

Genetic Clustering for the Multi-depot Vehicle Routing Problem

Libor Novák

Abstract—In this project, the multi depot vehicle routing problem (MDVRP) is examined. The task is split into two subsidiary tasks. First, the customers are clustered in order to determine the depot from which each customer will be served. Then, the classical vehicle routing problem (VRP) is applied and solved for each of the clusters. The goal of this project is to suggest a new clustering method based on genetic algorithms for solving the first part of the MDVRP task.

I. ASSIGNMENT

One way to approach the Multi Depot Vehicle Routing Problem (MDVRP) is to employ two steps. First, the customers are clustered - assigned to each depot and, subsequently, for each cluster the classical Vehicle Routing Problem (VRP) is solved. Because the cluster centers are in this case given by the positions of the depots, all regular clustering methods yield the same solutions, which does not have to be ideal for the final solution of this task. Suggest, implement and test a new clustering method specifically for the MDVRP using genetic algorithms.

II. INTRODUCTION

The vehicle routing problem (VRP) is a task, where, given a set of customers C and a depot, we are asked to determine the optimal routes of trucks to serve these customers and return to the depot. The goal is to minimize the cost of travel in terms of distance traveled, while meeting other constraints. The multi depot vehicle routing problem (MDVRP) is an extension of the basic problem, where we consider the customers to be served from several available depots $d_m \in D$. The positions of the customers $c_n \in C$ and the positions of the depots $d_m \in D$ are specified as well as the routes and the lengths $r_{ij}; i, j \in C \cup D$ between the customers and the depots.

In this semestral work we will address the problem of clustering the customers. The VRP for each depot is simplified to the Traveling Salesman Problem (TSP) since we do not consider maximum capacity of each truck. Therefore one truck is considered for each depot. The only constraint we consider is the maximum supply of each depot, excluding this constraint would lead to always having one single cluster of all customers.

The importance of the used clustering method is obvious. Once we create the clusters, the VRP can only solve the subtasks within those clusters, therefore optimal clustering of the customers is crucial to obtain good results of the MDVRP. In most cases, the customers are assigned to the closest depot (given Euclidean distance) available as in Surekha and Sumathi [1], which is nothing more than K-Means clustering. This is possible because the distances between the customers c_n and the depots d_m are given by their Euclidean distance on a 2D map

$$\text{dst}(c_i, d_j) = \|p(c_i) - p(d_j)\|,$$

where $p(\cdot)$ denotes the position on the map and the map can be represented as a complete undirected graph. Nevertheless, in this clustering approach the total demand of the customers in each cluster (the workload of each depot) is not considered, neither are the distances between the customers themselves.

In this semestral project, we exploit the adaptability of genetic algorithms and suggest a new clustering method for the MDVRP. In the next section, the optimization problem is stated and the genetic algorithm described, followed by comparison of obtained results in the subsequent section, followed by discussion.

III. PROBLEM STATEMENT AND SOLUTION

A. Problem Statement

We are solving a special case of the MDVRP, where we consider only one truck to be launched from each depot. Therefore for the set of trucks T it holds that $|T| = |D|$, i.e. the number of trucks is the same as the number of depots. The goal is to minimize the total route length, which can be stated as

$$\begin{aligned} \min \quad & \sum_{i=1}^{|T|} r_i, \\ \text{s.t.} \quad & \text{dem}(d_i) \leq \text{Thr} \quad \forall i = 1, \dots, |D|, \end{aligned}$$

where r_i is the length of the route of the truck i and $\text{dem}(d_i)$ is the demand covered by the depot d_i , which has to be lower or equal to the maximum depot coverage Thr .

The MDVRP is a NP hard problem, therefore several heuristics are used in order to make this task solvable in a reasonable time frame. The usual heuristic used to approach MDVRP is to split the customers $C = C_1 \cup C_2 \cup \dots \cup C_{|D|}$ into several sets - clusters $d_m \in D$, which are assigned to depots (each set of customers $C_m \subset C$ is served by exactly one depot d_m). This simplifies the problem to $|D|$ VRPs, which are then solvable either using direct tree search methods, linear programming, Clarke and Wright [2] method or other, as described by Laporte [3]. In this semestral work we use the Clarke and Wright method.

The clustering itself is of a special kind because we are not seeking the optimal positions of the cluster centers as in the classical clustering task, but the cluster centers are given by the depot positions $p(d_m)$. We seek the optimal assignment of the customers $c_n \in C$ to the depots $d_m \in D$. This difference is however crucial, the most commonly used K-means, EM and graph clustering methods [4] assign the customers always to the closest center, therefore yielding the same results. These approaches can be disadvantageous when the depots are not evenly distributed over the map.

B. Solution

Since we would like to examine the possibility of creating different kinds of clusters, we will make use of the flexibility of genetic algorithms, i.e. their fitness functions, which allows us to specify different clustering objectives than just distance to the depot (cluster center). Several versions and ideas can be found in [5], [6].

For our experiments, we used an altered version of the general genetic algorithm, as described in Figure 1.

a) Representation: Each individual in the population is an assignment of the customers to the depots, that is its chromosome is for example $[1, 4, 2, 3, 3, 2, 4]$, where the position (locus) is the customer index and the number at that position (gene) is the depot index to which the customer is assigned.

```

Randomly initialize population.
Repair the individuals in the population.
For  $g = 1, \dots, G$ :
    • Compute fitness of all individuals.
    • Elitism.
    • Tournament selection.
    • Uniform crossover.
    • Mutation.
    • Repair the individuals.
    • Replace the whole population.
Output the best individual.
```

Fig. 1: Genetic algorithm for clustering, pseudo code.

b) Initialization: The population is initialized randomly with equiprobable distribution.

c) Elitism: From each population, we directly select the individual with the best fitness for mating.

d) Selection: Tournament selection is used in our implementation, where the best out of 5 randomly selected individuals is chosen as a future parent.

e) Crossover: Each pair of parents has a probability of undergoing crossover. We use uniform crossover - for each locus we determine with probability 0.5 whether the gene of the individuals will be kept or swapped. Both children are then added to the new population.

f) Mutation: After a new population is generated by selection and crossover, the individuals go through mutation with certain probability. We use a simple bit flip, which means that each locus has a certain probability to experience mutation and if mutation is carried out for the given locus, then we randomly select one depot (gene) to be put in that position.

g) Replacement: The whole population is then replaced with the newly generated individuals - generational replacement.

One can notice that there is an extra step in the pseudo code (Figure 1) for repairing the population. This step assures that all individuals in the population meet the condition on the maximum capacity of the depots. We created a function that carries out the fixing, which is described in Figure 2.

C. Fitness Computation - Exclusive Pairing

We propose a new fitness function that uses nearly exclusive pairing of the customers assigned to each depot. The motivation behind it is to assimilate the routing process without actually solving it. Instead of taking into account the distance of the customers to

```

Given:  $D = \{d_1, \dots, d_m\}$ ,  $C_i = \{c_{i1}, \dots, c_{in_i}\}, max\_demand$ 
Initialize  $D_e = \{\text{all depots exceeding their capacity}\}$ .
While  $notempty(D_e)$ :
    • Take a depot  $i$  from  $D_e$ .
    • Random shuffle customers  $C_i$ .
    • While  $demand(d_i) > max\_demand$ 
        ◦ For  $j = 1, \dots, n_i$ :
            ▪ Try to place the customer  $c_{ij}$  into one of the depots from  $D \setminus D_e$ .
            ▪ If it is possible, move it, otherwise leave it in the depot  $d_i$ .
    • Remove  $d_i$  from  $D_e$ .  $D_e = D_e \setminus \{d_i\}$ .
```

Fig. 2: Individual fixing process.

Given: $D = \{d_1, \dots, d_m\}$, $C_i = \{c_{i1}, \dots, c_{in_i}\}$
Initialize $fitness = 0$.
For $i = 1, \dots, m$:

- Random shuffle customers C_i .
- Set $CM_i = \emptyset$.
- For $j = 1, \dots, n_i$:
 - Find the closest customer c_{ic} to the customer c_{ij} from the set $C_i \setminus CM_i$.
 - Add the distance $d(c_{ic}, c_{ij})$ to the total fitness function
$$fitness = fitness + d(c_{ic}, c_{ij}).$$
- Set the current customer as already paired $CM_i = CM_i \cup \{c_{ij}\}$.

Output $fitness$.

Fig. 3: Exclusive Pairing fitness computation pseudo code.

the cluster center, we use the distance between the customers themselves.

We pair the customers in each cluster according to their distance (closest ones are a pair), however once we pair a customer with another one, the latter cannot select the former as its pair anymore, therefore perhaps the second nearest must be selected as the paired one.

Let us have one individual, denote C_i the set of customers assigned to the depot d_i . For simplicity we will describe only individual in one generation and therefore leave out the subscripts of the generation and indices denoting the individual. Denote CM_i the set of already paired customers served by the depot d_i , which is empty in the beginning. For each depot we randomly order the customers. Then we take one by one customer c_{ij} and always select the closest neighbor (customer) from the set of not yet paired customers $C_i \setminus CM_i$. Add their distance to the total fitness function and add the customer c_{ij} to CM_i . This is done for all customers and all depots. Pseudo code is shown in Figure 3.

We then minimize this fitness function. It can be noticed that the exclusivity of the pairing in the fitness function is questionable because there can be more customers, which have the same nearest neighbor. Here we exclude only mutually nearest neighbors, which as an approximation works well.

Because the value of the fitness function actually depends on the order in which the customers are paired, we carry out this computation several times with different random orderings of the customers, which gives us a higher chance of finding a value, which is closer to the optimum. This method can be understood as a variant of RANSAC [7].

IV. EXPERIMENTS

For testing we used the set of Cordeau's instances (<http://www.bernabe.dorronsoro.es/vrp/>), which is a well known benchmark for MDVRP that was also used for instance by Surekha and Sumanthi [1]. It contains several problems from several up to hundredths of customers. We present our results on the instances p07 (Figure 4) and p07altered (Figure 5), where we modified the positions of the depots.

We performed three runs through clustering since the results differ for each run due to randomness of genetic algorithms. We then took the best found solution and compared with each other. In each run we used the population of 100 individuals and the evolution was ran for 99 generations. The probability of crossover was set to 0.8 and the probability of mutation of each gene to 0.02. The tournament selection was carried out on 5 randomly selected individuals.

As a baseline we took K-Means clustering. In fact, the regular K-Means clustering could not be used since it does not take into account the maximum depot capacity constraint. Therefore a genetic algorithm with a fitness function - Distance to The Depot was used to assimilate K-Means clustering. The results of this method are shown in Figure 6. On the left, we can see the clusters created by this method, middle shows the solution to the MDVRP. On the right we see how the fitness evolved in each population.

MDVRP results with our competing genetic clustering method are shown in Figure 7. It is obvious that the results of MDVRP are in this case inferior to the K-Means clustering results. However, the nice feature of this fitness function is that it actually allows to create clusters which intersect and therefore the routes in the final routing will cross, which cannot happen for K-Means. It gives us a higher planning flexibility by also not fixing the distances to the depots. Clearly, this approach would not be ideal when considering multiple vehicles from each depot, nonetheless that was not our assignment.

V. DISCUSSION

As is obvious from the results, it was not shown that our suggested fitness function performs better than the original K-Means clustering (Distance to The Depot fitness function). The comparison of the results with used fitness functions is displayed in Table I.

TABLE I: Comparison of suggested clustering fitness function with K-Means (Distance to The Depot)

	Distance to The Depot	Exclusive Pairing
Instance p07:	735.02	792.03
Instance p07altered:	739.73	741.85

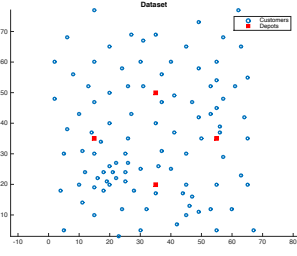


Fig. 4: Cordeau's instance p07

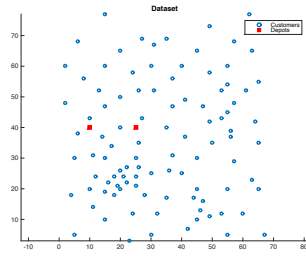


Fig. 5: Cordeau's instance p07altered

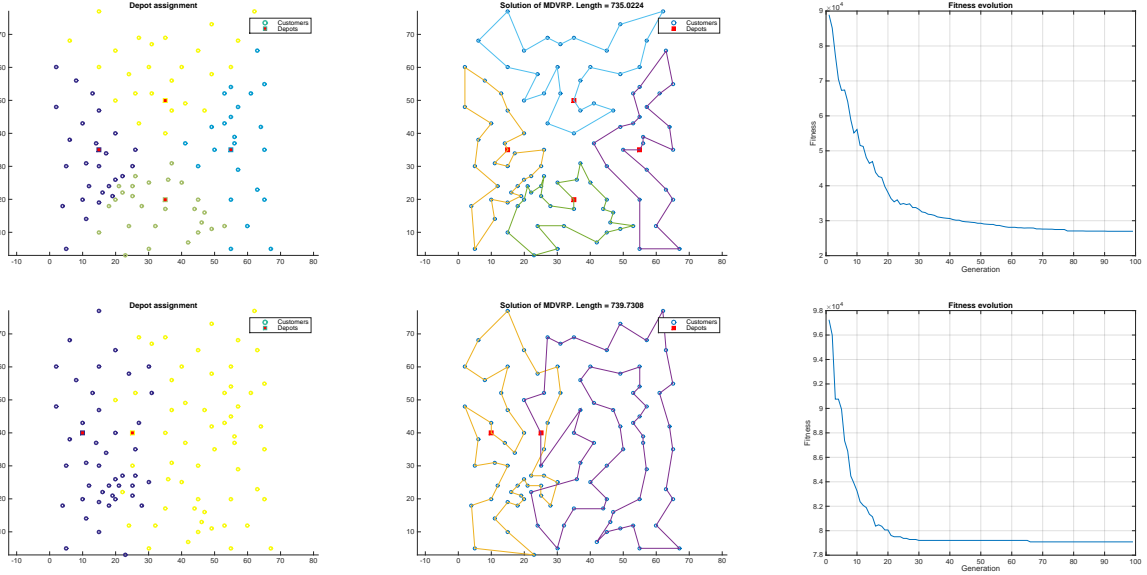


Fig. 6: Sample results of genetic clustering with Distance to The Depot fitness (p07 - top, p07altered - bottom)

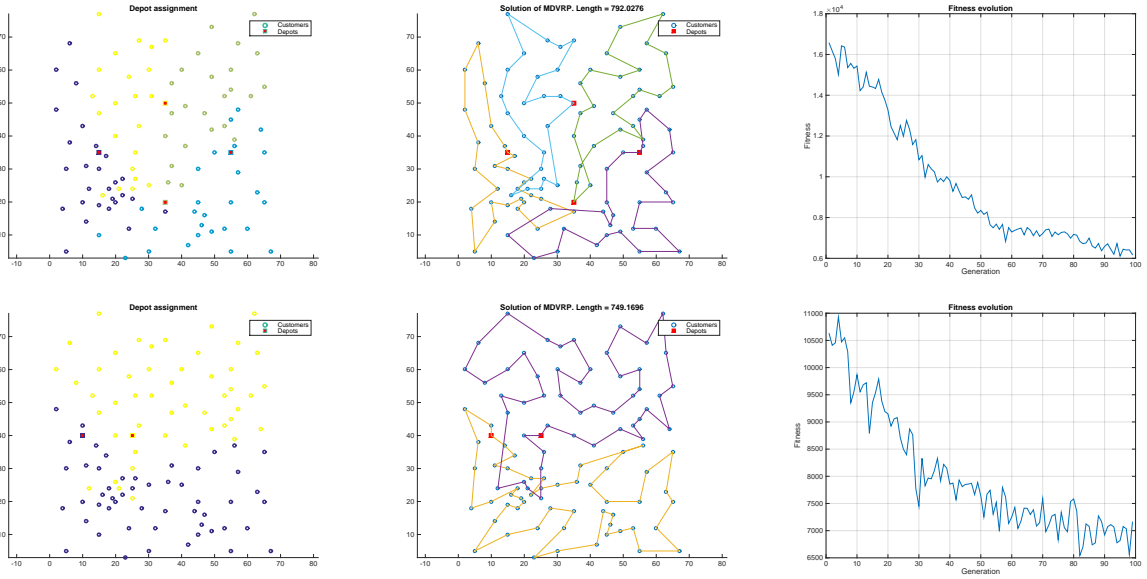


Fig. 7: Sample results of genetic clustering with Exclusive Pairing fitness (p07 - top, p07altered - bottom)

However, we can conclude that for a more difficult depot positions our results are very similar to the K-Means ones. The Exclusive Pairing fitness function showed some interesting possibilities for multi depot vehicle routing that the previous methods do not allow. For example creating clusters, which allow for intersecting roads. We believe that a further investigation of this method could lead in some instances to better results than the classical and for MDVRP so widely used K-Means clustering.

VI. CONCLUSION

In this semestral work we explored the impact of different clustering methods on the solution of MDVRP. We explained that all regular clustering methods yield the same clusters since the positions of the depots (cluster centers) are given. Hence we proposed a new clustering method that uses genetic algorithms and has a better insight in what actually is the purpose of the clustering.

From the presented results we can conclude that our proposed genetic clustering was not superior to the K-Means strategy. Even though, we showed an interesting new method, which allows for a higher cluster variability than K-Means. It is important because our goal is not to create compact clusters, but clusters suitable for routing.

REFERENCES

- [1] P. Surekha and S. Sumathi, "Solution to multi-depot vehicle routing problem using genetic algorithms," *World Applied Programming*, vol. 1, no. 3, pp. 118–131, 2011.
- [2] G. U. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research*, vol. 12, no. 4, pp. 568–581, 1964.
- [3] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992.
- [4] N. Alldrin, A. Smith, and D. Turnbull, "Clustering with em and k-means," *University of San Diego, California, Tech Report*, pp. 261–95, 2003.
- [5] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Parallel Problem Solving from Nature PPSN VI*. Springer, 2000, pp. 385–394.
- [6] C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 825–832, 1996.
- [7] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [8] G. Laporte, Y. Nobert, and S. Taillefer, "Solving a family of multi-depot vehicle routing and location-routing problems," *Transportation science*, vol. 22, no. 3, pp. 161–172, 1988.
- [9] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, no. 30, pp. 105–119, 1997.