# OPTIMIZATION OF THE DISTRIBUTION OF PARCEL AT DRY CLEANING SERVICES WITH MULTIPLE VEHICLES

## ASSIGNMENT 4: SOLVING A COMPLEX PROBLEM USING METAHEURISTICS

MODULE: Metaheuristics - Evolutionary and Bio-Inspired Algorithms

**PROGRAMME : Master Intelligence Artificielle et Robotique**

SUBMITTED BY:

SUNGKUR SEEKCHA,

FERIO RASAMBATA

MUNISH DAWOONAH

## Acknowledgement

We would like to thank Mr. Amic Seeven, senior lecturer at Université des Mascareignes for teaching us the basics and importance of metaheuristics, its uses, and his guidance throughout the whole course. The management of dry-cleaning services for placing faith and confidence that we could help them with their problems and our parents who is with us in every step guiding us and being supportive. Lastly of our colleagues and lecturers who provided their support to the best of their abilities.

# Table of Contents

# List of figures

## 1.0 Introduction

Dry cleaning services (DCS), a company built in the 1950s, which offers services like dry cleaning and stains removal for many big companies and hotels. Parcel delivery problems is an NP-hard problem, most precisely an integer programming problem, with one fix depot that is DCS, the initial location where the vehicles leave for delivery and come back with soiled linen. Usually, the task of route planning is done by the employees, which relies on their experience to get the job done. However, due to the Covid-19 pandemic, DCS had to let go of their long-time employees leaving the young ones to fend for their own. This situation has only worsened as the number of complaints from customers all around the country has escalated. As a fellow employee of DCS and with the help of my trusted friends, we wanted to help with the exasperated situation.

## 2.0 Problem

Although the company has 70 years of existence, there are debates about vehicle routing for deliveries and pickups almost every morning between the managers and route planners. It seems ironic to take routing as a problem as it is a repetitive task done daily, by the same people. Therefore, we thought why not instead of being planned by a person, we could make a program decide which routes to take as it will provide proofs that the routes that are being proposed are the best ones. This proof will be in terms of the distance covered and the number of vehicles dispatched.

# 3.0 Aims and objective

By using different algorithms, we could also determine which one is more efficient and if possible, we want the company to print the route proposed to give to the driver to minimize confusion. If successful, this software can be sold to the company, as Munish Dawoonah, who is an employee there discussed this possibility with the management. Later the system can be expanded and helps in a more crucial matter like the shortest path in an ambulance path system. The objective of the parcel delivery problem is to deliver a set of destinations in a short duration of time and returning to the depot.

# 4.0 Methodology

## 4.1 NP-Hard Problem of parcel delivery problem

NP stands for “nondeterministic polynomial time”. The origin of this word was defined according to nondeterministic machines, that is, machines that have more than one possible move from a given configuration. A problem is said to be NP-hard when it solves in polynomial time would make it possible to solve all problems in class NP in polynomial time.

The parcel delivery problem is classified as an NP-hard problem as the solution time normally increase a lot with size. The number of possible solutions to the parcel delivery problem is of the order of n!, where n is the number of nodes (locations the vehicle must reach) in the network. Given the factorial growth rate of possible solutions, and assuming a computer used for enumeration (parcel delivery problem) can process a billion computations per second, the total time is taken to solve the parcel delivery problem for different numbers of nodes using brute-force approach can be illustrated as follows (Analytics, 2019).

| Problem Size (Number of Nodes) | Approximate Solution Time |
|---|---|
| 10 | 3 milli-seconds |
| 20 | 77 years |
| 25 | 490 million years |
| 30 | $8.4 \times 10^{15}$ years |
| 50 | $9.6 \times 10^{47}$ years |

Aforementioned, the process is an enumeration process, where the solution time gets out of control with the increasing size of the locations.

## 4.1 Tabu Search

Tabu Search is a metaheuristic procedure for solving optimization problem designed to guide other methods to escape the trap of local minima. Tabu Search is used to find an optimal solution for a large range of problems. (Kaleia, M. 2018).

1. Initial Solution – the first node that is the depot is the starting and finished node. The list below is the tabu list. The starting node in the list looks for the nearest node. The experiment is repeated that is, the current node is looking for the nearest node and the previous node is removed from the list. The process ends when all the unvisited nodes are visited. (See Figure 1)

```
Initial Solution:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [2, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [2, 5, 6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [2, 6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [6, 7, 8, 9, 10, 11, 12]
Initial Solution:  [6, 8, 9, 10, 11, 12]
```

*Figure 1: Remove the visited node from Init_list for Tabu Search*

2. Neighborhood – The neighbor of the given solution is another provided solution obtained by the pair of wise exchanges of two nodes in the solution. In the situation, the fixed node 1 is at the beginning node and end node, therefore the N nodes problems are $^{N-1}C_2$. At each iteration, the neighborhood with the best minimum distance is selected. (See figure 2

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best neighbour:  1
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best neighbour:  3
[2, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best neighbour:  4
[2, 5, 6, 7, 8, 9, 10, 11, 12]
```

*Figure 2: Best neighbor for Tabu Search*

3. Tabu list - A tabu list consists of solutions that have changed by the process of moving from one solution to another. Here the attribute used is a pair of nodes that have recently be exchanged. The tabu list stores the number of iterations.

```
Selected Neighbor:  1
Tabu List [0, 1]
Selected Neighbor:  3
Tabu List [0, 3]
Selected Neighbor:  4
Tabu List [0, 4]
```

*Figure 3: Store in tabu list possibilities for Tabu Search*

4. Termination criteria: The algorithm terminates if a pre-specified number of iterations is reached in this case 10000 seconds.

```
ans = tabuSearch(distance_matrix, 10000, 100, 2, numVehicles)
```

```
Time for Tabu Search algorithm to run =  0.03812360763549805 s
Total route duration =  5959

Total route duration =  3985

Total route duration =  5536
```

*Figure 4: maxDuration as Termination criteria for Tabu Search*

## 4.2 Simulated Annealing

The solution illustrations and the algorithm used to determine the initial solution for Simulated Annealing (SA) are the same as the Tabu Search. The steps for SA in parcel delivery problem are as follows:

1. Neighborhood – Same as tabu search, the best neighbor, that is, minimum duration in the combination of the proposed list is selected.

```
Initial Solution for SA:  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best Neighbour:  1
Initial Solution for SA:  [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best Neighbour:  3
Initial Solution for SA:  [2, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Best Neighbour:  4
Initial Solution for SA:  [2, 5, 6, 7, 8, 9, 10, 11, 12]
Best Neighbour:  5
```

*Figure 5: Best neighbor for Simulated Annealing*

2. Termination criteria
   a) It reaches a maxDuration

```
ans = simulatedAnnealing(distance_matrix, 10000, 100, 6, numVehicles)
```

```
Time for Simulated Annealing algorithm to run =  0.027539491653442383 s
Optimal route = Total route distance =  89.012 kilometers
Total route duration =  1:39:19

Optimal route = Total route distance =  55.895 kilometers
Total route duration =  1:06:25

Optimal route = Total route distance =  90.608 kilometers
Total route duration =  1:32:16
```

*Figure 6: maxDuration as Termination criteria for Simulated Annealing*

   b) There is no improvement in the solution list.

```
Initial Solution for SA:  [8, 9, 10, 11, 12]
Best Neighbour:  9
Initial Solution for SA:  [8, 9, 10, 11, 12]
Best Neighbour:  9
Initial Solution for SA:  [8, 9, 10, 11, 12]
Best Neighbour:  9
Initial Solution for SA:  [8, 9, 10, 11, 12]
Best Neighbour:  9
Initial Solution for SA:  [8, 9, 10, 11, 12]
Best Neighbour:  9
```

*Figure 7: Solution list without improvement*

## 4.3 Solution

Given a set of vehicles and a set of locations, and assuming a fixed cost (time duration) of traversing any location-location pair, to find a path that reaches all the location at a minimum cost. (Analytics, 2019). In this case, the depot (initial point) is DNS and the number of locations was 12 as it took a lot of time to manually create a database (JSON format) with the name of the location, longitude-latitude, city name, and id (see JSON file in Annex). Also, the maximum number of vehicles to be used is 6, result as follows for 6 vehicles.

<u>Tabu search</u>

```
Time for Tabu Search algorithm to run =  0.020912647247314453 s
Optimal route = DCS ---> Voila Hotel ---> Intercontinental ---> DCS
Optimal route = DCS ---> Lagoon Attitude Hotel ---> Sofitel  ---> DCS
Optimal route = DCS ---> ClubMed  ---> SSR International Airport ---> Moka E.N.T Hospital ---> DCS
Optimal route = DCS ---> LUX Grand Gaube Resort & Villas ---> DCS
Optimal route = DCS ---> One&Only Le Saint GÃ©ran ---> DCS
Optimal route = DCS ---> Holiday inn ---> Dr AG Jeetoo Hospital ---> DCS
```

*Figure 8: Tabu Search result: 6 vehicles*

Simulated Annealing

```
Time for Simulated Annealing algorithm to run =  0.030950069427490234 s
Optimal route = DCS ---> Voila Hotel ---> ClubMed  ---> DCS
Optimal route = DCS ---> Lagoon Attitude Hotel ---> Intercontinental ---> DCS
DCS ---> LUX Grand Gaube Resort & Villas ---> DCS
DCS ---> One&Only Le Saint GÃ©ran ---> DCS
DCS ---> Holiday inn ---> Dr AG Jeetoo Hospital ---> DCS
DCS ---> ClubMed  ---> SSR International Airport ---> Moka E.N.T Hospital ---> DCS

Process finished with exit code 0
```

*Figure 9: Tabu Search result: 6 vehicles*

The distance matrix was obtained by using real-time data. The time and distance were obtained from google API. The route duration is modified every time the code runs as the API takes into consideration the fastest route rather than the distance it has to cover. The following figure is the method of calling the API. (See figure 10)

```
# Calling the distance matrix API
gmaps = googlemaps.Client(key='AIzaSyAEw2KJbW6FOzY5T2TkUpkAXfAqG9jfL68')
```

```
for i in range(0, len(locs['locations'])):
    for j in range(0, len(locs['locations'])):
        # adding Json data obtained in get_distance_between_coords in distance_matrix
        distance_matrix[i][j] = get_distance_between_coords(locs['locations'][i]['coords'],
                                                            locs['locations'][j]['coords'])
```

*Figure 10: Distance matrix API*

## 5.0 Results and Discussion

The algorithm for SA and Tabu search consists of strategies described above which were implemented in python. The results obtained for both implementation in the parcel delivery problem show that they provide a good solution. The solution mostly remains the same as it depends on the database which remains the same and despite the duration time changes, it does not change significantly to change the results. However, with more location in the database, the results would be more accurate.

The code was tested with a maximum number of 6 vehicles and 11 cities. (See figure 11)

```
Maximum number of vehicle = 6
VEHICLES NUMBERS : 6
```

*Figure 11: Number of vehicles*

For Tabu Search, the six-route obtained was as follows. (See figure 12). The results show the time the code was run, how long it took to run, the vehicle routes with its distance and duration.

```
PROCESSING TIMES :
 |>Begin at  =  18:00:23
 |>End at    =  18:01:23
 |>Time for Tabu Search algorithm to run =  0:01:00

VEHICLE 1 :
 |>Optimal route = DCS ---> Voila Hotel ---> Intercontinental ---> DCS
 |>Total route distance =  46.355 kilometers
 |>Total route duration =  0:58:19


VEHICLE 2 :
 |>Optimal route = DCS ---> Lagoon Attitude Hotel ---> Sofitel  ---> DCS
 |>Total route distance =  105.005 kilometers
 |>Total route duration =  2:02:46


VEHICLE 3 :
 |>Optimal route = DCS ---> ClubMed  ---> SSR International Airport ---> Moka E.N.T Hospital ---> DCS
 |>Total route distance =  143.504 kilometers
 |>Total route duration =  2:20:52


VEHICLE 4 :
 |>Optimal route = DCS ---> LUX Grand Gaube Resort & Villas ---> DCS
 |>Total route distance =  55.268 kilometers
 |>Total route duration =  0:59:08


VEHICLE 5 :
 |>Optimal route = DCS ---> One&Only Le Saint GÃ©ran ---> DCS
 |>Total route distance =  64.658 kilometers
 |>Total route duration =  1:14:10


VEHICLE 6 :
 |>Optimal route = DCS ---> Holiday inn ---> Dr AG Jeetoo Hospital ---> DCS
 |>Total route distance =  44.619 kilometers
 |>Total route duration =  0:54:19
```

*Figure 12: Results of the Tabu search*

For SA, the six-route obtained was as follows. (See figure 13). The results show the time the code was run, how long it took to run, the vehicle routes with its distance and duration.

```
PROCESSING TIMES :
 |>Begin at = 18:01:23
 |>End at =  18:02:22
 |>Time for Simulated Annealing algorithm to run =  0:00:59

VEHICLE 1 :
 |>Optimal route = DCS ---> Voila Hotel ---> Intercontinental ---> DCS
 |>Total route distance =  46.355 kilometers
 |>Total route duration =  0:58:19


VEHICLE 2 :
 |>Optimal route = DCS ---> Lagoon Attitude Hotel ---> Sofitel  ---> DCS
 |>Total route distance =  105.005 kilometers
 |>Total route duration =  2:02:46


VEHICLE 3 :
 |>Optimal route = DCS ---> ClubMed  ---> SSR International Airport ---> Moka E.N.T Hospital ---> DCS
 |>Total route distance =  143.504 kilometers
 |>Total route duration =  2:20:52


VEHICLE 4 :
 |>Optimal route = DCS ---> LUX Grand Gaube Resort & Villas ---> DCS
 |>Total route distance =  55.268 kilometers
 |>Total route duration =  0:59:08


VEHICLE 5 :
 |>Optimal route = DCS ---> One&Only Le Saint GÃ©ran ---> DCS
 |>Total route distance =  64.658 kilometers
 |>Total route duration =  1:14:10


VEHICLE 6 :
 |>Optimal route = DCS ---> Holiday inn ---> Dr AG Jeetoo Hospital ---> DCS
 |>Total route distance =  44.619 kilometers
 |>Total route duration =  0:54:19
```

*Figure 13: Results of SA*

The comparison of the algorithm of both the algorithm was done in terms of time the code was run 9 times in a loop, to show the time it takes to run. (See figure 14) Since the distance matrix stays the same, the amount of the processing of the data remains the same but not between the two algorithms. The two algorithms have the same principle with the minimum time duration and since there are only 11 cities, the results of the code are almost the same. However, the Tabu search takes more time to run than simulated Annealing.

```
Comparing the running time for algorithm

Number of time the function ran:  1
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.2879362106323242 1875 s

Number of time the function ran:  2
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  3
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  4
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  5
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  6
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  7
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  8
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s

Number of time the function ran:  9
Time for Tabu Search algorithm to run =  60.68928933143616 s
Time for Simulated Annealing algorithm to run =  59.287936210632324 21875 s
```

*Figure 14: Results of comparison*

## 6.0 Future work

For future work, an implementation of a pick-up system can also be implemented. An interface either mobile or web version can be added to improve the user interface with a google map where the user can pin several destinations, the vehicles will have to visit on a specific day. The pin data will be store in the database and run with the code. Also, an estimation of unloading and loading the parcel time can be added to avoid overtime or miss deadlines. There providing better services to the customers which is the focus of DCS.

# 7.0 Reference

Analytics, O. (2019). Vehicle Routing Problems 101. [online] Medium. Available at: https://medium.com/opex-analytics/opex-101-vehicle-routing-problems-262a173f4214#:~:text=In%20mathematical%20terms%2C%20the%20VRP [Accessed 25 Oct. 2020].

Kaleia, M. (2018). Tabu Search — gentle introduction. [online] Medium. Available at: https://medium.com/@ms.kaleia/tabu-search-gentle-introduction-46c479eb6525 [Accessed 25 Oct. 2020].

# ANNEX

## JSON File


```
{
   "locations":[
      {
         "name":"DCS",
         "coords":[
            -20.16782,
            57.48314
         ],
         "city":"Port Louis",
         "id":"14"
      },
      {
         "name":"Voila Hotel",
         "coords":[
            -20.22442,
            57.49452
         ],
         "city":"Bagatelle",
         "id":"1"
      },
      {
         "name":"Intercontinental",
         "coords":[
            -20.08292,
            57.517429
         ],
         "city":"Balaclava",
         "id":"2"
      },
      {
         "name":"Lagoon Attitude Hotel",
         "coords":[
            -20.01511,
            57.58171
         ],
         "city":"Anse La Raie",
         "id":"3"
      },
      {
         "name":"ClubMed ",
         "coords":[
            -20.00447,
            57.56293
         ],
         "city":"Pointe aux Canonniers",
         "id":"4"
      },
      {
         "name":"LUX Grand Gaube Resort & Villas",
         "coords":[
            -20.02255,
            57.65055
```

```
      ],
      "city":"Grand Gaube",
      "id":"6"
   },
   {
      "name":"One&Only Le Saint Géran",
      "coords":[
         -20.0924,
         57.69941
      ],
      "city":"Pointe de Flacq",
      "id":"7"
   },
   {
      "name":"Holiday inn",
      "coords":[
         -20.2833,
         57.549999
      ],
      "city":"Plaine Magnien ",
      "id":"8"
   },
   {
      "name":"Sofitel ",
      "coords":[
         -20.27865,
         57.37215
      ],
      "city":"Flic en Flac ",
      "id":"9"
   },
   {
      "name":"The St. Regis Mauritius Resort",
      "coords":[
         -20.43979,
         57.322
      ],
      "city":"Le Morne",
      "id":"10"
   },
   {
      "name":"SSR International Airport",
      "coords":[
         -20.429712,
         57.66075
      ],
      "city":"Plaine Magnien ",
      "id":"11"
   },
   {
      "name":"Dr AG Jeetoo Hospital",
      "coords":[
         -20.17258,
         57.505
      ],
      "city":"Port Louis",
```

```
            "id":"12"
        },
        {
            "name":"Moka E.N.T Hospital",
            "coords":[
                -20.21743,
                57.49222
            ],
            "city":"Moka",
            "id":"13"
        }
    ]
}
```