

A Hybrid Heuristic for the Probabilistic Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints

Soumaya Sassi Mahfoudh¹[0000-0003-2672-881X] and Monia Bellalouna²[0000-0002-3261-9452]

University of Manouba, National School of Computer Science, Laboratory CRISTAL-GRIFT, Tunisia

¹ soumaya.lsm@gmail.com

² monia.bellalouna@gmail.com

Abstract. The Probabilistic Capacitated Vehicle Routing Problem (PCVRP) is a generalization of the classical Capacitated Vehicle Routing Problem (CVRP). The main difference is the stochastic presence of the customers, that is, the number of them to be visited each time is a random variable, where each customer associates with a given probability of presence.

We consider a special case of the PCVRP, in which a fleet of identical vehicles must serve customers, each with a given demand consisting in a set of rectangular items. The vehicles have a two-dimensional loading surface and a maximum capacity.

The resolution of problem consists in finding an a priori route visiting all customers which minimizes the expected length over all possibilities. We propose a hybrid heuristic, based on a branch-and-bound algorithm, for the resolution of the problem. The effectiveness of the approach is shown by means of computational results.

Keywords: vehicle routing · bin packing · probability · hybrid heuristic · exact algorithms

1 Introduction

In the last decades, several variants of Vehicle Routing Problem (VRP) have been studied since the initial work of Dantzig and Ramser [9]. In its classical form, the VRP consists in building routes starting and ending at a depot, to satisfy the demand of a given set of customers, with a fleet of identical vehicles. In particular, the capacitated Vehicle Routing Problem (CVRP) is a well known variation of the VRP [25], defined as follows : we are given a complete undirected graph $G = (V, E)$ in which V is the set of $n + 1$ vertexes corresponding to the depot and the n customers. For each vertex v_i is associated a demand d_i . The demand of each customer is generally expressed by a positive integer that represents the weight of the demand. A set of K identical vehicles is available at the depot. Each vehicle has a capacity Q . The CVRP calls for the determination

of vehicles routes with minimal total cost.

Recently, mixed vehicle routing and packing problems have been studied [16, 15, 8]. This problem is called CVRP with two-dimensional loading constraints (2L-CVRP). In 2L-CVRP, two-dimensional packing problems are considered where all vehicles have a single rectangular loading surface whose width and height are equal to W and H , respectively. Each customer is associated with a set of rectangular items whose total weight is equal to d_i . Solving 2L-CVRP consists in determining the optimal set of vehicles routes of minimal cost to satisfy the demand of the set of customers and without overlapping the loading surface. It is clear that 2L-CVRP is strongly NP-Hard since it generalizes the CVRP.

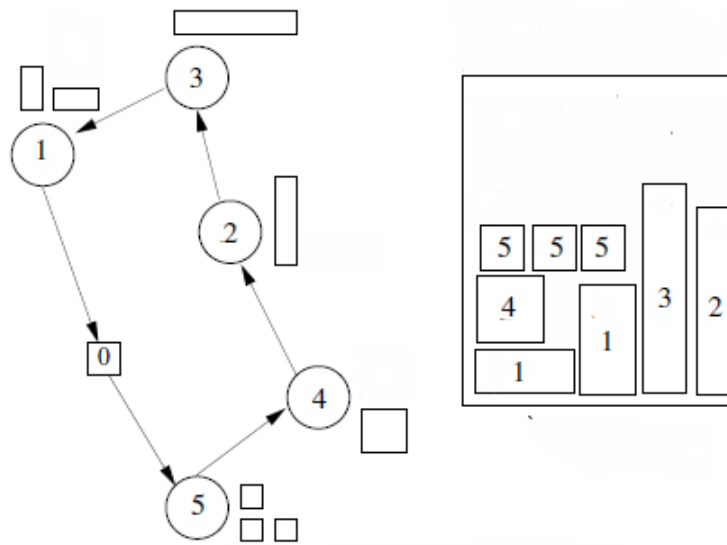


Fig. 1. Example of solution of 2L-CVRP

The 2L-CVRP has an important applications in the fields of logistics or goods distribution. However, in almost all-real world applications, randomness is an inherent characteristic of the problems. For example, in practice whenever a company, in a given day, is faced with only deliveries to a random subset of its usual set of customers, it will not be able to redesign the routes every day since it is not sufficiently important to justify the required effort and cost. Besides, the system's operator may not have the resources for doing so. Even more importantly the operator may have other priorities.

In both cases, the problem of designing routes can very well modeled by introducing a new variant of 2L-CVRP with the above described constraints, denoted as 2L-PCVRP. Among several motivations, 2L-PCVRP is introduced to formulate and analyze models which are more appropriate for real world problems. 2L-

PCVRP is characterized by the stochastic presence of the customers. It means that the number of customers to be visited each time is a random variable, where the probability that a certain customer requires a visit is given. This class of problems differs from the Stochastic Vehicle Routing Problem (SVRP) in the sense that here we are concerned only with routing costs without the introduction of additional parameters ([5, 3, 22, 23]).

2L-PCVRP has not been previously studied in the literature. The only closely related work we are aware of is the deterministic version 2L-CVRP [16] and Probabilistic Vehicle Routing Problem (PVRP) [4].

In fact, compared with the number of studies available on classical CVRP, relatively few papers have been published on optimizing both routing of vehicles and loading constraints. For deterministic 2L-CVRP, Iori et al. [16] presented an exact algorithm for the solution of the problem. The algorithm is making use of both classical valid inequalities from CVRP literature and specific inequalities associated with infeasible loading constraints. The algorithm was evaluated on benchmark instances from the CVRP literature, showing a satisfactory behavior for small-size instances. In order to deal with more larger sized instances, heuristic algorithms have been proposed. Gendreau et al. [13] developed a Tabu search for the 2L-CVRP. The resolution of the routing aspect is handled through the use of Taburoute, a Tabu search heuristic developed by Gendreau et al. [12] for the CVRP. Some improvements were proposed to the above Tabu search, Fuellerer et al. [10] have proposed an Ant Colony Optimization algorithm, as a generalization of savings algorithm by Clarke and Wright [7] through the addition of loading constraints.

Our aim is to develop a hybrid heuristic for the 2L-PCVRP, to solve the perturbation of the initial problem 2L-CVRP, through a combination of sweep heuristic [14] and a branch and bound algorithm[1].

This paper is organized as follows : section 2 exhibits a detailed description of the problem. Section 3 presents the proposed algorithm for the solution of the 2L-PCVRP. In section 4, computational results are discussed and section 5 draws some conclusions.

2 Problem Description

2L-PCVRP is a NP-Hard since it generalizes two NP-Hard problems that have been treated separately 2L-CVRP and PVRP [4]. In fact, 2L-PCVRP belongs to the class of Probabilistic Combinatorial Optimization Problems (PCOP) which was introduced by Jaillet [17] and studied in ([4, 6, 1]).

So, 2L-PCVRP is formulated as follows. We consider a complete undirected graph $G = (V, E)$, in which V defines the set of $n + 1$ vertex corresponding to the depot v_0 and the n customers (v_1, \dots, v_n) . Each customer v_i is associated with a set of m_i rectangular items, whose total weight is equal to his demand d_i , and each having specific width and height equal to w_{il} and h_{il} , ($l = 1, \dots, m_i$). A fleet of identical vehicles is available. Each vehicle has a capacity Q and a rectangular loading surface S , for loading operations, whose width and height

are respectively equal to W and H .

Let P be the probability distribution defined on the subset of $V : \mathbb{P}(V)$, τ be a given a priori route through V . Let L_τ be a real random variable defined on, $\mathbb{P}(V)$, which in an a priori route τ and for each S of $\mathbb{P}(V)$, associates the length of the route through S . For each subset $S \subseteq V$, we consider \mathcal{U} a modification method, it consists in erasing from τ the absent vertices by remanning in the same order. The solution of the problem consists on finding an a priori route visiting all points that minimizes the expected length \mathbb{E} of a route τ [17, 4].

$$\mathbb{E}(L_{\tau, \mathcal{U}}) = \sum_{S \subseteq V} p(S) L_{\tau, \mathcal{U}}(S) \quad (1)$$

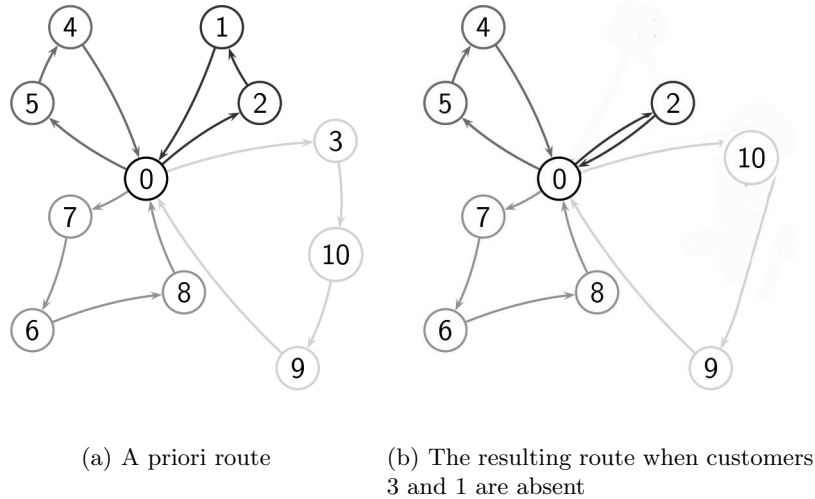


Fig. 2. Example of a priori route for PVRP

In our approach, the priori route is built satisfying the next conditions :

1. Each route starts and ends at the depot.
2. The total customers demands on one route does not exceed vehicle routing capacity Q .
3. All the items of a given customer must be loaded on the same vehicle (each customer is visited once).
4. Items don't have a fixed orientation (a rotation of 90° is allowed).
5. The items delivered on each route must fit within the loading surface S of the vehicle, their total weight should not exceed capacity Q .
6. The loading and unloading surface side of the vehicle is placed at height H .

An alternative approach to define 2L-PCVRP is as a combination of the probabilistic travelling salesman problem (PTSP), with a loading constraints which are closely related to the classical and extensively studied 2BPP ([20, 21]). The PTSP is the probabilistic version of the well-known problem Travelling salesman problem (TSP), which was introduced by Jaillet ([17, 18, 6]). PTSP calls for the determination of priori route of minimal expected total length and 2BPP occurs in determination of packing the given set of rectangular items into the loading surface of the vehicle.

3 Hybrid Heuristic for 2L-PCVRP

Few papers have proposed methods of resolution for PCOPs ([4, 24, 1]), in this section we present a hybrid heuristic for the solution of 2L-PCVRP based on a branch and bound algorithm for PTSP which proved to be a successful technique. The hybrid heuristic proceed in two stages :

1. Sweep Algorithm [14] : we determine clusters (groups of customers) satisfying the six conditions cited above.
2. Each obtained cluster is considered an instance of PTSP, we solve it with the branch and bound algorithm [1].

3.1 Sweep Algorithm

The customers are swept in a clockwise direction around a center gravity which is the depot and assigned to groups. Specifically, the sweep algorithm is the following :

Step 1 : Calculate for each customer, his polar coordinate θ_i in relation to the depot. Renumber the customers according to polar coordinates so that :

$$\theta_i < \theta_{i+1}, 1 \leq i \leq n \quad (2)$$

Step 2 : Start from the non clustered customer j with smallest angle θ_j construct a new cluster by sweeping consecutive customers $j+1, j+2 \dots$ until the capacity and the bounding of loading surface constraints will not allow the next customer to be added. This means that each cluster must lead to a feasible packing (items of cluster customers are packed without overlapping the loading surface).

It is clear that it is complex to take into account the 2BPP with the loading of the items into the vehicles, with the additional side constraint that all the items of any given customer must be loaded into the same bin. Concerning the complexity, note that the 2BPP is an NP-hard combinatorial optimization problem (see [11]). To this end, simple packing heuristic Bottom-Left [2] is used to solve 2BPP instances. Bottom-Left consists in packing the current item in the lowest possible position, left justified of open bin; if no bin can allocate it, a new one is initialized. The algorithm has $\mathcal{O}(n^2)$ time complexity.

Step 3 : Continue Step 2 until all customers are included in a cluster.

Step 4 : Solving each cluster, is tantamount solving an instance of PTSP, where each customer has a probability of presence p_i . The solution consists in finding a priori route visiting the cluster customers which minimizes the expected total length of the route τ .

3.2 Probabilistic Branch & Bound Algorithm

The overall aim is to perform a depth-first traversal of a binary tree by assigning to each branch a probabilistic evaluation. We consider M the distance matrix between the customers. This exact algorithm for solution of the PTSP, is based

Table 1. Matrix example

	A	B	C	D
A	∞	d_{AB}	d_{AC}	d_{AD}
B	d_{BA}	∞	d_{BC}	d_{BD}
C	d_{CA}	d_{CB}	∞	d_{CD}
D	d_{DA}	d_{DB}	d_{DC}	∞

on the expected length of a route introduced by Jaillet [17]. Let $d(i, j)$ be a distance between the customers $i = ABCD \dots = v_1 v_2 v_3 \dots$, $j = ABCD \dots = v_1 v_2 v_3 \dots$, and $p = P(k) \forall k \in \tau$, $q = 1 - p$ where p is the probability of presence, the expected length of a route τ is shown by (3).

$$E(L_\tau) = p^2 \sum_{r=0}^{n-2} q^r \sum_{i=1}^n d(i, T^r(i)) \quad (3)$$

Where $T^r(i)$ is the successor number r of i in the route τ .

This design takes the form of "Branch and Bound of Little et al. [19]" but in the probabilistic framework, by deriving the equations of the evaluations, in order to direct the search space towards the promising sub-spaces (i.e., the possibility of finding the optimal solution is very likely).

In the same manner of Littel's algorithm for the TSP, we reduce the matrix. The lower bound for the TSP equals Ev_{TSP} , which will help us to calculate the initial evaluation for the PTSP.

$$Ev_{TSP}(n) = \sum_{i=1}^n \min R_i + \sum_{j=1}^n \min C_j \quad (4)$$

Where R_i is the i^{th} row and C_i is the j^{th} column.

Let $G = (V, E, M)$ be a graph such as $|V| = n$, V is the set of vertices, E the set of edges and M is distance matrix. The probabilistic evaluation $P.E_{PTSP}$: which is defined as follows is considered as a lower evaluation for the PTSP.

$$P.E_\Omega = P.Ev_{PTSP}(n) = Ev_{TSP}(n) \left(p^2 \sum_{r=0}^{n-2} q^r \right) = Ev_{TSP}(n) p(1 - q^{n-1}) \quad (5)$$

This first evaluation associated with the root Ω of the tree shown in Fig. 3. Then, for the next two nodes of the tree, the next two transitional probabilistic evaluations are given due to choice of an arc, according to its effect on the construction of the optimal route. For the arc AB (the same for other arcs) :

1. Choose AB : increase the expected length of the route at least by

$$P.Ev_{AB} = P.Ev_{\Omega} + p^2 \sum_{r=1}^{n-2} q^r [\min_{X \neq A}^{(r)} d(A, X)] + p^2 Ev_{TSP_{Next}} \quad (6)$$

Where $Ev_{TSP_{Next}}$ is the evaluation of resulting matrix for the TSP where row A and column B are removed.

2. Not choose AB : increase the expected length of the route at least by

$$P.Ev_{\overline{AB}} = P.Ev_{\Omega} + p^2 [\min_{K \neq B}^{(1)} d(A, K) + \min_{K \neq A}^{(1)} (d(K, B))] \quad (7)$$

$$+ p^2 \sum_{r=2}^{n-2} q^r [\min_{X \neq K}^{(r)} d(A, X) + \min_{X \neq B}^{(r)} d(K, X)]$$

$P.Ev_{\overline{AB}}$ represents the probabilistic penalty cost for the arc \overline{AB} , $\min^{(i)} d(A, X)$ is the i^{th} minimum of row A, n is the size of the initial matrix. These formulas are valid for all iterations.

The construction starts from the root of the tree, which equals $P.E_{\Omega}$. The problem is divided into two sub-problems with the approach (depth-first, breadth-first) according to the probabilistic penalties cost. After the penalty calculation, it is easily to get the biggest probabilistic penalty cost. So, we separate according to this arc. First remove the row, column and replace the chosen arc by ∞ to prohibiting the parasitic circuits.

Table 2. Probabilistic penalties

	A	B	C	D
A	∞	$(P.Ev_{\overline{AB}})0(P.Ev_{AB})$	-	-
B	-	∞	-	-
C	-	-	∞	-
D	-	-	-	∞

$M=M_{reduced} =$

According to this probabilistic penalty calculation, we construct the first branching of the tree, which is shown in Fig. 3.

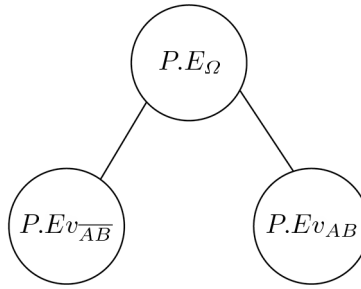


Fig. 3. Branching of the tree for the PTSP

The search continues until all branches that have been visited are either eliminated or the end of the process is reached. That is, the present evaluation is less than the all evaluations, which are defined by the expected length of each final branch by profiting that the expected length can be calculated in $O(n^2)$ time Jaillet [17].

4 Computational Results

In this section, we first compare our method over the benchmark proposed by Iori et al. [16] for the deterministic 2L-CVRP (we suppose that all customers are present). Next, we exhibit the new instances for 2L-PCVRP, before analyzing the performance of the proposed heuristic on those instances. The final results are reported at the end. The algorithm was coded in Java and all tests were performed on a typical PC with i5-5200U CPU (2.2GHz) under Windows 8 system.

4.1 Comparison on the 2L-CVRP

By assuming that all customers have a probability of occurrence ($p = 1$), deterministic instances of 2L-CVRP are obtained. In this case, our algorithm is only compared to the approach proposed by Iori et al. [16]. The algorithm of Iori was coded in C and was run on a PC with CPU 3GHz. Because Gendreau et al. [13] and Fuellerer et al. [10] do not solve the problem under the same constraints as we do, direct comparisons with their algorithms can not be made.

To test their algorithm, Iori et al. proposed a benchmark of five classes, for a total of 180 instances. The benchmark presents the coordinates (x_i, y_i) , the demand d_i and the items m_i of each customer.

Five classes are considered, which differ in number of customers, number of items, generated items and vehicle capacity. In the first class of instances, each customer has only one item of width and height equal to 1. However, we observed that packing was not constraining so the instances are reduced to the classical CVRP. For the other classes, each customer has 1 to r items in type r

($1 \leq r \leq 5$). The number of items and items dimensions were randomly generated. For all five classes the loading surface of vehicles was fixed to $W = 20$ and $H = 40$. The largest instance have up to 255 customers, 786 items and a fleet of 51 vehicles. Table 3 summarizes the different benchmark classes.

Table 3. Classes Used for the generation of items

Class	m_i	w_{il}	h_{il}
1	1	1	1
2	[1,2]	$[\frac{W}{10}, \frac{2W}{10}]$, $[\frac{2W}{10}, \frac{5W}{10}]$, $[\frac{4W}{10}, \frac{9W}{10}]$	$[\frac{4H}{10}, \frac{9H}{10}]$, $[\frac{2H}{10}, \frac{5H}{10}]$, $[\frac{H}{10}, \frac{2H}{10}]$
3	[1,3]	$[\frac{W}{10}, \frac{2W}{10}]$, $[\frac{2W}{10}, \frac{4W}{10}]$, $[\frac{3W}{10}, \frac{8W}{10}]$	$[\frac{3H}{10}, \frac{8H}{10}]$, $[\frac{2H}{10}, \frac{4H}{10}]$, $[\frac{H}{10}, \frac{2H}{10}]$
4	[1,4]	$[\frac{W}{10}, \frac{2W}{10}]$, $[\frac{W}{10}, \frac{4W}{10}]$, $[\frac{2W}{10}, \frac{7W}{10}]$	$[\frac{2H}{10}, \frac{7H}{10}]$, $[\frac{H}{10}, \frac{4H}{10}]$, $[\frac{H}{10}, \frac{2H}{10}]$
5	[1,5]	$[\frac{W}{10}, \frac{2W}{10}]$, $[\frac{W}{10}, \frac{3W}{10}]$, $[\frac{W}{10}, \frac{6W}{10}]$	$[\frac{H}{10}, \frac{6H}{10}]$, $[\frac{H}{10}, \frac{3H}{10}]$, $[\frac{H}{10}, \frac{2H}{10}]$

Table 4 presents the comparison of the two approaches. Since the unloading constrains are not considered in our approach, we only compared the total time spent by routing procedures. In each case, we indicate the number of solved instances by both algorithms as well as the average of routing CPU time in seconds for solving those instances.

Table 4. Comparison of two algorithms for deterministic 2L-CVRP

Type	Solved instances	Iori et al.	Our Hybrid heuristic
		T_{Rout}	T_{Rout}
1	10	38.33	9.49
2	11	620.77	7.30
3	11	1804.50	8.17
4	10	436.91	8.35
5	11	134.38	11.57
Average		624.99	8.79

Overall, our algorithm was able to solve instances with up to 255 customers and 786 items while the algorithm of Iori et al. was limited to maximum of 35 customers and 114 items. For the 53 instances solved by both algorithms, our hybrid heuristic took only few seconds compared to hundreds and thousands for the algorithm of Iori et al. This is a very considerable gain, if we take also into account the different performances of the machines used to run the two algorithms.

4.2 Probabilistic Instances

We exploited the 2L-CVRP benchmark described in the previous section to generate our probabilistic instances. Since packing problems do not occur for the

class of type 1, when solving all its instances, so we omit it from tests. For each of the 4 classes, we performed tests on 14 instances that differ in numbers of customers varying in $\{15, 20, 25, 30, 35, 45, 50, 75, 100, 120, 134, 150, 200, 255\}$ (255 the largest number of customers in the benchmark). For the $14 * 4 = 56$ instances, we generated 5 different probabilistic instances, for a total of 280, by varying the probability of presence $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

The computational results are summarized in Table 5. Each instance is represented by a four digits number : the first two digits corresponds to the instance number, while the last two digits identify the instance class. The column denoted "n" represents the number of customers. The table shows T_{clust} representing the CPU time used by the sweep algorithm. T_{rout} gives the CPU time for generating routes to all clusters using the branch and bound algorithm. T_{tot} reports the total CPU time used by the overall heuristic ($T_{tot} = T_{clust} + T_{rout}$). All times are expressed in seconds.

Table 5: Summary of final results

n	Q	p	Class 2			Class 3			Class 4			Class 5						
			Inst	T_{clust}	T_{rout}	T_{tot}	Inst	T_{clust}	T_{rout}	T_{tot}	Inst	T_{clust}	T_{rout}	T_{tot}	Inst	T_{clust}	T_{rout}	T_{tot}
15	90	0.1		0.41	0.31	0.73		0.39	0.41	0.80		0.50	0.34	0.84		0.56	0.43	1.00
		0.3		0.42	0.30	0.72		0.40	0.39	0.80		0.51	0.27	0.78		0.57	0.42	0.99
		0.5	0102	0.42	0.29	0.72	0103	0.40	0.38	0.79	0104	0.51	0.20	0.72	0105	0.57	0.41	0.98
		0.7		0.46	0.24	0.71		0.40	0.38	0.78		0.51	0.20	0.72		0.65	0.29	0.95
		0.9		0.45	0.23	0.68		0.39	0.34	0.73		0.46	0.19	0.65		0.56	0.27	0.83
20	85	0.1		0.59	0.35	0.94		0.46	0.67	1.13		0.51	0.70	1.22		0.54	1.18	1.73
		0.3		0.58	0.28	0.86		0.45	0.60	1.05		0.53	0.67	1.20		0.63	0.90	1.54
		0.5	0302	0.56	0.18	0.75	0303	0.45	0.52	0.97	0304	0.50	0.69	1.19	0305	0.94	0.40	1.35
		0.7		0.41	0.19	0.61		0.46	0.43	0.89		0.46	0.70	1.17		0.81	0.35	1.16
		0.9		0.41	0.06	0.48		0.43	0.49	0.92		0.48	0.635	1.11		0.76	0.32	1.08
25	48	0.1		0.47	0.27	0.75		0.48	0.69	1.18		0.51	0.54	1.05		0.59	0.70	1.30
		0.3		0.45	0.25	0.71		0.46	0.61	1.08		0.54	0.48	1.02		0.60	0.67	1.28
		0.5	0902	0.45	0.25	0.70	0903	0.48	0.51	0.99	0904	0.53	0.45	0.98	0.90	0.95	0.21	1.17
		0.7		0.45	0.24	0.69		0.45	0.44	0.90		0.54	0.43	0.98		0.60	0.54	1.15
		0.9		0.45	0.23	0.68		0.46	0.40	0.87		0.55	0.39	0.95		0.70	0.44	1.14
30	68	0.1		0.61	0.37	0.98		0.40	0.95	1.36		0.59	0.59	1.19		0.68	0.98	1.67
		0.3		0.61	0.34	0.95		0.45	0.80	1.25		0.61	0.53	1.14		0.68	0.81	1.50
		0.5	1202	0.60	0.31	0.91	1203	0.43	0.64	1.08	1204	0.58	0.50	1.09	1205	0.67	0.66	1.34
		0.7		0.58	0.29	0.87		0.54	0.37	0.91		0.64	0.40	1.04		0.88	0.29	1.17
		0.9		0.50	0.26	0.76		0.50	0.34	0.84		0.60	0.41	1.02		0.90	0.26	1.16
35	67	0.1		0.43	0.41	0.84		0.51	0.68	1.19		0.88	0.47	1.36		0.95	0.61	1.57
		0.3		0.30	0.35	0.65		0.54	0.56	1.10		0.67	0.57	1.24		0.92	0.45	1.37
		0.5	1602	0.18	0.28	0.46	1603	0.51	0.50	1.02	1604	0.66	0.46	1.12	1605	0.98	0.31	1.29
		0.7		0.11	0.16	0.28		0.65	0.27	0.93		0.79	0.21	1.01		0.90	0.30	1.21
		0.9		0.10	0.17	0.28		0.50	0.41	0.91		0.71	0.25	0.96		0.78	0.24	1.02

45	60	0.1	1702	0.63	0.51	1.15	1703	0.52	0.77	1.29	1704	0.75	0.99	1.74	1705	0.82	0.40	1.23
		0.3		0.58	0.43	1.01		0.48	0.63	1.12		0.73	0.79	1.52		0.91	0.26	1.17
		0.5		0.53	0.34	0.88		0.57	0.36	0.94		0.65	0.65	1.31		0.85	0.17	1.02
		0.7		0.48	0.26	0.74		0.50	0.26	0.76		0.65	0.44	1.10		0.88	0.09	0.98
		0.9		0.23	0.12	0.36		0.22	0.18	0.40		0.62	0.38	1.00		0.74	0.14	0.88
50	160	0.1	1902	0.58	0.56	1.15	1903	0.62	0.79	1.41	1904	0.96	0.38	1.34	1905	0.96	1.24	2.21
		0.3		0.58	0.55	1.13		0.60	0.78	1.38		0.91	0.40	1.31		1.18	0.89	2.07
		0.5		0.64	0.46	1.11		0.67	0.68	1.35		0.92	0.36	1.29		9.85	0.94	1.92
		0.7		0.65	0.43	1.08		0.70	0.62	1.32		0.89	0.39	1.28		1.01	0.77	1.78
		0.9		0.64	0.35	0.99		0.75	0.54	1.29		0.87	0.38	1.26		1.03	0.71	1.75
75	100	0.1	2402	0.70	0.72	1.43	2403	0.73	0.59	1.33	2404	1.22	0.38	1.61	2405	1.29	0.67	1.96
		0.3		0.73	0.53	1.26		0.75	0.57	1.32		1.12	0.44	1.57		1.20	0.72	1.93
		0.5		0.68	0.42	1.10		0.81	0.49	1.30		1.14	0.38	1.53		1.18	0.69	1.89
		0.7		0.69	0.40	1.10		0.76	0.52	1.29		1.01	0.47	1.49		1.19	0.66	1.86
		0.9		0.54	0.39	0.94		0.75	0.50	1.25		1.28	0.19	1.47		1.28	0.51	1.79
100	112	0.1	2702	0.88	0.61	1.49	2703	0.87	1.34	2.22	2704	1.56	0.40	1.96	2705	1.71	0.69	2.41
		0.3		0.87	0.54	1.41		0.89	1.08	1.97		1.65	0.24	1.90		1.95	0.90	2.85
		0.5		0.78	0.54	1.33		0.92	0.81	1.73		1.53	0.29	1.83		1.79	0.50	2.30
		0.7		0.79	0.49	1.24		0.89	0.60	1.49		1.43	0.32	1.76		1.73	0.51	2.24
		0.9		0.73	0.35	1.08		0.92	0.47	1.39		1.34	0.30	1.64		1.75	0.38	2.13
120	200	0.1	2802	0.89	0.60	1.50	2803	1.00	1.74	2.74	2804	1.80	1.74	3.54	2805	2.19	1.17	3.36
		0.3		0.89	0.55	1.44		1.05	1.25	2.31		1.68	1.64	3.33		2.38	0.70	3.08
		0.5		0.88	0.50	1.39		0.96	0.91	1.88		1.62	1.12	2.74		2.38	0.42	2.80
		0.7		0.95	0.38	1.33		1.01	0.43	1.45		2.13	0.39	2.53		2.39	0.13	2.52
		0.9		0.95	0.36	1.32		1.03	0.31	1.34		1.17	0.55	1.73		1.45	0.52	1.97
134	2210	0.1	2902	1.01	0.50	1.51	2903	0.86	0.81	1.68	2904	1.10	2.38	3.49	2905	1.78	1.55	3.34
		0.3		0.96	0.49	1.46		0.92	1.06	1.98		1.34	1.54	2.88		1.71	1.24	2.96
		0.5		0.90	0.51	1.41		0.90	1.32	2.23		1.34	0.93	2.27		1.89	0.69	2.58
		0.7		0.88	0.53	1.41		0.95	1.51	2.47		1.33	0.33	1.66		1.78	0.41	2.20
		0.9		0.87	0.49	1.36		0.89	1.81	2.71		1.26	0.23	1.50		1.54	0.23	1.78
150	200	0.1	3002	0.93	0.64	1.57	3003	0.92	1.35	2.27	3004	1.28	1.72	3.00	3005	1.94	1.39	3.33
		0.3		0.96	0.55	1.52		0.95	1.01	1.96		1.18	1.64	2.83		2.08	1.07	3.15
		0.5		0.87	0.61	1.48		0.96	0.69	1.66		1.25	1.19	2.44		1.97	1.00	2.97
		0.7		0.87	0.61	1.48		0.96	0.71	1.68		1.21	0.82	2.04		2.00	0.97	2.79
		0.9		0.87	0.61	1.49		0.98	0.39	1.38		1.54	0.45	2.00		2.15	0.63	2.79
200	200	0.1	3302	1.06	1.53	2.59	3303	1.24	1.31	2.35	3304	1.63	1.41	3.05	3305	2.25	2.14	4.39
		0.3		1.06	1.39	2.45		1.26	1.05	2.31		1.45	1.44	2.89		2.06	1.93	4.00
		0.5		1.23	1.06	2.29		1.21	1.06	2.27		1.39	1.34	2.74		1.97	1.63	3.60
		0.7		1.20	0.94	2.14		1.25	0.97	2.23		1.56	1.18	2.74		1.97	1.24	3.21
		0.9		1.18	0.84	2.03		1.56	0.44	2.01		1.96	0.62	2.58		2.48	0.59	3.07

255	1000	0.1	3602	1.28	1.65	2.93	3603	1.62	1.21	2.83	3604	1.69	1.55	3.25	3605	2.68	1.93	4.62
		0.3		1.30	1.49	2.80		1.62	1.03	2.66		1.67	1.46	3.14		2.53	1.73	4.26
		0.5		1.31	1.35	2.67		1.39	1.08	2.48		1.59	1.48	3.07		2.46	1.44	3.91
		0.7		1.26	1.27	2.53		1.26	1.04	2.30		1.48	1.52	3.00		2.61	0.95	3.56
		0.9		1.23	1.25	2.49		1.75	0.54	2.29		1.94	0.92	2.87		2.53	1.19	3.72

By observing Table 5, we may see that the proposed heuristic was able to solve all instances with up to 255 customers within moderate computing time. We observed that the heuristic is sensitive to the type of items. In fact, T_{clust} arise increasingly from class 2 to class 5 and this is explained by the fact that class 5 is characterized by a large number of items compared to the rest of classes, if we consider the same instance. This is a typical feature of 2BPP.

As matter of fact, T_{clust} , including the CPU time for packing customers items, absorbs a very large part of T_{tot} .

When addressing larger instances of 2L-PCVRP, the complexity of the problem increases consistently. This is, however, not surprising because it is well known that the two problems PTSP and 2BPP are NP-hard. We can observe, for the same probability of presence, an increase of T_{tot} going to a factor of 4 (from 1s to 4s).

Table 5 shows that a higher percentage of stochastic customers (low values of probability of presence p) increases the complexity of the problem, as indicated by T_{tot} . For example, when the probability of presence equals 0.1, T_{tot} increases from 166ms to 1.55s through all tested instances.

5 Conclusion and Future Work

This paper has introduced a probabilistic variant of 2L-CVRP, where each customer has a probability of presence. 2L-PCVRP combines the well known probabilistic travelling salesman problem and the two-dimensional bin packing problem.

A hybrid heuristic was presented to deal with the new variant, 2L-PCVRP. The heuristic, is consisted of two phases, where in the first phase, the sweep algorithm was used to generate clusters of customers. Each cluster is solved using a branch and bound algorithm.

The proposed heuristic solved successfully all instances derived for 2L-PCVRP, involving up to 255 customers and 786 items. On the deterministic 2L-CVRP, the heuristic outperformed another state-of-the-art of an exact algorithm.

Future work will consider different variants where, for example, stochastic customers and items are considered.

Acknowledgement

Thanks to Mohamed Abdellahi Amar for his contribution to this article.

References

1. Abdellahi Amar, M., Khaznaji, W., Bellalouna, M.: An exact resolution for the probabilistic traveling salesman problem under the a priori strategy. *Procedia Computer Science* **108**, 1414 – 1423 (2017), international Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland
2. Baker, B., Coffman, E., Rivest, R.: Orthogonal packing in two dimensions. *SIAM Journal Computing* pp. 846–855 (1980)
3. Berhan, E., Beshah, B., Kitaw, D., Abraham, A.: Stochastic vehicle routing problem: A literature survey. *Journal of Information & Knowledge Management* **13**(3) (2014)
4. Bertsimas, D.: The probabilistic vehicle routing problem. Ph.D. thesis, Sloan School of Management, Massachusetts Institute of Technology (1988)
5. Bertsimas, D.: A vehicle routing problem with stochastic demand. *Operations Research* **40**(3), 574–585 (1992)
6. Bertsimas, D., Jaillet, P., Odoni, A.: A priori optimization. *Operations Research* **38**(6), 1019–1033 (1990)
7. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12**(4), 568–581 (1964)
8. Côté, J., Potvin, J., Gendreau, M.: The Vehicle Routing Problem with Stochastic Two-dimensional Items. *CIRRELT (Collection)*, CIRRELT (2013)
9. Dantzig, G., Ramser, J.: The truck dispatching problem. *Management science* **6**(1), 80–91 (1959)
10. Fuellerer, G., Doerner, K., Hartl, R., Iori, M.: Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research* **36**(3), 655–673 (2009)
11. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA (1979)
12. Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem. *Management Science* **40**(10), 1276–1290 (1994)
13. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks* **51**(1), 4–18 (2008)
14. Gillett, E., Miller, R.: A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* **22**(2), 340–349 (1974)
15. Iori, M., Martello, S.: Routing problems with loading constraints. *TOP* **18**(1), 4–27 (2010)
16. Iori, M., Salazar-González, J., Vigo, D.: An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science* **41**(2), 253–264 (2007)
17. Jaillet, P.: Probabilistic traveling Salesman Problem. Ph.D. thesis, MIT, Operations Research center (1985)
18. Jaillet, P.: A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research* **36**(6), 929–936 (1988)
19. Little, J.D.C., Murty, K.G., Sweeney, D.W., Karel, C.: An algorithm for the traveling salesman problem. *Operations research* **11**(6), 972–989 (1963)
20. Lodi, A., Martello, S., Monaci, M.: Two-dimensional packing problems: A survey. *European Journal of Operational Research* **141**(2), 241–252 (2002)
21. Lodi, A., Martello, S., Monaci, M., Vigo, D.: *TwoDimensional Bin Packing Problems*, pp. 107–129. Wiley-Blackwell (2014)

22. Oyola, J., Arntzen, H., Woodruff, D.: The stochastic vehicle routing problem, a literature review, part i: models. *EURO Journal on Transportation and Logistics* (2016)
23. Oyola, J., Arntzen, H., Woodruff, D.: The stochastic vehicle routing problem, a literature review, part ii: solution methods. *EURO Journal on Transportation and Logistics* **6**(4) (2017)
24. Rosenow, S.: Comparison of an Exact Branch-and-Bound and an Approximative Evolutionary Algorithm for the Probabilistic Traveling Salesman Problem, pp. 168–174. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
25. Toth, P., Vigo, D.: The vehicle routing problem. chap. An Overview of Vehicle Routing Problems, pp. 1–26 (2001)