IT 3708: Project 1

Solving The Multiple Depots Vehicle Routing Problem (MDVRP) Using Genetic Algorithm (GA)

Lab Goals

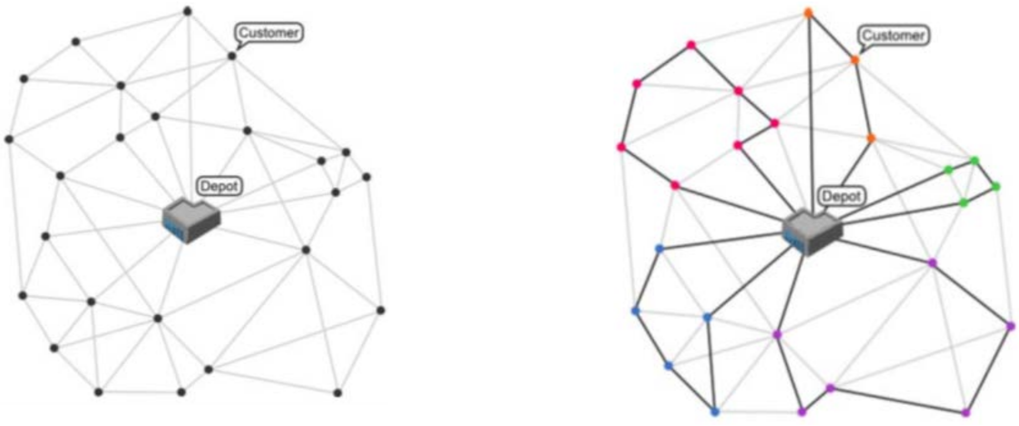
- Implement genetic algorithm (GA) to solve a NP-hard combinatorial optimization problem ─ The Multiple Depots Vehicle Routing Problem (MDVRP).
- Compare the performance of your solutions on several benchmark problems.
- Test and analyze effects of genetic operators and related parameters.

Groups Allowed? Yes. For this project, you may work alone or in groups of two. However, even though, you worked in a group, **you must attend the demo individually as well as submit the report individually**.

Deadline: February 15, 2018 (Thursday) at 08: 00 AM.

Assignment Details

Vehicle routing problems (VRPs) are classical combinatorial optimization problems which have received much attention in recent years due to their wide applicability and economic importance. VRP formulations are used to model an extremely broad range of issues in many application fields, transportation, supply chain management, production planning, and telecommunication, to name but a few. A typical VRP can be stated as follows: a set of geographically dispersed customers with known demands are to be serviced by a homogenous fleet of vehicles with limited capacity. Each customer is to be fully serviced exactly once and each vehicle is assumed to start and end at the same depot, and the **primary objective is to minimize the total distance travelled by all vehicles**. However, in a large number of practical situations and to satisfy real-life scenarios, additional constraints are usually defined for variants of the VRP.



(a) (b)

Fig. 1: A hypothetical instance of a VRP (a) and its solution (b)

In this project, you need to solve the multi-depot VRP (MDVRP), which is an extension of the classical VRP with the exception that there are multiple depots with a number of vehicles and every customer should be serviced by a vehicle based at one of several depots. Similar to the typical VRP, in the MDVRP every vehicle route also must start and end at the same depot. The MDVRP is NP-hard, which means that an efficient algorithm for solving the problem to optimality is unavailable. Therefore, solving the MDVRP by traditional algorithms is time-consuming and computationally intractable. For this reason, heuristic/meta-heuristic algorithms are good choices to solve the MDVRP. **In this project, you will solve the MDVRP using a well-known bio-inspired algorithm (also meta-heuristic), called the genetic algorithm (GA)**.

Problem Formulation:

The MDVRP describes the vehicle scheduling challenge for a transportation company. The transportation company has multiple depots from which their vehicles depart and arrive, and has multiple customers being served from the different depots. **The challenge is to make a schedule for each vehicle individually** so that the vehicles drive in the most efficient way optimizing one or several objectives.

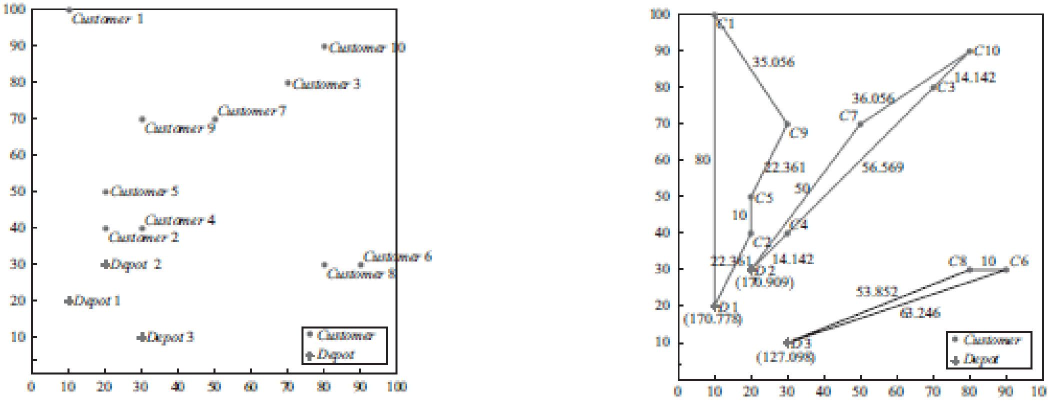
Formally, the MDVRP can be defined as follows. We are given a set of depot locations and a set of customer locations, which are assumed to be disjoint (even if two points share the same physical coordinates, they are still handled as different entities). Each customer is characterized by their own demand. A fleet of vehicles with limited capacity is based at each depot. Each vehicle originates from one depot, service the customers assigned to that depot, and returns to the same depot. The MDVRP consists of determining the routes for multiple depots with multiple vehicles per depot in parallel. Further, each depot has a set of customers. The route should also optimize predefined objective(s) as well as satisfying the following conditions:

(i) every customer appears on exactly one route.

(ii) every route starts and ends at the originating depot.

(iii) capacity limit: the total demand of the customers on any route does not exceed a vehicle's capacity.

(iv) route limit: the total duration of a route does not exceed a preset value *(for this project, it is only for those problems for which this value is mentioned in the test data)*.



(a)

(b)

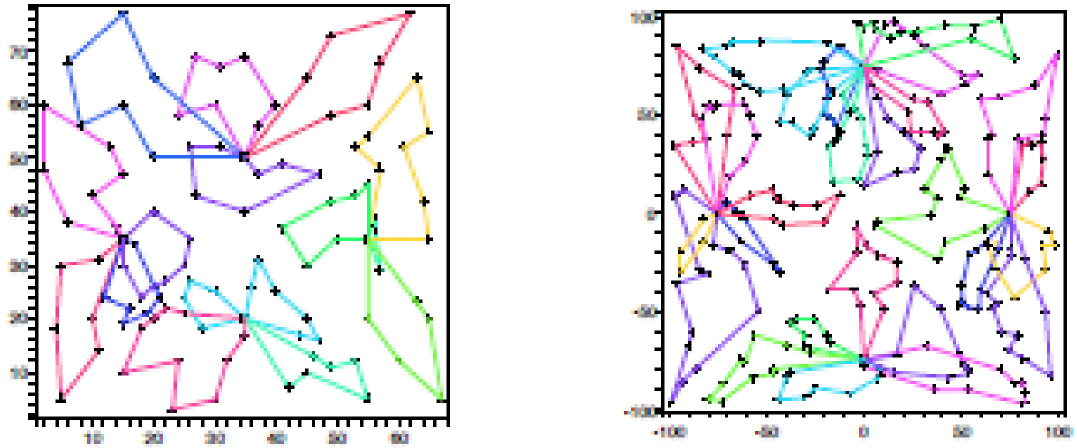
Fig. 2: A hypothetical instance of a MDVRP: (a) locations of customers and depots, (b) one solution

Fig. 2 presents an example for MDVRP with an arbitrary solution. Generally, the objectives of the MDVRP are to minimize the total combined route duration for all required vehicles from all depots, to minimize the time spent in serving all customers, or to minimize the number of vehicles needed in serving all customers. In this project, **your main goal is to minimize the total distance travelled by all vehicles across the depots to serve all customers**.

Algorithm

As mentioned earlier, to solve the MDVRP, you need to implement the genetic algorithm (GA) that has been already discussed in lectures. In order to get the optimal/near-optimal results, you may check several forms of representation, genetic operators, and selection mechanism. It would be beneficial to test whether *elitism* gives a better solution.

Note that, GA parameter values (population size, generation number, crossover rate, and mutation rate) are correlated and your GA will successfully find the optimal values if you use appropriate parameter values. However, there is no definite rule to find the appropriate parameter values. Therefore, you should test different set of parameter values to decide the appropriate values.



(a) Solution for a test problem with 100 customers, depot no. 2,3,4 have the capacity limit of 100

(b) Solution for a test problem with 249 customers, 4 depots with capacity limit of 500 and route limit of 310

Fig. 3: Solution representation

Things To Do

The 15 points total for this project is 15 of the 100 points available for this course. The 15 points will be distributed on two parts: (i) demo (09 points) and (ii) report (06 points). To test your code, we uploaded several benchmark test data and their solutions. The description of problem and solution data file formats is also included. For the demo session, **your code must have the option to read the test data according to the given format**. Note that, your implementation must produce the solution strictly following the graphical (Fig. 3) and the text format (Fig. 4, also mentioned in the provided solution data files). **You must use different colours to represent different routes from a single depot** as shown in Fig. 3.

(a) Demo (09p):

There will be a demo session where you will show us the running code and we will verify that it works. In the demo session, you need to describe how you designed and implemented your GA. Also, you have to test you code by running 04 (four) test problems that you will be supplied during the demo (3 test problems to check the performance of your implementation + 1 test problems to explain your implementation).

```
576.87
1  1  60.06  71  0 44 45 33 15 37 17 0
1  2  66.55  79  0 42 19 40 41 13 0
1  3  47.00  78  0 25 18 4 0
2  1  53.44  73  0 6 27 1 32 11 46 0
2  2  79.47  80  0 48 8 26 31 28 22 0
2  3  81.40  77  0 23 7 43 24 14 0
2  4  23.50  54  0 12 47 0
3  1  50.41  75  0 9 34 30 39 10 0
3  2  25.22  54  0 49 5 38 0
4  1  47.67  67  0 35 36 3 20 0
4  2  42.14  69  0 21 50 16 2 29 0
```

Fig. 4: Expected text format for your solution

The point distribution for the demo is as follows:

(1) Testing 3 test problems (7.5p = 2.5p x 3):

- Does your solution find the known optimal value for travel-distance value considering all vehicles across the depots in serving all customers? (2.5p)
    - If your value is within 5% of the known optimal value, you will get full points.
    - If your value is within 10% of the known optimal value, you will get 2 points.
    - If your value is within 20% of the known optimal value, you will get 1.5 point.
    - If your value is within 30% of the known optimal value, you will get 1 point.
    - Otherwise, you will get 0.

- **You can only get a maximum of 1 point per test if your output is not in the correct format as described (graphically as well as in text format).**

(2) Individual demo and explanation (1.5p):

- You will be provided with an extra test data that needs to be run using your submitted code. You can choose to make a couple of changes (or can be asked to change) to the run to see what effect(s) the change has on the score (the total distance travelled by all vehicles). If there is an improvement in the score, you need to be ready to explain the reason behind this improvement. If not why not.

(b) Report (06p):

You should write a report answering the points below. Your report must not exceed 02 (two) pages in total. **Over length reports will result in points being deducted from your final score**. Print on both sides of the sheet, preferably. **Bring a hard copy of your report to the demo session**.

1. Describe the Chromosome representation that you used in your implementation. Also, mention another representation that could be used for this problem and why this representation is also suitable. Defend your choice of the most suitable representation. (2p)
2. Describe whether the crossover and mutation operators will produce infeasible off-spring(s) after executing. If yes, how did you handle that? If not, why not? (2p)
3. In GA, often the parameter values (population size, generation number, crossover rate, and mutation rate) have some form of relationship. Considering that you tested several set of parameter values, what relationship did you observe among these parameter values? Also, describe the effects of these parameter values during the early and later stages of evolutionary cycle? (2p)

Delivery

You should deliver your report + a zip file of your code on *BlackBoard*. The submission system will be closed at 08:00 AM on February 15, 2018. **If you worked in a group, both you should submit individually,** mentioning your group member's name in the report (precisely as your group member). **You must attend the demo individually on the scheduled demo date,** which have been already declared on *BlackBoard***. No early or late demo will be entertained** except for extreme emergency.

NB: Every question regarding this project will **only be handled** through the Slack channel.