Decision Making: Applications in Management and Engineering
Vol. 2, Issue 2, 2019, pp. 112-125.
ISSN: 2560-6018
eISSN:2620-0104
DOI: https://doi.org/10.31181/dmame1902089b

A 2-OPT GUIDED DISCRETE ANTLION OPTIMIZATION ALGORITHM FOR MULTI-DEPOT VEHICLE ROUTING PROBLEM

Partha Sarathi Barma [1*], Joydeep Dutta [1] and Anupam Mukherjee [2]

[1] Department of Computer Science and Engineering, NSHM Knowledge Campus Durgapur, India
[2] Department of Mathematics, National Institute of Technology Durgapur, India

Received: 25 January 2019;
Accepted: 30 April 2019;
Available online: 10 May 2019.

Original scientific paper

Abstract: *The Multi-depot vehicle routing problem (MDVRP) is a real-world variant of the vehicle routing problem (VRP) where the customers are getting service from some depots. The main target of MDVRP is to find the route plan of each vehicle for all the depots to fulfill the demands of all the customers, as well as that, needs the least distance to travel. Here all the vehicles start from different depots and return to the same after serving the customers in its route. In MDVRP each customer node must be served by only one vehicle which starts from any of the depots. In this paper, we have considered a homogeneous fleet of vehicles. Here a bio-inspired meta-heuristic method named Discrete Antlion Optimization algorithm (DALO) followed by the 2-opt algorithm for local searching is used to minimize the total routing distance of the MDVRP. The comparison with the Genetic Algorithm, Ant colony optimization, and known best solutions is also discussed and analyzed.*

Key words: *Multi depot vehicle routing problem, Antlion Optimization (ALO), Bio-inspired Algorithm, Combinatorial Optimization.*

1. Introduction

Supply of goods from source to destination is a challenging operational process in the logistic distribution system. The products can be delivered either directly from the production center or from the stock points located nearby the production site or via distribution warehouses. Such kind of problems can be mathematically modeled as a particular type of VRP which belongs to the set of NP-hard problems. It consists of a single depot or warehouse to service the demands of different cities, but most of the cases the different company has more than one warehouse to serve the demands. In such a scenario the problem can be formulated using more than one depot that is

* Corresponding author.
E-mail addresses: partha4187@gmail.com (P. S. Barma), joy77deep@yahoo.co.in (J. Dutta), mukherjee.anupam.bnk@gmail.com (A. Mukherjee)

called Multi-Depot Vehicle Routing Problem, in short MDVRP. MDVRP deals with the delivery of items to all the customers with minimum cost or distance. VRP can be used to manage such kind of scenario efficiently.

The main task of the basic form of vehicle routing problem is to search the collection of paths to serve customers with some similar vehicles. In the classic form of VRP, a set of customer node is present, the demands of each node and other primary information such as the distance between all pair of nodes, the distance between nodes and depots, number of vehicles and vehicle capacity are known a priory. The VRP can be closed or open. In closed VRP (Laporte & Nobert, 1987) vehicles move from a central point called depot, serves each customer and back to the central position such that the total demand served by one conveyance is less than the vehicle capacity. Whereas in the case of open VRP (Li et al., 2007) after serving the customer the vehicle does not return to the depot.

There are many variants of VRP found in the literature; some of them are capacitated VRP (CVRP), VRP with time window (VRPTW), VRP that includes pickup and delivery, multi-depot VRP, stochastic VRP, etc. In this paper, we have focused on Multi-depot VRP (MDVRP). The pictorial representation of MDVRP is presented in Figure 1. In MDVRP, there will be more than one depot.

For solving MDVRP, the following two steps can be used:

I Clustering: Allocation of cities to a depot.

II Routing: Finding the optimum routes for each depot. This sub-problem is similar to VRP.

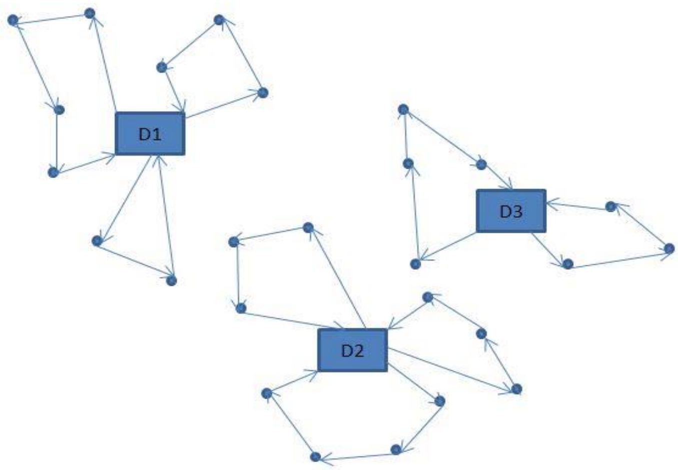

Figure1. Pictorial representation of MDVRP

MDVRP can be solved in two ways considering the two sub-problems, one is route first cluster second, and another is cluster first route second. Here we have discretized the Ant lion optimization (ALO) algorithm to solve the MDVRP. For local searching of routes, the 3-opt algorithm is used. The main contribution of this article is as follows: (1) An improved discrete ALO has been proposed to fit the MDVRP; (2) A new encoding scheme to form a solution (ant or antlion) and (3) A hybridization of ALO and 2 opt algorithm.

The paper is arranged as per the below sections. The literature review presents in section 2. The motivation behind this work is explained in section 3. Section 4 describes the mathematical model for the MDVRP. Section 5 deals with the proposed Discrete ALO. The result and discussion are presented in section 6. The conclusion is in section 7.

2. Literature Review

Some of the solving techniques for single depot VRP are exact algorithms like brunch and bound, branch and cut proposed by Fisher (1994) and Ladányi et al. (2001). Many heuristic algorithms like cluster first route second (Taillard, 1993), savings algorithm (Clarke & Wright, 1964) also found in the literature. Meta-heuristic like GA (Berger & Barkaoui, 2003), PSO (Chen et al., 2006), ACO (Reimann et al., 2004) are also used by many researchers to solve single depot VRP.

Laporte et al. (1984, 1988) formulated the integer linear programs for MDVRP containing degree constraints, sub-tour elimination constraints, chain-barring constraints, and integrality constraints and presented an exact solution. Renaud et al. (1996) presents a Tabu search heuristic for MDVRP. Chao et al. (1993) solved the MDVRP using a multi-phase heuristic approach. Ombuki-Berman and Hanshar (2009) applied a genetic algorithm to MDVRP. Vianna et al. (1999) proposed an evolutionary algorithm coupled with local search heuristic to minimize the total cost. Matos and Oliveira (2004) have to use ant colony optimization (ACO) to solve MDVRP. Guimarães et al. (2019) have published a paper on the multi-depot inventory-routing problem with the application on a two-echelon (2E) supply chain. It is also showing a stricter policy for inventory management. In 2017, a different version of MDVRP was developed that deals with hazardous materials by Yuan et al. (2017). It was solved using a two-stage heuristic method. In the same year, Rabbouch et al. (2017) have published a survey paper on MDVRP for heterogeneous vehicles. It also considered the time windows concept. Very recently Lalla-Ruiz and Voß (2020) have developed multi-depot cumulative capacitated VRP. It also designed a meta-heuristic approach (POPMUSIC) to solve it. In 2018, one more paper has also been published on MDVRP, and it has been solved using general variable neighborhood search meta-heuristic (Bezerra et al., 2018). It uses a local search method named randomized variable neighborhood descent. Li et al. (2018) have presented a paper on MDVRP with fuel consumption to make the benefits analysis. It finds the factors that affect the benefit ratio. In the same year, one more paper on MDVRP has also been published that deals with multi-compartment vehicles. It uses the hybrid adaptive large neighborhood search (Alinaghian & Shokouhi, 2018) to solve the problem. One more new variety of MDVRP has been proposed by Zhou et al. (2018). They have developed two –Echelon MDVRP that introduces the last mile distribution in the city logistics problem. It has been solved using a hybrid multi-population genetic algorithm. Silva et al. (2018) have presented a paper on multi-depot online vehicle routing with a soft boundary. Recently Zhang et al. (2019) have published an article on MDVRP for routing alternate fuel vehicles. They have used the ant colony method. Very recently Dutta et al. (2019) have designed a modified version of Kruskal's algorithm over the GA to solve OVRP for a single depot problem. Mukherjee et al. (2019) have developed a special version of the TSP problem that can be mapped on several real-life scenarios.

3. Motivation

There are several works that have already been published in the field of VRP using the exact method and meta-heuristics algorithms. But most of the real-life problems fit with the MDVRP. e.g., newspaper distribution, courier services, emergency services, taxi services, and refuse-collection management, etc. In literature, there are some works on MDVRP but in most of the cases they used meta-heuristic algorithms, and in few cases, exact algorithms were used. Exact algorithms give better result but take longer computational time. Meta-heuristic algorithms take less computational time but will not provide the best solution always. So finding good meta-heuristic to address the real-life problem which will give better result in reasonable computational time is a tough job. So here we try to find a hybrid algorithm which will combine an exact algorithm and one meta-heuristic algorithm to address MDVRP. Two competitive firms produce two substitute products and sell their products separately in the market.

4. Mathematical Model

The MDVRP can be represented using a graph G = (V, E) where V is the union of two subsets namely, *Vc = {V1, . . . ,Vn}* the set of city or customer and *Vd = {Vn+1,..., Vn+m}* the set of depots, and E is the edge set. A cost or distance matrix *C= {cij}* is the cost of traveling from city *i* to city *j*. Each city vi has a demand qi. In this paper symmetric cost or distance matrix is considered and triangular inequality also satisfied in C. Here all depots have a finite set of homogeneous vehicles with capacity Q. The solution to an MDVRP consists of a set of vehicle routes each starts and ends at the same depot, and each customer node is visited exactly once by only one vehicle. The total demand of customers in each route must not exceed the vehicle capacity Q. Here the goal is to minimize the total routing cost.

In this problem, n nodes are grouped into m cluster where each cluster contain ni: i = 1,2,...,m number of node and each ni clusters are again group by kj groups depending on the vehicle capacity.

The mathematical model for MDVRP proposed by Lang (2018) is given below.

$$Min\ Z = \sum_{p=1}^{m}\sum_{q=1}^{k_p}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ijpq} \qquad (1)$$

Subject to

$$\sum_{j=1}^{n} q_i y_{iqp} \leq Q \qquad (2)$$

$$0 \leq n_{jq} \leq n_j \qquad (3)$$

$$\sum_{q=1}^{k_p} n_{jq} = \ n_j \quad \forall\ j = 1\ tom \qquad (4)$$

$$a\sum_{j=1}^{m} n_j = \text{n} \qquad (5)$$

$$\sum_{p=1}^{m}\sum_{q=1}^{k_p} y_{iqp} = 1 \tag{6}$$

$$x_{ijqp} = \begin{cases} 1 \text{ if vehicle p in depot q travels from customer i to customer j} \\ 0 \quad otherwise \end{cases} \tag{7}$$

$$y_{ikm} = \begin{cases} 1 \text{ if vehicle k of depot m serves customer i} \\ 0 \quad otherwise \end{cases} \tag{8}$$

The equation (1) is the objective function, minimizes the total traveling distance or cost. Equation (2) ensures the capacity constraint of a vehicle. Equation (3) guarantees that the vehicles serving the number of customers must not exceed the number of customers in a depot. Equation (4) shows that the total number of customers served by the entire route must be equal to the sum of customers served by depot m. Each customer must be served from a single depot is ensured in Equation (5). Equation (6) shows that each customer is serviced not more than once. Equation (7) and (8) represents that the decision variables are binary.

5. Proposed Discrete ALO Algorithm

In this paper, we have used the Ant Lion Optimization algorithm proposed by Mirjalili (2015). ALO is a bio-inspired algorithm that mimics the foraging behavior of antlion. The steps of ALO are given below:

- Initialization of ant and antlions
- Random walk of ants
- Building traps by antlions
- Entrapment of ants in traps prepared by the antlion
- Catching preys by antlion
- Re-building traps.
- Elitism
- Here 2-opt algorithm is used to optimize each route covered by one vehicle.

5.1. Encoding Scheme

An MDVRP contains n cities and m depots. We have used cluster first, route second approach. So to represent an ant or ant lion one integer array A of size n is considered, and the array elements will be ranging from 1 to m. An element *A[i]* represents that ith city will be served from depot *A[i]*. As an example consider n as 10 and m as 3 then an ant or an antlion will be as in Figure 2.

2	1	1	3	3	2	2	1	3	2
1	2	3	4	5	6	7	8	9	10

Figure 2. Encoding of an ant

From Figure 2 it is clear that depot 1 will serve city 2, city 3 and city 8, depot 2 will serve city 1, city 6 and city 7 and depot 3 will serve city 4, city 5 and city 9.

5.2. Fitness Evaluation

In this paper, the fitness function is considered as same as the objective function. Now to evaluate the value of fitness function we have to find the depot corresponding to each city and the vehicle which will serve the city. From the encoding scheme stated above, it is clear that which city will be served from which depot. Then we have to find the vehicle routes starting from each depot. Here we have applied a very well-known 2-opt algorithm to find the shortest path starts and end in the same depot after serving all the cities in the route.

Therefore, Total fitness value = the total distances traveled by all the vehicles from all depots. Consider an ant A as follows.

3	1	3	1	2	1	2	3	2	2

Then Depot 1 will serve city 2, 4, 6; depot 2 will serve city 5, 7, 9, 10 and depot 3 will serve city 1, 3, 8. Now according to the vehicle capacity routes are to be decided from each vehicle from the depot. Assume one vehicle is required for depot 1. Then the initial route will be as {0, 2, 4, 6, 0} for depot 1. Now, this is very similar to the traveling salesman problem. Here we have used a 2-opt algorithm for local search to optimize the route length. A similar approach is taken for all the routes from the different depot, and finally, all the route lengths are added to get the fitness value.

5.3. Operators of ALO

The Antlion Optimizer does a mimic of the relationship of antlions and ants. The ants will move on the search space, and the antlions are building traps to hunt ants. After capturing an ant, the position of the Antlion is updated if it becomes fitter. The movement of ant for searching food is stochastic therefore a random walk is as follows

$$x(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)] \quad (9)$$

Where cumsum represents the cumulative sum where n represents the maximum iteration number and t, gives the step of random walk and r(t) is a random function given by:

$$r(t) = \begin{cases} 1\ if rand > 0.5 \\ 0\ if rand \leq 0.5 \end{cases} \quad (10)$$

The position of ant and antlions are stored in the following matrix respectively

$$M_{Ant} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,d} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,d} \end{bmatrix} \quad (11)$$

$$M_{Antlion} = \begin{bmatrix} Al_{1,1} & \cdots & Al_{1,d} \\ \vdots & \ddots & \vdots \\ Al_{n,1} & \cdots & Al_{n,d} \end{bmatrix} \quad (12)$$

A fitness function is used to identify the quality of ant and antlion during the optimization process. Two different matrices MOA and MOAL are used to store the fitness of all ant and antlion respectively. The matrices are as follows.

$$M_{OA} = \begin{bmatrix} f([A_{1,1}, A_{1,2}, \dots\dots, A_{1,\mathrm{d}}]) \\ f([A_{2,1}, A_{2,2}, \dots\dots, A_{2,\mathrm{d}}]) \\ \vdots \\ \vdots \\ f([A_{\mathrm{n},1}, A_{\mathrm{n},2}, \dots\dots, A_{\mathrm{n},\mathrm{d}}]) \end{bmatrix} \quad (13)$$

$$M_{OAL} = \begin{bmatrix} f([Al_{1,1}, Al_{1,2}, \dots\dots, Al_{1,\mathrm{d}}]) \\ f([Al_{2,1}, A_{2,2}, \dots\dots, Al_{2,\mathrm{d}}]) \\ \vdots \\ \vdots \\ f([Al_{\mathrm{n},1}, Al_{\mathrm{n},2}, \dots\dots, Al_{\mathrm{n},\mathrm{d}}]) \end{bmatrix} \quad (14)$$

Where *f* is the objective function. *Ai,j* gives the value of the *j*th dimension of *i*th ant, *n* represents the total number of ants and is similar for antlions.

The ALO (Mirjalili, 2015) was designed to solve continuous problems. In this paper, we are focused on solving MDVRP which is one combinatorial optimization problem. So the operators used in original ALO may not work as desired hence we have customized the operators according to our requirement.

Initialization

In this step, two populations of size N for ant and antlion are formed randomly. Let us assume n number of customers and m number of depots is present. Assume (*Al1, Al2,……, AlN*) and (*A1, A2,……, AN*) are the populations of antlion and ant respectively. Then each Alj and Aj represents the *j*th antlion and ant respectively. Both *Alj* and *Aj* are a one-dimensional array of size *n*, and the array elements will range from 1 to *m*.

Random walks of ants

In case of discrete problem random walk of an ant is implemented by inverting the entities of a randomly selected part of the string. The operation is demonstrated in Figure 3.

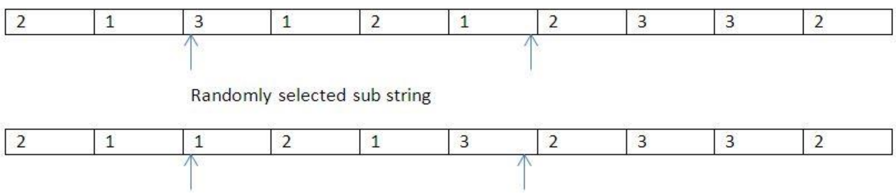

Figure 3. Random Walk of an Ant

Building traps by Antlion

In ALO, each antlion builds a trap to catch one ant. To implement this mechanism, we have used the Roulette-wheel selection mechanism to select Antlion. Roulette wheel selection chooses the fitter Antlions for catching ants with higher probability.

Entrapment of ants in traps

Ants are moving randomly in search of food while antlions build traps. The higher the fitness, the bigger the trap is. When an ant falls in the trap antlion shoot sand on it; as a result, the ant slides down towards the trap. To realize this scenario crossover operator of GA is used. In this step crossover between one selected antlion and one ant is performed. The operation is pictorially represented in Figure 4. One sub-string of an ant is selected randomly, and that substring is copied into the corresponding antlion.

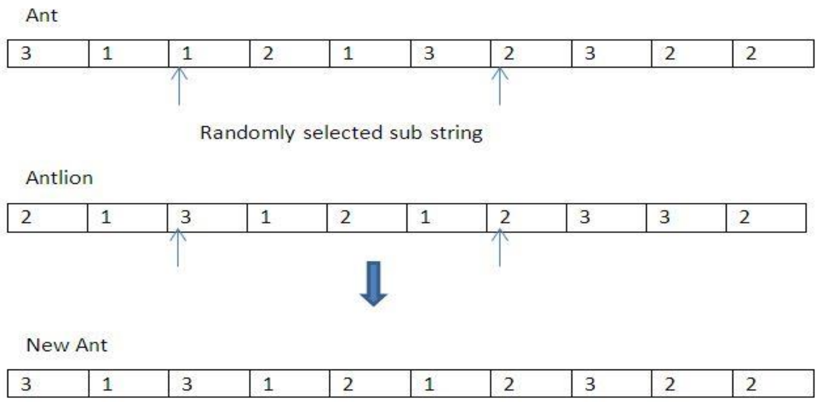

Figure 4. Representation of crossover operation

Catching of prey, re-construction of pit

The final step of ALO reaches after an antlion catches the prey. To mimic the step, it is considered that catching of ant happens when prey is going to be fitter than the corresponding antlion. Then the antlion will change the location to the corresponding ant to increase the chance of catching a new pre. The above scenario is mathematically represented by the equation (15).

$$Antlion_j^t = Ant_i^t \, if \, f(Ant_i^t) > f\left(Antlion_j^t\right) \qquad (15)$$

where t shows the current iteration, Antlion $_j{}^t$ shows the position of selected jth antlion at t^{th} iteration, and Ant^t_j indicates the position of ith ant at tth iteration.

Elitism is one of the most important properties of evolutionary algorithms. Elitism allows preservation of one or more good solution(s) in one generation for the next generation. In continuous ALO it is assumed that the elite solution will influence random walk of every ant. In this paper, we have chosen 5% solutions from the population of Antlion as elite, and they replace the worst antlions after the selection for the next generation.

5.4. Pseudo codes the 2-opt algorithm

Croes et al. (1958) have developed the 2-opt technique to solve the TSP. It is a local search algorithm. The pseudo code for the 2-opt is given below.

```
Input: cost matrix C, number of city Nc
do {
minchange = 0;
 for (i = 0; i< Nc-2; i++)
      {
      for (j = i+2; j <Nc; j++)
              {
              change= C(i,j)+C(i+1,j+1)-C(i,i+1)-C(j,j+1);
              if (minchange> change)
                      {
                      minchange = change;
                      mini = i; minj = j;
                      }
              }
      }
```

} while (minchange< 0);

5.5. Pseudo codes the Discrete ALO algorithm

Input: Number of city n, Number of depot m, Cost matrix C, Number of vehicles available in each depot, Vehicle capacity Q.

- Perform a random Initialization of ant's population and antlions' population.
- Find the ant's fitness and the antlions' fitness
- Search the best antlion to make it elite
- while the termination condition is not satisfied
- for every ant in the population
- Select an antlion using Roulette wheel selection
- perform a random walk
- Update the position of the ant
- end for
- Calculate the fitness of all ants
- Replace an antlion with its corresponding ant if it becomes fitter using equation 15.
- Update elite if an antlion becomes fitter than the elite
- end while
- Return elite

6. Result and Discussion

The discrete ALO is implemented in C language on Intel Core i5 CPU (2.30 GHz), 4GB RAM. The performance of the MDVRP is evaluated using some of the benchmark problems proposed by Creviera et al. (2007) taken from http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/ online resource of University of Malaga, Spain. The specifica-tion of some of the benchmark problems is given in Table 1.

Table 1. Specification of benchmark instances

Instance	P01	P02	P03	P04	P06
Total Number of customer	50	50	75	100	100
Total Number of depots	4	4	5	2	3
Number of the vehicle in each depot	8	5	7	12	10
Vehicle capacity	80	100	140	100	100

The parameters for the proposed Discrete ALO are given in Table 2.

Table 2. Parameters of Discrete ALO

Parameter	Value
Population Size	70 if total customer<50 else 100
Iteration	2500 to 4000
Selection	Roulette wheel
Elitism	5% of total population size, i.e., 5

The solutions of instance p1 are given in Table 3.

Table 3. The solution of Instance P01

Depot	Routes
1	Vehicle 1: 0 25 18 4 0 Vehicle 2: 0 44 45 33 15 37 17 0 Vehicle 3: 0 42 19 40 41 13 0
2	Vehicle 1: 0 48 8 26 31 28 22 0 Vehicle 2: 0 6 27 1 32 11 46 0 Vehicle 3: 0 12 47 0 Vehicle 4: 0 23 7 43 24 14 0
3	Vehicle 1: 0 49 5 38 0 Vehicle 2: 0 9 34 30 39 10 0
4	Vehicle 1: 0 21 50 16 2 29 0 Vehicle 2: 0 35 36 3 20 0

The results of MDVRP instances using discrete ALO guided with 2-opt are compared with the exact solution, solution using Discrete ALO, GA and ACO are presented in Table 4.

Table 4. Comparison of solutions of MDVRP using discrete ALO with GA, ACO and exact solution

Instance	Exact Solution	Discrete ALO guided with 2-opt	Discrete ALO	GA	ACO
p01	576.87	576.87	591.45	598.45	576.87
p02	473.53	473.53	483.15	473.53	473.53
p03	641.15	641.15	694.49	641.18	645.15
p04	1001.04	1003.86	1011.36	1006.66	1001.04
p05	750.03	750.03	750.72	752.39	750.11
p06	876.5	876.5	882.48	877.84	876.5
p07	885.8	885.8	907.55	893.36	888.41
p08	4437.68	4449.65	4450.37	4474.23	4437.68
p09	3895.7	3895.7	4085.51	3900.22	3904.92
p10	3663.02	3663.02	3825.73	3680.02	3666.35
p11	3554.18	3554.18	3732.36	3593.37	3569.68
p12	1318.95	1318.95	1318.95	1318.95	1318.95
p13	1318.95	1318.95	1318.95	1318.95	1318.95
p14	1360.12	1360.12	1365.69	1365.69	1360.12
p15	2505.42	2505.42	2554.12	2549.65	2526.06
p16	2572.23	2572.23	2606.22	2606.22	2572.23
p17	2709.09	2709.09	2733.8	2733.8	2709.09
p18	3702.85	3702.85	3871.01	3781.66	3771.35
p19	3827.06	3827.06	3884.81	3884.81	3827.06
p20	4058.07	4058.07	4058.07	4094.86	4058.07
p21	5474.84	5474.84	5824.58	5668.97	5608.26
p22	5702.16	5702.16	5873.41	5873.41	5708.78
p23	6095.46	6095.46	6129.99	6159.9	6124.67

The percentage of the gap in the result found in the proposed method with the other method in the literature is given in Table 5. The gap is calculated using the following formula.

$$Gap = \frac{(Z_l - Z_p)}{Z_p} * 100 \qquad (26)$$

Where Z_p represents the objective value obtained by the proposed method, and Z_l is the objective value of the problem by the others method. Therefore, the posi tive gap represents the better performance of the proposed algorithm compared to others. Whereas negative gap represents the opposite fact.

Table 5. The percentage of Gap in the result in comparison with other methods

Instance	Exact Solution	Discrete ALO	GA	ACO
p01	0	2.527433	3.740877	0
p02	0	2.03155	0	0
p03	0	8.319426	0.004679	0.623879
p04	-0.28092	0.747116	0.278923	-0.28092
p05	0	0.091996	0.314654	0.010666
p06	0	0.682259	0.152881	0
p07	0	2.455408	0.853466	0.294649
p08	-0.26901	0.016181	0.552403	-0.26901
p09	0	4.872295	0.116025	0.236671
p10	0	4.441963	0.464098	0.090909
p11	0	5.013252	1.102645	0.436106
p12	0	0	0	0
p13	0	0	0	0
p14	0	0.409523	0.409523	0
p15	0	1.943786	1.765373	0.823814
p16	0	1.321421	1.321421	0
p17	0	0.912114	0.912114	0
p18	0	4.541367	2.128361	1.849926
p19	0	1.508991	1.508991	0
p20	0	0	0.906589	0
p21	0	6.388132	3.545857	2.436966
p22	0	3.003248	3.003248	0.116096
p23	0	0.566487	1.05718	0.479209
Average Gap %	-0.02391	2.251911	1.049535	0.297781

From the above table, we observe that 2-opt guided discrete ALO gives a better result than discrete ALO, GA, and ACO in most of the case. It is also found that the proposed algorithm fails to yield the exact solution always. The ACO gives a better result than Discrete ALO guided with the 2-opt technique in case of instance p04, p08.

7. Conclusion

In distribution logistics, two main decision problems are routing and scheduling. The cost of delivering an item from source to the destination is optimized only by efficient routing. Single depot VRP often fails to solve real-life scenario because there exists more than one depot. As an NP-hard problem, MDVRP is very difficult to solve and to find exact solutions by exact methods. In this paper, we proposed a 2-opt local exchange guided discrete antlion optimization algorithm to solve MDVRP. This amalgamation of heuristics with local search gives good result in case of MDVRP. Moreover, the algorithm can be applied to solve similar kind of problem like multi-depot location routing problem, waste collection problem, etc.

Author Contributions: Each author has participated and contributed sufficiently to take public responsibility for appropriate portions of the content.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alinaghian, M., & Shokouhi, N. (2018). Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. Omega, 76, 85-99.

Berger, J., & Barkaoui, M. (2003). A hybrid genetic algorithm for the capacitated vehicle routing problem. In Genetic and evolutionary computation conference (pp. 646-656). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bezerra, S., Souza, S., & Souza, M. (2018). A GVNS Algorithm for Solving the Multi-Depot Vehicle Routing Problem. Electronic Notes in Discrete Mathematics, 66, 167-174.

Chao, I. M., Golden, B. L., & Wasil, E. (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. American Journal of Mathematical and Management Sciences, 13, 371–406.

Chen, A. L., Yang, G. K., & Wu, Z. M. (2006). Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University-Science A, 7(4), 607-614.

Clarke, G., & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12(4), 568–581.

Creviera, B., Cordeaua, J. F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. European Journal of Operational Research, 176, 756-773.

Croes, G. A. (1958). A method for solving traveling salesman problems. Operations Research, 6, 791-812.

Dutta, J., Barma, P., Kar, S., & De, T. (2019). A Modified Kruskal's Algorithm to Improve Genetic Search for Open Vehicle Routing Problem. International Journal of Business Analytics, 6, 55-76.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. Operations research, 42(4), 626-642.

Guimarães, T. A., Coelho, L. C., Schenekemberg, C. M., & Scarpin, C. T. (2019). The two-echelon multi-depot inventory-routing problem. Computers & Operations Research, 101, 220-233.

Ladányi, L., Ralphs, T. K., & Trotter, L. E. (2001). Branch, cut, and price: Sequential and parallel. Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions, 223-260.

Lalla-Ruiz, E., & Voß, S. (2020). A popmusic approach for the multi-depot cumulative capacitated vehicle routing problem. Optimization Letters, 14(3), 671-691.

Lang, M. X. (2018). Study on the model and algorithm for multi-depot vehicle scheduling problem. Journal of Transportation Systems Engineering and Information Technology, 16(15), 65–70.

Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. Annals of Discrete Mathematics. Surveys in Combinatorial Optimization, 31, 147-184.

Laporte, G., Nobert, Y., & Arpin, D. (1984). Optimal solutions to capacitated multi-depot vehicle routing problems. Congressus Numerantium, 44, 283–292.

Laporte, G., Nobert, Y., & Taillefer, S. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. Transportation Science, 22, 161–172.

Li, F., Golden, B., & Wasil E. (2007). The Open Vehicle Routing Problem: Algorithms, Large scale Test Problems, and Computational Results. Computers and Operations Research, 34(10), 2918–2930.

Li, J., Wang, R., Li, T., Lu, Z., & Pardalos, P. (2018). Benefit analysis of shared depot resources for multi-depot vehicle routing problem with fuel consumption. Transportation Research Part D: Transport and Environment, 59, 417-432.

Matos, A. C., & Oliveira, R. C. (2004). An Experimental Study of the Ant Colony System for the Period Vehicle Routing Problem. In: Dorigo M., Birattari M., Blum C., Gambardella L.M., Mondada F., Stützle T. (eds) Ant Colony Optimization and Swarm Intelligence. ANTS 2004. Lecture Notes in Computer Science, vol 3172. Springer, Berlin, Heidelberg.

Mirjalili, S. (2015). The Ant Lion Optimizer. Advances in Engineering Software, 83, 80-98.

Mukherjee, A., Panigrahi, G., Kar, S., & Maiti, M. (2019). Constrained covering solid travelling salesman problems in uncertain environment. Journal of Ambient Intelligence and Humanized Computing, 10, 125-138.

Ombuki-Berman, B., & Hanshar, F. T. (2009). Using genetic algorithms for multi-depot vehicle routing. In Bio-inspired algorithms for the vehicle routing problem (pp. 77-99). Berlin, Heidelberg: Springer Berlin Heidelberg.

Rabbouch, B., Mraihi, R., & Saâdaoui, F. (2017). A Recent Brief Survey for the Multi-Depot Heterogeneous Vehicle Routing Problem with Time Windows. International Conference on Health Information Science, Hybrid Intelligent Systems, 147-157.

Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers and Operations Research, 31, 563–591.

Renaud, J., Laporte, G. & Boctor. F.F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. Computers and Operations Research, 23, 229–235.

Silva, Á., Ferreira, L., Pereira, M., & Moreira, F. (2018). A Model for the Multi-Depot Online Vehicle Routing Problem with Soft Deadlines. International Conference on Innovation, Engineering and Entrepreneurship HELIX 2018: Innovation, Engineering and Entrepreneurship, 818-824.

Taillard, E. D. (1993). Parallel iterative search methods for vehicle routing problems. Networks, 23, 661–673.

Vianna, D. S., Ochi, L. S., & Drummond, L. M. (1999). A parallel hybrid evolutionary metaheuristic for the period vehicle routing problem. In Rolim, J., Mueller, F., Zomaya, A.Y., Ercal, F., Olariu, S., Ravindran, B., Gustafsson, J., Takada, H., Olsson, R., & Kale, L.V. (Eds.), Parallel and distributed processing. Lecture notes in com-puter science. vol. 1586 (183–191). Berlin: Springer-Verlag.

Yuan, W., Wang. J., Li, J., Yan, B., & Wu, J. (2017). Two-stage heuristic algorithm for a new model of hazardous material multi-depot vehicle routing problem. UK Workshop on computational intelligence UKCI 2017: Advances in computational intelligence systems, 362-366.

Zhang, S., Zhang, W., Gajpal, Y., & Appadoo, S. (2019). Ant colony algorithm for routing alternate fuel vehicles in multi-depot vehicle routing problem. Decision Science, 251-260.

Zhou, L., Baldacci, R., Vigo, D., & Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. European Journal of Operational Research, 265(2), 765-778.