RAIRO-Oper. Res. 46 (2012) 63–82
DOI: 10.1051/ro/2012008

RAIRO Operations Research
www.rairo-ro.org

A HYBRID GENETIC ALGORITHM FOR THE VEHICLE ROUTING PROBLEM WITH THREE-DIMENSIONAL LOADING CONSTRAINTS

Lixin Miao[1], Qingfang Ruan[1], Kevin Woghiren[2] and Qi Ruo[1]

Abstract. This paper addresses a Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP) which combines a three-dimensional loading problem and vehicle routing problem in distribution logistics. The problem requires the combinatorial optimization of a feasible loading solution and a successive routing of vehicles to satisfy client demands, where all vehicles must start and terminate at a central depot. In spite of its clear practical significance in the real world of distribution management, 3L-CVRP in literature is very limited for its high combinatorial complexity. We solve this problem by a hybrid approach which combines Genetic Algorithm and Tabu Search (GATS). Genetic algorithm is developed for vehicle routing and tabu search for three-dimensional loading, while these two algorithms are integrated for the combinatorial problem. We computationally evaluate this hybrid genetic algorithm on all publicly available test instances, and obtain new best solutions for several instances.

Keywords. Vehicle routing, three-dimensional loading, genetic algorithm, tabu search.

Mathematics Subject Classification. 97M40.

1. Introduction

The Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP) is a very complex problem which combines two NP-hard problems called

Received January 28, 2011. Accepted April 2, 2012.

[1] Research Center for Modern Logistics, Graduate School at Shenzhen, Tsinghua University, 518055 Shenzhen, P.R. China. miaolx@sz.tsinghua.edu.cn; rqf09@mails.tsinghua.edu.cn; luoqi0601@gmail.com

[2] HSBC Business School, Graduate School at Shenzhen, Peking University, 518055 Shenzhen, P.R. China. Kwoghiren@gmail.com

Article published by EDP Sciences © EDP Sciences, ROADEF, SMAI 2012

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

three-dimensional loading and vehicle routing. 3L-CVRP is initially introduced in a seminal paper [16] and it requires the determination of a set of goods whose total weight and volume cannot exceed vehicle weight capacity and loading space respectively, a feasible placement of goods in the loading space and several routes transported by a vehicle fleet for shipping goods to a number of clients for the minimization of the number of vehicles and total transportation cost.

This combinatorial problem is of significant value from both practical and theoretical viewpoints. From the practical application, 3L-CVRP is especially relevant for the cases that suppliers have to deal with large goods and the operations of loading have to be considered seriously. *i.e.*, when one is distributing soft drinks, kitchen components, auto parts, and household appliances, the loading problem must be taken into account. It has become increasingly popular in recent years in the field of distribution management. Concerning theoretical importance, 3L-CVRP is a very challenging problem for it includes two well known NP-hard optimal problems: three-dimensional Loading Problem (3LP) and Capacitated Vehicle Routing Problem (CVRP).

The 3LP is particularly related to various container loading problems. It is constrained by the supporting surface, the fragility of goods, and the sequential loading or unloading which are considered in the literatures [4, 11, 25, 30, 36]. The 3LP calls for the determination of a given set of three-dimensional rectangular goods into the minimum number of three-dimensional containers (bins), while ensuring goods are completely contained within containers. Although the 3LP is solved by some effective heuristics and meta-heuristics in the literature [7,8,12,27], several instances in [27,28] with less than 50 goods are still unsolved to optimality due to their special difficult quality.

The CVRP is one of the most widely researched problems in combinatorial optimization. It calls for determining the set of routes of minimal transportation cost, where a set of consignments are shipped by a vehicle fleet. For each vehicle, the total weight volume of loaded goods cannot exceed vehicle weight capacity while the specific loading is not considered. Its thorough review is in the classical volume [34]. Different effective solution methods are introduced for CVRP: *i.e.*, an exact algorithm combining branch-and-cut and branch-and-price is presented in [15], some other meta-heuristics are proposed in [5, 6, 31, 35].

To the best of our knowledge, loading and routing problems have been studied widely but independently. Only few papers concern the combined optimization of vehicle routing and three dimensional loading problems. More precisely, these combinatorial problems are almost motivated for the practical cases. For example, a seminal paper [16] is exploded especially by a difficult loading problem with different shaped goods of high risk and the cost of being damaged during transportation; the paper [9] is studied for the delivery of timber chipboards, where Tabu Search (TS) and an Ant Colony Optimization (ACO) are presented to solve the problem; authors in [14] are capable of dealing with large-size instances, they introduce a meta-heuristics for 3L-CVRP to provide high quality solutions with reasonable

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

computational effort. A systematic survey of this field is presented recently in [21], which can be useful both to practitioners and academic researchers.

The 3L-CVRP generalizes another loading and routing problem known as the Two-dimensional Loading Capacitated Vehicle Routing Problem, denoted as 2L-CVRP. In 2L-CVRP, each vehicle has a two-dimensional rectangular space, goods with a two-dimensional rectangular space for being loaded to vehicles. The aim of 2L-CVRP is to optimize the distribution cost of the given goods which cannot be stacked one over the other. The 2L-CVRP proposed initially in [19, 20] is solved by exact and meta-heuristic algorithms. Subsequently it is improved by several methods such as the branch-and-cut algorithm [22], tabu search heuristics [17], GRASP × ELS approach [10], memetic algorithm [24], hybrid meta-heuristics [33], and ACO [13], respectively.

In this paper we propose a hybrid approach based on the Genetic Algorithm and Tabu Search (GATS) for 3L-CVRP. It is motivated by the excellent results produced from both the classical routing problem [2], where the competitive solution time and quality is obtained by genetic algorithm (GA), which owns some efficient features to be depicted in the third section, and three-dimensional loading [16], in which a feasible loading pattern can easily be obtained and the computational time can be rapidly reduced by tabu search (TS) and two types of loading heuristics possessing some strengths. Our aim is to generate high quality solutions with reasonable computational effort, therefore developing a hybrid genetic algorithm being able to solve large-size cases and being probably useful for real world delivery. We deal with this problem by carefully merging and tailoring techniques from the literature to the problem.

The main contribution of this paper is twofold. Firstly, using the hybrid genetic algorithm, we produce very good results on all publicly available test instances improving the best previous approaches on average by 0.82%, and in some cases even by more than 9%. Secondly, we show that these solutions can be found very fast by GATS, the average of computational times being in less than 60 s. This is in contrast to the current results in the literature where computation time is slower.

The remainder of this paper is outlined as follows: a more specific description of 3L-CVRP, its loading constraints and problem model, and an example are given in Section 2. A hybrid genetic algorithm is presented in Section 3. In Section 4 computational results are proposed and analyzed. Some conclusions are drawn in the final section.

2. Problem description

2.1. Problem statement

Let $A = (B, C)$ be a complete graph, where $B = \{0, 1, \ldots, n\}$ is a set of $n + 1$ vertices corresponding to a depot (vertex 0) and n clients (vertices $1, \ldots, n$), and C a complete set of edges connecting each vertex pair (i, j) $(i, j = 0, 1, \ldots, n)$.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

Each edge has an associated routing cost c_{ij}. Given a fleet of $v(p = 1, \ldots, v)$ identical vehicles, each of which has a weight capacity D and a three-dimensional rectangular loading space S addressed by width W, height H, and length L. Denote by $S = W \cdot H \cdot L$ the available rectangular loading space. Supposed each vehicle has an opening as large as the vehicle $(W \cdot H)$ on the rear for the loading and unloading operations. Each customer i $(i = 1, \ldots, n)$ requires a set of m_i three-dimensional items $I_{ik}(i = 1, \ldots, n, k = 1, \ldots, m_i)$ having width w_{ik}, height h_{ik}, and length l_{ik}, whose total weight is d_i. Let $s_i = \sum_{i=1}^{m_i} w_{ik} h_{ik} l_{ik}$ denote the total amount of loading space needed by customer i. The notation is based on [14].

The 3L-CVRP calls for the determination of a set of at most v routes entirely starting and ending at the depot, a feasible three-dimensional loading and the minimization of the total cost, given that there exists a placement of goods in the volume that satisfies 3L-CVRP constraints deriving from the nature of goods, the stability of goods, and other practical transportation regulations, which are presented as follows:

- Visiting rule: all items of each client only can be distributed once, hence split deliveries are not allowed.
- Containing request: all goods are completely contained in vehicles, no parts of goods outside of vehicles.
- Fragile requirement: goods are divided into two types: fragile and non-fragile. If I_{ik} is fragile, a fragility flag f_{ik} is equal to 1, and 0 otherwise $(i = 1, \ldots, n, k = 1, \ldots, m_i)$. Fragile and non-fragile goods can be stacked on top of each other respectively, however non-fragile items cannot be placed on top of fragile ones.
- Loading constraint: the loading must be orthogonal. Goods usually have a fixed top with respect to the height in transportation. Goods can only be rotated by 90° on the horizontal plane.
- Sequential loading policy: last-in-first-out policy introduced in [14,22] is a common request in the loading and delivering environments. It is denoted in the following as a sequential loading constraint. When visiting a client, all his goods must be unloaded without moving goods of clients visited later through a sequence of straight shifts parallel to the L-edge along the route. That is to say, no goods required by the successive clients can be placed between I_{ik} and the rear of vehicle or on top of I_{ik}.
- Up-down unloading limitation: if item i is on top of item j (the top of item j is under the bottom of item i, but does not necessarily touches it), item j cannot be unloaded before the unloading of item i, and the items which are on top item i.
- Supporting area constraint: when item I_{ik} is placed on top of other goods, its base should be supported by a minimum supporting area. In other words, the goods placed under I_{ik}, must form a cumulative area $\bar{A} \geq \lambda w_{ik} l_{ik}$, where $0 \leq \lambda \leq 1$ is a given parameter which represents a minimum fraction of area of I_{ik} to be supported. Obviously, if an item is placed on the base of the vehicle directly, the supporting area constraint will be always satisfied.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

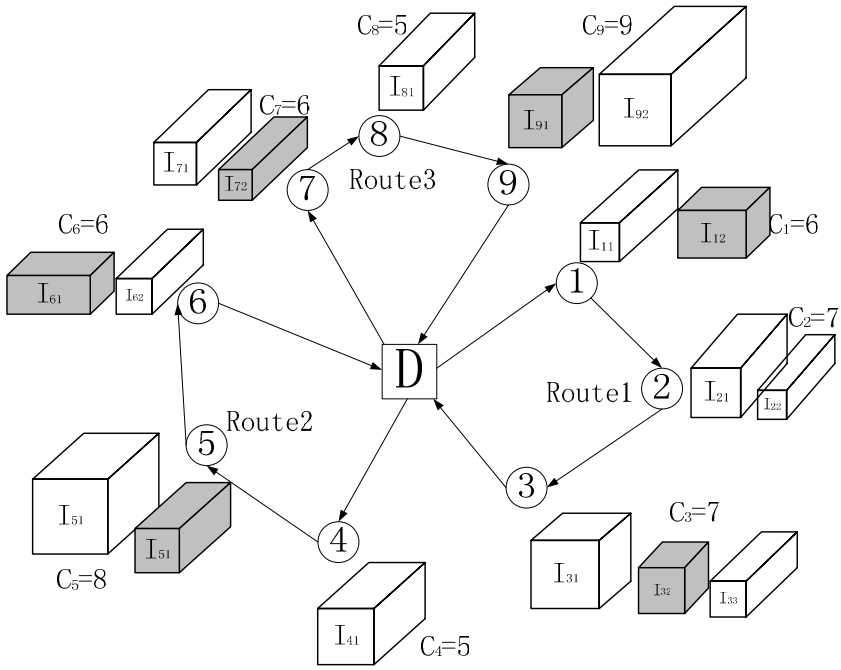


FIGURE 1. A simple 3L-CVRP example.

The goal of 3L-CVRP is twofold. First, it minimizes the number of usage of vehicles and the total transportation cost of routes. Secondly, it addresses a feasible loading stack in the first place. Hence, the 3L-CVRP model can be divided into two groups, one for VRP and the other for 3LP. The objective function and constraints of VRP model are mathematically formulated as shown in [1, 14, 23, 32, 33, 37].

2.2. Problem example

For the sake of good understanding of 3L-CVRP and its constraints, see, for example, Figure 1, where nine clients demand for a total of 17 weighted goods, to be delivered through vehicles based at a central depot. Fragile goods are depicted in grey color. The total volume of goods loaded in the vehicle does not exceed the vehicle weight $D = 20$. A possible solution formed by three routes is proposed.

In Figure 2, feasible three-dimensional loadings corresponding to the routes of Figure 1 are shown: no split delivery happens; all goods are completely contained in vehicles; no non-fragile item is placed on top of fragile goods; the loading is orthogonal; each item base is supported, partially or completely, by the surface of vehicles or other goods (see the loading of goods I_{11} and I_{32} of route 1, I_{72} and I_{91} of route 3 in Fig. 2); all items can be unloaded without moving items of the clients visited later. For instance, recalling the loading of route 2, all goods of client 4

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

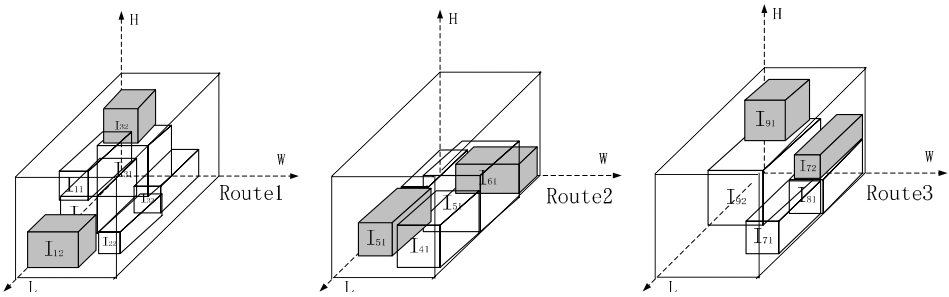


FIGURE 2. Feasible three-dimensional loadings for the routes of Figure 1.

can be unloaded without shifting goods from client 5 or 6, after distributing the goods of client 4, the goods of client 5 can be unloaded without moving any goods of client 6.

In the version of 3L-CVRP, when removing some of the constraints described above, some different loading and routing problems representing different practical problems in transportation will be obtained. It is also of interest to estimate the solution difference between one loading structure and another for understanding the cost implications of different constraints. We will analyze and estimate these different influences in Section 4.

3. A HYBRID GENETIC ALGORITHM

To our best knowledge, both the capacity vehicle routing problem and the three-dimensional problem are NP-hard problems, and the combinatorial problem 3L-CVRP is clearly also the case. Therefore, we prefer to employ heuristic and meta-heuristic methods instead of an exact algorithm to solve this combinatorial problem within reasonable time. A number of initial feasible solutions for 3L-CVRP are initially generated by Sweep Approach (SA) [18], which are optimized by a genetic algorithm (GA). We then solve the loading sub-problem using TS method. In Section 3.1 GA is set up to construct the solutions of VRP. In Section 3.2 TS is proposed to find a feasible solution of 3LP. In Section 3.3 a hybrid genetic algorithm is presented to solve 3L-CVRP, its procedure is introduced in detailed.

3.1. GA FOR VEHICLE ROUTING

At the start of the process of GA for VRP, we produce an initial population of structured solutions using SA proposed by [3, 26, 29]. Four stages are executed in SA. First, the consignments around the depot are numbered and sorted clockwise. Second, the vehicles are numbered and each of them is required to serve the consecutive customers from a randomly-selected customer base until a constraint violation occurs. Third, a given number of initial solutions are constructed to build a population, each of which is calculated through the routing distances and related

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

TABLE 1. The solution depiction of Figure 1.

Customer	1	2	3	4	5	6	7	8	9
Solution	1	1	1	2	2	2	3	3	3

TABLE 2. The application of 2-point crossover of the parents.

Customer	1	2	3	4	5	6	7	8	9
Parent 1	**1**	**1**	**3**	**2**	1	2	3	2	**3**
Parent 2	1	3	2	1	**3**	**1**	**2**	2	3

TABLE 3. The result of 2-point crossover for the offspring.

Customer	1	2	3	4	5	6	7	8	9
Offspring 1	**1**	**1**	**3**	**2**	**3**	**1**	**2**	**2**	**3**
Offspring 2	1	3	2	1	1	2	3	2	3

to a fitness value of GA. Fourth, for the sake of crossover of GA, the route of each solution of VRP is described as a line which is marked with the decimal number of vehicle. For example, the solution of Figure 1 is depicted as follows:

In Table 1, customers 1, 2, 3 are marked with vehicle 1, customers 4, 5, 6 with vehicle 2, customers 7, 8, 9 with vehicle 3. The chromosome of solution is depicted as line 1-1-1-2-2-2-3-3-3 according to the sequence number of customers.

The structured solutions of initial population will be optimized by a reproductive process and a replacement scheme of GA. In the reproductive process, two parent solutions in the population are chosen by roulette method [2], which is employed to generate offspring with a standard 2-point crossover procedure. These 2-points in the chromosome are selected randomly, see Table 2. Crossover points are generated randomly between customers 4 and 5, and between customers 8 and 9. Columns depicted as lines are inserted in the corresponding locations in Table 2 and are divided into two sectors. An application of 2-point crossover yields the following offspring, each having the vehicle allocations from parent 1 with one sector and from parent 2 within the other sector in Table 3.

Customers 1, 2, 5 are served by a same vehicle in Table 2, while customer 5 is replaced by customer 6 in Table 3. The former route for customers 1, 2, 5 is broken, and a new route must be set up for customers 1, 2, 6. In this paper, we make use of TS (readers can refer [16] for a detailed description) to find loadings of solution of each offspring. If offspring duplicate existing members of the population, it will be removed. If offspring have worse fitness than their parents, it will be deleted. If offspring violate volume and/or capacity constraints, it will be removed. If offspring have better gene values than their parents, they will be selected to stay in the population. In each step of iterative process of this algorithm, a simple mutation of chromosomes is applied in the population for the improvement of solutions.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

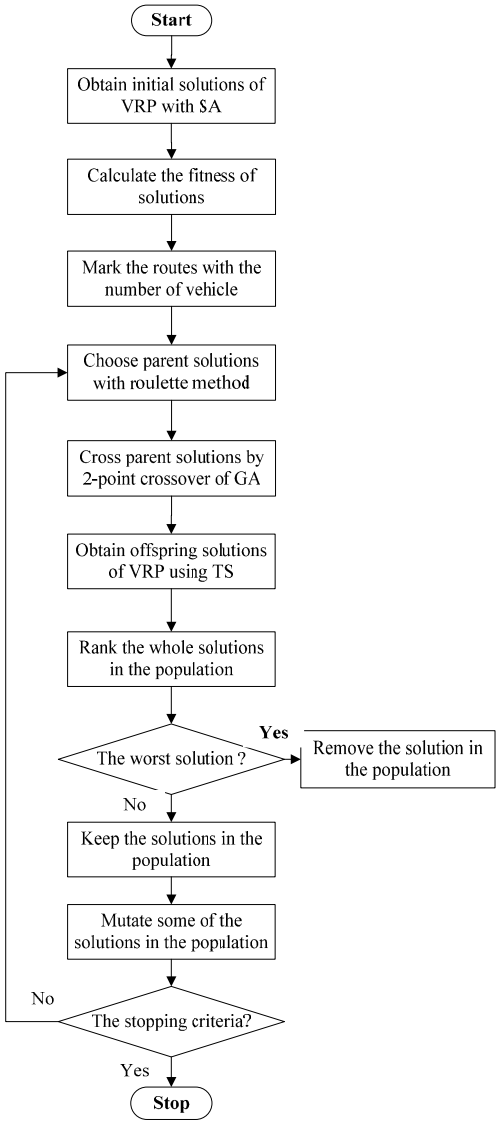


FIGURE 3. A process of GA for VRP.

In the replacement scheme, we give a size of population and keep it constant. When an eligible offspring enters the population of GA, the fitness of the whole members in the population will be calculated and ranked again. The member with the worst fitness will be replaced by an eligible offspring. With this replacement method feasible solutions will evolve and save computational time of algorithm. The procedure will continue until the terminal criterion is satisfied. Figure 3 shows a process of GA for VRP.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

3.2. TS FOR THREE-DIMENSIONAL LOADING

In the TS algorithm, items $I(k)$ to be loaded in the same vehicle k are sequenced in reversed order of visit considering loading constraints. Non-fragile items precede fragile items to ensure the former's top surface can be employed for carrying subsequent items of any fragility status. For example, Figure 1, where the loading sequence is $\{I_{31}, I_{33}, I_{32}, I_{21}, I_{22}, I_{11}, I_{12}\}$ for vehicle 1, $I_{62}, I_{61}, I_{51}, I_{52}, I_{41}$ for vehicle 2, and $I_{92}, I_{91}, I_{81}, I_{71}, I_{72}$ for vehicle 3. Given such a sequence, front-left-bottom and touching-area heuristics are executed respectively to find a feasible loading solution that minimizes the used space or maximizes the number of items to be loaded into vehicles. In this subsection, our aim is to use a loading space of maximum width W, maximum height H, and minimum length LL.

Front-Left-Bottom heuristics for three-dimensional loading problem (FLB_{3L}) [3] focuses on those positions where the bottom surface of item to be loaded (resp. left, resp. front) touches either the bottom of the vehicle container or the top of an already loaded item. FLB_{3L} scans the normal positions according to our aim and considers two feasible orientations on the W-L plane. FLB_{3L} chooses the first feasible loading as soon as it satisfies all packing constraints. Touching-Area heuristics for three-dimensional loading problem (TA_{3L}) [26] concerns the positions in which an item to be loaded maximizes the percentage of the item surface touching the vehicle container and other items already packed. TA_{3L} selects the position with the highest percentage. Each procedure of these two heuristics is iterated until all items are placed into vehicles or one of other stopping criteria is reached and the best loading solution is obtained. The stopping criteria say that a prefixed maximum number of generations is reached and the algorithmic convergence has been achieved.

If the minimum length LL is less than L for each vehicle, namely $LL \leq L$, a feasible loading is found and the algorithm terminates. If one vehicle cannot contain its items, namely $LL > L$, TS chooses a better loading solution obtained by the two loading heuristics and attempts to update this incumbent solution. For the vehicles where items cannot be fully loaded, items are divided into two types. Items u loaded entirely within the vehicle are defined as type T_1, $u \in T_1$. Items v having a portion outside the vehicle are as type T_2, $v \in T_2$. TS uses two tabu lists for these two types of items, The value of the tabu list length is experimentally determined as the minimum between 10 and half the number of goods to be loaded. TS will substitute a different pair (u, v) to execute all possible shifts for investigating the neighborhood thoroughly only if it improves the incumbent solution. The improvement is evaluated by a score $M(u, v)$, where $M(u, v) = S(u, v) + (\sigma(u) + \sigma(v))L$. $S(u, v)$ is the solution value of VRP after u and v are exchanged. $\sigma(u)$ and $\sigma(v)$ is respectively the rate of the number of times items u and v having been selected for exchanging in the previous movement of the number of exchanges executed. L is the length of vehicle container. We calculate the score $M(u, v)$ of each pair (u, v) and choose the best score for the improvement of loading problem. TS does not permit the exchange (u, v) until it has executed

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

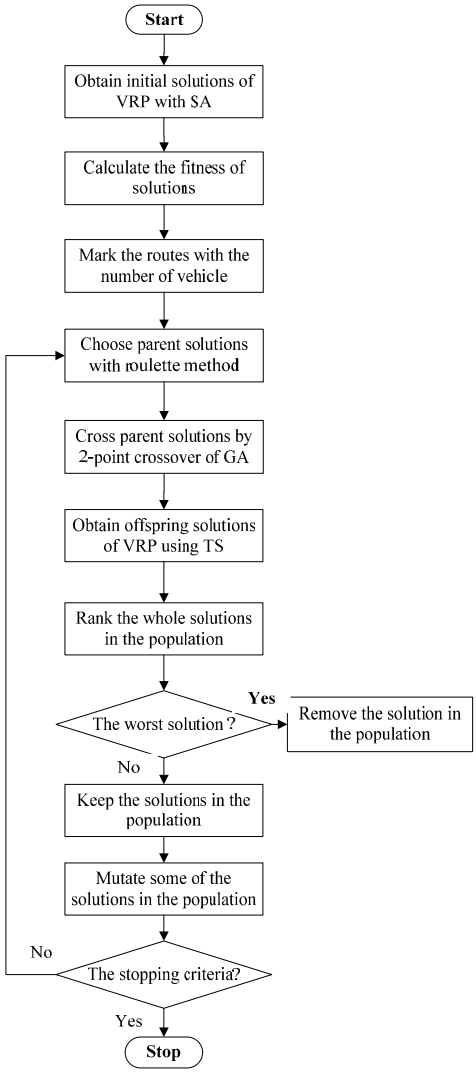


FIGURE 4. A process of TS for 3LP.

a prefixed number of iterations. TS will terminate as soon as a feasible loading is found or when the algorithm has finished a prefixed number of iterations. A procedure of TS is shown in Figure 4.

3.3. A HYBRID APPROACH FOR 3L-CVRP

In this subsection, we integrate GA and TS to be a hybrid approach for 3L-CVRP. First we apply SA to generate some initial solutions for VRP. Subsequently

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

we employ GA to optimize the solutions of VRP and rank the solutions of a given number in the population. Third we make use of FLB_{3L} and TA_{3L} (FT) to address 3LP. If a feasible loading of the 3LP is found by the two loading heuristics, the process of the hybrid algorithm will terminate. Otherwise, 3LP is optimized by TS. In the TS algorithm, items are divided into two types and exchanged with each other to improve the rate of space usage of vehicles. If a feasible loading solution is generated by TS, the hybrid algorithm terminates. Otherwise, the procedure is repeated for the successive top solution of VRP until a feasible solution is found. A pseudocode of the hybrid approach for 3L-CVRP is described in Table 4, while in the next two paragraphs some procedures of the hybrid approach are explained.

In Table 4, four different types of algorithms are integrated for 3L-CVRP, namely SA, GA, FT and TS. Two main processes are produced to solve vehicle routing sub-problem and three-dimensional loading sub-problem respectively. These four algorithms will be integrated since no feasible solution of 3L-CVRP is found in a whole procedure.

Some steps probably confusing readers during the process are explained as follows. The first point is step 13, we list the whole solutions of VRP for the choice of finding a feasible loading solution in sequence. The second one is step 19, the number of iterations is given, if TS cannot find a feasible loading solution on a top solution of VRP, the algorithm will halt and test the successive top solution. At last, the process will stop when either the number of generations reaches 1000 or the number of continuous iteration of non-improvement reaches 100.

4. Computational results

The Hybrid Approach is coded in C# and compiled with windows XP compiler. The algorithm is tested by computational experiments on a Pentium IV with 2.3 GHz and 1 GB of RAM, running under a windows operative system. It is tested on the set of instances proposed in [16], which can be downloaded from http://www.or.deis.unibo.it/research.html. These instances are the only 3L-CVRP instances available on the web. They provide an interesting test bed since heuristic solutions are available for comparison. In the instances, the graphs, the weight demanded by clients and the vehicle weight capacities are taken from 27 Euclidean CVRP instances (see [34] for a detailed description of CVRP test bed instances). The arc costs are determined as the Euclidean distances between client coordinates. The loading volume has dimensions $W = 25$, $H = 30$, and $L = 60$. For each client the number of requested goods 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. For presenting the advantage of this hybrid approach, we compare its solutions with TS [16], GTS [33], ACO [14], and HBMO [32]. The parameters setting for the GATS are given in Table 5, where the ones of TS are tested in [16,17,29,38], and the ones of GA are examined in [2].

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

TABLE 4. A pseudocode of the hybrid approach for 3L-CVRP.

```
SA-GA for CVRP, FT-TS for 3LP
Using SA to generate an initial population of structured solutions
Evaluate fitness value for each individual in the population
  Select two parents in the population with roulette method
  Produce two offspring by crossovering the parents
  Mutate offspring
  Evaluate fitness of offspring
  Rank the member of the population
  Remove the worst solution from the population
  Keep the size of population constant
  If the stopping criteria of GA are satisfied
    Go to step 15
  Else
    Go to step 03
  Endif
List the top solutions of GA (LTS)
Do i=1:LTS
    Load the items with FLB_3L and TA_3L respectively.
      If a feasible loading solution is obtained
        Go to step 35
      Else
        Do ii = 1 : NI          %NI: the number of iterations
          Employ two tabu lists for two types of items
          Calculate the improvement of M(u, v) for each pair
          Choose the best improvement and exchange the items pair (i, j)
          If a feasible loading solution is obtained
            Go to step 35
          Else
            go to step 21
          Endif
        Enddo
      Endif
Enddo
If no feasible solution for 3L-CVRP is found
  Go to step 1 and repeat the procedure until the solution of 3L-CVRP is found
Return the best solution found for 3L-CVRP
```

TABLE 5. The parameters of GA and TS.

GA			TS		
Parameter	Description	Value	Parameter	Description	Value
α	Iterations	1000	γ	Neighbor size	Min$\{n/4, 20\}$
β	Popular size	100	μ	Tabu list tenure	Min$\{n/10, 15\}$
λ	Crossover	0.75	τ	Tentative value	$1, c, 2c$
θ	Mutation	0.05	π	Average edge cost	

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

In Table 6, the first four columns report the index I of instances, the number of clients n, the total number of goods M ($M = \sum_{i=1}^{n} m_i$), and the number of vehicles employed. The next four columns report the solutions obtained by TS [16], GTS [33], ACO [14], and HBMO [32], respectively. The hybrid algorithm in this paper is run 10 times. For the GATS we report the average total travel costs u_{avg}, the computational time generating best solution of the instances sec_u, and the total running time for the solution sec_{tot}. Since the solutions found by TS in [16] is the first and standard results based on all publicly available test instances, we mainly compare GATS with the other four algorithms and give their gaps, evaluated as $\%g_{G\text{-}T} = 100((u_{\text{avg}} - x)/x$, $\%g_{G\text{-}G} = 100((u_{\text{avg}} - y)/y$, $\%g_{G\text{-}A} = 100((u_{\text{avg}} - z_{\text{avg}})/z_{\text{avg}}$, and $\%g_{G\text{-}H} = 100((u_{\text{avg}} - c)/c$. For each column the row AVG presents the average values on the 27 instances. Each vehicle satisfies the standard 3L-CVRP requirements of the loading constraints: fragile constraint, sequential loading, a minimum supporting of 75% of each goods base and rotation allowed only on the base.

In Table 6, in terms of solution quality the GATS meta-heuristics is clearly superior to the TS. In all the cases, the average of the solutions found by GATS is 8.14% better than the one found by the TS. Secondly, we find that the performance of GATS is 3.99% better than the performance of GTS, for the whole instances except for E023-03s, E031-09h, E036-11h, E041-14h, E076-14s, E101-08e, and E101-14s. We thirdly note that the GATS results in a 0.95% solution improvement compared with the ACO algorithm, 16 instances within the 27 can produce better solutions. Fourthly, we find that performance of GATS is 0.82% better than the performance of HBMO, which produced the best current solutions in literature. Finally, we focus on the computational times executed by GATS. We yield a very largely averaged reduction of computational times of GATS compared with the other four algorithms, reaching 97.40%, 97.78%, 96.93%, and 92.89%, respectively.

In Table 7, we execute the GATS approach to evaluate the effect of each loading constraint imposed by the problem on the total costs. In particular, we run our algorithm with four different loading constraint configurations, including the fragility, LIFO, and supporting area constraints. For the sake of simple comparison, we only give the average value of solutions generated by the algorithms. The first column reports the different loading constraints for the different algorithms, while the second to sixth columns report the solution under four different loading constraints. The seventh column reports the gaps between four different loading conditions of GATS, where $\%g_{\text{GATS}} = 100(u_i - u_{\text{avg}})/u_{\text{avg}}$, $u_i (i = 2, 3, 4, 5)$ is the solution value of the i th row generated by GATS. The last column reports the gaps between these five different algorithms, where $\%d_{G\text{-}T} = 100(u_{\text{avg}} - x)/x$, $\%d_{G\text{-}G} = 100(u_{\text{avg}} - y)/y$, $\%d_{G\text{-}A} = 100(u_{\text{avg}} - z_{\text{avg}})/z_{\text{avg}}$, $\%d_{G\text{-}H} = 100(u_{\text{avg}} - c)/c$. The first row shows the solutions of all constraints of TS, GTS, ACO, HBMO, GATS and two types of gaps, the second row to the fourth one does not consider the fragility, LIFO, and support area respectively. For the last figuration all three aforementioned constraints are ignored.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

TABLE 6. Comparisons of the GATS algorithm with TS, GTS, ACO, and HBMO on 3L-CVRP instances from the literature. All loading constraints imposed.

I	n	M	V	TS	GTS	ACO	HBMO	GATS			GAP			
				x	y	z_{avg}	c	u_{avg}	sec_u	sec_{tot}	$\%g_{G\text{-}T}$	$\%g_{G\text{-}G}$	$\%g_{G\text{-}A}$	$\%g_{G\text{-}H}$
E016-03m	15	32	4	316.3	321.5	305.4	303.2	303.7	2.6	2.9	−3.98	−5.52	−0.53	0.17
E016-05m	15	26	5	350.6	335.0	335.0	335.0	309.4	7.0	8.5	−11.75	−7.63	−7.63	−7.63
E021-04m	20	37	4	447.7	431.0	409.8	398.1	358.2	12.8	15.3	−20.01	−16.89	−12.60	−10.02
E021-06m	20	36	6	448.5	458.0	440.7	440.7	439.7	10.4	13.6	−1.96	−4.01	−0.23	−0.23
E022-04g	21	45	4	464.2	465.8	453.2	452.6	435.2	12.4	14.2	−6.25	−6.56	−3.97	−3.83
E022-06m	21	40	6	504.5	508.0	501.5	498.6	488.4	7.6	9.1	−3.17	−3.84	−2.60	−2.03
E023-03s	22	46	5	831.7	796.6	797.5	790.2	823.4	18.5	21.4	−0.99	3.37	3.26	4.20
E023-05s	22	43	5	871.8	880.9	820.7	820.7	813.5	12.4	13.8	−6.69	−7.66	−0.88	−0.88
E026-08m	25	50	8	666.1	642.2	635.5	635.5	625.3	11.9	13.8	−6.12	−2.63	−1.60	−1.60
E030-03g	29	62	6	911.2	884.7	841.1	836.2	799.6	21.4	23.8	−12.24	−9.62	−4.93	−4.37
E030-04g	29	58	5	819.4	873.4	821.0	825.8	767.0	19.4	21.6	−6.39	−12.19	−6.58	−7.12
E031-09h	30	63	9	651.6	624.2	629.1	626.6	635.5	15.2	18.7	−2.47	1.80	1.02	1.42
E033-03n	32	61	5	2928.3	2799.7	2739.8	2739.8	2736.7	28.8	29.1	−6.54	−2.25	−0.11	−0.11
E033-04s	32	72	6	1559.6	1504.4	1472.3	1469.4	1287.3	19.1	22.1	−17.46	−14.43	−12.56	−12.39
E033-05s	32	68	6	1452.3	1415.4	1405.5	1369.7	1289.4	22.4	24.2	−11.22	−8.91	−8.26	−5.87
E036-11h	35	63	11	707.9	698.6	698.9	703.2	706.7	19.8	23.6	−0.16	1.16	1.11	0.50
E041-14h	40	79	14	920.9	872.8	870.3	872.1	874.0	22.7	25.0	−5.09	0.14	0.42	0.23
E045-04f	44	94	8	1400.5	1296.6	1261.1	1250.9	1234.6	24.4	26.3	−11.85	−4.78	−2.10	−1.30
E051-05e	50	99	8	871.3	818.7	781.3	780.4	779.8	39.7	48.5	−10.50	−4.74	−0.19	−0.07
E072-04f	71	147	13	732.1	641.6	611.3	605.6	596.6	67.0	72.3	−18.51	−7.01	−2.40	−1.49
E076-07s	75	155	12	1275.2	1159.7	1124.6	1119.5	1138.7	54.2	57.1	−10.71	−1.82	1.26	1.72
E076-08s	75	146	13	1277.9	1245.4	1197.4	1167.3	1224.0	34.1	37.9	−4.22	−1.72	2.21	4.85
E076-10s	75	150	13	1258.2	1231.9	1171.8	1171.8	1211.1	40.7	47.3	−3.74	−1.69	3.36	3.36
E076-14s	75	143	15	1307.1	1202.0	1148.7	1136.3	1261.9	86.3	90.7	−3.46	4.99	9.86	11.06
E101-08e	100	193	16	1570.7	1457.5	1436.2	1426.3	1501.0	689.6	708.2	−4.44	2.99	4.51	5.24
E101-10c	100	199	18	1848.0	1711.9	1617.0	1585.5	1621.7	91.2	112.3	−12.24	−5.27	0.29	2.29
E101-14s	100	198	17	1747.5	1646.4	1573.5	1562.2	1588.3	55.8	58.3	−9.11	−3.53	0.94	1.67
AVG				1042.3	997.18	966.66	960.10	957.43	53.6	57.8	−8.14	−3.99	−0.95	−0.82

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

TABLE 7. Summarized comparison between these five different algorithms on different loading configurations.

I	TS		GTS	ACO		HBMO	GATS		GAP	GAP-value			
	x	sec_x	y	z_{avg}	sec_z	c	u_{avg}	sec_u	$\%g_{\text{GATS}}$	$\%d_{G\text{-}T}$	$\%d_{G\text{-}G}$	$\%d_{G\text{-}A}$	$\%d_{G\text{-}H}$
All constraints	1042.3	2058.9	997.2	966.7	1746.6	960.1	957.4	54.6	0	−8.14	−3.99	−0.95	−0.28
No fragility	1014.5	2410.8	972.7	945.1	1337.0	935.9	938.8	50.2	−1.95	−7.46	−3.49	−0.68	0.30
No LIFO	951.2	1709.8	950.6	916.3	676.6	912.5	911.3	51.4	−4.82	−4.19	−4.13	−0.54	−0.13
No support	939.5	1882.5	928.3	919.7	1340.0	915.2	913.0	53.2	−4.64	−2.82	−1.64	−0.73	−0.24
3D only	876.3	1567.4	876.4	856.7	689.3	851.1	849.2	48.8	−11.30	−3.09	−3.10	−0.87	−0.22

We firstly focus on the percentage difference $\%g_{\mathrm{GATS}}$, the gaps between all constraints and the other four constraints in Table 7. If we remove the LIFO constraint and supporting area constraint, the algorithm will lead to the larger reduction of 4.82% and 4.64%. While removing the fragility constraint can just find a weak reduction of 1.95% in the solution value. When we remove all three constraints, the hybrid approach will obtain an overall average solution value reduction of 11.30% which is by far better improvement than three others. Secondly, we concern the differences $\%d_{G\text{-}T}$, $\%d_{G\text{-}G}$, $\%d_{G\text{-}A}$, and $\%d_{G\text{-}H}$ between the average solutions found by TS, GTS, ACO, HBMO and GATS under different loading configurations. We find that when considering the whole constraints, the gaps among these three algorithms are the largest at 8.14%, 3.99%, 0.95%, and 0.28%. While relaxing some constraints, the gap will be reduced. We note that the smallest gaps between GATS and TS, GATS and GTS are 2.82% and 1.64% when the supporting area constraint is ignored, the smallest gap between GATS and ACO is 0.54% when we relax the LIFO constraint.

We finally test tabu search and two types of loading heuristics on the three-dimensional loading instances independently, and compare the results with the recent literature, such as [7,8]. We find tabu search integrating FLB_{3L} and TA_{3L} in this paper cannot produce better solutions than the results generated by [7,8] due to the lack of application of the Interval Graph representation of the packing and the size enlargement of the associated neighborhoods.

5. Conclusions and future researches

The problem in this paper combines three-dimensional loading problem and capacitated vehicle routing problem which both are NP-hard problems. Although the practical relevance is evident, we only can find very few papers devoted to the combination of loading and routing due to its sophisticated characteristics. There is no exact algorithm presented for this combinatorial problem, so it is difficult to achieve the optimality for the reasonably sized problems. A very good performance of GATS approach is presented with publicly available test instances comparing to recent TS, GTS, ACO, and HBMO meta-heuristics. We are able to improve the total routing cost on average by 0.82% comparing with the best current solutions.

Since 3L-CVRP can open new possibilities of solving real life problems in transportation logistics, it is valuable to study this field continuously. In our opinion, the future research benefit is threefold. Firstly, the performance of the hybrid approach for 3L-CVRP can be further improved possibly when considering different objective functions reflecting practical requirements. Secondly, 3L-CVRP can be extensively studied, where its formulation is extended to cover more operational constraints such as heterogeneous fleet of vehicles, pickup and delivery, time windows and the center of gravity of the load. Thirdly, it is also possible to integrate the model and method with location issues.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

Acknowledgements. The authors very thank two anonymous referees for their valuable comments and suggestions.

Appendix

(The routes of 27 instances: RI)

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

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

RI 20: 0-52-45-53-46-49-0|| 0-9-7-4-8-3-5-71-0|| 0-13-12-16-17-2-15-19-11-59-0||0-58-61-60-62-63-64-66-54-0||0-65-67-68-39-57-25-0||0-44-42-43-48-47-50-70-0||0-28-22-21-30-29-20-0||0-10-6-18-1-33-0||0-32-31-34-36-41-0||0-14-35-55-0||0-24-26-23-27-0||0-38-37-69-40-51-0||0-56-0||

RI 21: 0-55-25-50-18-24-49-16-3-0|| 0-56-23-63-33 -73-62-28-22-1-0||0-70-60-71-36-47-21-74-30-4-0||0-72-39-9-32-44-40-12-0||0-51-6-68-75-67-34-52-57-0||0-64-42-41-43-2-48-0||0-20-37-5-29-45-13-0||0-38-65-11-66-59-14-0||0-46-8-35-7-0||0-31-10-58-26-0||0-53-19-54-15-0||0-61-69-27-17-0||

RI 22: 0-13-27-52-34-46-8-35-14-54-0|| 0-17-51-6-68 -75-4-67-0||0-43-41-42-64-22-0||0-71-60-70-37-47-21-29-0||0-61-28-62-73-33-63-16-0||0-44-3-32-9-39-40-0||0-19-53-11-65-38-10-0||0-69-36-5-48-74-2-0||0-25-50-18-24-49-23-0||0-57-15-20-7-0||0-1-56-30-45-0||0-12-26-58-72-0||0-59-66-31-55-0||

RI 23: 0-39-9-32-44-3-16-63-0|| 0-12-40-17-51-6-75 -0||0-49-24-18-50-25-55-72-65-0||0-15-57-13-27-29-5-36-0||0-43-41-42-64-22-62-73-2-0||0-70-60-71-69-21-47-48-54-0||0-67-34-46-8-35-7-58-0||0-23-56-1-33-68-30-0||0-10-38-11-66-19-0||0-53-14-59-52-45-0||0-61-28-74-4-0||0-20-37-26-0||0-31-0||

RI 24: 0-53-14-19-8-13-0|| 0-71-60-70-20-37-5-69-0|| 0-25-50-18-24-49-1-0||0-43-41-42-64-22-73-23-0||0-10-58-72-39-9-0||0-36-47-21-74-28-0||0-32-44-3-16-63-51-0||0-26-67-34-46-0||0-4-75-68-6-17-0||0-2-30-48-29-27-0||0-52-45-15-57-54-12-0||0-7-35-11-65-38-0||0-59-66-31-55-0||0-33-62-61-56-0||0-40-0||

RI 25 : 0-42-87-97-95-94-6-96-0|| 0-77-3-79-33-81 -9-51-20-70-31-0||0-98-37-92-59-99-93-85-61-5-89-0||0-55-25-39-56-75-74-22-0||0-36-47-48-82-7-8-11-0||0-44-14-100-91-16-0||0-67-23-41-73-21-72-4-54-0||0-18-60-83-84-17-86-0||0-63-90-32-10-62-0||0-38-43-15-57-2-58-40-0||0-28-27-69-1-50-0||0-80-68-12-26-53-13-0||0-29-24-76-78-34-35-71-19-0||0-45-46-49-64-0||0-30-66-65-52-0||0-88-0||

RI 26: 0-73-70-71-76-78-81-63-0|| 0-44-45-46-48-51 -52-43-0||0-24-25-27-26-23-6-12-0||0-32-33-34-36-39-38-37-0||0-97-93-92-94-95-96-99-98-4-0||0-89-88-85-84-83-68-0||0-86-87-90-91-75-1-5-0||0-82-79-77-80-72-66-0||0-7-3-8-9-15-0||0-40-41-42-49-0||0-22-21-20-28-0||0-18-17-13-19-16-14-29-0||0-56-58-60-54-0||0-50-47-69-62-55-0||0-67-65-74-61-64-0||0-59-57-53-31-35-0||0-11-10-2-30-0||0-100-0||

RI 27: 0-35-71-9-81-33-79-51-0|| 0-92-59-99-96-94-6 -89-18-24-0||0-93-85-91-100-37-98-0||0-82-48-47-36-46-8-45-0||0-15-43-42-87-97-95-27-0||0-14-44-16-61-5-84-7-0||0-21-73-72-74-22-75-56-25-0||0-65-66-20-30-70-31-0||0-57-2-58-40-0||0-60-83-17-86-38-53-0||0-3-77-76-50-1-28-0||0-80-68-26-0||0-13-52-88-62-10-63-0||0-41-23-39-4-34-0||0-11-19-49-64-90-32-0||0-54-55-12-0||0-69-78-29-67-0||

References

[1] B. Andreas, A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Comput. Oper. Res.* **39** (2012) 2248–2257.

[2] B.M. Baker and M.A. Ayechew, A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.* **30** (2003) 787–800.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

[3] B.S. Baker, E.G. Coffman Jr. and R.L. Rivest, Orthogonal packings in two dimensions. *SIAM J. Comput.* **9** (1980) 846–855.
[4] A. Bortfeldt and H. Gehring, A hybrid genetic algorithm for the container loading problem. *Eur. J. Oper. Res.* **131** (2001) 143–161.
[5] J. Cordeau and G. Laporte, Tabu search heuristics for the vehicle routing problem. *Metaheuristic Optimization via Memory and Evolution* **30** (2005) 145–163.
[6] J. Cordeau, M. Gendreau, A. Hertz, G. Laporte and J. Sormany, New heuristics for the vehicle routing problem. *Logistics Systems: Design and Optimization* (2005) 279–297.
[7] T.G Crainic, G. Perboli and R. Tadei, Extreme point-based heuristics for three-dimensional bin packing. *Informs J. Comput.* **20** (2008) 368–384.
[8] T.G. Crainic, G. Perboli and R. Tadei, TS2PACK: A two-level tabu search for the three-dimensional bin packing problem. *Eur. J. Oper. Res.* **195** (2009) 744–760.
[9] K.F. Doerner, G. Fuellerer, R.F. Hartl, M. Gronalt, and M. Iori, Metaheuristics for the vehicle routing problem with loading constraints. *Networks* **49** (2007) 294–307.
[10] C. Duhamel, P. Lacomme, A. Quilliot and H. Toussaint, A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Comput. Oper. Res.* **38** (2011) 617–640.
[11] M. Eley, Solving container loading problems by block arrangement. *Eur. J. Oper. Res.* **141** (2002) 393–409.
[12] O. Faroe, D. Pisinger and M. Zachariasen, Guided local search for the three-dimensional bin-packing problem. *Informs J. Comput.* **15** (2003) 267–283.
[13] G. Fuellerer, K.F. Doerner, R.F. Hartl and M. Iori, Ant colony optimization for the two-dimensional loading vehicle routing problem. *Comput. Oper. Res.* **36** (2009) 655–673.
[14] G. Fuellerer, K.F. Doerner, R.F. Hartl and M. Iori, Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *Eur. J. Oper. Res.* **201** (2010) 751–759.
[15] R. Fukasawa, H. Longo, J. Lysgaard, M.P.D. Aragão, M. Reis, E. Uchoa and R.F.Werneck, Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Math. Program.* **106** (2006) 491–511.
[16] M. Gendreau, M. Iori, G. Laporte and S. Martello, A tabu search algorithm for a routing and container loading problem. *Trans. Sci.* **40** (2006) 342–350.
[17] M. Gendreau, M. Iori, G. Laporte and S. Martello, A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks* **51** (2008) 4–18.
[18] B.E. Gillett and R.M. Leland, A heuristic algorithm for the vehicle-dispatch problem. *Oper. Res.* **22** (1974) 340–349.
[19] M. Iori, *Metaheuristic algorithms for combinatorial optimization problems.* Ph.D. thesis, Italy (2004).
[20] M. Iori, Metaheuristic algorithms for combinatorial optimization problems. *Quart. J. Oper. Res.* **3** (2005) 163–166.
[21] M. Iori and S. Martello, Routing problems with loading constraints. *TOP* **18** (2010) 4–27.
[22] M. Iori, J.J. Salazar-Gonzalez and D. Vigo, An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Trans. Sci.* **41** (2007) 253–264.
[23] L. Junqueira, R. Morabito and D. Sato Yamashita, Three-dimensional container loading models with cargo stability and load bearing constraints. *Comput. Oper. Res.* **39** (2012) 74–85.
[24] S. Khebbache-Hadji, C. Prins, A. Yalaoui and M. Reghioui, Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. *Cent. Eur. J. Oper. Res.* (in press) 1–30.
[25] G. Levitin and R. Abezgaouz, Optimal routing of multiple-load AGV subject to LIFO loading constraints. *Comput. Oper. Res.* **30** (2003) 397–410.
[26] A. Lodi, S. Martello and D. Vigo, Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *Informs J. Comput.* **11** (1999) 345–357.
[27] S. Martello, D. Pisinger and D. Vigo, The three-dimensional bin packing problem. *Oper. Res.* **48** (2000) 256–267.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press

[28] S. Martello, D. Pisinger, D. Vigo, E. Den Boef and J. Korst, Algorithm 864: general and robot-packable variants of the three-dimensional bin packing problem. *ACM Trans. Math. Softw.* **33** (2007) 7–17.
[29] F. Pereira, J. Tavares, P. Machado and E. Costa, GVR: a new genetic representation for the vehicle routing problem. *Artificial Intelligence and Cognitive Science* **24** (2002) 95–102.
[30] D. Pisinger, Heuristics for the container loading problem. *Eur. J. Oper. Res.* **141** (2002) 382–392.
[31] M. Reimann, K. Doerner and R.F. Hartl, D-Ants: Savings based ants divide and conquer the vehicle routing problem. *Comput. Oper. Res.* **31** (2004) 563–591.
[32] Q. Ruan, Z. Zhang, L. Miao and H. Shen, A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Comput. Oper. Res.* (in press).
[33] C.D. Tarantilis, E.E. Zachariadis and C.T. Kiranoudis, *A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. IEEE Trans. Intell. Transp. Syst.* **10** (2009) 255–271.
[34] P. Toth and D. Vigo, The vehicle routing problem, in *SIAM Monographs on Discrete Mathematics and Applications*. Philadelphia (2002).
[35] F. Tricoire, K. Doerner, R. Hartl and M. Iori, Heuristic and exact algorithms for the multipile vehicle routing problem. *OR-Spectrum* **33** (2011) 931–959.
[36] H. Xu, Z. Chen, S. Rajagopal and S. Arunapuram, Solving a practical pickup and delivery problem. *Trans. Sci.* **37** (2003) 347–364.
[37] T. Yi and W. Fan, A New packing heuristic based algorithm for vehicle routing problem with three-dimensional loading constraints, in *IEEE Int. Conf. Automation Science and Engineering (CASE)* (2010) 972–977.
[38] E.E. Zachariadis, C.D. Tarantilis and C.T. Kiranoudis, A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *Eur. J. Oper. Res.* **195** (2009) 729–743.

https://doi.org/10.1051/ro/2012008 Published online by Cambridge University Press