Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 448–453

A Column Generation Based Heuristic for the Capacitated Vehicle Routing Problem with Three-dimensional Loading Constraints

Batoul Mahvash*, Anjali Awasthi**
Satyaveer Chauhan***

**Mechanical and Industrial Engineering Department, Concordia University Montreal,Canada (e-mail: b_mahvas@encs.concordia.ca).*
*** Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada (e-mail: anjali.awasthi@concordia.ca)*
**** John Molson School of Business, Concordia University, Montreal,Canada, (e-mail: Satyaveer.Chauhan@concordia.ca)}*

Abstract: This paper addresses an integrated problem of routing and loading known as the three-dimensional loading capacitated vehicle routing problem (3L-CVRP). 3L-CVRP consists of finding feasible routes with minimum total travel cost while satisfying customers' demands expressed in terms of cuboid and weighted items. Practical constraints related to connectivity, stability, fragility, and LIFO are considered as parts of the problem. 3L-CVRP is addressed by using a column generation (CG) technique based heuristic. To generate new columns, an integrated approach using the shortest path problem and 3D loading problem is applied. To speed up the CG technique, fast CG is also carried out by applying a heuristic pricing method. The CG technique outperforms the efficient tabu search technique proposed in the literature in terms of solution quality and execution time.

Keywords: three-dimensional loading problem, routing problem, column generation

1. 1. INTRODUCTION

Most vehicle routing problems consider only those capacity constraints which require that the total weight of products transported by a vehicle should not exceed its capacity (Cordeau et al., 2002). While in real world distribution systems, the shape of the products is also a key factor. Therefore, vehicle routing problems must take into account loading constraints that identify feasible placements of products within vehicles based on their shapes and properties.

In the literature, vehicle routing problems with loading constraints are classified into two types, two-dimensional and three-dimensional loading capacitated vehicle routing problems (2L-CVRP and 3L-CVRP respectively). In 2L-CVRP, the customer's demands are expressed in terms of distinct rectangular items. Items cannot be stacked on top of each other due to their fragility, weight, or large dimensions. One of the applications of this problem is the distribution of kitchen appliances such as refrigerators.

Iori et al. (2007) were the first to propose an exact approach to solve 2L-CVRP. In their approach, routing costs are minimized by a branch-and-cut algorithm, and loading aspects are iteratively checked by a branch-and-bound algorithm. The authors tested their approach on benchmark instances derived from the classical CVRP test problems involving up to 35 customers and more than 100 items. Gendreau et al. (2006) solved larger 2L-CVRP instances with up to 255 customers and 786 items by employing a tabu search (TS) method, where customers are relocated through a generalized insertion procedure (GENI) (Gendreau et al., 1992). Authors also applied a two-dimensional strip packing problem in order to guarantee the optimal placement of the load components. Interested reader may refer to Fuellerer et al. (2009), Zachariadis et al. (2009), Duhamel et al. (2011) for other methods for 2L-CVRP.

In 3L-CVRP, the demands of customers are expressed in terms of cuboid and weighted items. The aim of the problem is to find feasible routes with the minimum total traveling cost while satisfying customers' demands and practical loading constraints. 3L-CVRP, as one of the rich routing problems, draws a great deal of attention in supply chain management (Schmid et al., 2013). In particular, this problem has significance for applications that deal with many large items and in which the loading requirement is not trivial. Some examples include distribution of household appliances and mechanical components.

There is no exact algorithm for 3L-CVRP in the literature. Most researchers simply extend metaheuristic algorithms from 2L-CVRP to apply to 3L-CVRP. As mentioned by Vidal et al. (2013), most existing techniques for 3L-CVRP are based on TS combined with efficient packing heuristics. Gendreau et al. (2006) were the first to employ TS to tackle routing aspects of the problem and applied two packing heuristics to handle loading constraints. They evaluated their algorithm based on vehicle routing instances adapted from the literature as well as on new real-world instances. Tarantilis et al. (2009) described a hybrid metaheuristic methodology called GTS that combines the approaches of TS

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.122

and guided local search (GLS). They showed that GLS improves the solution attained by TS within a variable neighborhood search. The loading characteristics in this method were determined by employing a collection of packing heuristics. GTS improves average solution of TS of Gendreau et al. (2006) by 3.54%.

Fuellerer et al. (2010) solved 3L-CVRP by means of a highly efficient ant colony optimization (ACO) algorithm, adapted of the savings-based ants (Reimann et al., 2004). They handled routing aspects by an ant-based procedure. To deal with loading components, the ACO employs and iteratively invokes the local search and packing heuristics used in Gendreau et al. (2006). The ACO outperforms TS of Gendreau et al. (2006) in 26 out of 27 instances and GTS of Tarantilis et al. (2009) in 23 out of 27 instances. We refer the reader to (Wang et al., 2009) and (Iori and Martello, 2010) for a survey of 3L-CVRP.

In this paper we propose a column generation (CG) technique based heuristic to solve strong NP-hard 3L-CVRP. First 3L-CVRP is formulated as a set-partitioning model based on Dantzig-Wolfe decomposition. This model is then split into a master problem that is a linear relaxation of model and a sub-problem. Using CG technique, we in fact obtain a solution to master problem. If such solution is integer, then it would be a solution to the set-partitioning formulation. Sub-problem is used to discover new columns.

The valid columns are generated by applying two different methods. In the first method, an elementary shortest path problem is solved to obtain routes with negative reduced cost. Then an extreme point-based efficient heuristic is employed in order to verify the feasibility of obtained routes in terms of loading constraints. In the second method, an efficient heuristic pricing is applied to attain feasible routes in terms of loading constraints with negative reduced cost. If there is no improvement in the master problem solution then the column generation stops. After terminating CG technique, a branching rule in a heuristic framework is applied in case the solution value of the master problem is non-integer.

The main contribution of this paper is that using a CG based heuristic technique; we produce good results on benchmark instances. CG technique outperforms the tabu search(TS) from Gendreau et al. (2006) and Guided Tabu Search (GTS) from Tarantilis et al. (2009) in terms of solution quality and computation time.

The remainder of this paper is organized as follows: A description of the 3L-CVRP and corresponding set-partitioning formulation are presented in next section. In Section 3, CG technique including pricing problem is detailed. Computational results are presented and analysed in Section 4. Finally, conclusion and suggestions for future work are given in Section 5.

2. PROBLEM DISCRIPTION

Let us consider a complete graph G = (V, E) where V = {0, *1,…, n*} is the vertex set and $E = \{(i,j): i,j \in V, i \neq j\}$ is the set of edges. Vertex 0 corresponds to the central depot and the other vertices correspond to the customers. A cost c_{ij} is associated with each edge *(i, j)* that represents traveling cost from vertex *i* to vertex *j*. Assume that a fleet of *m* identical vehicles is available in the central depot. Each vehicle has a weight capacity *D* and a three-dimensional loading space of length *L*, width *W*, and height *H*. The demand of each customer *i (i = 1,…, n)* is expressed in terms of a set of cuboid items CI_i with a total volume s_i and a total weight capacity d_i. It is assumed that $s_i \leq L.W.H$ and $d_i \leq D$. Each item $I_{ik} \in CI_i$ *(k = 1, 2 . . . m)* is characterized by the length l_{ik}, width w_{ik}, height h_{ik}, and fragility status f_{ik} ($f_{ik} = 1$ for fragile items and 0 for non-fragile ones).

The aim of the 3L-CVRP is to identify a set of vehicle routes with the minimum total traveling cost while satisfying the following constraints:

- The number of vehicle routes selected in the solution must be less than or equal to the number of vehicles available in the depot;
- Each vehicle route must start and end at the depot;
- Each customer must be served by exactly one vehicle and visited only once;
- The total weight capacity and volume of the items placed in each vehicle must not exceed the weight capacity and volume of the vehicle; and
- All items must be orthogonally packed into the vehicles without overlapping, while satisfying rotation, stability, fragility, and last in first out (LIFO) constraints defined below.

Rotation constraints demand that items should be loaded with a fixed vertical orientation, that is, they can be rotated only by 90 degrees on the horizontal plane. In the fragility constraints, only fragile items can be stacked on other fragile items; any items can be stacked on the non-fragile items. To meet stability constraints, each item that is not packed directly on the vehicle floor should be stable in the vehicle and supported by a sufficient surface comprising other items. The supporting surface of an item should be at least 75% of the base area of the item. LIFO constraints require that all items packed into a vehicle should be directly unloaded through a sequence of straight movements parallel to the length of vehicle without repositioning other items. When a customer is visited, its items should not be blocked by or stacked under items that will be delivered to customers later on the route.

2.1 Set-partitioning Formulation

We express 3L-CVRP as set-partitioning formulation (Dantzig-Wolfe decomposition) in order to use CG technique. The key idea of set-partitioning formulation for the 3L-CVRP is to enumerate all feasible routes of the problem. A feasible route is defined as a vehicle trip that starts and ends at the depot, while visiting a subset of customers and satisfying capacity and loading constraints.

Let $\mathcal{R}$ be the set of all possible feasible routes; $c_r = \sum_{(i,j)\in r} c_{ij}$, the cost of feasible route $r \in \mathcal{R}$ that is sum of traveling costs along route r; a_{ir}, the binary parameter equal to 1 if customer i is along feasible route r and 0 otherwise; and x_r, the binary variable equal to 1 if feasible route r is selected in the solution and 0 otherwise.

The set-partitioning formulation of 3L-CVRP is:

$$\text{Min} \quad \sum_{r\in\mathcal{R}} c_r x_r \tag{1}$$

s.t

$$\sum_{r\in\mathcal{R}} a_{ir} x_r = 1 \qquad \forall\, i \in V\backslash\{0\} \tag{2}$$

$$\sum_{r\in\mathcal{R}} x_r \leq m \tag{3}$$

$$x_r \in \{0,1\} \qquad \forall\, r \in \mathcal{R} \tag{4}$$

A set of feasible routes with the minimum total traveling cost is determined in the objective function (1). Constraints (2) ensure that each customer is covered by exactly one of the feasible routes selected in the solution. Constraint (3) ensures that the number of vehicle routes selected in the solution does not exceed the number of vehicles available in the depot.

3. CG TECHNIQUE

CG technique has been advocated by several researchers as a very powerful technique for solving a variety of operation research problems to optimality. CG is capable of solving linear programming (LP) models with a large number of variables. The technique was introduced by Ford and Fulkerson [15] to address a multi-commodity network flow problem. Then, Dantzig and Wolfe [16] adapted it to LP problems with a decomposable structure. Furthermore, Gilmore and Gomory [17] demonstrated the efficiency of the CG technique as applied to a cutting stock problem. The CG technique has also found applications in relation to the bin packing problem, the generalized assignment problem, the vehicle routing problem, the crew scheduling problem, the coloring, p-median problem, and other integer-constrained problems.

Using CG technique, we obtain a solution to a LP relaxation of the set-partitioning formulation that is called master problem (MP). If such solution is integer, then it would be a solution to the set-partitioning formulation. Note that, the integrality constraints on x variables have been removed in the MP. To solve MP, CG technique starts with a small part of MP, specifically, a partial set of the variables in the model. The corresponding LP relaxation model to such partial set is called restricted master problem (RMP). Based on solution to RMP and analyzing the resultant partial solutions, more variables can be discovered and added to the RMP. Then, the expanded model is resolved.

Let π_i , $i \epsilon V\backslash\{0\}$ be the set of dual variables associated with the constraints (2) and μ dual variable associated with constraint (3). $\bar{c}_r = c_r - \mu - \sum_{i\in V\backslash\{0\}} a_{ir}\pi_i$ is defined as reduced cost associated with route $r \in \mathcal{R}$, that can be reformulated as follows: $\bar{c}_r = \sum_{(i,j)\in A, j\neq 0}\left(c_{ij} - \pi_j\right) + c_{k0} - \mu$. Here, A is the set of arcs associated with route $r \in \mathcal{R}$, and k is last customer node visited along route r.

In the CG technique, sub-problem is a new problem used to discover new variables. According to the current dual values, the reduced costs of new variables are considered entries of the objective function in the sub-problem. If variables with negative reduced cost are available, they are identified by the sub-problem. Such variables, expressed as new columns, are added to the RMP. This process is repeated until no father variables with a negative reduced cost are discovered. A basic scheme of CG technique is given in Algorithm 1.

Algorithm 1: Column Generation Scheme

1. Generate a partial set of vehicle routes (columns) by assigning one customer to one vehicle
2. Solve corresponding RMP and obtain dual values
3. Solve pricing problem to identify feasible vehicle routes with negative reduced cost due to obtained dual values
4. Add these vehicles routes to RMP as new columns and go to step 2
5. If there is no feasible route with negative reduced cost or no improvement in the RMP solution, then stop, the current solution to RMP is optimal to MP

As mentioned in Algorithm 1, CG technique is initialized by assigning one customer to just one vehicle. Since constraint (3) is not satisfied with such initial columns, a single dummy variable with coefficient 0 is added to constraint (3) with a large penalty in the objective function. This variable is going to be set to zero in the final integer solution.

3.1 Pricing Problem

The pricing problem (Sub-problem) consists of finding a feasible route r of a single vehicle with the minimum negative reduced cost $\bar{c}_r$ while satisfying all loading constraints. In fact, the kind of pricing problem arising in this work is an elementary shortest path problem with resource constraints (ESPPRC) such as capacity, volume and loading conditions.

With relaxing loading constraints, we first obtain routes with negative reduced cost by means of ESPPRC with just volume and capacity resources. The ESPPRC is solved by invoking a label correcting algorithm with dominance rules introduced in (Feillet et al., 2004). The loading aspects of obtained routes are then verified by the extreme point-based heuristic method explained in (Crainic et al., 2008). Such heuristic method is examined for a combination of three items sorting rules and different position selecting criteria. For all three sorting rules (called *Rule-1, Rule-2, Rule-3*), customers along the route are sorted by inverse order of visit. For each customer non-fragile items precede fragile ones. Each of the subsets of fragile and non-fragile items is sorted by following criteria for each sorting rule:

- *Rule-1*: Decreasing volume, breaking ties by decreasing height;
- *Rule-2*: Decreasing base area, breaking ties by decreasing height; and
- *Rule-3*: Decreasing height, breaking ties by decreasing base area.

The extreme points are scanned according to the six following position selection criteria:

- *Criterion-1*: Select the EP with the minimum Y coordinate value, breaking ties with the X coordinate value and then minimum Z-coordinate value;
- *Criterion-2*: Select the EP with the minimum X coordinate value, breaking ties with the minimum Y coordinate value and then minimum Z coordinate value;
- *Criterion-3*: Select the EP with the minimum Y coordinate value, breaking ties with the minimum Z coordinate value and then minimum X coordinate value;
- *Criterion-4*: Select the EP at which the item to be packed, touches the vehicle and other packed items more;
- *Criterion-5*: Select the EP with the minimum residual space along X and Y axes (Crainic et al., 2008); and
- Criterion-6: Select the EP with the minimum residual space (Crainic et al., 2008)

If the EP-based algorithm cannot give a feasible loading pattern for a route with a negative cost, then, a saving procedure is used. The saving procedure eliminates some customers from the route, which represent a smaller effect on the reduced cost in order to obtain a feasible loading pattern.

Let $(s, \dots, i, k, j, \dots e)$ be an elementary path with the negative reduced cost given by the label correcting algorithm. The saving procedure is based on notation of saving that is:

$$s_k = \bar{c}_{ij} - \bar{c}_{ik} - \bar{c}_{kj}.$$

According to the value of s_k, the least profitable customer along route $(s, \dots, i, k, j, \dots e)$ is selected and removed from the route. The loading feasibility of such route is then checked. If a valid loading pattern is not derived, then the next least profitable customer is removed from the path. The saving procedure stops when a valid loading pattern is achieved or the reduced cost of the path is positive.

3.2 *Heuristic Pricing Approach*

We have developed a greedy heuristic pricing approach to hasten the generation of valid columns with a negative reduced cost (see Algorithm 2). Algorithm 2 gives promising feasible routes with negative, but not necessarily minimal, reduced cost. For each $i \in V$, L_i is a list of $j \in V, j \neq i$, sorted by increasing order of $\bar{c}_{ij}$. The algorithm starts from L_d, which is a list of customers $j \in V \backslash d$ sorted by increasing order of $\bar{c}_{dj}$. Let k be the first entry in L_d. Items associated with customer k are inserted into a vehicle by the EP-based algorithm. Then, the first entry in list L_k is selected. If this entry is a customer whose items cannot be packed into the vehicle according to the loading constraints, then the second entry in L_k is selected. This process is repeated until the selected entry is depot. A feasible loading pattern with a negative reduced cost (called fLoading) is added to the set all-Loadings. For a feasible loading pattern with a positive reduced cost, the checkR procedure removes one or more customers with the lowest value of π from fLoading in order to obtain a feasible loading pattern with a negative reduced cost. We apply Algorithm 2 for different position criteria, as explained in the previous section.

Algorithm 2: Heuristic Pricing Method

Input *V*:the set of customers and depot; (π, μ): the dual values
Output *all-Loadings*: all feasible loading patterns with the negative reduced costs

```
1   Obtain L_i for each i ∈ V
2   for each customer k ∈ L_d do
3     Pack items of customer k by EP-based algorithm
4     Add customer k to fLoading
5     Set CL=L_k
6     for each j ∈ CL do
7         if j is depot then
8             Go to step 15
9         end if
10        if items of customer j can be packed then
11            Add customer j to fLoading
12            Set CL= L_j and go to step 6
13        end if
14    end for
15    if reduced cost of fLoading is negative then
16        Inverse the order of customers in fLoading
17        Add fLoading to all-Loadings
18    end if
19    if reduced cost of fLoading is positive then
20        checkR(fLoading)
21    end if
22  end for
23  return all-Loadings
```

4. DEPTH_FIRST HEURISTIC (D_FH) BRANCHING RULE

After terminating the CG technique, if the current solution to the RMP is an integer solution, then it is a solution to the set-partitioning model. Otherwise, an integer solution is determined through a branching method embedded in the CG technique. In this branching rule, variables with fractional values that are greater than a given specific value are set to one. If there are no variables with appropriate fractional values, then the variable with the largest fractional value is fixed to one. After fixing the variables, more columns are generated using pricing problem until there is little or no change in the RMP solution. Repeating this branching strategy leads either to an integer solution or an infeasible one.

5. COMPUTATIONAL RESULTS

The heuristic CG algorithm is coded in C++ and run on the Intel 2.20 GHz. The algorithm is tested on a set of 27 instances introduced Gendreau et al. (2006). The graph, the weight capacity demanded by the customers, and the vehicle weight capacity are derived from 27 Euclidean CVRP instances (see (Toth and Vigo, 2002)). The length L, width W, and height H of the vehicles (loading spaces) are set to 60, 25, and 30, respectively. For each customer i, the number of requested items (m_i) is stochastically generated within the interval [1, 3]. For each item I_{ik},($k = 1, \dots, m_i$), dimensions l_{ik}, w_{ik}, and h_{ik} are integer values within intervals [0.2L, 0.6L], [0.2W, 0.6W], and [0.2H, 0.6H], respectively.

The pricing problem in the CG technique is solved using two approaches, as follows: 1) the heuristic pricing method (HP), and 2) by solving an integrated problem of ESPPRC and the loading problem denoted as ESPPRC-L. The results of the CG technique using HP and ESPPRC-L are summarized in Table 1. The instance name, total number of customers (n), total number of items(m), and number of vehicles (v)are presented in the first four columns. For each pricing method, the total routing cost (z) resulted from CG and the execution times (sec) in seconds are reported. The results show that the performance of the CG technique with HP is better than with ESPPRC-L in terms of solution quality and execution time for all instances except E016-05m, E023-03g, and E026-08m.

It should be noted that ESPPRC stops when 200 labels with a negative reduced cost are extended to the depot node at each iteration of the CG technique. If no further columns with negative reduced costs can be found by pricing method or there are no improvements in the RMP solution, then the CG stops. Since the solutions to the RMP for all instances are non-integers, we applied branching rule D_FH to obtain an integer solution. D_FH with the limit value 0.9 leads to satisfactory integer solutions for most instances, but infeasible solutions for some others. For the instances with the infeasible solutions, all fixed variables in the branching are released and the RMP is solved to integrality by means of CPLEX.

Table 1: Results for the CG technique with HP and ESPPRC-L

Instances				HP		ESPPRC-L	
	n	m	v	z	sec_z	z	sec_z
E016- 03m	15	32	5	315.16	19.73	315.16	415.85
E016-05m	15	26	5	345.28	5.69	341.93	10.63
E021-06m	20	37	5	391.55	36.48	393.47	672.38
E021-06m	20	36	6	430.78	12.44	445.11	201.83
E022-04g	21	45	7	447.56	38.17	447.56	856.40
E022-06m	21	40	6	498.97	18.65	500.93	181.19
E023-03g	22	46	6	793.40	47.50	789.8	585.09
E023-05s	22	43	8	807.01	42.79	848.52	605.51
E026-08m	25	50	8	653.73	35.24	647.33	525.89
E030-03g	29	62	10	883.57	128.18	883.57	930.95
E030-04s	29	58	9	820.19	156.18	823.92	545.57
E031-09h	30	63	9	614.42	70.07	614.42	819.63
E033-03n	32	61	9	2735.18	185.79	2789.19	936.86
E033-04g	32	72	11	1504.25	295.60	1504.25	752.75
E033-05s	32	68	10	1412.89	330.06	1412.89	749.33
E036-11h	35	63	11	698.42	98.12	698.42	527.54
E041-14h	40	79	14	866.21	84.05	866.21	331.21
E045-04f	44	94	14	1275.85	606.64	1344.08	1980.04
E051-05e	50	99	13	799.37	1280.08	846.96	1956.77
E072-04f	71	147	20	632.43	1464.27	642.21	1723.30
E076-07s	75	155	18	1145.11	1700.46	1295.93	3244.25
E076-08s	75	146	19	1237.53	1180.41	1299.86	2758.94
E076-10e	75	150	18	1246.86	1353.24	1246.86	1101.85
E076-14s	75	143	18	1244.84	1089.99	1264.66	3814.46
E101-08e	100	193	24	1456.95	2435.53	1585.58	2299.33
E101-10c	100	199	28	1711.10	2815.44	1870.89	2076.56
E101-14s	100	198	25	1642.56	3553.39	1700.62	3318.475
Average				985.598	706.825	1015.567	1256.3938

The performance of the CG technique using HP is compared with the TS algorithm from Gendreau et al. (2006) and GTS from Tarantilis et al. (2009) in Table 2. We have re-implemented their approaches. The differences between total routing costs of CG and TS as $100(z_{CG}\text{-}z_{TS})/z_{TS}$ and for CG and GTS as $100(z_{GS}\text{-}z_{GTS})/z_{GTS}$ are represented in the *Gap* column. The overall results indicate that the CG technique outperforms TS and GTS in terms of quality and execution time. CG improves the average solution from TS by 5.04% and GTS by 1.5%. The quality of the solutions obtained using the CG technique is better than that of TS for all instances. The best CG technique solutions, in comparison with both the TS and GTS methods, are highlighted in bold in Table 2.

Table 2: Results for the CG, TS and GTS

CG		*TS*		*GTS*		*Gap %*	
z_{CG}	Sec_z	z_{TS}	Sec_z	z_{GTS}	Sec_z	*CG-TS*	*CG-GTS*
315.16	19.73	316.32	129.5	321.47	7.8	-0.37	-1.96
345.28	5.69	350.58	5.3	334.96	7.2	-1.51	3.08
391.55	36.48	447.73	461.1	430.95	352.6	-12.55	-9.14
430.78	12.44	448.48	181.1	458.04	204.0	-3.95	-5.95
447.56	38.17	464.24	75.8	465.04	61.3	-3.59	-3.76
498.97	18.65	504.46	1167.9	507.96	768.8	-1.09	-1.77
793.40	47.50	831.66	181.1	796.61	241.5	-4.60	-0.40
807.01	42.79	871.77	156.1	880.93	140.0	-7.43	-8.39
653.73	35.24	666.10	1468.5	642.22	604.7	-1.86	1.79
883.57	128.18	911.16	714.0	884.74	803.1	-3.03	-0.13
820.19	156.18	819.36	396.4	873.43	308.5	0.10	-6.10
614.42	70.07	651.58	268.1	624.24	180.8	-5.70	-1.57
2735.18	185.79	2928.34	1639.1	2799.74	1309.5	-6.60	-2.31
1504.25	295.60	1559.64	3451.6	1504.44	2678.1	-3.55	-0.01
1412.89	330.06	1452.34	2327.4	1415.42	1466.3	-2.72	-0.18
698.42	98.12	707.85	2550.3	698.61	2803.2	-1.33	-0.03
866.21	84.05	920.87	2142.5	872.79	1208.6	-5.94	-.75
1275.85	606.64	1400.52	1452.9	1296.59	1300.9	-8.90	-1.60
799.37	1280.08	871.29	1822.3	818.68	1438.4	-8.25	-2.36
632.43	1464.27	732.12	790.0	641.57	1284.8	-13.62	-1.42
1145.11	1700.46	1275.20	2370.3	1159.72	1704.8	-10.20	-1.26
1237.53	1180.41	1277.94	1611.3	1245.35	1663.5	-3.16	-0.63
1246.86	1353.24	1258.16	6725.6	1231.92	3048.2	-0.90	1.21
1244.84	1089.99	1307.09	6619.3	1201.96	2876.8	-4.76	3.57
1456.95	2435.53	1570.72	5630.9	1457.46	3432.0	-7.24	-0.03
1711.10	2815.44	1847.95	4123.7	1711.93	3974.8	-7.41	-0.05
1642.56	3553.39	1747.52	7127.2	1646.44	5864.2	-6.01	-0.24
Average 985.598	706.82	1042.26	2058.9	997.156	1471.0	-5.04	-1.50

5. CONCLUSION AND FUTUR WORK

This paper presents a column generation technique based heuristic for 3L-CVRP. 3L-CVRP consists of finding feasible routes with minimum total travel cost while satisfying customers' demands expressed in terms of cuboid and weighted items. To generate new columns, the pricing problem that is part of CG is solved by using two approaches: 1-by means of shortest path problem with resource constraints and 3D loading problem, and 2-a heuristic pricing method (HP). CG using HP with a simple scheme can attain solutions competitive with the efficient TS algorithm described in the literature.

The performance of the CG technique strongly depends on three elements: 1) the pricing problem, 2) the heuristic method of checking feasibility in terms of loading constraints, and 3) branching rules. It is suggested that future work should include the modification and development of efficient methods to deal with each of these elements. CG technique, as a flexible method, can be tested on the 3L-CVRP with more practical constraints such as fleet of heterogeneous vehicles, time windows and pickup and delivery.

REFRENCES

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., Semet, F., 2002. A Guide to Vehicle Routing Heuristics. *J. Oper. Res. Soc*. 53, 512–522.

Crainic, T.G., Perboli, G., Tadei, R., 2008. Extreme point-based heuristics for three-dimensional bin packing. *Inf. J. Comput*. 20, 368–384.

Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs. Oper. Res. 8, 101–111.

Duhamel, C., Lacomme, P., Quilliot, A., Toussaint, H., 2011. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Comput. Oper. Res.* 38, 617–640.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C., 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*. 44, 216–229.

Ford Jr, L.R., Fulkerson, D.R., 1958. A suggested computation for maximal multi-commodity network flows. *Manag. Sci.* 5, 97–101.

Fuellerer, G., Doerner, K.F., Hartl, R.F., Iori, M., 2009. Ant colony optimization for the two-dimensional loading vehicle routing problem. *Comput. Oper. Res.* 36, 655–673.

Fuellerer, G., Doerner, K.F., Hartl, R.F., Iori, M., 2010. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *Eur. J. Oper. Res.* 201, 751–759.

Gendreau, M., Hertz, A., Laporte, G., 1992. New insertion and postoptimization procedures for the traveling salesman problem. *Oper. Res.* 40, 1086–1094.

Gendreau, M., Iori, M., Laporte, G., Martello, S., 2006. A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transp. Sci.* 40, 342–350.

Gilmore, P.C., Gomory, R.E., 1961. A linear programming approach to the cutting-stock problem. *Oper. Res.* 9, 849–859.

Iori, M., Martello, S., 2010. Routing problems with loading constraints. *TOP* 18, 4–27.

Iori, M., Salazar-González, J.-J., Vigo, D., 2007. An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transp. Sci.* 41, 253–264.

Lodi, A., Martello, S., Vigo, D., 1999. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *Inf. J. Comput.* 11, 345–357.

Reimann, M., Doerner, K., Hartl, R.F., 2004. D-ants: Savings based ants divide and conquer the vehicle routing problem. *Comput. Oper. Res.* 31, 563–591.

Schmid, V., Doerner, K.F., Laporte, G., 2013. Rich routing problems arising in supply chain management. *Eur. J. Oper. Res*. 224, 435–448.

Tarantilis, C.D., Zachariadis, E.E., Kiranoudis, C.T., 2009. A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *Intell. Transp. Syst. IEEE Trans. On* 10, 255–271.

Toth, P., Vigo, D., 2002. *The vehicle routing problem.* Society for Industrial and Applied Mathematics, Philadelphia.

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

Wang, F., Tao, Y., Shi, N., 2009. A survey on vehicle routing problem with loading constraints, in: Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on. IEEE, pp. 602–606.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2009. A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *Eur. J. Oper. Res.* 195, 729–743.