An effective genetic algorithm for solving the capacitated vehicle routing problem with two-dimensional loading constraint

Ines Sbai*

LARODEC,
Institut Supérieur de Gestion de Tunis,
Université de Tunis,
41, Avenue de la Liberté, Cité Bouchoucha,
Le Bardo 2000, Tunisia
Email: sbai_ines@yahoo.fr
*Corresponding author

Olfa Limam

LARODEC,
Institut Supérieur d'Informatique Tunis,
Université de Tunis El Manar, Tunisia
Email: limemolfa@yahoo.fr

Saoussen Krichen

LARODEC,
Institut Supérieur de Gestion de Tunis,
Université de Tunis,
41, Avenue de la Liberté, Cité Bouchoucha,
Le Bardo 2000, Tunisia
Email: krichen_s@yahoo.fr

Abstract: In this article, we focus on the symmetric capacitated vehicle routing problem where customer demand is composed of two-dimensional weighted items. The objective consists in designing a set of trips, starting and terminating at a central depot, that minimise the total transportation cost with a homogenous fleet of vehicles based on a depot node. Items in each vehicle trip must satisfy the two-dimensional orthogonal packing constraint. The capacitated vehicle routing problem with two-dimensional loading constraint is an *NP*-hard problem of high complexity. Given the importance of this problem, many solution approaches have been developed. However, it still a challenging problem. Then, we propose to use a new heuristic based on an adaptive genetic algorithm in order to find better solution. Our algorithm is tested with 150 benchmark instances and compared with state-of-the-art approaches. Results shown that our proposed approach is competitive in terms of the quality of the solutions found.

Keywords: capacitated vehicle routing problem; CVRP; loading; genetic algorithm; GA; 2L-CVRP.

Reference to this paper should be made as follows: Sbai, I., Limam, O. and Krichen, S. (2020) 'An effective genetic algorithm for solving the capacitated vehicle routing problem with two-dimensional loading constraint', *Int. J. Computational Intelligence Studies*, Vol. 9, Nos. 1/2, pp.85–106.

Biographical notes: Ines Sbai is a PhD student in Computer Sciences, in the Higher Institute of Management of Tunis, ISG University of Tunis, Tunisia. She is currently a member of the Larodec laboratory (ISG). Her current research interests include optimisation algorithms (metaheuristics), especially genetic algorithms, and their application to real-parameter benchmarking and real-world problems in the domain of road-traffic optimisation. Because of this work, she has published several papers in referred international and national conferences.

Olfa Limam is an Assistant Professor in the Higher Institute of Computer Science ISI, University of Tunis ElManar, Tunisia and member of the Larodec laboratory in Higher Institute of Management ISG. She works on different areas of research namely multi-objective optimisation, fuzzy optimisation, coalition formation and clustering, brain image segmentation supply chain management. She has published several research papers on real-world problems using different metaheuristics.

Saoussen Krichen is a Professor in the Higher Institute of Management of Tunis, ISG University of Tunis, Tunisia, and member of the Larodec laboratory (ISG). She works on single and multi-objective, optimisation, dynamic optimisation, coalition formation and supply chain management. She has published several papers on real-world problems using bio-inspired algorithms. Her research activity has resulted in several scientific publications consisting of 15 articles in journals indexed by DBLP and Scopus, and more than 16 refereed conferences. Besides that, she has published three books and several book chapters.

1 Introduction

Vehicle routing problem (VRP) is the most studied combinatorial optimisation problems in transport and logistics. It consists in the distribution of goods between depots and customers (Toth and Vigo, 2002). The purpose of the VRP is to find a set of path-ways for a fleet of vehicle that satisfy customer demands where the objective is to minimise routing costs. In the last decades, several variants of the classical VRP have been introduced. The basic version of the VRP is the capacitated vehicle routing problem (CVRP). In the CVRP, vehicles are identical and are based at a single central depot, only the capacity restrictions for the vehicles are respected. The CVRP objective is also to minimise the total cost needed to serve all the customers.

Recently, the combination of the routing and packing problems have been modelled and introduced in the literature. The important characteristic of these problems is to optimise the vehicle routing operations where physical dimensions of transported goods are taken into account. So, in this paper, we study the well-known capacitated vehicle routing problem with two-dimensional loading constraints (2L-CVRP). It is considered as a combination of two optimisation NP-hard problems named the CVRP and the two dimensional bin packing problem (2BPP). Compared to the original CVRP, the

2L-CVRP is defined on an undirected connected graph between a depot and a set of customers or between two customers. The depot contains unlimited fleet of homogeneous vehicles with a maximum weight and a rectangular loading surface of length and width. The demand of each customer consists of a set of rectangular items of known length and width.

The 2L-CVRP can be found in many real-life situations related to the transportation of voluminous items where items must satisfy the two-dimensional of orthogonal packing constraints such as, household appliances, professional cleaning equipment, forklifts, etc. While the 2L-CVRP is known as a challenging *NP*-hard problem of high complexity, we propose a metaheuristic approach based on an 'adaptive genetic algorithm (AGA)' (Zachariadis et al., 2009).

The rest of this paper is organised as follows. A literature review is presented in Section 2. Then, a detailed description of the 2L-CVRP is given in Section 3. After that, the GA proposed methodology is introduced in detail in Sections 4 and 5. The extensive computational results for the benchmark instances and a statistical analysis are presented in Section 6. Finally, we present a conclusion and future work.

2 Literature review

In recent years the combination of the routing and packing problems have been introduced with the 2L-CVRP. The 2L-CVRP is a highly complex NP-hard problem (Wei et al., 2017). Given the importance of this problem, many solution approaches have been developed.

Iori et al. (2007) presented firstly the 2L-CVRP using an exact algorithm based on the branch-and-cut approach for the routing aspect and the branch-and-bound for loading check problem. They used the partial loading and they only considered the sequential variant of the problem and deal with only small sizes of instances.

While, the first meta-heuristic approach is proposed by Gendreau et al. (2008) to tackle larger size problem instances for the 2L-CVRP using a tabu search (TS) algorithm. To check the loading constraints for the sequential and unrestricted variants, they used two heuristics LH_{2SL} and the LH_{2UL}. Then, Zachariadis et al. (2009) proposed an other metaheuristic algorithm which incorporates the rationale of TS and guided local search. For the loading problem, they used five packing heuristics, and three neighbourhood search namely, customer relocation, route exchange and route interchanging, to generate the initial solution. In the same way, Fuellerer et al. (2009) presented an algorithm that scorches the space of routing solution based on an ant colony optimisation. They used two heuristics (bottom left fill and touching perimeter algorithm) to check the feasibility of the two dimensional loading problem. Then, an extended guided tabu search (EGTS) algorithm is developed by Leung et al. (2011), it incorporates the theories of TS and extended guided local search. Furthermore, to solve the 2BPP, they proposed a lowest line best-fit heuristic (LBFH). Duhamel et al. (2011) used the greedy randomised adaptive search procedure combined with evolutionary local search algorithm to obtain an intermediate solution then it is transformed to 2L-CVRP solution. In addition, Leung et al. (2013) proposed six packing heuristics to check the feasibility of loading and they developed a simulated annealing with heuristic local search (SA-HLS). In the same way, Zachariadis et al. (2013) presented a static move description algorithm. Dominguez et al.

(2015) studied the 2L-CVRP with heterogeneous fleet using the multi-start biased randomised algorithm combines the biased randomised algorithm with biased-randomised versions of the best-fit packing heuristic and the touching perimeter algorithms, Pollaris et al. (2015) presented a survey on loading and routing problems. Wei et al. (2015) proposed a variable neighbourhood search (VNS) approach for solving the 2L-CVRP and adapted the skyline heuristic to examine the loading constraints. VNS was also proposed by Pinto et al. (2017) to solve the pickup and delivery problem with loading constraints.

In this paper, we propose to use a metaheuristic solution based on AGA to solve the 2L-CVRP and an adaptive least wasted first heuristic (ALWF) to check the packing aspect. The GA metaheuristic is widely used to generate high quality solutions for combinatorial optimisation problems. Numerous papers studied the routing problems using the GA approach show the excellent solution quality and speed of GA such as Berger and Barkaoui (2003), Baker and Ayechew (2003) and Masum et al. (2011). In comparison with other available 2L-CVRP solutions our proposed AGA algorithm will show its strength in terms of solution quality and efficiency.

3 Problem description

In the 2L-CVRP, the demand of each customer are formed by a set of two-dimensional, rectangular, weighted items. The objective is to design a set of minimum cost routes, starting and terminating at the central depot, to satisfy the customer demands using a set of identical vehicles.

The 2L-CVRP may be defined as follows Iori et al. (2007):

Given a complete undirected graph $G = (V, E)$, in which V defines the set of $n + 1$ vertices corresponding to the depot (vertex 0) and to the customers (vertices 1, ..., n). An associated travelling cost, c_e, is defined for each edge $e \in E$. In the following, a given edge e can also be represented by its end point vertices (i, j).

A set of K identical vehicles is available at the depot. Each vehicle has the same weight capacity Q and a rectangular loading surface that respectively equal to W and H. $A = W * H$ is the total area of the loading surface.

Each customer i ($i = l, ..., n$) is associated with a set of D_i, rectangular items, whose total weight is equal to q_i and each having specific width and height equal to w_t^i and h_t^i, ($t = 1, ..., |D_i|$), respectively. Each item will be denoted by a pair of indices (i, t).

The total area of the items of customer i is denoted by $a_i = \sum_{l=1}^{|D_i|} w_t h_t$.

Finally, the total number of items is denoted by $M = \sum_{i=1}^{n} D_i$. The decision variables of the problem are x_{ijk} where it take 1 if the vehicle k travels from customer i to j (0 otherwise). In order to effectively manage the placement of the item into the vehicle k we define the variable y_{tk} which takes 1 if the item t is inside the vehicle k (0 otherwise).

In the 2L-CVRP, the feasible loading must satisfy the following constraints (Leung et al., 2011):

a all orders related to a customer should be loaded on the same vehicle

b items have a fixed orientation and must be loaded with their sides parallel to the sides of the loading surface c each route must choose the central depot as the starting and the ending point

d all customers must be visited once and only once

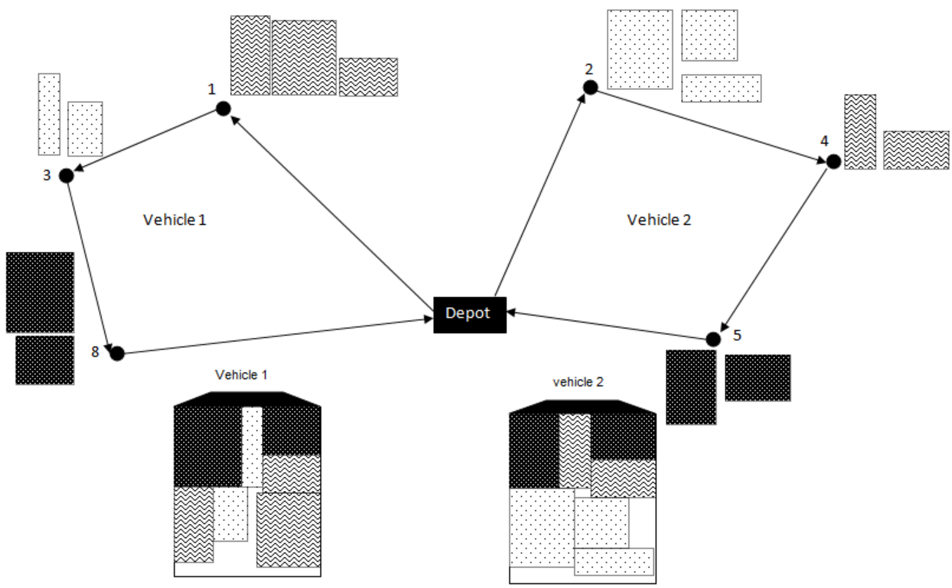
e the total weight of all items in a route must not exceed the capacity *D* of the vehicle

f all items of each customer must be completely loaded on the surface of the vehicle

g no two items can overlap in the same route.

Figure 1 presents an example solution of a 2L-CVRP instance involving six customers, 14 items and two vehicles.

Figure 1 An example of 2L-CVRP solution



4 Problem formulation

The 2L-CVRP is stated as follows (Khebbache et al., 2010):

$$Min \sum_{k=1}^{k} \sum_{i=0}^{n} \sum_{j=0, j \neq i}^{n} c_{ij} x_{ij}^{k} \tag{1}$$

Subject to:

$$\sum_{j=1}^{n} x_{0j}^{k} = \sum_{i=1}^{n} x_{i0}^{k} = 1, \forall k \in \{1 \ldots n\} \tag{2}$$

$$\sum_{j=0, j \neq i}^{n} \sum_{k=1}^{n} x_{ij}^{k} = 1, \forall i \in \{1 \ldots n\} \tag{3}$$

$$\sum_{j=0,i\neq j}^{n} x_{ij}^{k} = \sum_{j=0,j\neq i}^{n} x_{ji}^{k}, \forall k \in \{1\ldots n\}, i \in \{1\ldots n\} \tag{4}$$

$$\sum_{i\in St}\sum_{j\in St} x_{ij}^{k} \leq |St|-1, \forall k \in \{1\ldots n\}, St \subseteq V \tag{5}$$

$$\sum_{i=1}^{n}\sum_{j=0,j\neq i}^{n} x_{ij}^{k} q_i \leq Q, \forall k \in \{1\ldots n\} \tag{6}$$

$$\sum_{t=1}^{M} y_{tk} w_t^i * h_t^i \leq W * H, \forall k \in \{1\ldots n\} \tag{7}$$

$$\sum_{j=1}^{n} x_{ij}^{k} = y_{tk}, \forall k \in \{1\ldots n\} \forall t \in D_i \tag{8}$$

$$u_{tk} + w_t \leq W + M\left(1 - y_{tk}\right), \forall k \in \{1\ldots n\}, \forall t = 1\ldots M \tag{9}$$

$$v_{tk} + h_t \leq H + M\left(1 - y_{tk}\right), \forall k \in \{1\ldots n\}, \forall t = 1\ldots M \tag{10}$$

$$x_{ijk} \in 0, 1 \forall i = 0\ldots n, \forall j = 0\ldots n, i \neq j, \forall k \in \{1\ldots n\} \tag{11}$$

$$y_{tk} \in 0, 1 \forall t = 1\ldots m, \forall k \in \{1\ldots n\} \tag{12}$$

- Objective function (1) consists in minimising the total cost (Faiz and Krichen, 2012).
- Constraint (2) expresses that each travel should begin and end at the depot.
- Constraint (3) provides that a single vehicle leaves each client *i*.
- Constraint (4) guarantees the continuity of a tour.
- Constraint (5) eliminates the sub-tour ($St \subseteq V$) with $|St| = i$ is $C_n^i * K$.
- Constraint (6) provides that the vehicle weight is not exceeded.
- Constraint (7) guarantees that the surface of a vehicle is not exceeded.
- Constraint (8) provides that all items of a given customer should be assigned to the same vehicle.
- Constraint(9) and (10) denote that each item is placed in the vehicle that actually transports it.
- Constraints (11) and (12) define the fields and variables signs.

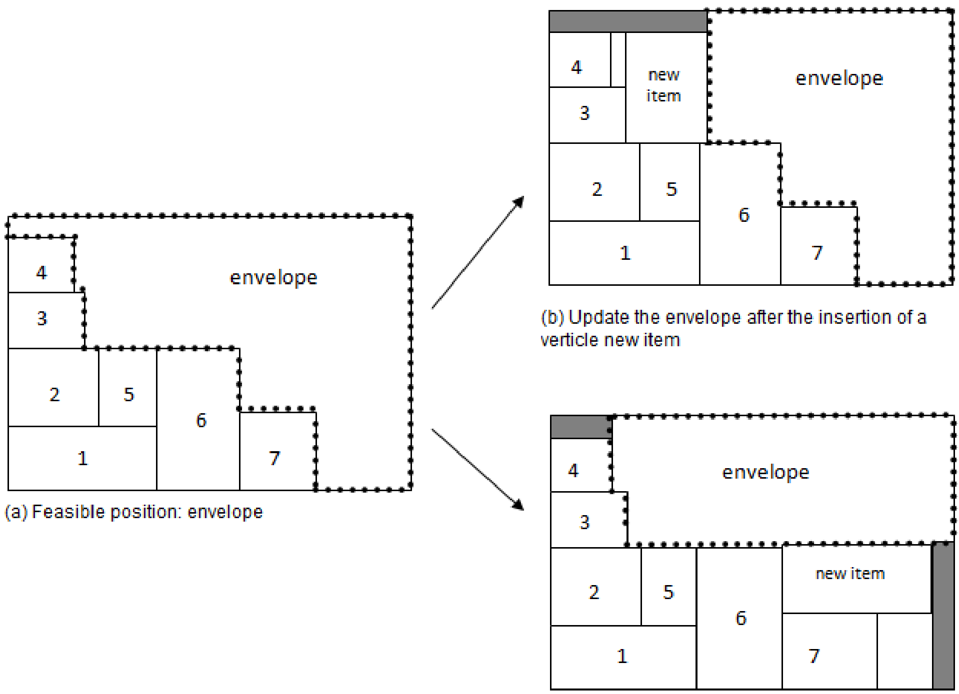
5 An AGA for the 2L-CVRP

In the present section, we describe the steps of our AGA for solving the 2L-CVRP and the employed ALWF heuristic to handle the loading constraints.

5.1 Heuristics for the two-dimensional loading problems

To determine whether a route-sequence of customers is feasible in terms of the loading constraints of the examined problem, we designed an ALWF heuristic.

Figure 2 Envelope update



(a) Feasible position: envelope

(b) Update the envelope after the insertion of a verticle new item

(c) Update the envelope after the insertion of an horizontal new item

First, for each sorted sequence of items, we use the method of 2D-CORNERS (Martello et al., 2000) to find a position to pack an item into the vehicle presented as follow: the first items R_i is placed with its bottom left corner at the origin (0, 0) of the vehicle k with width W and height H and its four sides parallel to X and Y. The packing must satisfy the following constraints:

1 each item can be horizontally or vertically packed into the vehicle it means that the rectangles are rotatable

2 each edge of an item packed should be parallel to an edge of the vehicle, which is also called orthogonal packing

3 each item packed should be completely packed within the vehicle.

Then, a new item must be placed at a position where any packed item is below or to the left of it. The feasible regions where the remaining items can be packed in is called the envelope, an example is shown in Figure 2(a) where seven items named 1–7 are packed into the vehicle one by one and the feasible envelope is enclosed by the broken line. Then to place a new item at the feasible position as illustrated in Figures 2(b) and 2(c), we test if it oversteps or not the border of the vehicle. When the gap between a feasible position and vehicle's border is less than smallest edge of the unpacked item, we call this region bad region because none unpacked items can be packed at this position.

Algorithm 1 The 2D-loading algorithm

```
Begin
succes ← True
y_min ← 0
B_c ← ø
R ← A
k ← 1
while (k ≤ n) do
    Select randomly a_i from R
    y_i ← itemor
    y_i ← y_min
    B_c ← B_c ∪ {a_i}
    R ← R / {a_i}
    if max(y_min + h_i) > H then
      Stop ← True
      succes ← False
    end if
end while
k ← k + 1
END
```

Algorithm 1 presents the loading aspect of items into the vehicle using the following instructions:

- *success:* takes 1 if i is assigned to vehicle K and 0 otherwise
- $\{y_1 \dots y_n\}$: the vertical axis for a_i
- $y_{\min}$: the smallest ordinate to which a new object can be placed given the objects already placed
- B_c: the set of item packed so far
- *stop:* stop condition
- R: the set of objects i which can be placed in $(x_i, y_{\min})$.

The 2D-loading algorithm is initialised with $y_{\min} = 0$ and $R = A$ (where A is the ordered set of items). At each iteration, an object a_i of R is chosen randomly and positioned in $(x_i, y_{\min})$. The value of $y_{\min}$ is then recalculated to take account of this new and the set R is updated. We continue iteratively until we have placed objects or until it is no longer possible to insert a new object without exceeding the width of the vehicle.

5.2 ALWF heuristic for the packing

The second step is to apply iteratively the proposed ALWF proposed by Wei et al. (2009) which determines the feasibility of packing a set of oriented two-dimensional items into a set of identical two-dimensional bins. Algorithm 2 describes the proposed heuristic using the 2D-loading (R, callmax, samemax) procedure. Where, lmax is the number of the calls of the 2D-loading. (R, callmax, samemax) procedure, callmax, samemax are the parameters used to control the number of calls of the 2D-loading procedure, best is used to record the best result found so far. totarea is the total area of items in R. In the LWF-2D-Loading(R, lmax, callmax, samemax) procedure the 2D-loading(R, callmax, samemax) procedure is called for each iteration and the best result found is saved until the number of calls exceeds lmax or an optimal solution is found.

Algorithm 2 Least wasted first heuristic algorithm

```
Begin
2D-loading (R, callmax, samemax)
best ← 0
for i: = 0 to lmax do
    area ← 2D-loading(R, callmax, samemax)
    if (area > best) then
        best ← area
        if (W *  H) or best = totarea then
            break
        end if
    end if
end for
return best
END
```

5.3 Design of the proposed AGA algorithm

The genetic algorithm (GA) was first proposed by Holland (1975). It is a search algorithm based on an evolutionary principle.

To solve the 2L-CVRP problem, we used an adaptive GA and the different steps of our proposed AGA are illustrated in Algorithm 3. The algorithm starts by generating a randomly number of individuals, and evaluating them afterwards using the fitness function that coincide (objective functions). Two solutions are selected randomly from the population using the tournament selection (Ombuki et al., 2006). Therefore each parent is selected and submitted to a recombination in pairs (crossover) to generate offspring solution, by using the ordered crossover (OX). The best obtained offspring goes through a mutation step using the exchange and inversion operator. The last step is to replace the old population by new population of offspring solutions using the elitism strategy which means that the best solution ever computed from a previous generation is automatically replicated and inserted as a member of the next generation. Thus, the new population is created and the algorithm iterates until a maximum number of generations is reached. We detailed the different steps of our approach in the next section.

Algorithm 3 The proposed AGA for 2L-CVRP

Input:

N: set of customer; *Q*: vehicle capacity,

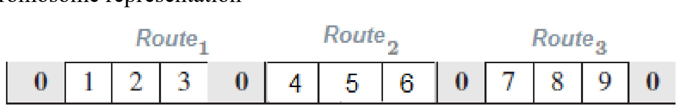
population *P*(*t*), *t*: number of generation {GA parameters}

```
Begin
Create an initial population
Evaluate each chromosome in P(t) {the fitness function is evaluated for the initial
population}
while stopping criterion is not satisfied do
  Initialise a temporary population p'
  for i = 1 to –P(t)– do
    Select two parents from P(t) {using the tournament selection}
    Apply crossover (offspring) {using the ordered crossover}
    if the offspring and parents are identical then
      Improve each offspring by the mutation operator {using the exchange and inversion
      mutation}
    end if
  end for
  Replace the old population P(t) by the new P'; {using the elitism replacement strategy}
end while
Output: Best found solution {routes with shortest total travelled distance}
END
```

5.3.1 Solution encoding and evaluation

The 2L-CVRP solution encoding is based on indexed array to present the chromosomes and each chromosome includes a set of customers, to be visited by an assigned vehicle. The 2L-CVRP possible solution is represented by a set of chromosomes, it can be considered as valid solution if all constraints for the loading and routing problems are satisfied. Figure 3 presents an example of a chromosome of three routes with nine customers where the node 0 indicates the centre depot.

Figure 3 Chromosome representation



5.3.2 Initial population

The initial population is generated randomly in order to scan a larger portion of the search space: first, a customer is selected at random and placed as the first location to be visited on the first route. Then, a different random customer is chosen and, if the capacity and two-dimensional loading constraints would be met, it is placed on the current route after the previous customer. If any of the constraints are not met, a new route is created and this customer is the first location to be visited on that route. This process is repeated until all customers have been assigned to a route.

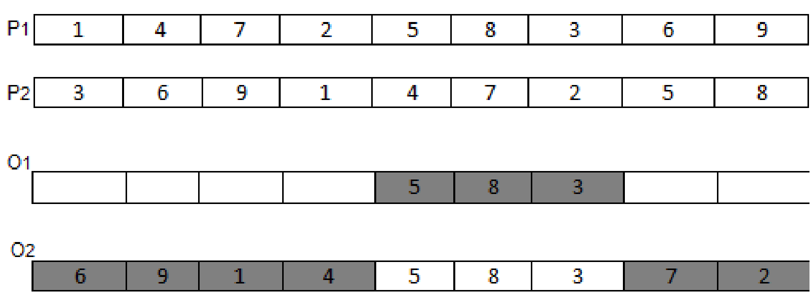
5.3.3 Selection

In order to select parent chromosomes and offspring, every individual is evaluated in terms of its fitness value based to the objective function (Masum et al., 2011). Many methods are used to select an individuals such as roulette wheel method, ranking method, in this paper we use an efficient combination of selection and ranking methods namely the tournament selection where parent is selected by a subgroup from the population or by choosing the best individual from a set of individuals.

5.3.4 Crossover

Crossover reproduces a duplication between two parents used to vary the programming of chromosomes from one generation to the next one. It recombines a pair of solutions in a certain way generating one or more offspring's. To create a new child, both implemented crossovers take information from one individual and insert into an other individual. They do not perform mutual exchange of genetic material between two parents. As a crossover operator we use the ordered crossover (OX) (Karakatic and Podgorelec, 2015) with the crossover probability is 1. Figure 4 presents an example of the ordered crossover.

Figure 4 Ordered crossover (OX)



The first step on the OX is to select two points randomly. Then, the substring between the two sections points on the first parent is copied to the offspring. After that, replace the remaining positions by following the customer order on the second parent, starting at the position just after the second cross point. The procedure resumes at position 1 when the end of the sequence is obtained.

5.3.5 Mutation

The mutation operator brings random changes into the new chromosome generated by crossover. It is applied to a single solution with a certain probability. We select two random positions in new chromosome and interchanged those positions using exchange and inversion mutation (Karakatic and Podgorelec, 2015). A simple example of the exchange and inversion mutation operator is shown in Figure 5 and Figure 6 respectively.

Figure 5 Exchange

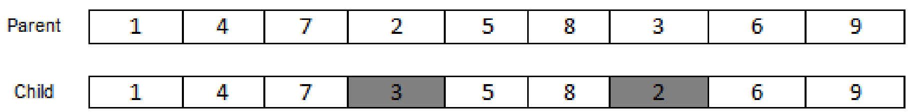
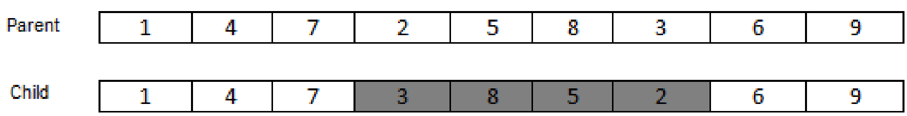


Figure 6 Inversion



5.3.6 Elitism replacement strategy

We applied the elitism strategy as a replacement strategy which means that the best solution ever computed from a previous generation is automatically replicated and inserted as a member of the next generation.

The stopping criterion can be defined in the number of generations with no improvement in the best solution found and in terms of elapsed computer time. In the proposed AGA the stopping criteria is the number of 1,000 generations.

6 Computational experiments

The proposed AGA is implemented using Java language version 7, and it was executed using the Netbeans 8.0.2 integrated development environment. All our experiments were performed on a PC equipped Intel(R) Core (TM) i3-4005U CPU with 4 Go RAM under Microsoft Windows 7.

Table 1 reports the parameters of our AGA algorithm. The computational environments for these approaches are summarised in Table 2. All these approaches were executed 30 times for each instance by setting different random seeds.

In order to assess the performance of our algorithm, we compare our approach with the following approaches:

- *GRASP * ELS:* by Duhamel et al. (2011)
- *PRMP:* by Zachariadis et al. (2013)
- *VNS:* by Wei et al. (2015).

Table 1 A meta-tuning of the AGA

Parameters	*Values*
Population size (N)	100
Selection	The tournament selection
Crossover rate	0.65
Mutation rate	0.20
Offsprings	300
Replacement strategy	The elitism operator
Maximum number of generation	1,000

6.1 Benchmark

To test the performance and robustness of the proposed AGA, we tested it over 150 2L-CVRP benchmark problems. All instances are introduced by Iori et al. (2007) and Gendreau et al. (2008). These instances were derived from 30 CVRP instances, by expressing the customer demand as a set of two-dimensional, weighted and rectangular items. Five classes of the item demand characteristics introduced by Iori et al. (2007) are generated, it may be downloaded from http://www.or.deis.unibo.it/research.html. Some details on the instances generated in this way are reported in Table 3 by generating the number of vehicles K.

The dimensions values of loading surface of each vehicle are $L = 40$ and $W = 20$ for all instances.

Table 2 Computational environments for different methods

Methods	*CPU*	*RAM*
GRASP-ELS	Opteroun 2.1 GHz	-
PRMP	Intel Core 2 Duo E6600 2.4 GHz	-
VNS	Intel XeonE5430 (QuadCore)	-
AGA	Intel(R) Core(TM) i3 CPU170 GHz	4Go

Table 4 denotes the characteristics of 2L-CVRP benchmark instances. The first column gives the name of the original CVRP instance and the second one the number of customers n. The five following groups of four columns give, for each class, the total number of items m and the number of vehicle k. We used the same number of customers for the five classes and the position and demand value of each customer is presented also in the same Table 3.

- *Class 1:* the problems of class 1 are in fact standard CVRP instances, as every customer sequence is feasible in terms of the loading constraints of the problem examined. Each customer is associated a single item of width and length equal to nil. They are used to test the algorithmic effectiveness in terms of the routing aspects of the problem examined.

- *For classes 2 to 5:* each of the five classes of instances has the same number of customers, and the position and demand value of each customer is also in Table 3. The number of items demanded by customer i, m_i is a random value in a given interval.

Table 3 Characteristics of the 2L-CVRP benchmark instances

Instance		*n*	*Class 1*		*Class 2*		*Class 3*		*Class 4*		*Class 5*	
			m	*K*	*m*	*K*	*m*	*K*	*m*	*K*	*m*	*K*
1	E016-03m	15	15	3	24	3	31	3	37	3	45	4
2	E016-05m	15	15	5	25	5	31	5	40	5	48	5
3	E021-03m	20	20	4	29	5	46	5	44	5	49	5
4	E021-06m	20	20	6	32	6	43	6	50	6	62	6
5	E022-06g	21	21	4	31	4	37	4	41	4	57	5
6	E022-06m	21	21	6	33	6	40	6	57	6	56	6
7	E023-03g	22	22	3	32	5	41	5	51	5	55	6
8	E023-05s	22	22	5	29	5	42	5	48	5	52	6
9	E026-08m	25	25	8	40	8	61	8	63	8	91	8
10	E030-03g	29	29	3	43	6	49	6	72	7	86	7
11	E030-04s	29	29	4	43	6	62	7	74	7	91	7
12	E031-09h	30	30	9	50	9	56	9	82	9	101	9
13	E033-03n	32	32	3	44	7	56	7	78	7	102	8
14	E033-04g	32	32	4	47	7	57	7	65	7	87	8
15	E033-05s	32	32	5	48	6	59	6	84	8	114	8
16	E036-11h	35	35	11	56	11	74	11	93	11	114	11
17	E041-14h	40	40	14	60	14	73	14	96	14	127	14
18	E045-04f	44	44	4	66	9	87	10	112	10	122	10
19	E051-05e	50	50	5	82	11	103	11	134	12	157	12
20	E072-04f	71	71	4	104	14	151	15	178	16	226	16
21	E076-07s	75	75	7	114	14	164	17	168	17	202	17
22	E076-08s	75	75	8	112	15	154	16	198	17	236	17
23	E076-10e	75	75	10	112	14	155	16	179	16	225	16
24	E076-14s	75	75	14	124	17	152	17	195	17	215	17
25	E101-08e	100	100	8	157	21	212	21	254	22	311	22
26	E101-10c	100	100	10	147	19	168	20	247	20	310	20
27	E101-14s	100	100	14	152	19	211	22	245	22	320	22
28	E121-07c	120	120	7	183	23	242	25	299	25	384	25
29	E135-07f	134	134	7	197	24	262	26	342	28	422	28
30	E151-12b	150	150	12	225	29	298	30	366	30	433	30

Table 4 The item characteristics for classes 2–5 instances

Class	*m*	*Vert*		*Homog*		*Horiz*	
		L	*w*	*L*	*w*	*L*	*w*
2	[1, 2]	[0:4, 0:9]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]
3	[1, 3]	[0:4, 0:8]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]
4	[1, 4]	[0:4, 0:7]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]
5	[1, 5]	[0:4, 0:6]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]	[0:4, 0:9]	[0:1, 0:2]

Results obtained by our proposed AGA are reported in Tables 4–6 in order to compare GRASP * ELS (Duhamel et al., 2011), PRMP (Zachariadis et al., 2013) and the VNS (Wei et al., 2015).

Moreover, we used a %gap that describes the percent gap between our solution scores and BKS and is defined as follows:

$$\%gap := 100\left(\left(BKS - C_{GA}\right)/BKS\right) \tag{13}$$

where

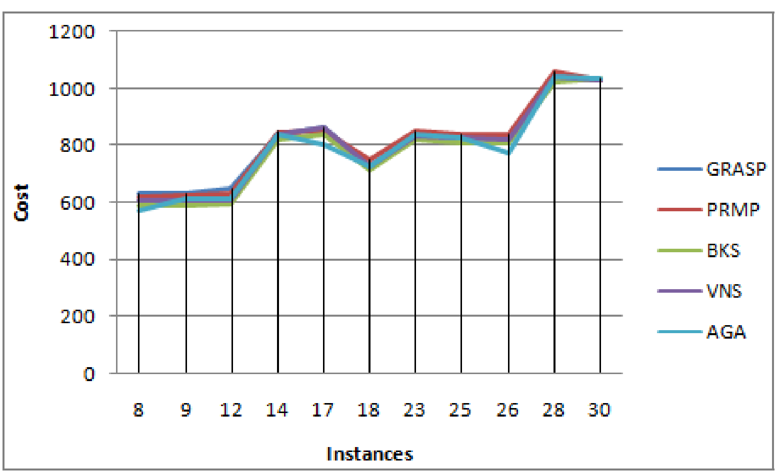
Best known solution (*BKS*) is the best algorithmic solution (among EGTS + LBFH, ACO, GRASP * ELS, PRMP)

Average (*Avg*) is the sum of a list of numbers divided by the number of numbers in the list.

Table 4 presents all the results obtained by our AGA from class 1 to class 5.

Results obtained by our algorithm on the CVRP instances of class 1 are reported in Table 5.

Figure 7 Comparison of algorithms performance from class 1 (see online version for colours)



We compare our results with existing methods GRASP * ELS (Duhamel et al., 2011), PRMP (Zachariadis et al., 2013) and VNS (Wei et al., 2015). Table 6 shows that the best results are obtained by our method over ten runs on each instance and for 30 instances from class 1. Our approach GA provides the highest quality solutions for two examined instances (17 and 26) when it reaches the minimum average cost. In addition, our approach robustly matched the best value for 26 instances compared to the best ones obtained by previous approaches (BKS). Regarding the VNS proposed by Wei et al. (2015), it matches the worst results for five instances 10, 28, 31, 32 and 36 where the %gap are –0.01, –0.14, –0.22, –0.26, –0.18, respectively. Moreover, it provides the worst average scores achieved *Avg.* of %gap is equal to –0.02%. Whereas, our AGA gives the best *Avg.* equal to 0.001.

Table 5 Results for the 2L-CVRP from class 1 to 5

Instance		*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*
1	E016-03m	278.73	278.73	282.52	284.95	278.73
2	E016-05m	334.96	332.96	354.16	336.96	332.96
3	E021-03m	358.40	387.72	394.70	362.45	360.40
4	E021-06m	430.88	430.89	430.89	447.37	430.86
5	E022-06g	375.28	375.28	381.72	383.88	375.28
6	E022-06m	495.85	495.85	498.16	497.22	495.85
7	E023-03g	568.56	725.46	678.75	700.72	657.75
8	E023-05s	568.56	674.54	738.43	692.47	609.90
9	E026-08m	607.65	607.65	607.65	621.25	607.65
10	E030-03g	535.74	689.82	624.49	710.87	678.66
11	E030-04s	505.01	693.45	706.73	759.00	624.82
12	E031-09h	610.00	610.57	610.00	614.24	586.07
13	E033-03n	2,006.37	2,527.72	2,468.06	2,548.06	2,334.78
14	E033-04g	837.67	1,040.72	1,003.52	981.00	875.00
15	E033-05s	837.67	1,017.95	1,155.22	1,181.30	1,120.33
16	E036-11h	698.61	698.61	698.61	680.69	698.61
17	E041-14h	861.78	863.67	861.79	861.79	861.79
18	E045-04f	723.54	1,004.99	1,070.43	1,082.48	917.94
19	E051-05e	524.61	757.44	750.7	775.87	644.59
20	E072-04f	241.97	531.22	521.31	537.59	471.64
21	E076-07s	687.60	997.63	1,117.59	973.03	877.75
22	E076-08s	740.66	1,030.71	1,056.11	1,054	932.38
23	E076-10e	835.26	1,035.18	1,059.79	1,071.30	935.33
24	E076-14s	1,024.69	1,122.95	1,080.88	1,108.34	1,042.83
25	E101-08e	826.14	1,409.24	1,369.26	1,391.13	1,150.69
26	E101-10c	819.53	1,272.87	1,345.45	1,394.19	1,234.65
27	E101-14s	1,082.65	1,313.12	1,370.40	1,316.19	1,249.17
28	E121-07c	1,040.70	2,555.29	2,610.56	2,602.93	2,308.18
29	E135-07f	1,162.96	2,201.34	2,087.15	2,246.75	2,128.84
30	E151-12b	1,028.42	1,808.77	1,825.97	1,826.10	1,521.00

According to Table 5, %gap are less than 3%. In summary, the %gap between the proposed GA algorithm and the other three methods for class 1 shows that the proposed algorithm is promising in terms of solutions founded.

Figure 7 shows the performance of our algorithm against GRASP, PRMP, BKS and VNS for class 1.

Table 6 Comparative results on standard CVRP instances of class 1

Instance		*GRASP (1)*	*PRMP (2)*	*BKS*	*VNS (3)*	*Proposed AGA*	*%Gap*
1	E016-03m	278.73	278.73	278.73	278.73	278.73	0.0
2	E016-05m	334.96	334.96	334.96	334.96	334.96	0.0
3	E021-03m	358.40	358.40	358.40	358.40	358.40	0.0
4	E021-06m	430.88	430.88	430.88	430.89	430.88	0.0
5	E022-06g	375.28	375.28	375.28	375.28	375.28	0.0
6	E022-06m	495.85	495.85	495.85	495.85	495.85	0.0
7	E023-03g	568.56	568.56	568.56	568.56	568.56	0.0
8	E023-05s	568.56	568.56	568.56	568.56	568.56	0.0
9	E026-08m	607.65	607.65	607.65	607.65	607.65	0.0
10	E030-03g	535.80	535.80	535.74	535.80	535.74	0.0
11	E030-04s	505.01	505.01	505.01	505.01	505.01	0.0
12	E031-09h	610.00	610.00	610.00	610.00	610.00	0.0
13	E033-03n	2,006.34	2,006.34	2,006.34	2,006.34	2,006.37	–0.001
14	E033-04g	837.67	837.67	837.67	837.67	837.67	0.0
15	E033-05s	837.67	837.67	837.67	837.67	837.67	0.0
16	E036-11h	698.61	698.61	698.61	698.61	698.61	0.0
17	E041-14h	861.79	861.79	861.79	861.79	*861.78*	0.001
18	E045-04f	723.54	723.54	723.54	723.54	723.54	0.0
19	E051-05e	524.61	524.61	524.61	524.61	524.61	0.0
20	E072-04f	241.97	241.97	241.97	241.97	241.97	0.0
21	E076-07s	687.60	687.60	687.60	687.60	687.60	0.0
22	E076-08s	740.66	740.66	740.66	740.66	740.66	0.0
23	E076-10e	835.26	835.26	835.26	835.26	835.26	0.0
24	E076-14s	1,024.60	1,024.69	1,024.69	1,024.69	1,024.69	0.0
25	E101-08e	827.39	826.14	826.14	826.14	826.14	0.0
26	E101-10c	819.56	819.56	819.56	819.56	*819.53*	0.003
27	E101-14s	1,082.65	1,082.65	1,082.65	1,082.65	1,082.65	0.0
28	E121-07c	1,042.12	1,042.12	1,040.70	1,042.12	1,040.70	0.0
29	E135-07f	1,162.96	1,162.96	1,162.96	1,162.96	1,162.96	0.0
30	E151-12b	1,033.42	1,028.42	1,028.42	1,028.42	1,028.42	0.0
Avg.		697.82	721.73	721.68	721.73	*721.67*	0.001

Notes: (1) Duhamel et al. (2011), (2) Zachariadis et al. (2013) and (3) Wei et al. (2015).

Also, to empirically test the effectiveness of our AGA algorithm for class 2 to 5, we used 144 benchmark problems. It has also been executed 30 times on each of the 120 2L-CVRP benchmark instances of classes 2–5. The results are summarised in Table 7.

Table 7 Comparative results on standard CVRP instances for classes 2–5

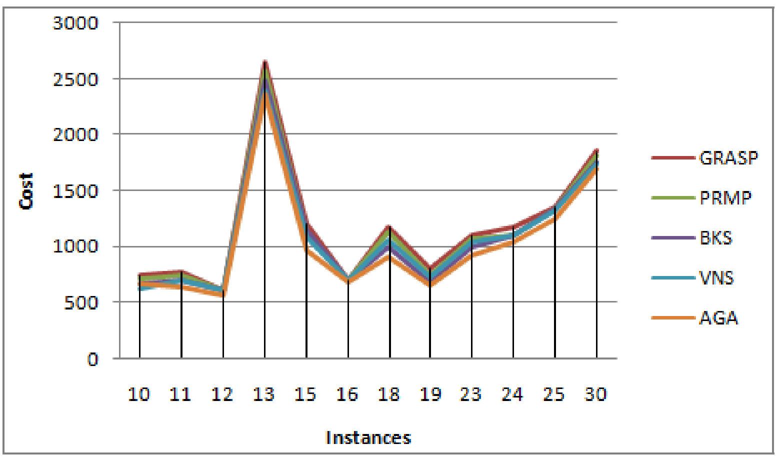
Instance		*GRASP (1)*	*PRMP (2)*	*BKS*	*VNS (3)*	*Proposed AGA*	*%Gap*
1	E016-03m	282.65	281.23	281.23	281.23	281.23	0.0
2	E016-05m	339.26	339.26	339.26	339.26	339.26	0.0
3	E021-03m	376.32	376.32	376.32	376.32	376.32	0.0
4	E021-06m	435.01	435.01	435.00	435.01	435.00	0.0
5	E022-06g	379.03	379.03	379.03	379.03	379.04	–0.002
6	E022-06m	497.04	497.04	496.77	497.04	496.77	0.0
7	E023-03g	691.11	690.67	690.67	690.67	690.67	0.0
8	E023-05s	678.84	678.84	678.84	678.84	678.84	0.0
9	E026-08m	612.01	612.01	611.05	612.01	611.05	0.0
10	E030-03g	675.79	676.75	675.59	674.92	675.96	–0.05
11	E030-04s	705.95	703.22	702.96	702.47	*696.00*	0.99
12	E031-09h	611.26	611.26	611.26	611.20	*605.22*	0.988
13	E033-03n	2,490.62	2,491.18	2,486.17	2,484.16	*2,469.55*	0.668
14	E033-04g	984.42	975.88	974.04	975.06	975.06	–0.10
15	E033-05s	1,144.69	1,132.91	1,128.86	1,128.60	*1,118.70*	0.90
16	E036-11h	699.79	699.79	699.79	699.79	*694.13*	0.808
17	E041-14h	864.05	864.05	862.26	864.05	862.26	0.0
18	E045-04f	1,029.71	1,031.95	1,028.61	1,027.98	*1,018.96*	0.938
19	E051-05e	739.19	741.78	739.15	737.73	*732.15*	0.947
20	E072-04f	522.68	515.44	515.44	515.92	515.44	0.0
21	E076-07s	994.58	992.78	991.50	991.63	991.50	0.0
22	E076-08s	1,021.45	1,023.01	1,017.33	1,019.03	1,018.30	–0.095
23	E076-10e	1,038.16	1,032.36	1,032.36	1,036.40	*1,025.40*	0.674
24	E076-14s	1,107.93	1,104.64	1,099.57	1,102.53	*1,088.75*	0.984
25	E101-08e	1,345.08	1,341.26	1,340.18	1,333.76	*1,330.08*	0.753
26	E101-10c	1,317.41	1,311.79	1,311.79	1,306.60	1,311.79	0.0
27	E101-14s	1,323.54	1,318.04	1,318.04	1,311.27	*1,312.22*	0.441
28	E121-07c	2,560.06	2,530.46	2,530.46	2,519.35	*2,519.24*	0.443
29	E135-07f	2,191.46	2,173.02	2,173.02	2,166.14	*2,166.02*	0.322
30	E151-12b	1,775.44	1,760.59	1,760.16	1,746.82	*1,745.46*	0.835
Avg.		981.151	977.385	976.223	974.827	*972.012*	0.431

Notes: (1) Duhamel et al. (2011), (2) Zachariadis et al. (2013) and (3) Wei et al. (2015).

We can notice that the proposed AGA is stable in terms of the quality of the solutions generated in each run. The % gap between the average and highest quality solution provides the highest quality solutions for 14 examined instances (11, 12, 13, 15, 16, 18, 19, 23, 24, 25, 27, 28, 29 and 30).

According to Table 7, the %gap are less than 10%. In addition, the VNS proposed by Wei et al. (2015) matches the worst results for eight instances (6, 9, 14, 17, 20, 21, 22 and 24), where the %gap are equal to –0.05, –0.16, –0.11, –0.21, –0.09, –0.01, –0.17, –0.27, respectively compared to our AGA which provides worse results for just four instances (5, 10, 14, 22) for %gap equal to (–0.002, –0.05, –0.10 and –0.095 respectively). Note that our AGA robustly matched the best value for 13 instances compared to the best ones obtained by previous approaches (BKS). Figure 8 shows the performance of our algorithm against GRASP, PRMP, BKS and VNS for class 2 to 5.

Figure 8 Comparison of algorithms performance from class 2 to 5 (see online version for colours)



In summary, the %gap between the proposed AGA algorithm and the other three methods for class 2–5 shows that the performance of our proposed algorithm is better. The results deemed reasonable, taking into account both the complexity of the 2L-CVRP model and the high quality of the obtained results.

6.2 Statistical analysis

The statical analysis aims to compare the performance of the proposed AGA against GRASP, PRMP, BKS and VNS solutions using an analysis test namely, a one way analysis of variance (ANOVA) test based on the numerical data provided by different performance measures.

We use the Friedman test to carry out multiple comparisons among our AGA, GRASP, PRMP, BKS and VNS. This test is useful to detect differences among the set of algorithms considered for the analysis.

If the difference between the variance of values provided by the different algorithms (AGA, GRASP, PRMP, BKS and VNS) is less than the confidence level (α), this leads to the rejection of H_0. Otherwise, it is concluded that the results are statistically significant.

The first step of Friedman test is to consider the following hypothesis:

H_0: Null hypothesis — There is no significant difference among the algorithms

H_1: Alternate hypotheis — There is a significant difference among the algorithms

The level of significance (α) is set to 0.05. The degree of freedom is 4 ($k - 1$, where k is the number of algorithm to be compared).

6.2.1 Friedman test of the 2L-CVRP for class 1

The objective function values generated by GRASP, PRMP, BKS, VNS and the proposed AGA for class 1 is presented in Table 6. Table 8 defines the Friedman test result of the 2L-CVRP for class 1 where n is the number problems, χ^2 is the calculated value, *dof* is the degree of freedom and ρ is the confidence level.

n (=30) is sufficiently large therefore the sampling distribution of χ^2 is a close approximation of the sampling distribution of χ^2 with degree of freedom of 4.

Table 8 Friedman test results of the 2L-CVRP for class 1

Mean ranks for samples	GRASP	PRMP	BKS	VNS	AGA
	2.1	2.1	2.06	2.16	2.03
Statistics test	k	n	χ^2	*dof*	ρ
	5	30	9.322	4	0.0067

Regarding Table 8, we show that the table value for 0.05 level of significance and the degree of freedom of 4 is 7.472, which is less than χ^2 the calculated value, the null hypothesis is rejected. The conclusion is that there is a significant difference among the algorithms considered for analysis. Based on the results provided by this test, our proposed AGA is effective in terms of solution found for the class 1.

6.2.2 Friedman test of the 2L-CVRP from class 2 to 5

The objective function values generated by GRASP, PRMP, BKS, VNS and the proposed AGA for class 1 is presented in Table 7. Table 9 defines the Friedman test result of the 2L-CVRP for class 1 where n is the problem numbers, χ^2 is the calculated value, *dof* is the degree of freedom and ρ is the confidence level.

n (=30) is sufficiently large therefore the sampling distribution of χ^2 is a close approximation of the sampling distribution of χ^2 with degree of freedom of 4.

Table 9 Friedman test results of the 2L-CVRP from class 2–5

Mean ranks for samples	GRASP	PRMP	BKS	VNS	AGA
	3.04	2.6	1.96	1.96	1.43
Statistics test	k	n	χ	*dof*	ρ
	5	30	8.58	4	0.0052

Regarding Table 8, we show that the table value for 0.05 level of significance and the degree of freedom of 4 is 7.472, which is less than χ^2 the calculated value, the null hypothesis is rejected. The conclusion is that there is a significant difference among the algorithms considered for analysis. Based on the results provided by this test, our proposed AGA is effective in terms of solution found for the class 2–5.

The results described in this paper, show that the proposed AGA yields consistently better results than GRASP, PRMP, BKS and VNS solutions. The results of this experimentation demonstrate that the proposed AGA is an effective instrument to find high-quality solutions for the 2L-CVRP. This is clear in all tables, graphs and validated by an ANOVA test.

7 Conclusions

An AGA for solving the capacitated vehicle routing problem with two-dimensional loading constraints (2L-CVRP) is proposed in this paper. The proposed approach is developed to mainly address the routing and the packing aspects using a diversification operators to obtain the optima solution.

The effectiveness of our approach is tested through experiments on widely used benchmark instances. Results showed that our proposed approach is competitive in terms of the quality of the solutions found. This is clear in all tables, graphs and validated by an ANOVA test.

For future work, we can apply our method of AGA for solving the CVRP with 2BPP under multi-objective framework denoted by an AGA approach for solving the 2L-CVRP with multi-objective case.

References

Baker, B. and Ayechew, M. (2003) 'A genetic algorithm for the vehicle routing problem', *Computers & Operations Research*, Vol. 30, No. 5, pp.787–800.

Berger, J. and Barkaoui, M. (2003) 'A hybrid genetic algorithm for the capacitated vehicle routing problem', in *Genetic and Evolutionary Computation Conference*, Springer Berlin Heidelberg, pp.646–656.

Dominguez, O., Juan, A.A., Barrios, B., Faulin, J. and Agustin, A. (2015) 'Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet', *Annals of Operations Research*, Vol. 236, No. 2, pp.383–404.

Duhamel, C., Lacomme, P., Quilliot, A. and Toussaint, H. (2011) 'A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem', *Computers & Operations Research*, Vol. 38, No. 3, pp.617–640.

Faiz, S. and Krichen, S. (2012) *Geographical Information Systems and Spatial Optimization*, CRC Press, Boca Raton.

Fuellerer, G., Doerner, K., Hartl, R. and Iori, M. (2009) 'Ant colony optimization for the two-dimensional loading vehicle routing problem', *Computers & Operations Research*, Vol. 36, No. 3, pp.655–673.

Gendreau, M., Iori, M., Laporte, G. and Martello, S. (2008) 'A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints', *Networks: An International Journal*, Vol. 51, No. 1, pp.4–18.

Holland, J.H. (1975) 'Adaptations in natural and artificial systems: an introductory analysis with applications to biology', *Control and Artificial Intelligence*, p.228, MIT Press, Cambridge.

Iori, M., Salazar, J.J. and Vigo, D. (2007) 'An exact approach for the vehicle routing problem with two-dimensional loading constraints', *J. Transportation Science*, Vol. 41, No. 2, pp.253–264.

Karakatic, S. and Podgorelec, V. (2015) 'A survey of genetic algorithms for solving multi depot vehicle routing problem', *Applied Soft Computing*, Vol. 27, No. C, pp.519–532.

Khebbache, S., Prins, C., Yalaoui, F. and El Mhamedi, A. (2010) 'Problmes de tournes de vhicules avec chargement et fenłtres horaires (2L-CVRPTW)', in *8th International Conference of Modeling and Simulation*, p.11.

Leung, S., Zhang, Z., Zhang, D., Hua, X. and Lim, M. (2013) 'A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints', *European Journal of Operational Research*, Vol. 225, No. 2, pp.199–210.

Leung, S.C., Zhou, X., Zhang, D. and Zheng, J. (2011) 'Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem', *Computers Operations Research*, Vol. 38, No. 1, pp.205–215.

Martello, S., Pisinger, D. and Vigo, D. (2000) 'The three-dimensional bin packing problem', *Operations Research*, Vol. 48, No. 2, pp.256–267

Masum, A.K.M., Shahjalal, M., Faruque, F. and Sarker, I.H. (2011) 'Solving the vehicle routing problem using genetic algorithm', *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 7, pp.126–131.

Ombuki, B., Ross, B.J. and Hanshar, F. (2006) 'Multi-objective genetic algorithms for vehicle routing problem with time windows', *Applied Intelligence*, Vol. 24, No. 1, pp.17–30.

Pinto, T., Alves, C. and de Carvalho, J.V. (2017) 'Variable neighborhood search algorithms for pickup and delivery problems with loading constraints', *Electronic Notes in Discrete Mathematics*, Book Section 34, Vol. 58, pp.111–118.

Pollaris, H., Braekers, K., Caris, A., Janssens, G.K. and Limbourg, S. (2015) 'Vehicle routing problems with loading constraints: state-of-the-art and future directions', *OR Spectrum*, Vol. 37, No. 2, pp.297–330.

Toth, P. and Vigo, D. (2002) 'The vehicle routing problem', in *The Vehicle Routing Problem*, pp.1–26, Society for Industrial and Applied Mathematics, Philadelphia.

Wei, L., Zhang, D. and Chen, Q. (2009) 'A least wasted first heuristic algorithm for the rectangular packing problem', *Computers & Operations Research*, Vol. 36, No. 5, pp.1608–1614.

Wei, L., Zhang, Z., Zhang, D. and Leung, S.C. (2017) 'A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints', *European Journal of Operational Research*, Vol. 265, No. 3, pp.843–859.

Wei, L., Zhang, Z., Zhang, D. and Lim, A. (2015) 'A variable neighborhood search for the capacitated vehicle routing problem with two-dimensional loading constraints', *European Journal of Operational Research*, Vol. 243, No. 3, pp.798–814.

Zachariadis, E., Tarantilis, C. and Kiranoudis, C. (2009) 'A guided tabu search for the vehicle routing problem with two-dimensional loading constraints', *European Journal of Operational Research*, Vol. 195, No. 3, pp.729–743.

Zachariadis, E., Tarantilis, C. and Kiranoudis, C. (2013) 'Integrated distribution and loading planning via a compact metaheuristic algorithm', *J. European Journal of Operational Research*, Vol. 228, No. 1, pp.56–71.