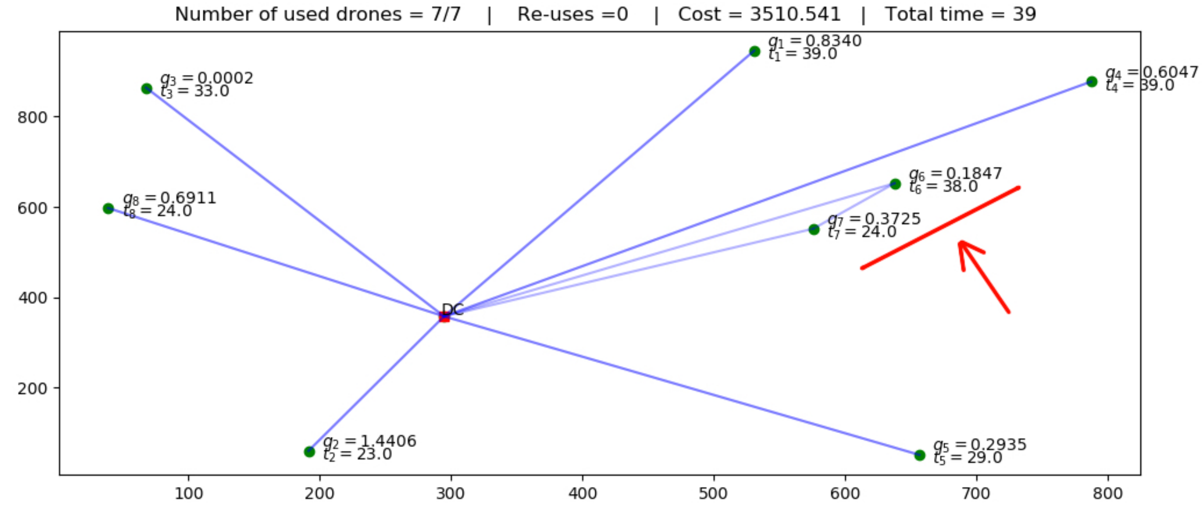

Figure 4.3: Time optimization, 7 drones for 8 destinations

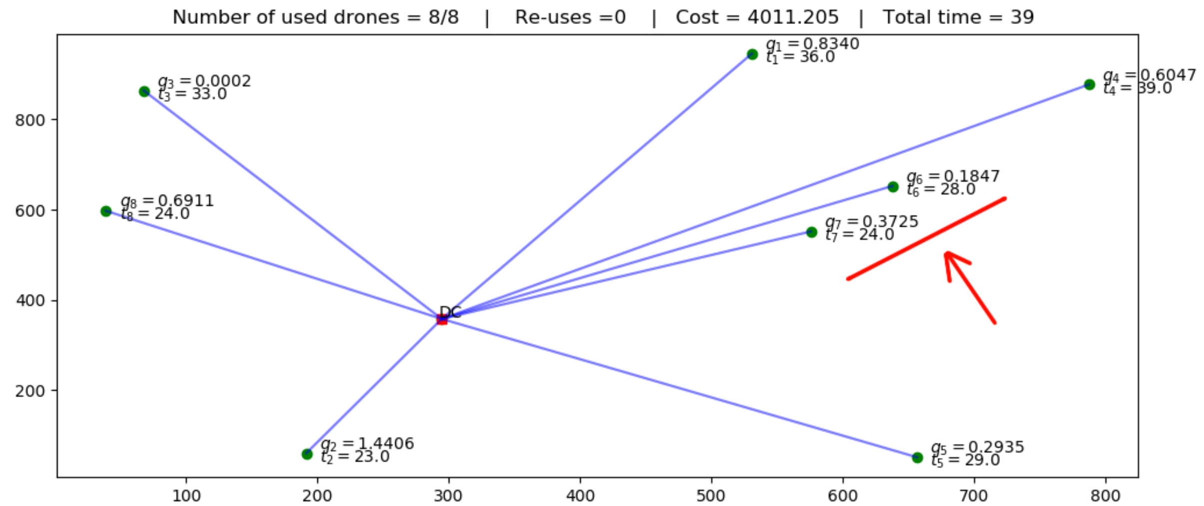

Figure 4.4: Time optimization, 8 drones for 8 destinations

5 Conclusions

After doing the validation of the model and going through the sensitivity analysis, some elements of its performance have to be analyzed.

First, about the reusability feature of the model, it is always used when possible during cost optimisations. Unless it's not taken into account in the cost function, the model will always try to optimise for one drone. However, when optimising for time, reusability is mostly ignored, unless there are few drones available and the destinations are close enough to the depot to allow for it.

Second, the algorithm is not sensible to time efficiency logic when optimizing for costs. This results in the solution taking the whole time limit to complete the routes because the simulation will automatically wait until critical time values to begin the deliveries. In practical terms, the real effective time begins when the drone starts flying, so the waiting time can simply be ignored. However, for correctness in the values an option could be to set a specific delivery time for each destination and thus include an extra set of constraints.

Third, from the computation times, we can derive that the model works efficiently for a small range of destinations when doing cost optimisation. An alternative to improve its computation performance with large sets of destinations would be to simply take away reusability, but then having a considerable increase in costs. For the case of time optimisation, computation times are acceptable as long as proportional amounts of drones and destinations are given.

Fourth, for typical drone applications, this model shows that energy costs have little contribution to the cost function as compared to the cost of the drones themselves. On the other side, if the models were to be tested under more critical circumstances involving greater distances and capacities, then the energy costs could increase enough to change the solution or even make it infeasible. However, the model here used relies on linear approximations and assumptions about drone specifications when computing the energy consumption, so that energy constraints might need to be re-defined for a more reliable solution when used in the above mentioned critical scenarios.

Fifth, from figure 4.1 and 4.2 the logical conclusion would be to use drones with high capacity and velocity. However, physical limitations, as supported by [Liong et al][4], leave place to choose between two possible scenarios: increasing velocity without compromising capacity (more powerful batteries thus more elevated costs) or increasing velocity by compromising capacity (bigger batteries, so less payload) . Which explains why time and cost optimisations are treated separately since a trade-off between both of them has to be made at the beginning when setting the values of the parameters.

Summarizing, the model demonstrated to work effectively. Its innovative inclusion of energy costs and reusability feature puts the model closer to a reliable real-life mathematical representation of the drone delivery VRP. Although its application is bounded to drones, it paves the way into defining similar constraints for other types of vehicles.

Bibliography

[1] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313, 2016.

[2] Kevin Dorling, Jordan Heinrichs, Geoffrey G Messier, and Sebastian Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47 (1):70–85, 2016.

[3] Gerald Lieberman. *Introduction to Operations Research*. McGraw-Hill, 10 edition, 6 2014.

[4] C.-Y Liong, I. Wan, and Khairuddin Omar. Vehicle routing problem: Models and solutions. *Journal of Quality Measurement and Analysis*, 4:205–218, 01 2008.

A

Labels used in LP file for the constraints

Constraint	Label in LP file
2.1	Routes_To_i
2.2	Routes_From_i
2.3	Leave_Loc_i
2.4	Reusability_a_i
2.5	Reusability_b_i
2.6	Reusability_c
2.7	Demand_i
2.8	Payload_on_route_ij
2.9	Travel_Timeo_ij
2.10	Delivery_Time_i
2.11	Reuse_Time_ij
2.12	Max_Time_i
2.13	Time_Limit
2.14	Capacity_Restriction_ij
2.15	Find_zeta_a_i
2.16	Find_zeta_b_ij
2.17	Set_q_ij
2.18	Set_q_i0
2.19	Set_f_i_j
2.20	Set_z_a_i
2.21	Set_b_i
2.22	Cost_function
2.23	Budget

Table A.1: Relation between each of the constraints and the name given in the LP file

B

LPsolve file problem

This LPsolve problem represents the scenario from Figure 3.2.

```
Minimize objective:
I

Subject to:

Routes_To_1: x01 + x21 + x31 + x41 + x51 = 1
Routes_To_2: x02 + x12 + x32 + x42 + x52 = 1
Routes_To_3: x03 + x13 + x23 + x43 + x53 = 1
Routes_To_4: x04 + x14 + x24 + x34 + x54 = 1
Routes_To_5: x05 + x15 + x25 + x35 + x45 = 1

Routes_From_1: x10 + x12 + x13 + x14 + x15 = 1
Routes_From_2: x20 + x21 + x23 + x24 + x25 = 1
Routes_From_3: x30 + x31 + x32 + x34 + x35 = 1
Routes_From_4: x40 + x41 + x42 + x43 + x45 = 1
Routes_From_5: x50 + x51 + x52 + x53 + x54 = 1

Leave_Loc_0: x01 + x02 + x03 + x04 + x05 − x10 − x20 − x30 − x40 − x50 = 0
Leave_Loc_1: x10 + x12 + x13 + x14 + x15 − x01 − x21 − x31 − x41 − x51 = 0
Leave_Loc_2: x20 + x21 + x23 + x24 + x25 − x02 − x12 − x32 − x42 − x52 = 0
Leave_Loc_3: x30 + x31 + x32 + x34 + x35 − x03 − x13 − x23 − x43 − x53 = 0
Leave_Loc_4: x40 + x41 + x42 + x43 + x45 − x04 − x14 − x24 − x34 − x54 = 0
Leave_Loc_5: x50 + x51 + x52 + x53 + x54 − x05 − x15 − x25 − x35 − x45 = 0

Reusability_a_1: si11 + si12 + si13 + si14 + si15 − x10 <= 0
Reusability_a_2: si21 + si22 + si23 + si24 + si25 − x20 <= 0
Reusability_a_3: si31 + si32 + si33 + si34 + si35 − x30 <= 0
Reusability_a_4: si41 + si42 + si43 + si44 + si45 − x40 <= 0
Reusability_a_5: si51 + si52 + si53 + si54 + si55 − x50 <= 0

Reusability_b_1: si11 + si21 + si31 + si41 + si51 − x01 <= 0
Reusability_b_2: si12 + si22 + si32 + si42 + si52 − x02 <= 0
Reusability_b_3: si13 + si23 + si33 + si43 + si53 − x03 <= 0
Reusability_b_4: si14 + si24 + si34 + si44 + si54 − x04 <= 0
Reusability_b_5: si15 + si25 + si35 + si45 + si55 − x05 <= 0

Reusability_c: x01 + x02 + x03 + x04 + x05  − si21 − si31 − si41 − si51 − si12 − si32 − si42
− si52 − si13 − si23 − si43 − si53 − si14 − si24 − si34 − si54 − si15 − si25 − si35 − si45 <= 3

Domand_1: y01 + y21 + y31 + y41 + y51 − y10 − y12 − y13 − y14 − y15 = 0.2934971060451444
Domand_2: y02 + y12 + y32 + y42 + y52 − y20 − y21 − y23 − y24 − y25 = 0.2934971060451444
Domand_3: y03 + y13 + y23 + y43 + y53 − y30 − y31 − y32 − y34 − y35 = 0.2934971060451444
Domand_4: y04 + y14 + y24 + y34 + y54 − y40 − y41 − y42 − y43 − y45 = 0.2934971060451444
Domand_5: y05 + y15 + y25 + y35 + y45 − y50 − y51 − y52 − y53 − y54 = 0.2934971060451444

Payload_on_route_10: y10 − 10000 x10 <= 0
Payload_on_route_20: y20 − 10000 x20 <= 0
```

```
Payload_on_route_30: y30 - 10000 x30 <= 0
Payload_on_route_40: y40 - 10000 x40 <= 0
Payload_on_route_50: y50 - 10000 x50 <= 0
Payload_on_route_01: y01 - 10000 x01 <= 0
Payload_on_route_21: y21 - 10000 x21 <= 0
Payload_on_route_31: y31 - 10000 x31 <= 0
Payload_on_route_41: y41 - 10000 x41 <= 0
Payload_on_route_51: y51 - 10000 x51 <= 0
Payload_on_route_02: y02 - 10000 x02 <= 0
Payload_on_route_12: y12 - 10000 x12 <= 0
Payload_on_route_32: y32 - 10000 x32 <= 0
Payload_on_route_42: y42 - 10000 x42 <= 0
Payload_on_route_52: y52 - 10000 x52 <= 0
Payload_on_route_03: y03 - 10000 x03 <= 0
Payload_on_route_13: y13 - 10000 x13 <= 0
Payload_on_route_23: y23 - 10000 x23 <= 0
Payload_on_route_43: y43 - 10000 x43 <= 0
Payload_on_route_53: y53 - 10000 x53 <= 0
Payload_on_route_04: y04 - 10000 x04 <= 0
Payload_on_route_14: y14 - 10000 x14 <= 0
Payload_on_route_24: y24 - 10000 x24 <= 0
Payload_on_route_34: y34 - 10000 x34 <= 0
Payload_on_route_54: y54 - 10000 x54 <= 0
Payload_on_route_05: y05 - 10000 x05 <= 0
Payload_on_route_15: y15 - 10000 x15 <= 0
Payload_on_route_25: y25 - 10000 x25 <= 0
Payload_on_route_35: y35 - 10000 x35 <= 0
Payload_on_route_45: y45 - 10000 x45 <= 0

Travel_Time_01: t0 -t1 + 10000 x01 <= 9907
Travel_Time_21: t2 -t1 + 10000 x21 <= 9967
Travel_Time_31: t3 -t1 + 10000 x31 <= 9873
Travel_Time_41: t4 -t1 + 10000 x41 <= 9847
Travel_Time_51: t5 -t1 + 10000 x51 <= 9841
Travel_Time_02: t0 -t2 + 10000 x02 <= 9907
Travel_Time_12: t1 -t2 + 10000 x12 <= 9967
Travel_Time_32: t3 -t2 + 10000 x32 <= 9857
Travel_Time_42: t4 -t2 + 10000 x42 <= 9849
Travel_Time_52: t5 -t2 + 10000 x52 <= 9847
Travel_Time_03: t0 -t3 + 10000 x03 <= 9907
Travel_Time_13: t1 -t3 + 10000 x13 <= 9873
Travel_Time_23: t2 -t3 + 10000 x23 <= 9857
Travel_Time_43: t4 -t3 + 10000 x43 <= 9885
Travel_Time_53: t5 -t3 + 10000 x53 <= 9864
Travel_Time_04: t0 -t4 + 10000 x04 <= 9930
Travel_Time_14: t1 -t4 + 10000 x14 <= 9847
Travel_Time_24: t2 -t4 + 10000 x24 <= 9849
Travel_Time_34: t3 -t4 + 10000 x34 <= 9885
Travel_Time_54: t5 -t4 + 10000 x54 <= 9967
Travel_Time_05: t0 -t5 + 10000 x05 <= 9922
Travel_Time_15: t1 -t5 + 10000 x15 <= 9841
Travel_Time_25: t2 -t5 + 10000 x25 <= 9847
Travel_Time_35: t3 -t5 + 10000 x35 <= 9864
Travel_Time_45: t4 -t5 + 10000 x45 <= 9967

Delivery_Time_1: t1 - a1 + 10000 x10 <= 9907
Delivery_Time_2: t2 - a2 + 10000 x20 <= 9907
Delivery_Time_3: t3 - a3 + 10000 x30 <= 9907
Delivery_Time_4: t4 - a4 + 10000 x40 <= 9930
Delivery_Time_5: t5 - a5 + 10000 x50 <= 9922

Reuse_Time_21: a2 - t1 + 10000 si21 <= 9907
Reuse_Time_31: a3 - t1 + 10000 si31 <= 9907
Reuse_Time_41: a4 - t1 + 10000 si41 <= 9907
Reuse_Time_51: a5 - t1 + 10000 si51 <= 9907
Reuse_Time_12: a1 - t2 + 10000 si12 <= 9907
Reuse_Time_32: a3 - t2 + 10000 si32 <= 9907
Reuse_Time_42: a4 - t2 + 10000 si42 <= 9907
Reuse_Time_52: a5 - t2 + 10000 si52 <= 9907
Reuse_Time_13: a1 - t3 + 10000 si13 <= 9907
Reuse_Time_23: a2 - t3 + 10000 si23 <= 9907
```

```
Reuse_Time_43: a4 - t3 + 10000 si43 <= 9907
Reuse_Time_53: a5 - t3 + 10000 si53 <= 9907
Reuse_Time_14: a1 - t4 + 10000 si14 <= 9930
Reuse_Time_24: a2 - t4 + 10000 si24 <= 9930
Reuse_Time_34: a3 - t4 + 10000 si34 <= 9930
Reuse_Time_54: a5 - t4 + 10000 si54 <= 9930
Reuse_Time_15: a1 - t5 + 10000 si15 <= 9922
Reuse_Time_25: a2 - t5 + 10000 si25 <= 9922
Reuse_Time_35: a3 - t5 + 10000 si35 <= 9922
Reuse_Time_45: a4 - t5 + 10000 si45 <= 9922

Max_Time_1: t1 -l <= 0
Max_Time_2: t2 -l <= 0
Max_Time_3: t3 -l <= 0
Max_Time_4: t4 -l <= 0
Max_Time_5: t5 -l <= 0

Time_Limit: l <= 2000

Capacity_Restriction_10: q10 + y10 - 5 x10 <= 0
Capacity_Restriction_20: q20 + y20 - 5 x20 <= 0
Capacity_Restriction_30: q30 + y30 - 5 x30 <= 0
Capacity_Restriction_40: q40 + y40 - 5 x40 <= 0
Capacity_Restriction_50: q50 + y50 - 5 x50 <= 0
Capacity_Restriction_01: q01 + y01 - 5 x01 <= 0
Capacity_Restriction_21: q21 + y21 - 5 x21 <= 0
Capacity_Restriction_31: q31 + y31 - 5 x31 <= 0
Capacity_Restriction_41: q41 + y41 - 5 x41 <= 0
Capacity_Restriction_51: q51 + y51 - 5 x51 <= 0
Capacity_Restriction_02: q02 + y02 - 5 x02 <= 0
Capacity_Restriction_12: q12 + y12 - 5 x12 <= 0
Capacity_Restriction_32: q32 + y32 - 5 x32 <= 0
Capacity_Restriction_42: q42 + y42 - 5 x42 <= 0
Capacity_Restriction_52: q52 + y52 - 5 x52 <= 0
Capacity_Restriction_03: q03 + y03 - 5 x03 <= 0
Capacity_Restriction_13: q13 + y13 - 5 x13 <= 0
Capacity_Restriction_23: q23 + y23 - 5 x23 <= 0
Capacity_Restriction_43: q43 + y43 - 5 x43 <= 0
Capacity_Restriction_53: q53 + y53 - 5 x53 <= 0
Capacity_Restriction_04: q04 + y04 - 5 x04 <= 0
Capacity_Restriction_14: q14 + y14 - 5 x14 <= 0
Capacity_Restriction_24: q24 + y24 - 5 x24 <= 0
Capacity_Restriction_34: q34 + y34 - 5 x34 <= 0
Capacity_Restriction_54: q54 + y54 - 5 x54 <= 0
Capacity_Restriction_05: q05 + y05 - 5 x05 <= 0
Capacity_Restriction_15: q15 + y15 - 5 x15 <= 0
Capacity_Restriction_25: q25 + y25 - 5 x25 <= 0
Capacity_Restriction_35: q35 + y35 - 5 x35 <= 0
Capacity_Restriction_45: q45 + y45 - 5 x45 <= 0

Find_zeta_a_1: 0.0015384615384615385 z1 - zeta1 + 10000 x10 <= 10000
Find_zeta_a_2: 0.0015384615384615385 z2 - zeta2 + 10000 x20 <= 10000
Find_zeta_a_3: 0.0015384615384615385 z3 - zeta3 + 10000 x30 <= 10000
Find_zeta_a_4: 0.0015384615384615385 z4 - zeta4 + 10000 x40 <= 10000
Find_zeta_a_5: 0.0015384615384615385 z5 - zeta5 + 10000 x50 <= 10000

Find_zeta_b_21: zeta2 -zeta1 + 10000 x12 <= 10000
Find_zeta_b_31: zeta3 -zeta1 + 10000 x13 <= 10000
Find_zeta_b_41: zeta4 -zeta1 + 10000 x14 <= 10000
Find_zeta_b_51: zeta5 -zeta1 + 10000 x15 <= 10000
Find_zeta_b_12: zeta1 -zeta2 + 10000 x21 <= 10000
Find_zeta_b_32: zeta3 -zeta2 + 10000 x23 <= 10000
Find_zeta_b_42: zeta4 -zeta2 + 10000 x24 <= 10000
Find_zeta_b_52: zeta5 -zeta2 + 10000 x25 <= 10000
Find_zeta_b_13: zeta1 -zeta3 + 10000 x31 <= 10000
Find_zeta_b_23: zeta2 -zeta3 + 10000 x32 <= 10000
Find_zeta_b_43: zeta4 -zeta3 + 10000 x34 <= 10000
Find_zeta_b_53: zeta5 -zeta3 + 10000 x35 <= 10000
Find_zeta_b_14: zeta1 -zeta4 + 10000 x41 <= 10000
Find_zeta_b_24: zeta2 -zeta4 + 10000 x42 <= 10000
Find_zeta_b_34: zeta3 -zeta4 + 10000 x43 <= 10000
```

```
Find_zeta_b_54: zeta5 -zeta4 + 10000 x45 <= 10000
Find_zeta_b_15: zeta1 -zeta5 + 10000 x51 <= 10000
Find_zeta_b_25: zeta2 -zeta5 + 10000 x52 <= 10000
Find_zeta_b_35: zeta3 -zeta5 + 10000 x53 <= 10000
Find_zeta_b_45: zeta4 -zeta5 + 10000 x54 <= 10000

Set_q_01: q01 - zeta1 - 10000 x01 >= -10000
Set_q_21: q21 - zeta1 - 10000 x21 >= -10000
Set_q_31: q31 - zeta1 - 10000 x31 >= -10000
Set_q_41: q41 - zeta1 - 10000 x41 >= -10000
Set_q_51: q51 - zeta1 - 10000 x51 >= -10000
Set_q_02: q02 - zeta2 - 10000 x02 >= -10000
Set_q_12: q12 - zeta2 - 10000 x12 >= -10000
Set_q_32: q32 - zeta2 - 10000 x32 >= -10000
Set_q_42: q42 - zeta2 - 10000 x42 >= -10000
Set_q_52: q52 - zeta2 - 10000 x52 >= -10000
Set_q_03: q03 - zeta3 - 10000 x03 >= -10000
Set_q_13: q13 - zeta3 - 10000 x13 >= -10000
Set_q_23: q23 - zeta3 - 10000 x23 >= -10000
Set_q_43: q43 - zeta3 - 10000 x43 >= -10000
Set_q_53: q53 - zeta3 - 10000 x53 >= -10000
Set_q_04: q04 - zeta4 - 10000 x04 >= -10000
Set_q_14: q14 - zeta4 - 10000 x14 >= -10000
Set_q_24: q24 - zeta4 - 10000 x24 >= -10000
Set_q_34: q34 - zeta4 - 10000 x34 >= -10000
Set_q_54: q54 - zeta4 - 10000 x54 >= -10000
Set_q_05: q05 - zeta5 - 10000 x05 >= -10000
Set_q_15: q15 - zeta5 - 10000 x15 >= -10000
Set_q_25: q25 - zeta5 - 10000 x25 >= -10000
Set_q_35: q35 - zeta5 - 10000 x35 >= -10000
Set_q_45: q45 - zeta5 - 10000 x45 >= -10000

Set_q_10: q10 - zeta1 - 10000 x10 >= -10000
Set_q_20: q20 - zeta2 - 10000 x20 >= -10000
Set_q_30: q30 - zeta3 - 10000 x30 >= -10000
Set_q_40: q40 - zeta4 - 10000 x40 >= -10000
Set_q_50: q50 - zeta5 - 10000 x50 >= -10000

Set_f_0_1: f0 - f1 + 20.253333333333334 q01 + 20.253333333333334 y01 + 10000 x01 <= 9983
Set_f_2_1: f2 - f1 + 7.2695 q21 + 7.2695 y21 + 10000 x21 <= 9994
Set_f_3_1: f3 - f1 + 27.739833333333333 q31 + 27.739833333333333 y31 + 10000 x31 <= 9977
Set_f_4_1: f4 - f1 + 33.27333333333333 q41 + 33.27333333333333 y41 + 10000 x41 <= 9972
Set_f_5_1: f5 - f1 + 34.503 q51 + 34.503 y51 + 10000 x51 <= 9971
Set_f_0_2: f0 - f2 + 20.253333333333334 q02 + 20.253333333333334 y02 + 10000 x02 <= 9983
Set_f_1_2: f1 - f2 + 7.2695 q12 + 7.2695 y12 + 10000 x12 <= 9994
Set_f_3_2: f3 - f2 + 31.103333333333335 q32 + 31.103333333333335 y32 + 10000 x32 <= 9974
Set_f_4_2: f4 - f2 + 32.839333333333336 q42 + 32.839333333333336 y42 + 10000 x42 <= 9973
Set_f_5_2: f5 - f2 + 33.27333333333333 q52 + 33.27333333333333 y52 + 10000 x52 <= 9972
Set_f_0_3: f0 - f3 + 20.253333333333334 q03 + 20.253333333333334 y03 + 10000 x03 <= 9983
Set_f_1_3: f1 - f3 + 27.739833333333333 q13 + 27.739833333333333 y13 + 10000 x13 <= 9977
Set_f_2_3: f2 - f3 + 31.103333333333335 q23 + 31.103333333333335 y23 + 10000 x23 <= 9974
Set_f_4_3: f4 - f3 + 25.02733333333333 q43 + 25.02733333333333 y43 + 10000 x43 <= 9979
Set_f_5_3: f5 - f3 + 29.69283333333333 q53 + 29.69283333333333 y53 + 10000 x53 <= 9975
Set_f_0_4: f0 - f4 + 15.19 q04 + 15.19 y04 + 10000 x04 <= 9988
Set_f_1_4: f1 - f4 + 33.27333333333333 q14 + 33.27333333333333 y14 + 10000 x14 <= 9972
Set_f_2_4: f2 - f4 + 32.839333333333336 q24 + 32.839333333333336 y24 + 10000 x24 <= 9973
Set_f_3_4: f3 - f4 + 25.02733333333333 q34 + 25.02733333333333 y34 + 10000 x34 <= 9979
Set_f_5_4: f5 - f4 + 7.2695 q54 + 7.2695 y54 + 10000 x54 <= 9994
Set_f_0_5: f0 - f5 + 17.070666666666668 q05 + 17.070666666666668 y05 + 10000 x05 <= 9986
Set_f_1_5: f1 - f5 + 34.503 q15 + 34.503 y15 + 10000 x15 <= 9971
Set_f_2_5: f2 - f5 + 33.27333333333333 q25 + 33.27333333333333 y25 + 10000 x25 <= 9972
Set_f_3_5: f3 - f5 + 29.69283333333333 q35 + 29.69283333333333 y35 + 10000 x35 <= 9975
Set_f_4_5: f4 - f5 + 7.2695 q45 + 7.2695 y45 + 10000 x45 <= 9994

Set_z_a_1: f1 -z1 + 20.253333333333334 q10 + 20.253333333333334 y10 + 10000 x10 <= 9983
Set_z_a_2: f2 -z2 + 20.253333333333334 q20 + 20.253333333333334 y20 + 10000 x20 <= 9983
Set_z_a_3: f3 -z3 + 20.253333333333334 q30 + 20.253333333333334 y30 + 10000 x30 <= 9983
Set_z_a_4: f4 -z4 + 15.19 q40 + 15.19 y40 + 10000 x40 <= 9988
Set_z_a_5: f5 -z5 + 17.070666666666668 q50 + 17.070666666666668 y50 + 10000 x50 <= 9986

Set_z_b_1: z1 - 10000 x10 <= 0
```

```
Set_z_b_2: z2 – 10000 x20 <= 0
Set_z_b_3: z3 – 10000 x30 <= 0
Set_z_b_4: z4 – 10000 x40 <= 0
Set_z_b_5: z5 – 10000 x50 <= 0

Cost_Function: 500 x01 +500 x02 +500 x03 +500 x04 +500 x05 – 500 si21 –500 si31 –500 si41
–500 si51 –500 si12 –500 si32 –500 si42 –500 si52 –500 si13 –500 si23 –500 si43 –500 si53
–500 si14 –500 si24 –500 si34 –500 si54 –500 si15 –500 si25 –500 si35 –500 si45 + 0.1 z1
+0.1 z2 +0.1 z3 +0.1 z4 +0.1 z5 – c = 0

Budget: c <= 300000

binary
x10 x01 x20 x02 x30 x03 x40 x04 x50 x05 x01 x10 si11 si11 x21 x12 si12 si21 x31 x13 si13
si31 x41 x14 si14 si41 x51 x15 si15 si51 x02 x20 x12 x21 si21 si12 si22 si22 x32 x23 si23
si32 x42 x24 si24 si42 x52 x25 si25 si52 x03 x30 x13 x31 si31 si13 x23 x32 si32 si23 si33
si33 x43 x34 si34 si43 x53 x35 si35 si53 x04 x40 x14 x41 si41 si14 x24 x42 si42 si24 x34
x43 si43 si34 si44 si44 x54 x45 si45 si54 x05 x50 x15 x51 si51 si15 x25 x52 si52 si25 x35
x53 si53 si35 x45 x54 si54 si45 si55 si55

end
```

C

LPsolve file solution

This is the solution of the LPsolve file in Appendix B.


```
<?xml version = ”1.0” encoding=”UTF–8” standalone=”yes”?>
<CPLEXSolution version=”1.2”>
 <header
   problemName=”C:\Users\ricca\OneDrive\Desktop\problem.lp”
   solutionName=”incumbent”
   solutionIndex=”–1”
   objectiveValue=”126”
   solutionTypeValue=”3”
   solutionTypeString=”primal”
   solutionStatusValue=”101”
   solutionStatusString=”integer␣optimal␣solution”
   solutionMethodString=”mip”
   primalFeasible=”1”
   dualFeasible=”1”
   MIPNodes=”0”
   MIPIterations=”170”
   writeLevel=”1”/>
 <quality
   epInt=”1.0000000000000001e–05”
   epRHS=”9.9999999999999995e–07”
   maxIntInfeas=”0”
   maxPrimalInfeas=”0”
   maxX=”2093.0000000000014”
   maxSlack=”298485.20800313848”/>
 <linearConstraints>
  <constraint name=”Routes_To_1” index=”0” slack=”0”/>
  <constraint name=”Routes_To_2” index=”1” slack=”0”/>
  <constraint name=”Routes_To_3” index=”2” slack=”0”/>
  <constraint name=”Routes_To_4” index=”3” slack=”0”/>
  <constraint name=”Routes_To_5” index=”4” slack=”0”/>
  <constraint name=”Routes_From_1” index=”5” slack=”0”/>
  <constraint name=”Routes_From_2” index=”6” slack=”0”/>
  <constraint name=”Routes_From_3” index=”7” slack=”0”/>
  <constraint name=”Routes_From_4” index=”8” slack=”0”/>
  <constraint name=”Routes_From_5” index=”9” slack=”0”/>
  <constraint name=”Leave_Loc_0” index=”10” slack=”0”/>
  <constraint name=”Leave_Loc_1” index=”11” slack=”0”/>
  <constraint name=”Leave_Loc_2” index=”12” slack=”0”/>
  <constraint name=”Leave_Loc_3” index=”13” slack=”0”/>
  <constraint name=”Leave_Loc_4” index=”14” slack=”0”/>
  <constraint name=”Leave_Loc_5” index=”15” slack=”0”/>
  <constraint name=”Reusability_a_1” index=”16” slack=”0”/>
  <constraint name=”Reusability_a_2” index=”17” slack=”1”/>
  <constraint name=”Reusability_a_3” index=”18” slack=”1”/>
  <constraint name=”Reusability_a_4” index=”19” slack=”1”/>
  <constraint name=”Reusability_a_5” index=”20” slack=”0”/>
  <constraint name=”Reusability_b_1” index=”21” slack=”1”/>
```

<constraint name=”Reusability_b_2” index=”22” slack=”0”/>
<constraint name=”Reusability_b_3” index=”23” slack=”1”/>
<constraint name=”Reusability_b_4” index=”24” slack=”0”/>
<constraint name=”Reusability_b_5” index=”25” slack=”1”/>
<constraint name=”Reusability_c” index=”26” slack=”0”/>
<constraint name=”Domand_1” index=”27” slack=”0”/>
<constraint name=”Domand_2” index=”28” slack=”0”/>
<constraint name=”Domand_3” index=”29” slack=”0”/>
<constraint name=”Domand_4” index=”30” slack=”0”/>
<constraint name=”Domand_5” index=”31” slack=”0”/>
<constraint name=”Payload_on_route_10” index=”32” slack=”0”/>
<constraint name=”Payload_on_route_20” index=”33” slack=”10000”/>
<constraint name=”Payload_on_route_30” index=”34” slack=”10000”/>
<constraint name=”Payload_on_route_40” index=”35” slack=”10000”/>
<constraint name=”Payload_on_route_50” index=”36” slack=”0”/>
<constraint name=”Payload_on_route_01” index=”37” slack=”9999.4130057879102”/>
<constraint name=”Payload_on_route_21” index=”38” slack=”0”/>
<constraint name=”Payload_on_route_31” index=”39” slack=”0”/>
<constraint name=”Payload_on_route_41” index=”40” slack=”0”/>
<constraint name=”Payload_on_route_51” index=”41” slack=”0”/>
<constraint name=”Payload_on_route_02” index=”42” slack=”0”/>
<constraint name=”Payload_on_route_12” index=”43” slack=”9999.7065028939542”/>
<constraint name=”Payload_on_route_32” index=”44” slack=”0”/>
<constraint name=”Payload_on_route_42” index=”45” slack=”0”/>
<constraint name=”Payload_on_route_52” index=”46” slack=”0”/>
<constraint name=”Payload_on_route_03” index=”47” slack=”9999.7065028939542”/>
<constraint name=”Payload_on_route_13” index=”48” slack=”0”/>
<constraint name=”Payload_on_route_23” index=”49” slack=”0”/>
<constraint name=”Payload_on_route_43” index=”50” slack=”0”/>
<constraint name=”Payload_on_route_53” index=”51” slack=”0”/>
<constraint name=”Payload_on_route_04” index=”52” slack=”0”/>
<constraint name=”Payload_on_route_14” index=”53” slack=”0”/>
<constraint name=”Payload_on_route_24” index=”54” slack=”0”/>
<constraint name=”Payload_on_route_34” index=”55” slack=”0”/>
<constraint name=”Payload_on_route_54” index=”56” slack=”9999.7065028939542”/>
<constraint name=”Payload_on_route_05” index=”57” slack=”9999.4130057879102”/>
<constraint name=”Payload_on_route_15” index=”58” slack=”0”/>
<constraint name=”Payload_on_route_25” index=”59” slack=”0”/>
<constraint name=”Payload_on_route_35” index=”60” slack=”0”/>
<constraint name=”Payload_on_route_45” index=”61” slack=”0”/>
<constraint name=”Travel_Time_01” index=”62” slack=”0”/>
<constraint name=”Travel_Time_21” index=”63” slack=”9934”/>
<constraint name=”Travel_Time_31” index=”64” slack=”9873”/>
<constraint name=”Travel_Time_41” index=”65” slack=”9814”/>
<constraint name=”Travel_Time_51” index=”66” slack=”9856”/>
<constraint name=”Travel_Time_02” index=”67” slack=”10033”/>
<constraint name=”Travel_Time_12” index=”68” slack=”0”/>
<constraint name=”Travel_Time_32” index=”69” slack=”9890”/>
<constraint name=”Travel_Time_42” index=”70” slack=”9849”/>
<constraint name=”Travel_Time_52” index=”71” slack=”9895”/>
<constraint name=”Travel_Time_03” index=”72” slack=”0”/>
<constraint name=”Travel_Time_13” index=”73” slack=”9873”/>
<constraint name=”Travel_Time_23” index=”74” slack=”9824”/>
<constraint name=”Travel_Time_43” index=”75” slack=”9852”/>
<constraint name=”Travel_Time_53” index=”76” slack=”9879”/>
<constraint name=”Travel_Time_04” index=”77” slack=”10056”/>
<constraint name=”Travel_Time_14” index=”78” slack=”9880”/>
<constraint name=”Travel_Time_24” index=”79” slack=”9849”/>
<constraint name=”Travel_Time_34” index=”80” slack=”9918”/>
<constraint name=”Travel_Time_54” index=”81” slack=”15”/>
<constraint name=”Travel_Time_05” index=”82” slack=”0”/>
<constraint name=”Travel_Time_15” index=”83” slack=”9826”/>
<constraint name=”Travel_Time_25” index=”84” slack=”9799”/>
<constraint name=”Travel_Time_35” index=”85” slack=”9849”/>
<constraint name=”Travel_Time_45” index=”86” slack=”9919”/>
<constraint name=”Delivery_Time_1” index=”87” slack=”11907.000000000002”/>
<constraint name=”Delivery_Time_2” index=”88” slack=”1874.0000000000018”/>
<constraint name=”Delivery_Time_3” index=”89” slack=”1907.0000000000018”/>
<constraint name=”Delivery_Time_4” index=”90” slack=”0”/>
<constraint name=”Delivery_Time_5” index=”91” slack=”9844”/>
<constraint name=”Reuse_Time_21” index=”92” slack=”7906.9999999999982”/>

<constraint name=”Reuse_Time_31” index=”93” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_41” index=”94” slack=”9804”/>
<constraint name=”Reuse_Time_51” index=”95” slack=”10000”/>
<constraint name=”Reuse_Time_12” index=”96” slack=”7939.9999999999982”/>
<constraint name=”Reuse_Time_32” index=”97” slack=”7939.9999999999982”/>
<constraint name=”Reuse_Time_42” index=”98” slack=”9837”/>
<constraint name=”Reuse_Time_52” index=”99” slack=”10033”/>
<constraint name=”Reuse_Time_13” index=”100” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_23” index=”101” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_43” index=”102” slack=”9804”/>
<constraint name=”Reuse_Time_53” index=”103” slack=”10000”/>
<constraint name=”Reuse_Time_14” index=”104” slack=”7962.9999999999982”/>
<constraint name=”Reuse_Time_24” index=”105” slack=”7962.9999999999982”/>
<constraint name=”Reuse_Time_34” index=”106” slack=”7962.9999999999982”/>
<constraint name=”Reuse_Time_54” index=”107” slack=”10056”/>
<constraint name=”Reuse_Time_15” index=”108” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_25” index=”109” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_35” index=”110” slack=”7906.9999999999982”/>
<constraint name=”Reuse_Time_45” index=”111” slack=”9804”/>
<constraint name=”Max_Time_1” index=”112” slack=”33”/>
<constraint name=”Max_Time_2” index=”113” slack=”0”/>
<constraint name=”Max_Time_3” index=”114” slack=”33”/>
<constraint name=”Max_Time_4” index=”115” slack=”0”/>
<constraint name=”Max_Time_5” index=”116” slack=”48”/>
<constraint name=”Time_Limit” index=”117” slack=”1874”/>
<constraint name=”Capacity_Restriction_10” index=”118” slack=”0”/>
<constraint name=”Capacity_Restriction_20” index=”119” slack=”4.910295535207263”/>
<constraint name=”Capacity_Restriction_30” index=”120” slack=”4.9344631146289251”/>
<constraint name=”Capacity_Restriction_40” index=”121” slack=”4.9276721676791819”/>
<constraint name=”Capacity_Restriction_50” index=”122” slack=”0”/>
<constraint name=”Capacity_Restriction_01” index=”123” slack=”4.3233013231169739”/>
<constraint name=”Capacity_Restriction_21” index=”124” slack=”0”/>
<constraint name=”Capacity_Restriction_31” index=”125” slack=”0”/>
<constraint name=”Capacity_Restriction_41” index=”126” slack=”0”/>
<constraint name=”Capacity_Restriction_51” index=”127” slack=”0”/>
<constraint name=”Capacity_Restriction_02” index=”128” slack=”0”/>
<constraint name=”Capacity_Restriction_12” index=”129” slack=”4.6167984291621185”/>
<constraint name=”Capacity_Restriction_32” index=”130” slack=”0”/>
<constraint name=”Capacity_Restriction_42” index=”131” slack=”0”/>
<constraint name=”Capacity_Restriction_52” index=”132” slack=”0”/>
<constraint name=”Capacity_Restriction_03” index=”133” slack=”4.6409660085837805”/>
<constraint name=”Capacity_Restriction_13” index=”134” slack=”0”/>
<constraint name=”Capacity_Restriction_23” index=”135” slack=”0”/>
<constraint name=”Capacity_Restriction_43” index=”136” slack=”0”/>
<constraint name=”Capacity_Restriction_53” index=”137” slack=”0”/>
<constraint name=”Capacity_Restriction_04” index=”138” slack=”0”/>
<constraint name=”Capacity_Restriction_14” index=”139” slack=”0”/>
<constraint name=”Capacity_Restriction_24” index=”140” slack=”0”/>
<constraint name=”Capacity_Restriction_34” index=”141” slack=”0”/>
<constraint name=”Capacity_Restriction_54” index=”142” slack=”4.6341750616340374”/>
<constraint name=”Capacity_Restriction_05” index=”143” slack=”4.3406779555888928”/>
<constraint name=”Capacity_Restriction_15” index=”144” slack=”0”/>
<constraint name=”Capacity_Restriction_25” index=”145” slack=”0”/>
<constraint name=”Capacity_Restriction_35” index=”146” slack=”0”/>
<constraint name=”Capacity_Restriction_45” index=”147” slack=”0”/>
<constraint name=”Find_zeta_a_1” index=”148” slack=”10000.089704464794”/>
<constraint name=”Find_zeta_a_2” index=”149” slack=”0”/>
<constraint name=”Find_zeta_a_3” index=”150” slack=”0”/>
<constraint name=”Find_zeta_a_4” index=”151” slack=”0”/>
<constraint name=”Find_zeta_a_5” index=”152” slack=”10000.07232783232”/>
<constraint name=”Find_zeta_b_21” index=”153” slack=”0”/>
<constraint name=”Find_zeta_b_31” index=”154” slack=”10000.024167579422”/>
<constraint name=”Find_zeta_b_41” index=”155” slack=”10000.017376632471”/>
<constraint name=”Find_zeta_b_51” index=”156” slack=”10000.017376632471”/>
<constraint name=”Find_zeta_b_12” index=”157” slack=”10000”/>
<constraint name=”Find_zeta_b_32” index=”158” slack=”10000.024167579422”/>
<constraint name=”Find_zeta_b_42” index=”159” slack=”10000.017376632471”/>
<constraint name=”Find_zeta_b_52” index=”160” slack=”10000.017376632471”/>
<constraint name=”Find_zeta_b_13” index=”161” slack=”9999.9758324205777”/>
<constraint name=”Find_zeta_b_23” index=”162” slack=”9999.9758324205777”/>
<constraint name=”Find_zeta_b_43” index=”163” slack=”9999.993209053051”/>

<constraint name=”Find_zeta_b_53” index=”164” slack=”9999.993209053051”/>
<constraint name=”Find_zeta_b_14” index=”165” slack=”9999.9826233675285”/>
<constraint name=”Find_zeta_b_24” index=”166” slack=”9999.9826233675285”/>
<constraint name=”Find_zeta_b_34” index=”167” slack=”10000.006790946949”/>
<constraint name=”Find_zeta_b_54” index=”168” slack=”10000”/>
<constraint name=”Find_zeta_b_15” index=”169” slack=”9999.9826233675285”/>
<constraint name=”Find_zeta_b_25” index=”170” slack=”9999.9826233675285”/>
<constraint name=”Find_zeta_b_35” index=”171” slack=”10000.006790946949”/>
<constraint name=”Find_zeta_b_45” index=”172” slack=”0”/>
<constraint name=”Set_q_01” index=”173” slack=”0”/>
<constraint name=”Set_q_21” index=”174” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_31” index=”175” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_41” index=”176” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_51” index=”177” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_02” index=”178” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_12” index=”179” slack=”0”/>
<constraint name=”Set_q_32” index=”180” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_42” index=”181” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_52” index=”182” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_03” index=”183” slack=”0”/>
<constraint name=”Set_q_13” index=”184” slack=”–9999.9344631146287”/>
<constraint name=”Set_q_23” index=”185” slack=”–9999.9344631146287”/>
<constraint name=”Set_q_43” index=”186” slack=”–9999.9344631146287”/>
<constraint name=”Set_q_53” index=”187” slack=”–9999.9344631146287”/>
<constraint name=”Set_q_04” index=”188” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_14” index=”189” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_24” index=”190” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_34” index=”191” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_54” index=”192” slack=”0”/>
<constraint name=”Set_q_05” index=”193” slack=”0”/>
<constraint name=”Set_q_15” index=”194” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_25” index=”195” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_35” index=”196” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_45” index=”197” slack=”–9999.9276721676797”/>
<constraint name=”Set_q_10” index=”198” slack=”–9999.9102955352064”/>
<constraint name=”Set_q_20” index=”199” slack=”0”/>
<constraint name=”Set_q_30” index=”200” slack=”0”/>
<constraint name=”Set_q_40” index=”201” slack=”0”/>
<constraint name=”Set_q_50” index=”202” slack=”–9999.9276721676797”/>
<constraint name=”Set_f_0_1” index=”203” slack=”0”/>
<constraint name=”Set_f_2_1” index=”204” slack=”9985.2143161807944”/>
<constraint name=”Set_f_3_1” index=”205” slack=”9983.4337687629886”/>
<constraint name=”Set_f_4_1” index=”206” slack=”9968.7909726335583”/>
<constraint name=”Set_f_5_1” index=”207” slack=”9976.4503370230104”/>
<constraint name=”Set_f_0_2” index=”208” slack=”10022.491087688344”/>
<constraint name=”Set_f_1_2” index=”209” slack=”1.8189894035458565e–12”/>
<constraint name=”Set_f_3_2” index=”210” slack=”9989.2194525821942”/>
<constraint name=”Set_f_4_2” index=”211” slack=”9978.5766564527657”/>
<constraint name=”Set_f_5_2” index=”212” slack=”9986.236020842216”/>
<constraint name=”Set_f_0_3” index=”213” slack=”0”/>
<constraint name=”Set_f_1_3” index=”214” slack=”9970.5662312370114”/>
<constraint name=”Set_f_2_3” index=”215” slack=”9958.7805474178058”/>
<constraint name=”Set_f_4_3” index=”216” slack=”9969.3572038705715”/>
<constraint name=”Set_f_5_3” index=”217” slack=”9974.0165682600218”/>
<constraint name=”Set_f_0_4” index=”218” slack=”10021.914431235578”/>
<constraint name=”Set_f_1_4” index=”219” slack=”9975.2090273664417”/>
<constraint name=”Set_f_2_4” index=”220” slack=”9967.4233435472343”/>
<constraint name=”Set_f_3_4” index=”221” slack=”9988.6427961294285”/>
<constraint name=”Set_f_5_4” index=”222” slack=”0”/>
<constraint name=”Set_f_0_5” index=”223” slack=”1.8189894035458565e–12”/>
<constraint name=”Set_f_1_5” index=”224” slack=”9965.5496629769896”/>
<constraint name=”Set_f_2_5” index=”225” slack=”9957.763979157784”/>
<constraint name=”Set_f_3_5” index=”226” slack=”9975.9834317399782”/>
<constraint name=”Set_f_4_5” index=”227” slack=”9985.3406356105479”/>
<constraint name=”Set_z_a_1” index=”228” slack=”9952.2945961308633”/>
<constraint name=”Set_z_a_2” index=”229” slack=”0”/>
<constraint name=”Set_z_a_3” index=”230” slack=”0”/>
<constraint name=”Set_z_a_4” index=”231” slack=”0”/>
<constraint name=”Set_z_a_5” index=”232” slack=”9960.7449331538719”/>
<constraint name=”Set_z_b_1” index=”233” slack=”0”/>
<constraint name=”Set_z_b_2” index=”234” slack=”9941.6920978847211”/>

```
  <constraint name=”Set_z_b_3” index=”235” slack=”9957.4010245088011”/>
  <constraint name=”Set_z_b_4” index=”236” slack=”9952.9869089914682”/>
  <constraint name=”Set_z_b_5” index=”237” slack=”0”/>
  <constraint name=”Cost_Function” index=”238” slack=”0”/>
  <constraint name=”Budget” index=”239” slack=”298485.20800313848”/>
 </linearConstraints>
 <variables>
  <variable name=”l” index=”0” value=”126”/>
  <variable name=”x01” index=”1” value=”1”/>
  <variable name=”x21” index=”2” value=”-0”/>
  <variable name=”x31” index=”3” value=”-0”/>
  <variable name=”x41” index=”4” value=”-0”/>
  <variable name=”x51” index=”5” value=”-0”/>
  <variable name=”x02” index=”6” value=”-0”/>
  <variable name=”x12” index=”7” value=”1”/>
  <variable name=”x32” index=”8” value=”-0”/>
  <variable name=”x42” index=”9” value=”-0”/>
  <variable name=”x52” index=”10” value=”-0”/>
  <variable name=”x03” index=”11” value=”1”/>
  <variable name=”x13” index=”12” value=”-0”/>
  <variable name=”x23” index=”13” value=”-0”/>
  <variable name=”x43” index=”14” value=”-0”/>
  <variable name=”x53” index=”15” value=”-0”/>
  <variable name=”x04” index=”16” value=”0”/>
  <variable name=”x14” index=”17” value=”-0”/>
  <variable name=”x24” index=”18” value=”-0”/>
  <variable name=”x34” index=”19” value=”0”/>
  <variable name=”x54” index=”20” value=”1”/>
  <variable name=”x05” index=”21” value=”1”/>
  <variable name=”x15” index=”22” value=”-0”/>
  <variable name=”x25” index=”23” value=”-0”/>
  <variable name=”x35” index=”24” value=”-0”/>
  <variable name=”x45” index=”25” value=”-0”/>
  <variable name=”x10” index=”26” value=”-0”/>
  <variable name=”x20” index=”27” value=”1”/>
  <variable name=”x30” index=”28” value=”1”/>
  <variable name=”x40” index=”29” value=”1”/>
  <variable name=”x50” index=”30” value=”0”/>
  <variable name=”si11” index=”31” value=”0”/>
  <variable name=”si12” index=”32” value=”-0”/>
  <variable name=”si13” index=”33” value=”-0”/>
  <variable name=”si14” index=”34” value=”-0”/>
  <variable name=”si15” index=”35” value=”-0”/>
  <variable name=”si21” index=”36” value=”-0”/>
  <variable name=”si22” index=”37” value=”0”/>
  <variable name=”si23” index=”38” value=”-0”/>
  <variable name=”si24” index=”39” value=”-0”/>
  <variable name=”si25” index=”40” value=”-0”/>
  <variable name=”si31” index=”41” value=”-0”/>
  <variable name=”si32” index=”42” value=”-0”/>
  <variable name=”si33” index=”43” value=”0”/>
  <variable name=”si34” index=”44” value=”-0”/>
  <variable name=”si35” index=”45” value=”-0”/>
  <variable name=”si41” index=”46” value=”-0”/>
  <variable name=”si42” index=”47” value=”-0”/>
  <variable name=”si43” index=”48” value=”0”/>
  <variable name=”si44” index=”49” value=”0”/>
  <variable name=”si45” index=”50” value=”-0”/>
  <variable name=”si51” index=”51” value=”-0”/>
  <variable name=”si52” index=”52” value=”-0”/>
  <variable name=”si53” index=”53” value=”-0”/>
  <variable name=”si54” index=”54” value=”0”/>
  <variable name=”si55” index=”55” value=”0”/>
  <variable name=”y01” index=”56” value=”0.58699421209028879”/>
  <variable name=”y21” index=”57” value=”0”/>
  <variable name=”y31” index=”58” value=”0”/>
  <variable name=”y41” index=”59” value=”0”/>
  <variable name=”y51” index=”60” value=”0”/>
  <variable name=”y10” index=”61” value=”0”/>
  <variable name=”y12” index=”62” value=”0.2934971060451444”/>
  <variable name=”y13” index=”63” value=”0”/>
```

```
<variable name=”y14” index=”64” value=”0”/>
<variable name=”y15” index=”65” value=”0”/>
<variable name=”y02” index=”66” value=”0”/>
<variable name=”y32” index=”67” value=”0”/>
<variable name=”y42” index=”68” value=”0”/>
<variable name=”y52” index=”69” value=”0”/>
<variable name=”y20” index=”70” value=”0”/>
<variable name=”y23” index=”71” value=”0”/>
<variable name=”y24” index=”72” value=”0”/>
<variable name=”y25” index=”73” value=”0”/>
<variable name=”y03” index=”74” value=”0.2934971060451444”/>
<variable name=”y43” index=”75” value=”0”/>
<variable name=”y53” index=”76” value=”0”/>
<variable name=”y30” index=”77” value=”0”/>
<variable name=”y34” index=”78” value=”0”/>
<variable name=”y35” index=”79” value=”0”/>
<variable name=”y04” index=”80” value=”0”/>
<variable name=”y54” index=”81” value=”0.2934971060451444”/>
<variable name=”y40” index=”82” value=”0”/>
<variable name=”y45” index=”83” value=”0”/>
<variable name=”y05” index=”84” value=”0.5869421209028879”/>
<variable name=”y50” index=”85” value=”0”/>
<variable name=”t0” index=”86” value=”0”/>
<variable name=”t1” index=”87” value=”93”/>
<variable name=”t2” index=”88” value=”126”/>
<variable name=”t3” index=”89” value=”93”/>
<variable name=”t4” index=”90” value=”126”/>
<variable name=”t5” index=”91” value=”78”/>
<variable name=”a1” index=”92” value=”2093.0000000000014”/>
<variable name=”a2” index=”93” value=”2093.0000000000014”/>
<variable name=”a3” index=”94” value=”2093.0000000000014”/>
<variable name=”a4” index=”95” value=”196”/>
<variable name=”a5” index=”96” value=”0”/>
<variable name=”q10” index=”97” value=”0”/>
<variable name=”q20” index=”98” value=”0.089704464792737068”/>
<variable name=”q30” index=”99” value=”0.065536885371074868”/>
<variable name=”q40” index=”100” value=”0.072327832320818145”/>
<variable name=”q50” index=”101” value=”0”/>
<variable name=”q01” index=”102” value=”0.089704464792737068”/>
<variable name=”q21” index=”103” value=”0”/>
<variable name=”q31” index=”104” value=”0”/>
<variable name=”q41” index=”105” value=”0”/>
<variable name=”q51” index=”106” value=”0”/>
<variable name=”q02” index=”107” value=”0”/>
<variable name=”q12” index=”108” value=”0.089704464792737068”/>
<variable name=”q32” index=”109” value=”0”/>
<variable name=”q42” index=”110” value=”0”/>
<variable name=”q52” index=”111” value=”0”/>
<variable name=”q03” index=”112” value=”0.065536885371074868”/>
<variable name=”q13” index=”113” value=”0”/>
<variable name=”q23” index=”114” value=”0”/>
<variable name=”q43” index=”115” value=”0”/>
<variable name=”q53” index=”116” value=”0”/>
<variable name=”q04” index=”117” value=”0”/>
<variable name=”q14” index=”118” value=”0”/>
<variable name=”q24” index=”119” value=”0”/>
<variable name=”q34” index=”120” value=”0”/>
<variable name=”q54” index=”121” value=”0.072327832320818145”/>
<variable name=”q05” index=”122” value=”0.072327832320818145”/>
<variable name=”q15” index=”123” value=”0”/>
<variable name=”q25” index=”124” value=”0”/>
<variable name=”q35” index=”125” value=”0”/>
<variable name=”q45” index=”126” value=”0”/>
<variable name=”z1” index=”127” value=”0”/>
<variable name=”zeta1” index=”128” value=”0.089704464792737068”/>
<variable name=”z2” index=”129” value=”58.307902115279092”/>
<variable name=”zeta2” index=”130” value=”0.089704464792737068”/>
<variable name=”z3” index=”131” value=”42.598975491198665”/>
<variable name=”zeta3” index=”132” value=”0.065536885371074868”/>
<variable name=”z4” index=”133” value=”47.013091008531795”/>
<variable name=”zeta4” index=”134” value=”0.072327832320818145”/>
```

```
  <variable name=”z5” index=”135” value=”0”/>
  <variable name=”zeta5” index=”136” value=”0.072327832320818145”/>
  <variable name=”f0” index=”137” value=”0”/>
  <variable name=”f1” index=”138” value=”30.705403869137552”/>
  <variable name=”f2” index=”139” value=”39.491087688343534”/>
  <variable name=”f3” index=”140” value=”24.271635106149827”/>
  <variable name=”f4” index=”141” value=”33.914431235578569”/>
  <variable name=”f5” index=”142” value=”25.255066846127203”/>
  <variable name=”c” index=”143” value=”1514.791996861501”/>
 </variables>
</CPLEXSolution>

end}
```