IT 3708 2019 : Project 1

Solving a variant of the Multiple Depots Vehicle Routing Problem (MDVRP) Using a Genetic Algorithm (GA)

Lab Goals

- Implement a genetic algorithm (GA) to solve an 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 the effects of the genetic operators and related parameters.

Groups Allowed? Groups are allowed for the **implementation phase**, max 2 persons. **Every student** must attend the demo day. The members of a group should sign up for the same time slot. **Every student** must write and submit their own **independent report**, providing their individual answers to the project questions in part b.

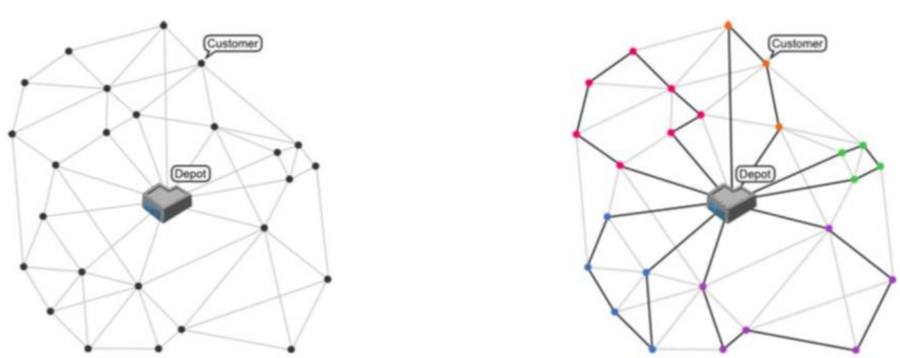
Deadline: February 11, 2019 (Monday) at 08: 00 AM.

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. In a large number of practical situations and to satisfy real-life scenarios, additional / revised constraints are defined for variants of the VRP.

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.
- There are a number of vehicles assigned to the depot
- A vehicle starts and has to return to the depot
- The primary objective is to minimize/maximise some goal, such as, the total distance travelled by all vehicles.



(a) (b)

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

In this project, you need to solve a variant of the multi-depot VRP (MDVRP):

- Multiple depots
- There are a number of vehicles assigned to each depot
- Each vehicle starts at its assigned depot and returns to **either the same or a different depot**.
- Every customer should be serviced by a vehicle based at one of several depots.

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, services the customers assigned to that depot, and returns to the same depot or a different depot.

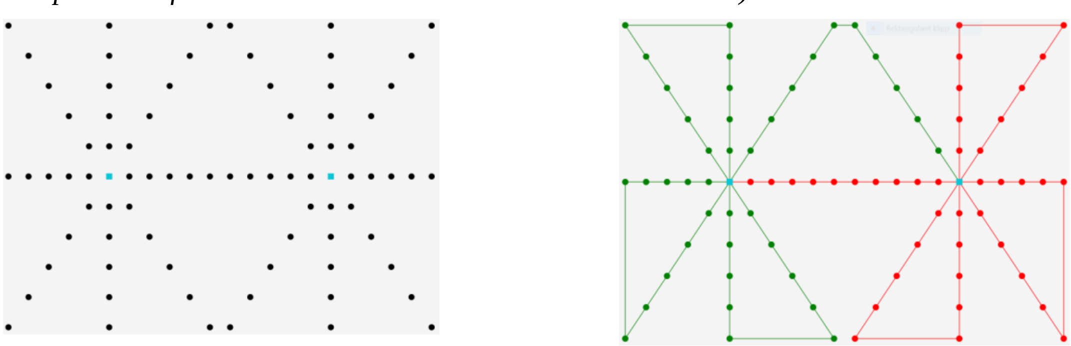
The MDVRP consists of determining the routes for multiple depots with multiple vehicles per depot in parallel. Further, each depot has a set of customers. The route should also optimize predefined objective(s) as well as satisfying the following conditions:

(i) every customer appears on exactly one route.

(ii) every route starts and ends at a depot (either the originating or another 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)

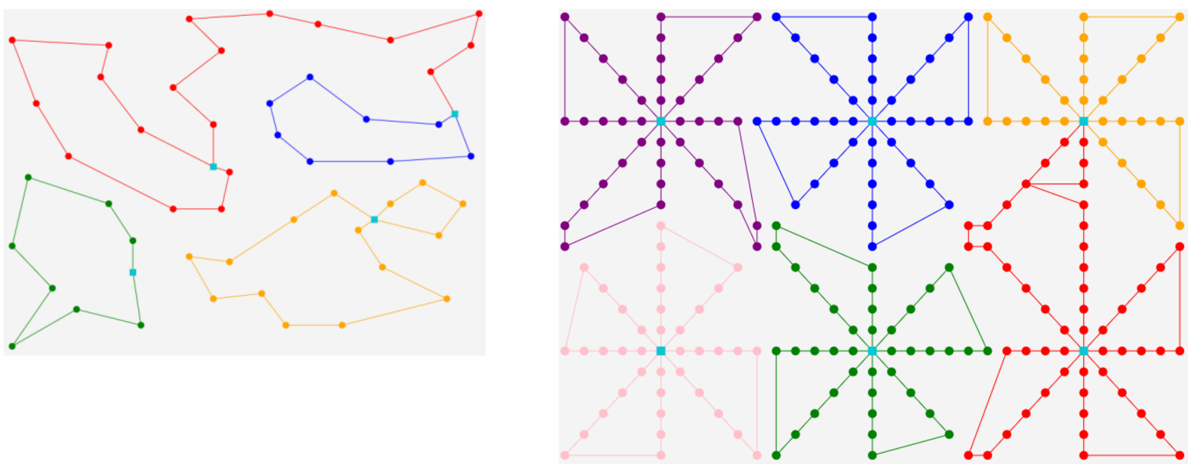
Fig. 2: A hypothetical instance of a MDVRP: (a) locations of customers and depots, customers as black dots and depots as blue rectangles, (b) one solution, color indicating routes belonging to one depot

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) as presented in the lectures. In order to get the optimal/near-optimal results, you may check several forms of representation, genetic operators, and selection mechanisms. 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 as to how to find the appropriate parameter values. Therefore, you should test different sets of parameter values to decide the appropriate values. Study closely the independence between the parameter values in your chosen implementation.



(a) Solution for a test problem with 50 customers, depot no. 1, 2, 3, 4 have a route limit of 160

(b) Solution for a test problem with 240 customers, 6 depots with capacity limit of 60 and route limit of 180

Fig. 3: Solution representation

Things To Do

The 20 points total for this project is 20 of the 100 points available for this course. The 20 points will be distributed on two parts: (i) demo (11 points) and (ii) report (09 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 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). **Each depot should have a unique color for its routes,** as shown in Fig. 3.

When validating how good a solution distance is, refer to the **benchmark table** supplied in the test data. The benchmarks are distances that should be obtainable, while the solution files are the best optimal values found that can be very hard to achieve.

It is not recommended to implement this algorithm in Python, as it requires too much computation and Python runs too slow, compared to other languages e.g. Java.

(a) Demo (11p):

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

Fig. 4: Expected text format for your solution

The point distribution for the demo is as follows:

1. **Group:** Testing 3 test problems (9 p = 3p x 3):
   * Does your solution find the benchmark value for travel-distance value considering all vehicles across the depots in serving all customers? (3p)
     - If your value is within 5% of the benchmark value, you will get full points.
     - If your value is within 10% of the benchmark value, you will get 2,5 points.
     - If your value is within 20% of the benchmark value, you will get 2 points.
     - If your value is within 30% of the benchmark value, you will get 1,5 points.
     - 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 work:** demo and explanation (2p):
   - You will be asked to make a couple of changes 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 (09p): Individual work:

You should write a report answering the points below. Your report must not exceed 2 (two) pages in total. **Over length reports will result in points being deducted from your final score**.

1. Describe the Chromosome representation that you used in your implementation. Also, describe another representation that could be used for this problem and why this representation is also suitable. Discuss the advantages and disadvantages of these two representations for this problem and defend which representation you believe to be better suited and why. (3p)
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.
   Select two of these parameters to compare.
   Run simple tests to provide comparison data and describe what the data illustrates wrt the relationship between these parameters. A tabular presentation of data to highlight the points that you wish to draw is sufficient. Graphs are not required, although you are welcome to use them. (2p)
   Describe the effects of these parameter values during the early and later stages of evolutionary cycle? Again use your data to illustrate your points. (2p)

   Questions regarding this assignment will ONLY be answered at the lab hours or through the Blackboard Forum.

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 11, 2019**.

The Demoday is also February 11. A signup schedule will be announced one week before. Please follow Blackboard for details.

Every student must submit both their code and their individual report. Ensure that your report provides the name of your partner, where relevant**. You must attend the demo individually on the scheduled demo date. No early or late submission or demo will be entertained** except in an emergency.