

# Multi-Phase Meta-Heuristic for Multi-Depots Vehicle Routing Problem

Jianping Luo, Xia Li, Min-Rong Chen

College of Information Engineering, Shenzhen University, Shenzhen, 518060, China.

Received 2013

## ABSTRACT

In this work, we present a multi-phase hybrid algorithm based on clustering to solve the multi-depots vehicle routing problem (MDVRP). The proposed algorithm initially adopts K-means algorithm to execute the clustering analyses, which take the depots as the centroids of the clusters, for the all customers of MDVRP, then implements the local depth search using the Shuffled Frog Leaping Algorithm (SFLA) for every cluster, and then globally re-adjusts the solutions, i.e., rectifies positions of all frogs by the extremal optimization (EO). The processes will continue until the convergence criterions are satisfied. The results of experiments have shown that the proposed algorithm possesses outstanding performance to solve the MDVRP.

**Keywords:** Combinatorial Optimization; Vehicle Routing Problem; Shuffled Frog Leaping Algorithm

## 1. Introduction

Multi-depots vehicle routing problem (MDVRP) is a complex NP-hard problem. To solve the MDVRP, Many researchers have proposed various solutions. Shuffled Frog Leaping Algorithm (SFLA) which was developed by Eusuff and Lansey in 2003[1], belongs to the MA family. It is a meta-heuristic optimization method inspired from the memetic evolution of a group of frogs when seeking for food. In this algorithm evolution of memes is driven by exchange of information among the interactive individuals. It combines the advantages of the genetic-based MA and the social behavior-based Particle Swarm Optimization (PSO) algorithm. SFLA has been tested on several combinatorial problems and found to be effective in searching the global solutions.

In this paper, an improved multi-phase SFLA based on clustering is presented to solve the multi-depots vehicle routing problem (MDVRP). To further improve the local search ability of SFLA, the extremal optimization(EO) local search is introduced to the solving framework.

This paper is organized as follows: Section 2 describes the hierarchy of decisions in the MDVRP. The method of solving MDVRP using SFLA is presented in Section 3. The multi-phase SFLA algorithm for MDVRP is introduced in Section 4. Experimental evaluations and result discussions are shown in Section 5. Finally, we draw the conclusions.

## 2. Multi-Depots Vehicle Routing Problem

Consider a distribution network in which one product is shipped from some depots to a set of customers. The MDVRP can be defined on multi-graph  $G\{C, D, E\}$ , where  $C$  is customer vertex set which includes  $N$  customers,  $D$  is the depot vertex set which includes  $M$  depots. The set  $V = \{C \cup D\}$  is the node set, and  $E$  is the arc set.  $E = \{(i, j) | i, j \in V\} \setminus \{(i, j) | i, j \in D\}$ . The demand of the  $i$ -th customer is  $q_i$ . The distance between point  $i$  and point  $j$  is  $c_{ij}$ . The system has  $L$  vehicles, and the  $K = \{k_1, k_2, \dots, k_L\}$  is the set of all vehicles. The maximum capacity of vehicle  $i$  is  $Q_i (i=1, 2, \dots, L)$ . We use  $K(d)$  to denote the vehicle set of the depot  $d (d \in D)$ , use  $K_d$  to denote the maximum vehicle number of depot  $d$ , and  $C(d)$  denotes the customer set of the depot,  $d (d \in D)$ . Each vehicle starts the travel from a depot and has to return to the same depot in completion of services to customers. The objective is to determine a viable delivery sequence of each route which minimizes the delivery distance or time spent in serving all customers.

Decision variables:

$$x_{ijk} = \begin{cases} 1 & \text{if point } i \text{ immediately precedes point } j \text{ on route } k; \\ 0 & \text{otherwise} \end{cases}$$

$$y_{kd} = \begin{cases} 1 & \text{if vehicle } k \text{ is allocated to depot } d; \\ 0 & \text{otherwise} \end{cases}$$

The MDVRP can then be stated as follows:

$$Z = \min \sum_{d \in D} \sum_{k \in K(d)} \sum_{i \in (C(d) \cup \{d\})} \sum_{j \in (C(d) \cup \{d\})} c_{ij} x_{ijk} \quad (1)$$

S.t.

$$\sum_{k \in K} \sum_{j \in C} x_{djk} \leq K_d \quad \forall d \in D \quad (2)$$

$$\sum_{d \in D} K_d \leq L \quad (3)$$

$$\sum_{d \in D} y_{kd} = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{k \in K} \sum_{j \in C} x_{ijk} = 1 \quad \forall i \in C \quad (5)$$

$$\sum_{i \in C \setminus p} x_{ipk} = \sum_{j \in C \setminus p} x_{pjk} \quad \forall p \in C, \forall k \in K \quad (6)$$

$$\sum_{j \in C} q_j \sum_{i \in V} x_{ijk} \leq Q_k \quad \forall k \in K \quad (7)$$

$$\sum_{j \in C} x_{djk} y_{kd} = 1 \quad \forall k \in K, \forall d \in D \quad (8)$$

$$\sum_{i \in C} x_{idk} y_{kd} = 1 \quad \forall k \in K, \forall d \in D \quad (9)$$

In the above, Eq.(1) which is the objective function is to minimize the total cost (distance or time) by all vehicles. Eqs.(2) and (3) require that the number of vehicles which are arranged is less than the maximum number of vehicles. Eq.(4) requires that each customer to be assigned to a single route. Eqs.(5) and (6) assure that each route can be served at most once. Eq.(7) is the capacity constraint set for vehicles. Eqs.(8) and (9) require that every route is circular path.

### 3. Multi-Phase Shuffled Frog Leaping Algorithm for MDVRP

Our previous work[2] for solving the MDVRP belongs to single-phase algorithm, whose solving process relies on the framework of SFLA. One frog represents one solution. The evolution and learning of algorithm take the whole solution as a unit, and take all information, e.g., all customers, all paths, all vehicles, etc., into account. The solving time increases rapidly with the increasing of the scale of MDVRP. However, for the updating of a solution within one iteration of SFLA, only a little information needs to be changed, i.e., only a little part of the paths changes, and most of components of individual keep unchanged. Therefore, the solving efficiency of algorithm is very low. In this work, we present a multi-phase SFLA based on clustering to solve the MDVRP more quickly. The improved algorithm initially adopts  $K$ -means algorithm to execute the clustering analyses for all the customers, which take the depots as the centroids of the clusters, then implements the local depth search by using the SFLA for each cluster, and then globally re-adjusts the solutions, i.e., rectifies positions of all frogs by the EO. Next step, the clustering

analyses will be repeated to generate new clusters according to the best solution gotten by preceding process. The improved paths information is inherited to the new clusters, and the local search by using SFLA for each cluster is used again. The processes will continue until the convergence criterions are satisfied.

#### 3.1. Initialization Using the K-Means Clustering

$K$ -means is one of unsupervised learning algorithms that solves the well known clustering problem.  $K$ -means clustering is a method of cluster analysis which aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume  $k$  clusters) fixed a priori. The main idea is to define  $k$  centroids, one for each cluster. These centroids should be placed in a cunning way because different locations cause different results. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate  $k$  new centroids as barycenters of the clusters resulting from the previous step. After we have these  $k$  new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the  $k$  centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (10)$$

where  $\|x_i^{(j)} - c_j\|^2$  is a chosen distance measure between a data point  $x_i^{(j)}$  and the cluster centre  $c_j$ , is an indicator of the distance of the  $n$  data points from their respective cluster centres.

In the multi-phase algorithm called multi-phase SFLA in this work, we use  $K$ -means clustering analyses to generate the clusters initially. All the customers are allocated to clusters which take the depots, some unchanged points, as centroids of clusters by using  $k$ -means method. So, after finishing allocation only once, the clustering process is completed.

#### 3.2. The Multi-Phase SFLA for Solving MDVRP

After finishing the clustering analyses for all the customers, every cluster, which includes some customers and

whose centroid is a depot, can be seen as a VRP. Therefore, MDVRP can be divided into several VRPs. The single-phase method is applied to solve the VRP, where the number of depots  $M$  is equal to 1 as the cluster includes only one depot. Each cluster generates several vehicle routes. The SFLA is regarded as the local search approach to implement in every cluster. The number of customers of every cluster is usually far smaller than the number of all customers, so the number of frogs of SFLA for one cluster is usually small, both the numbers of memplexes and the number of frogs in each memplex can be reduced  $M$  times. Therefore, the local search of SFLA for every cluster is very quick, and the whole solving process is very time-saving. For an MDVRP instance, there are  $M$  depots, so the number of frog in every cluster can be set as:

$$F_i = \frac{F}{M^2} \quad (11)$$

where  $F_i$  is the number of frogs in  $i$ -th ( $1 \leq i \leq M$ ) cluster;  $F$  is the total number of frog population.

It is important to note that the clustering analyses may not be accurate enough to divide the customers to every depot, so the vehicle route generated by the SFLA in every cluster may not be the optimal solution. Therefore, we take global information, which contains all customers and depots into account, re-adjust every path produced by the preceding local search, and get the better solution. In this work, we re-adjust the each route by using extremal optimization (EO) (Please refer to our earlier study[3]).

After finishing the readjustments for all routes, the algorithm repeats the preceding three stages: clustering analyses, SFLA local search in cluster and global adjusting. It should be noted that the frog population ought to be regenerated. Our algorithm does the new clustering analyses by re-adjusting the clusters according to the best solution gotten by preceding process. These stages continue until convergence criteria are satisfied.

## 4. Experiments and Results

Note that our algorithms were run on the hardware (i.e., Intel Pentium 4 with 2.8 GHz CPU and 512M memory) and software (i.e., VC++) platform. The parameters setting of single-phase SFLA are performed:  $m = 30$ ,  $n = 20$ ,  $q = 12$ , and the generation number for the sub-memplex is equal to 10. These values are found suitable to produce good solutions in terms of the processing time and the quality of the solution in accordance with our experiments observation. For the multi-phase version, the SFLA is regarded as the local search to use in every cluster, the parameters of SFLA will be reduced  $M$  times,  $M$  is the number of depots. For instance, when

$M = 4$ , the parameters are set as:  $m = 8, n = 5, q = 3$ , and the generation number for the sub-memplex is 3. In experiments, the solution stops when the following criterion is satisfied: the objective function value is not improved in 50 successive shuffling iterations (for single-phase version) or 50 successive global adjustments (for multi-phase version).

### 4.1. Results for the MDVRP

**Table 1** presents the results of two benchmark heuristics and the proposed multi-phase/single-phase algorithm on 21 MDVRP instances, respectively. The problem instances used for the analysis originate from Cordeau, Laporte, and Mercier (2001) and are available on the internet at <http://www.hec.ca/chairedistributique/data>. The first three columns of **Table 1** describe the benchmark instances with instance name, the number of customers which have to be served and the number of depots denoted by Name, n and d, respectively. In **Table 1**, we present the best, average result and average run-time over 10 runs for all the randomized methods, the time is recorded in minutes. The cost of a solution is defined as the total distance traveled by the vehicles. The ALNS[4] framework is an extension of the large neighborhood search framework by Shaw [Using constraint programming and local search methods to solve vehicle routing problems. In: CP-98, Fourth international conference on principles and practice of constraint programming, Lecture notes in computer science, vol. 1520, 1998. p.17–31] with an adaptive layer (hardware: 3 GHz Pentium 4). This layer adaptively chooses among a number of insertion and removal heuristics to intensify and diversify the search. Juan et al. 5] propose GRASP/VND which contains two metaheuristics based on greedy randomized adaptive search procedures (GRASP), variable neighborhood descent (VND) and evolutionary local search (ELS) to solve MDVRP (hardware: 3.4GHz Pentium D). Note that GRASP/VND[14] didn't solve the instances of capacity constrained and route duration constrained MDVRP, which are in the last 10 rows, i.e., from pr01 to pr10, we use the sign “-” to indicate it in **Table 1**.

The meta-heuristic proposed in this article outperform the other two meta-heuristics. Among the proposed meta-heuristics, our algorithms is both faster and more accurate than GRASP/VND and ALNS. The multi-phase algorithm presented in this paper got 17 out of 21 best solutions and is faster than GRASP/VND and ALNS. For example, for the pr06 instance, the run-time of multi-phase version is less than one second of the run-time of ALNS, but the former gets better BEST value. Also, the table also shows that our algorithms are quite stable as the average solution is very close to the best solution and the average deviations never surpass 1% in all instances. Therefore, the multi-phase version is very suitable for

**Table 1. Comparisons of some algorithms applied to the MDVRP.**

Instance			ALNS[4]			GRASP/VND[5]			multi-phase SFLA		
Name	n	d	AVE	BEST	TIME	AVE	BEST	TIME	AVE	BEST	TIME
p01	50	4	576.87	<b>576.87</b>	0.48	576.87	<b>576.87</b>	0.89	576.87	<b>576.87</b>	<b>0.26</b>
p02	50	4	473.53	<b>473.53</b>	0.47	473.53	<b>473.53</b>	1.12	473.53	<b>473.53</b>	<b>0.38</b>
p03	75	5	641.19	<b>641.19</b>	1.07	643.44	<b>641.19</b>	2.09	641.19	<b>641.19</b>	<b>0.58</b>
p04	100	2	1006.09	<b>1001.04</b>	1.47	1008.1	1003.62	3.5	1003.97	<b>1001.04</b>	<b>0.69</b>
p05	100	2	752.34	751.26	2	752.54	751.15	3.13	751.98	<b>750.03</b>	<b>0.92</b>
p06	100	3	883.01	<b>876.7</b>	1.55	884.42	880.69	3.51	882.64	<b>876.7</b>	<b>0.75</b>
p07	100	4	889.36	<b>881.97</b>	1.47	892.59	888.65	3.55	889.08	<b>881.97</b>	<b>0.90</b>
p12	80	2	1319.13	<b>1318.95</b>	1.25	1318.95	<b>1318.95</b>	2.2	1319.40	<b>1318.95</b>	<b>0.85</b>
p15	160	4	2519.64	<b>2505.42</b>	4.22	2508.05	<b>2505.42</b>	9.3	2512.87	2505.47	<b>2.98</b>
p18	240	6	3736.53	<b>3702.85</b>	6.98	3737.64	3702.82	22.72	3734.68	3709.29	<b>3.47</b>
p21	360	9	5501.58	<b>5474.84</b>	9.7	5522.02	5490.55	52.96	5500.06	<b>5482.55</b>	<b>4.58</b>
pr01	48	4	861.32	<b>861.32</b>	0.5	-	-	-	861.32	<b>861.32</b>	<b>0.36</b>
pr02	96	4	1308.17	<b>1307.34</b>	1.72	-	-	-	1308.10	<b>1307.34</b>	<b>0.75</b>
pr03	144	4	1810.66	<b>1806.6</b>	3.57	-	-	-	1809.15	<b>1806.6</b>	<b>1.57</b>
pr04	192	4	2073.16	2060.93	4.93	-	-	-	2070.84	<b>2059.99</b>	<b>1.95</b>
pr05	240	4	2350.31	2337.84	6.2	-	-	-	2351.24	2341.22	<b>3.11</b>
pr06	288	4	2695.74	2687.6	7.75	-	-	-	2699.37	<b>2687.43</b>	<b>3.69</b>
pr07	72	6	1089.56	<b>1089.56</b>	0.97	-	-	-	1089.92	<b>1089.56</b>	<b>0.82</b>
pr08	144	6	1675.74	<b>1664.85</b>	3.45	-	-	-	1677.33	<b>1664.85</b>	<b>1.98</b>
pr09	216	6	2144.84	<b>2136.42</b>	5.83	-	-	-	2148.58	<b>2136.42</b>	<b>3.52</b>
pr10	288	6	2905.43	<b>2889.82</b>	7.58	-	-	-	2903.25	2892.14	<b>3.14</b>

solving the large scales problem.

## 5. Conclusion

In this paper, we present a multi-phase hybrid algorithm based on clustering to solve the MDVRP. The proposed algorithm initially adopts K-means algorithm to execute the clustering analyses, which take the depots as the centroids of the clusters, for all the customers of MDVRP, then implements the local depth search using the Shuffled Frog Leaping Algorithm (SFLA) for every cluster, and then globally re-adjusts the solutions, i.e., rectifies positions of all frogs by the extremal optimization (EO). The processes will continue until the convergence criterions are satisfied. Compared to other algorithms, the multi-phase algorithm is very fast and can get better solution because it adopts strategy of first clustering, second local search and third adjustment. Therefore, the multi-phase version is very suitable for solving the large scales problem. The results of experiments have shown the proposed algorithm possesses outstanding perform-

ance to solve the MDVRP.

## 6. Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.60772148 and No.61005049, and the Shenzhen Natural Science Foundation under Grant No. JCYJ20120613161222123.

## REFERENCES

- [1] Eusuff M, Lansey K. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. *Journal of Water Resources Planning and Management* 2003; 129 (3):10–25.
- [2] Luo Jian-ping, Li Xia, Chen Min-rong. A Novel Meta-heuristic for the Multi-Depot Vehicle Routing Problem. *Communications in Computer and Information Science* 2012; 307:216-224.
- [3] Luo Jian-ping, Li Xia, Chen Min-rong. Improved Shuffled Frog Leaping Algorithm for Solving CVRP. *Journal of Electronics & Information Technology* 2011; 33(2):

- 30-434.
- [4] Crevier B, Cordeau J, Laporte G. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research* 2007; 176(2):756–773.
- [5] Villegas J G, Christian P. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence* 2010; 23(5):780–794.