A genetic algorithm for the

Multi-vehicle Covering Tour Problem

Abstract

This study considers the application of a genetic algorithm to a generalization of the vehicle routing problem in which the constraint of visiting all customers is not valid. The studied variant is called the multi-vehicle covering tour problem (m-CTP). This work tackles a particular case of the m-CTP where the number of vertices on each route does not exceed a given value p. In this paper, we use a genetic algorithm to solve the problem. Experiments are conducted on benchmark instances from TSP Library and show that our approach is competitive with the evolutionary local search (ELS) metaheuristic in terms of solution time and quality.

Keywords: Genetic algorithm, vehicle routing problem, covering tour problem.

1 Introduction

This paper aims at solving the multi-vehicle covering tour problem (m-CTP) where the number of vertices on each route does not exceed a given value p (The m-CTP-p). Different problem definitions are existing in the literature. Hà et al. (2013) define the m-CTP-p as an undirected graph $\mathrm{G} = (\mathrm{V} \cup \mathrm{W}, \mathrm{E})$, where $\mathrm{V} \cup \mathrm{W}$ is the vertex set and $\mathrm{E} = \{(\mathrm{v_i}, \mathrm{v_j}) : \mathrm{v_i}, \mathrm{v_j} \in \mathrm{V} \cup \mathrm{W}, \mathrm{i} < j\}$ is the edge set. Different kinds of locations have been considered: the set of vertices that can be visited (V), the set of vertices that must be covered (W), and the set of vertices that must be visited including the depot $(\mathrm{T} \subseteq \mathrm{V})$. Whereas, Hodgson et al.(1998) define the graph as $\mathrm{G} = (\mathrm{W}, \mathrm{E})$ where they does not include a subset that must be visited $(T = \emptyset)$ and consider that all customers must be covered and all may be visited $(\mathrm{V} \subseteq \mathrm{W})$. Each vertex of V has a unit demand and each vehicle has a capacity of p. The demand of each customer could be satisfied directly by visiting the customer along the tour or by covering it. Covered vertices must be within a predefined distance d from the tour. The coverage concept is used to deal with numerous real situations where there exists a restriction in some resources (time, vehicle,..) that prevents the visit of some customers. For example researchers apply the m-CTP to solve the problem of disaster relief and also to solve the problem of vaccination campaigns.

In this work, we consider a m-CTP-p variant that consists in finding m vehicle routes while minimizing the total cost and satisfying the following constraints:

(i) Each vehicle route starts and ends at the depot,

(ii) Each vertex of T belongs to exactly one route while each vertex of V \ T belongs to at most one route,

(iii) Each vertex of W must be covered by a route,

(iv)The number of vertices on a route (excluding the depot) is less than a given value p while the constraint on the length of each route is relaxed.

3 A Genetic algorithm for the m-CTP-p

The Genetic Algorithms (GA) prove their effectiveness in many combinatorial problems, including vehicle routing problems (VRP). The GA is a population based approach where a set of genetic operators is applied to select parent solutions from the population and produce offspring solutions. This work describes a GA developed to solve the m-CTP-p. We assume that the total distance travelled represents the fitness of each solution. In the following we start by defining the different elements of the solution in the population, then we present the different genetic operators used in our algorithm. After that, we evaluate the fitness of each individual in the population.

- **Solution representation**

The initial population was built by generating randomly an initial set of solutions. Each solution contains a set of vertices that must be visited, T, and some vertices from V to cover the total vertices in W. We evaluate the fitness function of each individual from the population and initialize the best one.

- **Genetic operators**

In this step we present three genetics operators: Selection, Crossover and mutation. The first one consists of selecting two parent solutions from the initial population. We use the roulette wheel selection to select these parents. Then, as a second operator, we use one point crossover procedure to produce an offspring. The procedure start by selecting randomly a crossing point cp from the first parent and then producing an offspring by copying the part of the chromosome before the cp from the first parent into the offspring, after that we delete those genes in the offspring from the second parent and we insert the gene from the second parent into the offspring and repeat this step until a feasible offspring is reached.

. Let I be the set of genes in the chromosome and $R = V \setminus I$. We remove randomly some genes from the chromosome and then we select a customer from R and insert it randomly in the chromosome. We repeat this process until the chromosome became feasible.

Each solution was evaluated respecting the capacity constraint. If the capacity constraint was violated, a penalty was added to our evaluation function. Based on the fitness function, we compare the worst individual with the new offspring and keep the best one inside the population.

4 Numerical results

The experiments are performed on laptop ASUS Intel Core i5-4200U, 2.3 Ghz processor and 6 GB memory and the proposed algorithm has been coded in C++ programming language. Our algorithm is then tested on the benchmark instances KroA100, KroB100, KroC100 and KroD100 of TSPLIB.

In the following, we present an example of computational results for KroA100 instances . In Table 1, we report small instances results and compare them with the existing methods in the literature. Column “Data” show the name of instance, column “Result” represents the objective value of the solution, column “Time” is the running time in seconds and finally the column “Gap” shows the deviation between the value of the solution given by the metaheuristic based on the evolutionary local search (ELS) developed by Hà et al. (2013) and the value obtained by our proposed approach.

We use the following genetic parameters in our experimentation. First we fixed the population size $S = 10$, then we define the probability of genetic operations. We consider $P_m = 0.1$ the probability of mutation operator and $P_c = 0.9$ the probability of crossover operator. The algorithm was executed until 100 generations were performed.

Table 1 Comparison between the ELS and the GA algorithm with 100 vertices

Data	GA		ELS (Hà et al. 2013)		
	Result	Time	Result	Time	Gap
A1-1-25-75-4	8479	0.016	8479	0.16	0
A1-1-25-75-5	8479	0.025	8479	0.17	0
A1-1-25-75-6	8479	0.06	8479	0.17	0
A1-1-25-75-8	7985	0.104	7985	0.16	0
A1-1-50-50-4	10271	0.182	10271	0.80	0
A1-1-50-50-5	9220	0.472	9220	0.78	0
A1-1-50-50-6	9130	0.385	9130	0.81	0
A1-1-50-50-8	9130	0.06	9130	0.81	0
B1-1-25-75-4	7146	0.067	7146	0.22	0
B1-1-25-75-5	6901	0.093	6901	0.18	0
B1-1-25-75-6	6450	0.038	6450	0.23	0
B1-1-25-75-8	6450	0.077	6450	0.20	0
B1-1-50-50-4	10107	0.455	10107	0.62	0
B1-1-50-50-5	9723	0.002	9723	0.64	0
B1-1-50-50-6	9382	0.441	9382	0.58	0
B1-1-50-50-8	8348	37.078	8348	0.58	0
C1-1-25-75-4	6161	0.002	6161	0.16	0
C1-1-25-75-5	6161	0.001	6161	0.16	0
C1-1-25-75-6	6161	0.001	6161	0.15	0
C1-1-25-75-8	6161	0.001	6161	0.17	0

The GA were able to produce competitive solutions compared with existing algorithms.

5 Conclusions and perspectives

Experimental results show that the GA performs reasonably well . It was considered as a strong competitor with other metaheuristics for the routing problem. Further work will be on the improvement of our algorithm by adding a local search approach.

References

Hà, M.H. , Bostel, N. , Langevin, A. , Rousseau, L.-M. , 2014. An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. Comput. Oper. Res. 43, 9–19 .

Hodgson, M.-J. , Laporte, G. , Semet, F. , 1998. A covering tour model for planning mobile health care facilities in suhum district gana. J. Regional Sci. 38, 621–638 .