Artificial Bee Colony Algorithm for Three-Dimensional Loading Capacitated Vehicle Routing Problem

Bin Wu, Jin-guo Lin, and Min Dong

Abstract The artificial bee colony algorithm (ABC) hybrid two loading heuristics for the three-dimensional loading capacitated vehicle routing problem (3L-CVRP) is presented in the paper. The 3L-CVRP is a combination of two well-known NP-hard problems, the capacitated vehicle routing problem, and the three-dimensional bin packing problem. It is very difficult to get a good performance solution in practice for these problems. The problem is solved by different heuristics for the loading part, and by artificial bee colony algorithm for the overall optimization. To solve the representation problem of the solution, a novel real encoding method is presented to represent the solution for ABC. The effectiveness of the proposed algorithm is tested, and proven by extensive computational experiments on benchmark instances.

Keywords 3L-CVRP • Artificial bee colony algorithm • Bin packing problem • Vehicle routing problem

1 Introduction

The Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP) is a highly complex problem combining three dimensional loading and vehicle routing, which was introduced by Gendreau et al. (2006). The 3L-CVRP calls for the determination of the routes traveled by a fleet of homogeneous vehicles that deliver items to customers such that the total distance traveled by all vehicles is minimized. In addition, the three-dimensional loading plan for each vehicle must be formulated while fulfilling a number of constraints that address issues such as the stability of the items, packing requirements for fragile items, and the convenience of loading

B. Wu (✉) • J. Lin • M. Dong
Department of Industrial Engineering, School of Economics and Management, Nanjing University of Technology, Nanjing 210009, China
e-mail: seasimen@yahoo.com.cn

E. Qi et al. (eds.), *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management*, DOI 10.1007/978-3-642-40063-6_81,

and unloading. The problem is of practical interest in freight distribution since the combination of the vehicle routing problem and the 3-D container loading problem with realistic constraints closely models real-life situations, especially when the delivery involves multiple large items of dissimilar dimensions. Examples are the distributions of household appliances, kitchen components, mechanical components and furniture.

On the other hand, the 3L-CVRP is an extremely hard and challenging optimization problem since it generalizes two of the most well known problems in combinatorial optimization: the Capacitated Vehicle Routing Problem (CVRP), and the Three Dimensional Bin Packing Problem (3D-BPP). During the past decades, in the area of combinatorial optimization, CVRP and 3D-BPP problems have been researched intensively but independently. About CVRP and related vehicle routing problems, e.g. the VRP with time windows (VRPTW), we recommend that readers survey the books by Toth and Vigo (2001) and Golden et al. (2008) and recent reviews (Laporte 2009; Vidal et al. 2013). The 3D-BPP problem and its variants, e.g. 3D strip packing problem (3D-SPP), have been solved to optimality so far. Exact methods for the 3D-BPP were proposed by Martello et al. (2000). Heuristic and meta-heuristic methods for the 3D-BPP were developed by Faroe et al. (2003), Crainic et al. (2008), Egeblad and Pisinger (2009).

Only in the recent years have some researchers pay attention to this combined optimization problem. Gendreau et al. (2006) were the first to consider the vehicle routing problem with three-dimensional loading constraints. A tabu search approach was proposed to address the 3L-CVRP problem, where the three-dimensional loading sub-problem was also solved by tabu search algorithm. Tarantilis et al. (2009) designed a hybrid procedure combining the strategies tabu search and guided local search, and they also devised collection of plain packing heuristics. Fuellerer et al. (2010) generalized their ant colony algorithm to the three-dimensional case by checking loading feasibility through fast and effective packing heuristics. Zhu et al. (2012) improved two packing heuristics, namely the Deepest-Bottom-Left-Fill heuristic and the Maximum Touching Area heuristic, for the three-dimensional loading sub-problem and provided efficient implementations. Based on these two new heuristics, an effective tabu search algorithm is given to address the overall problem. Bortfeldt (2012) presented a new hybrid algorithm including a tabu search algorithm for routing and a tree search algorithm for packing boxes into vehicles for the 3L-CVRP. Qingfang Ruan et al. (2013) proposed a hybrid approach which combines Honey Bee Mating Optimization and six loading heuristics to solve the 3L-CVRP. Moura and Oliveira (2009) studied the combination of the Vehicle Routing Problem with Time Window (VRPTW) and the Container Loading Problem. The number of vehicles is to be minimized with higher priority, whereas the total travel distance is to be minimized with lower priority. They proposed a genetic algorithm with a hybrid meta-heuristic methodology combining the strategies of tabu search to solve the problem. Then both a sequential and a hierarchical approach were proposed.

To the best of our knowledge, there is no detailed work that describes the use of the ABC algorithm to deal with 3L-CVRP. We present ABC algorithm with two loading heuristic to solve the 3L-CVRP. A novel real encoding method is presented

to represent the solution. The ABC algorithm is mainly searches the space of routing solutions, while checking the feasibility of the three-dimensional loading of each route by means of two loading heuristic deepest-bottom-left fill (DBLF) and max touching area (MTA). Our aim is to provide good solutions to the instances, leading to the algorithm of practical use in transportation.

The remainder of this paper is organized as follows: in Sect. 2, a detailed description of the problem is provided. In Sect. 3 the proposed algorithm is described, and followed by the computational results in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Problem Description

Let $G = (V,E)$ be an undirected graph, where $V = (0,1,2, \ldots,n)$ is the set of $n + 1$ vertices corresponding to a depot, represented by vertex 0, and n clients, denoted by vertices $1, \ldots, n$ and E is the set of edges. The cost of an edge (i,j) is denoted by c_{ij}. There are K identical vehicles available; each vehicle has a weight capacity D and a three-dimensional rectangular loading space $S = W \times H \times L$ defined by width W, height H and length L. Each client $i(1,2, \ldots,n)$ requires the delivery of a set of m_i three-dimensional items $I_{it}(t = 1, 2, \ldots, m_i)$ having width w_{it}, height h_{it} and length l_{it} with total weight d_i.

In 3L-CVRP, we assume all items are rectangular boxes. The items can only be placed orthogonally inside a vehicle; however, items can be rotated 90° by on the width-length plane. Some items are fragile; only fragile items can be placed on top of other fragile items, whereas any item can be placed on top of a non-fragile item. The stability of the packed items is important; one method to ensure stability is to require items that are placed on top of other items to have sufficient supporting areas. A packing is feasible if all items are either placed directly on the floor of the vehicle or on top of other items with total supporting area of at least α percent of their base areas. Another important requirement is to ensure that items can be easily unloaded; this is approximated by a Last-In-First-Out (LIFO) policy. When client i is visited, all of its corresponding items I_{it} must not be stacked beneath nor be blocked by items of clients that are to be visited later; an item A is beneath B if the interior of the projections of their bases on the vehicle floor intersect, and the top of A is not higher than the bottom of B in the vertical direction. An item is also considered blocked if it will overlap any item of a later client when it is moved along the L axis toward the rear door.

The objective of 3L-CVRP is to find a set of at most K routes (one per vehicle) such that:

1. Every vehicle starts from the depot, visits a sequence of clients and returns to the depot.
2. All clients are served, and every client is served by exactly one vehicle.
3. No vehicle carries a total weight that exceeds its capacity.

4. All items demanded by all the clients served by a vehicle can be orthogonally packed into that vehicle while satisfying the following loading constraints:
   (1) (Fragility constraint) no non-fragile items are placed on top of fragile items.
   (2) (Supporting area constraint) all items have a supporting area of at least a percent of their base area.
   (3) (LIFO constraint) all items fulfill the LIFO policy.
5. The total cost of all edges included in the routes is minimized.

3 The Proposed Algorithm

3.1 Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) algorithm is a new swarm intelligence technique inspired by intelligent foraging behavior of honey bees. The first framework of ABC algorithm was presented by Karaboga and Basturk (2007, 2008). In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. In the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. At the first step, the ABC generates a randomly distributed initial population of *SN* solutions (food source positions) according Eq. (1).

$$x_i^j = x_{\min}^j + rand\,() \left(x_{\max}^j - x_{\min}^j \right) \quad i \in SN, j \in D \tag{1}$$

Where *SN* denotes the size of population. Each solution x_i where $i = 1, 2, \ldots SN$ is a *D*-dimensional vector. Here, *D* is the number of optimization parameters. x_{max}^j and x_{min}^j are lower and upper bounds of parameter *j*, respectively.

After initialization, the population of the positions (solutions) is subjected to repeated cycles $C = 1, 2 \ldots MCN$ of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) using the formula (2) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher

than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory.

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} - x_{kj}\right) \tag{2}$$

Where a food source v_i is determined by changing one parameter of x_i. $k \in \{1, 2, \ldots SN\}$ and $j \in \{1, 2, \ldots D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. φ_{ij} is a random number between [−1,1]. It controls the production of neighbor food sources around x_{ij} and represents the comparison of two food positions visible to a bee.

After all employed bees complete the search process; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability value associated with that food source p_i calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{3}$$

Where fit_i is the fitness value of solution i.As in the case of the employed bee, it produces a modification on the position using formula (3) in its memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. If a position cannot be improved further through a predetermined value named *"limit"*, then that food source is assumed to be abandoned. The corresponding employed bee becomes a scout. The abandoned position will be replaced with a new food source found by the scout.

3.2 ABC for 3L-CVRP

For these state-of-arts algorithms such as ABC, encoding method is the key problem and also is a kind of art problem. When ABC was used to solve the combinatorial optimization problems, usually has two encoding method: real number encoding and integer number encoding. For the real number encoding method, such as random-key procedure is always used to solve the combinatorial optimization problems. For the integer number coding method, they are more used since can avoid many coding issues. For ABC algorithm, the new operators should be designed when the integer number was used. To avoid the problem, a new real number encoding method was presented in the paper.

Table 1 Example for Step2

	1	2	3	4	5	6	7	8
{X1}	{1245	3422	2345	3004	5434	6759	4355	8432}
{X2}	(63	94	61	53	32	87	59	23)

Table 2 Example for Step3

	1	2	3	4	5	6	7	8
$X1' = \{X1\} \bmod 8$	{5	6	1	4	2	7	3	0}
$X1' + 1$	{6	7	2	5	3	8	4	1}
(X2) mod 3	(0	1	1	2	2	0	2	2)

3.2.1 Solution Representations and Decoding Methods

Solution Representation

For n customers (non depot) k vehicles 3L-CVRP problem, the n real numbers as the representation has the form of a vector of length n.

For example: 8 customers, 3 vehicles.

$X(x_1,x_2 \ldots,x_8) = (1.24563,\ 3.42294,\ 0.0234561,\ 300.453,\ 54.3432,\ 6759.87,\ 43.5559,\ 8.43223)$.

Decoding Procedure

The next procedure is to decode the representation to the solution. The detail procedure is below:

***Step1*:** Remove the decimal point of x_i; transform x_i to integer number. For example: Change X to [X] = (124563, 342294, 234561, 300453, 543432, 675987, 435559, 843223)

***Step2*:** Segregate x_i two parts: {X1} and (X2). {X1} has four significant figures and (X2) has two significant figures. You can also adjust the number according to the instance size. For example (Table 1):

***Step3*:** Modulo operation. the first part modulo n, the second part modulo the number of vehicles k. Then, the first part is plus 1. For example (Table 2).

***Step4*:** Map the element of $X1'$ to the current vehicle in the current position X2, and the visited order is the order of customer appearance. So, the customer 6 corresponds to the vehicle 0, and customer 6 is the first visited in the route 0.

The solution is below:
Route 0: 0-6-8-0 Route 1: 0-7-2-0 Route 2: 0-5-3-4-1-0.

Revise the Solutions and the Representations

After decoding, maybe some solutions are infeasible. Two issues should be considered. (1) Maybe some customers are duplicated and some missed. (2) Maybe some route will exceed the vehicle capacity. So, we should check the solution, and revised the duplicated customers to the missed customers.

3.2.2 Local Search Procedure for Route

In previous works, many researchers used the meta-heurist algorithms hybrid the local search operator for the route. The local search operator can improve the solution quality very much. In the paper, 2-Opt and 3-Opt were used to improve the solution after decoding and check the solution. The 2-Opt is used to search in the inter route and intra route. The 3-Opt is mainly used to search in the intra route.

3.2.3 Algorithms for Three-Dimension Loading

In this subsection, we want to determine whether all the items needed by customers in a given route can be loaded into a single vehicle while satisfying all loading constraints depicted in Sect. 2. The heuristics presented to solve this problem must be flexible because they should be adaptable to consider all loading constraints or only a subset of them. We make use of the following procedures, aiming at a feasibly loading of items demanded by customers along the route.

To compute the potential infeasibility of a route, we first easily check if the weight constraint is not violated, and then apply the lower bounds for the 3BPP by Martello et al. (2000). If the lower bounds do not prove the infeasibility, then we repeatedly apply the two heuristics: deepest-bottom-left-fill algorithm and max touching area algorithm. These heuristics place the items according to a given sequence, one at a time, into a container of width W, height H and infinite length. Their aim is to find a feasible loading of minimum length, which does not exceed the vehicle length L. By changing the sequence of the items and again invoking the heuristics one can obtain new solutions. This process is iterated θ times or until a feasible solution has been found.

The initial loading sequence is constructed by taking the various loading constraints into account. We sort the items by applying the following sorting rules in order:

***SR1*:** (applies only if the LIFO constraint is to be enforced) Sort the items in reverse order of the clients to be visited. Hence, the items for later customers will be loaded first. Ties are broken using the next rule SR2.

***SR2*:** (applies only if fragility constraint is to be enforced) Sort the items so that non-fragile items are before fragile items. We would like to load the non-fragile items first since fragile item can be placed on top of non-fragile items, but the reverse is prohibited. Ties are broken using the next rule SR3.
***SR3*:** Sort the items by decreasing order of volume.

The deepest-bottom-left-fill algorithm packs the current item into the normal position which has lowest width, breaking ties by lowest height, breaking ties by lowest length. For each position, both feasible orientations on the *w*-*l* plane are considered. The first packing satisfying the loading conditions is selected and the process is iterated until all items are packed or no feasible packing exists for an item.

The max touching perimeter algorithm generates among all the possible normal positions for the current item, selects the one maximizing the percentage of the item surface touching the container and other items already packed. In this case too, the process is iterated until all items are packed or no feasible packing exists for an item.

4 Computational Results

In this section, the proposed algorithm is analyzed using several benchmarking problems. The proposed algorithm was coded in Visual C++, executed on an Intel Core2 i5, 2.5 GHz with 4 GB of RAM under Windows 7. The algorithm was tested on the classical 3L-CVRP instances, which were proposed by Gendreau et al. (2006). And they are available at http://www.or.deis.unibo.it/research.html. In the instances, the graphs, customers demand, and vehicle weight capacity are taken from CVRP instances. The arc costs are determined as the Euclidean distances between coordinates of customers. The vehicle loading volume has dimensions $W = 25$, $H = 30$, and $L = 60$. For each customer the number of required items is randomly generated according to a uniform distribution between 1 and 3. Each item dimension is randomly generated according to a uniform distribution in the interval between 20 and 60 % of the corresponding vehicle dimension. The minimum supporting area is set equal to 0.75. In the experiment, we select 10 instances from the benchmark.

The parameters of the proposed algorithm have remarkable effect on the quality and effectiveness of the algorithm. Therefore, with the help of the initial experiments, the algorithm parameters are set as follows: the population size *NP* is 60, the max iteration *MCN* is 500, the max try number *limit* is 50. In the following experiments, each problem is independently run 10 times. The results for every instance are the mean values of 10 times.

In Table 3, we compare its performance of ABC with the solutions solved by TS, GTS, ACO approaches proposed in the literatures (Gendreau et al. 2006; Tarantilis et al. 2009; Fuellerer et al. 2010), respectively. The first columns indicate the instance name, the next two columns report the best results obtained by TS and

Table 3 Comparisons of ABC with TS, GTS, ACO on 3L-CVRP instances from the literature

Instances	TS (Gendreau et al. 2006)	GTS (Tarantilis et al. 2009)	ACO (Fuellerer et al. 2010)	ABC
E016-03m	316.01	321.47	305.35	**302.46**
E021-06m	448.48	458.04	440.68	**439.43**
E026-08m	666.10	642.22	635.50	**631.88**
E030-04g	819.36	873.43	821.04	**793.62**
E036-11h	707.85	698.61	698.92	698.93
E041-14h	920.87	872.79	870.33	**868.74**
E045-04f	1400.52	1296.59	1300.9	**1258.51**
E051-05e	871.29	818.68	781.29	**778.76**
E072-04f	732.12	641.57	611.26	**605.31**
E101-10c	1847.95	1711.93	1616.99	**1597.23**

Table 4 Performance of the ABC algorithm for different loading configurations

Instances	No fragility	No LIFO	No support	3D loading only
E016-03m	302.46	297.13	296.32	295.72
E021-06m	439.43	430.88	430.88	430.88
E026-08m	631.88	615.76	617.07	609.43
E030-04g	793.62	761.28	750.73	714.55
E036-11h	698.93	698.61	698.92	698.61
E041-14h	861.57	859.21	862.16	854.74
E045-04f	1250.73	1224.17	1198.54	1157.73
E051-05e	740.26	734.43	732.91	671.91
E072-04f	576.26	562.37	562.74	515.39
E101-10c	1529.43	1454.75	1403.68	1403.33
gap	1.54 %	3.6 %	4.22 %	6.8 %

GTS (Because only best results were reported in their papers). The average results produced by ACO and our proposed ABC algorithm are in the last two columns. The best results of every instance for all the algorithms are denoted by bold.

In terms of solution quality the ABC algorithm is clearly superior to the others. The average solution values found by the ABC is worse than the one found by the ACO and GTS in just one case (E036-11h), while in all other cases it is better. In all instance, the average of solutions found by ABC is better 7.32 % than the one found by TS. For GTS except the instance E036-11h, the average performance of ABC is better than 4.13 %. We lastly note that the average quality of ABC is a little better 1.11 % than the ACO.

In Table 4 we examine the effect of the loading constraints discussed in Sect. 2. We run ABC with four different loading constraint configurations: (1) without the fragility constraint; (2) without the supporting area constraint; (3) without the LIFO policy, and (4)with only the 3D loading constraint, it means all three aforementioned constraints are not taken in to account.

From the results, we can see that although all three constraints have a significant effect on the cost of the final solution, the LIFO and supporting area constraints have a more significant effect than the fragility constraint. We compute the percentage differences between the average solution values found for each loading configuration and the ones found for the standard configuration (gap). The standard constraints include all the constraints and the solutions have been given in last column of the Table 3. The removal of the LIFO constraint and supporting area constraint leads to the larger reduction of 3.6 and 4.22 %, respectively. Removing the fragility constraint leads to the lowest reduction. While removing the fragility constraint only finds a weak reduction of 1.54 % in the average solution value. The configuration with no operational constraints has by far better solution values, with cost reductions close to 6.8 %.

5 Conclusion

In this paper, the 3L-CVRP, as a new variant of vehicle routing problem, is studied. This is a very interesting problem in terms of both theoretical complexity and practical applications. We have developed an ABC algorithm with two loading heuristics to solve the problem. To the best of our knowledge, it is the first time for ABC meta-heuristics to be employed in this combinatorial problem. A novel real number encoding method is presented for the solution representation. Two types of loading heuristics are presented as soon as we obtain the solution of the vehicle routing sub-problem. For testing the robustness and effectiveness of the hybrid approach, 10 benchmark instances of 3L-CVRP are solved. From the computational results, it is shown that the hybrid approach obtains many more optimal solutions.

Acknowledgment This work is supported by Humanities and Social Science Foundation of Ministry of Education (Grant No. 11YJCZH184), Natural Science Foundation of Jiangsu Province (Grant No. BK2010555) and "Outstanding Youth Plan of Nanjing University of Technology".

References

Bortfeldt A (2012) A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. Comput Oper Res 39:2248–2257

Crainic TG, Perboli G, Tadei R (2008) Extreme point-based heuristics for three-dimensional bin packing. Informs J Comput 20:368–384

Egeblad J, Pisinger D (2009) Heuristic approaches for the two- and three-dimensional knapsack packing problem. Comput Oper Res 36:1026–1049

Faroe O, Pisinger D, Zachariasen M (2003) Guided local search for the three-dimensional bin-packing problem. Informs J Comput 15:267–283

Fuellerer G, Doerner KF, Hartl RF, Iori M (2010) Metaheuristics for vehicle routing problems with three-dimensional loading constraints. Eur J Oper Res 201:751–759

Gendreau M, Iori M, Laporte G, Martello S (2006) A tabu search algorithm for a routing and container loading problem. Transp Sci 40:342–350

Golden B, Raghavan S, Wasil E (2008) The vehicle routing problem: latest advances and new challenges. Springer Press, Berlin

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. J Global Optim 39:459–471

Karaboga D, Basturk B (2008) On the performance of artificial bee colony algorithm. Appl Soft Comput 8:687–697

Laporte G (2009) Fifty years of vehicle routing. Transp Sci 43:408–416

Martello S, Pisinger D, Vigo D (2000) The three-dimensional bin packing problem. Oper Res 48:256–267

Moura A, Oliveira JF (2009) An integrated approach to the vehicle routing and container loading problems. OR Spectr 31:775–800

Qingfang Ruan, Zhengqian Zhang, Lixin Miao, Haitao Shen (2013) A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. Comput Oper Res 40:1579–1589

Tarantilis CD, Zachariadis EE, Kiranoudis CT (2009) A hybrid meta-heuristic algorithm for the integrated vehicle routing and three-dimensional container loading problem. IEEE Trans Intell Transp Syst 10:255–271

Toth P, Vigo D (2001) The vehicle routing problem. SIAM Press, Philadelphia

Vidal T, Crainic TG, Gendreau M, Prins C (2013) Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. Eur J Oper Res 231(1):1–21

Wenbin Zhu, Hu Qin, Andrew Lim, Lei Wang (2012) A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. Comput Oper Res 39:2178–2195