International Conference
on Industrial Engineering and Systems Management

IESM'2011

May 25 - May 27
METZ - FRANCE

Heuristic column generation for the truck and trailer routing problem [⋆]

Juan G. VILLEGAS [a] Christian PRINS [b] Caroline PRODHON [b]
Andrés L. MEDAGLIA [c] Nubia VELASCO [c]

[a] *Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

[b] *Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[c] *Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

Abstract

We present a heuristic column generation for the truck and trailer routing problem (TTRP) in which the routes of the local optima of a GRASP/VNS are used as columns of a set-partitioning formulation of the TTRP. This approach outperforms the previous state-of-the-art methods and improves the best-known solutions for several test instances from the literature.

Key words: Truck and trailer routing problem (TTRP), matheuristics, set-partitioning problem, greedy randomized adaptive search procedures (GRASP), variable neighborhood search (VNS).

1 Introduction

In the last years a new generation of hybrid optimization methods known as matheuristics has emerged. Matheuristics combine elements of exact mathematical programming algorithms and metaheuristics in a cooperative way [3,10,12]. Recently, Villegas et al. [17,18] presented a very effective GRASP/VNS with path relinking for the solution of the truck and trailer routing problem (TTRP). In their experiments a GRASP/VNS that uses path relinking as post-optimizer offered a good trade-off between solution quality and running time. This shows that GRASP/VNS is able to generate diverse high-quality solutions for the TTRP that can be used as input of a post-optimization phase. Therefore, in this paper we solve the TTRP using a hybrid method that combines a GRASP/VNS metaheuristic and a set-partitioning formulation of the problem.

The remainder of this paper is organized as follows. Section 2 describes the TTRP and gives a brief literature review of the methods to solve it. Section 3 presents a set-partitioning formulation of the TTRP. Section 4 describes the elements of the proposed matheuristic. Computational results are presented in Section 5 followed by the conclusions in Section 6.

[⋆] This paper was not presented at any other revue. Corresponding author J. G. Villegas. Tel. +574-2195575.
Email addresses: `jvillega@udea.edu.co` (Juan G. VILLEGAS), `christian.prins@utt.fr` (Christian PRINS), `caroline.prodhon@utt.fr` (Caroline PRODHON), `amedagli@uniandes.edu.co` (Andrés L. MEDAGLIA), `nvelasco@uniandes.edu.co` (Nubia VELASCO).

2 Problem definition and literature review

In the truck and trailer routing problem a heterogeneous fleet composed of m_t trucks and m_r trailers ($m_r < m_t$) serves a set of customers $N = \{1 \ldots, n\}$ from a main depot, denoted as node 0. Each customer $i \in N$ has a non-negative demand $q_i > 0$; the capacities of the trucks and the trailers are Q_t and Q_r, respectively; and the distance c_{ij} between any two points $i, j \in N \cup \{0\}$ is known. Some customers with limited maneuvering space or accessible through narrow roads must be served only by a truck, while other customers can be served either by a truck or by a complete vehicle (i.e., a truck pulling a trailer). These incompatibility constraints create a partition of N into two subsets: the subset of truck customers N_t accessible only by truck; and the subset of vehicle customers N_v accessible either by truck or by a complete vehicle. A distinguishing feature of the TTRP is that vehicle-customer locations can be used to park the trailer before serving truck customers. This possibility gives rise to complex routes with a main tour and one or more subtours.

The objective of the TTRP is to find a set of routes of minimum total distance such that: each customer is visited in a route performed by a compatible vehicle; the total demand of the customers visited in a route does not exceed the capacity of the allocated vehicle; and the number of required trucks and trailers is not greater than m_t and m_r, respectively.

Most of the methods to solve the TTRP follow the cluster-first, route-second approach. Chao [7] proposed a tabu search that solves a relaxed generalized assignment problem (RGAP) to allocate customers to routes in the clustering phase and uses an insertion heuristics to sequence the customers within the routes. Then, a multi-neighborhood search embedded within a hybrid tabu search/deterministic annealing method improves the initial solution. Likewise, Scheuerer [14] uses two constructive heuristics called T-Sweep and T-Cluster, that follow the cluster-first route-second principle, to find an initial solution that is improved with a tabu search. More recently, Caramia and Guerriero [6] have presented a mathematical programming-based heuristic that solves two subproblems sequentially. First, the customer-route assignment problem (CAP) assigns customers to valid routes. Then, given the assignment of customers to routes, the route-definition problem (RDP) minimizes the total length of each route.

In contrast, few researchers have tackled the TTRP with route-first, cluster-second methods. Lin et al. [11] developed a simple and effective simulated annealing that works on a permutation of the customers with additional dummy zeros to separate routes, along with a vector of binary variables of length $|N_v|$ representing the type of vehicle used to serve each customer in N_v. Whereas, Villegas et al. [17,18] solved the TTRP using a route-first, cluster-second procedure embedded within a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), variable neighborhood search (VNS), and path relinking. Likewise, Villegas et al. [16] solved the single truck and trailer routing problem with satellite depots (STTRPSD) with a multi-start evolutionary local search and a hybrid GRASP/VND. These metaheuristics use a route-first cluster-second procedure and a VND as building blocks. For an updated review of the TTRP and other related problems the reader is referred to [11,18].

3 A set-partitioning formulation of the TTRP

Figure 1 illustrates the three types of routes in a TTRP solution. Let Ω be the set of feasible routes. Let $\Gamma \subseteq \Omega$ be the set of *pure truck routes* performed by trucks visiting customers in N_t and N_v; let $\Psi \subseteq \Omega$ be the set of *pure vehicle routes* performed by complete vehicles serving only customers in N_v; and finally let $\Pi \subseteq \Omega$ be the set of *vehicle routes with subtours*. The routes in the latter set Π involve a main tour performed by a complete vehicle visiting customers in N_v, and one or more subtours, in which the trailer is detached at a vehicle customer location followed by visits (only with the truck) to one or more customers in N_t and probably some customers in N_v. Let us define the binary parameter a_{ij} that takes the value of 1 if customer $i \in N$ is visited in pure truck route $j \in \Gamma$; it takes the value of 0, otherwise. Likewise, b_{ik} takes the value of 1 if customer $i \in N_v$ is visited in pure vehicle route $k \in \Psi$; it takes the value of 0, otherwise; and e_{il} takes the value of 1 if customer $i \in N$ is visited in vehicle route with subtours $l \in \Pi$, it takes the value of 0, otherwise.

Let us define the binary variables x_j that takes the value of 1 if and only if route $j \in \Gamma$ is used in the solution of the TTRP (it takes the value of 0, otherwise); y_k takes the value of 1 if and only if route $k \in \Psi$ is used (it takes the value of 0, otherwise); and z_l takes the value of 1 if and only if route $l \in \Pi$ is included in the TTRP solution (it takes the value of 0, otherwise). Let c_r be the cost (total distance) of any route $r \in \Omega (= \Gamma \cup \Psi \cup \Pi)$. The TTRP formulated as a set-partitioning problem (hereafter labeled as SPP) follows.

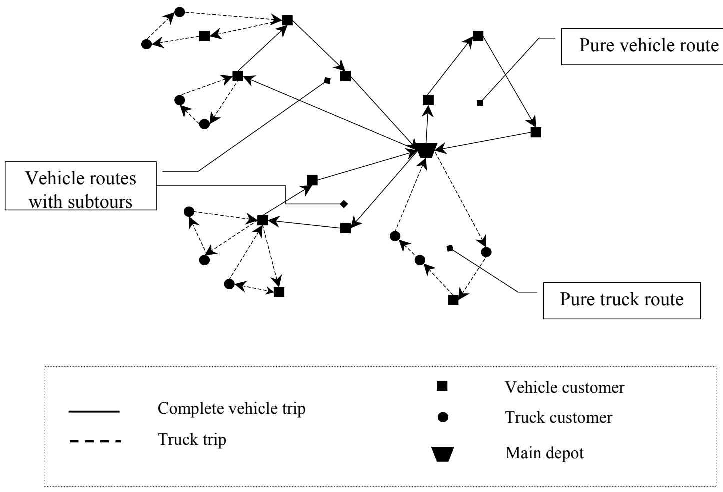

Fig. 1. Type of routes of a TTRP solution

$$\min \sum_{j=1}^{|\Gamma|} c_j x_j + \sum_{k=1}^{|\Psi|} c_k y_k + \sum_{l=1}^{|\Pi|} c_l z_l \tag{1}$$

subject to:

$$\sum_{j=1}^{|\Gamma|} a_{ij} x_j + \sum_{k=1}^{|\Psi|} b_{ik} y_k + \sum_{l=1}^{|\Pi|} e_{il} z_l = 1, \qquad \forall i \in N_v \tag{2}$$

$$\sum_{j=1}^{|\Gamma|} a_{ij} x_j + \sum_{l=1}^{|\Pi|} e_{il} z_l = 1, \qquad \forall i \in N_t \tag{3}$$

$$\sum_{j=1}^{|\Gamma|} x_j + \sum_{k=1}^{|\Psi|} y_k + \sum_{l=1}^{|\Pi|} z_l \leq m_t, \tag{4}$$

$$\sum_{k=1}^{|\Psi|} y_k + \sum_{l=1}^{|\Pi|} z_l \leq m_r, \tag{5}$$

$$x_j \in \{0,1\}, \qquad \forall j \in \Gamma \tag{6}$$

$$y_k \in \{0,1\}, \qquad \forall k \in \Psi \tag{7}$$

$$z_l \in \{0,1\}, \qquad \forall l \in \Pi \tag{8}$$

The objective function (1), to be minimized, is the total distance of the solution. Constraints (2) assure that each vehicle customer is visited exactly once; whereas, constraints (3) assure that each truck customer is visited exactly once by a truck route or a vehicle route with subtours. Constraints (4) and (5) impose an upper bound on the number of trucks and trailers that can be used, respectively. Since the number of feasible routes is huge, using this formulation directly is not possible. Therefore, we designed a heuristic solution approach that solves SPP over the subset of routes $\overline{\Omega} = \overline{\Gamma} \cup \overline{\Psi} \cup \overline{\Pi}$, where $\overline{\Gamma} \subseteq \Gamma$, $\overline{\Psi} \subseteq \Psi$ and $\overline{\Pi} \subseteq \Pi$.

4 Solution approach

The proposed heuristic follows a two-stage approach. In the first stage, a pool of columns (routes) $\overline{\Omega}$ is built by extracting the routes of the local optima visited during the search of a hybrid metaheuristic; then, in the second stage, we solve a SPP over the routes in $\overline{\Omega}$ to produce a possibly better solution. Different variants of this approach have been used before to solve several vehicle routing problems such as the capacitated vehicle routing problem [8,13], the vehicle routing problem with time windows [4], the split delivery vehicle routing problem [2], the heterogeneous fixed fleet vehicle routing problem [15], and a min-max selective vehicle routing problem [1]. In fact, one of the best solution methods for the capacitated vehicle routing problem is the adaptive memory programming approach of Rochat and Taillard [13] that uses a set-partitioning problem as post-optimizer.

4.1 GRASP/VNS for the TTRP

In this section, we briefly outline the main components of the hybrid GRASP/VNS, mechanism responsible of filling the pool of routes $\overline{\Omega}$ of the heuristic column generation approach. For a detailed description the reader is referred to [18].

4.1.1 Greedy Randomized Construction

The greedy randomized construction is performed by a tour splitting procedure (*Split*) that follows the route-first, cluster-second principle [5]. A giant tour T that visits all the customers in N is found using a randomized nearest neighbor heuristic with a restricted candidate list of size κ. Then, a solution S of the TTRP is derived from $T = (0, \dots, t_i, \dots, t_n, 0)$ by means of a tour splitting procedure. This procedure constructs one auxiliary acyclic graph $H = (X, U, W)$, where the set of nodes contains a dummy node 0 and n nodes numbered 1 through n, and node i represents the customer in the i-th position of T. The arc set U contains arc $(i-1, j)$ if and only if the subsequence $(t_i, \dots, t_j)$ can be served by a feasible route. Finally, the weight $w_{i-1,j}$ of arc $(i-1, j)$ is the total distance of the corresponding route. To derive S it is necessary to find the shortest path between 0 and n in H. The cost of the shortest path corresponds to the total distance of S and the arcs in the shortest path represent its routes.

The tour splitting procedure for the solution of the TTRP takes into account the heterogeneous fixed fleet by solving a resource-constrained shortest-path problem, where the resources are the available trucks and trailers. Moreover, if the arc $(i-1, j)$ represents a vehicle route with subtours its weight $w_{i-1,j}$ is found via a dynamic programming method that solves a restricted version of the single truck and trailer routing problem with satellite depots [16]; whereas for pure truck routes and pure vehicle routes it is easily calculated given their simple structure. However, it may be difficult to find feasible solutions with the tour splitting procedure in problems with a tight ratio between the total demand and the total capacity. Hence, if for a given giant tour there is no feasible solution (with at most m_t trucks and m_r trailers), an infeasible solution is obtained solving an unrestricted shortest-path problem.

4.1.2 Variable neighborhood search (VNS)

The VNS procedure takes the incumbent solution S produced by the tour-splitting mechanism and tries to improve it by performing iteratively three steps: (i) randomly exchanges p pairs of customers from its giant tour T to obtain a new giant tour T'; (ii) derives a new solution S' by applying the tour splitting procedure to T'; and (iii) applies a variable neighborhood descent (VND) to S'. The VND explores sequentially the following five neighborhoods using a best-improvement strategy: Or-opt (in single routes and subtours), node exchange (in single routes/subtours and between pairs of routes/subtours), 2-opt (in single routes/subtours and between pairs of routes/subtours), node relocation (in single routes/subtours and between pairs of routes/subtours), and a root refining procedure for subtours described in [7]. The VNS is repeated over ni iterations, and the value of p is controlled dynamically between 1 and p_{max}. Since infeasible solutions are accepted as initial solutions and also during the search of the VNS, the incumbent solution of VNS is replaced by S' if $f(S') < f(S)$ and its infeasibility $\Phi(S') = \max\{0, \frac{ut(S')}{m_t} - 1\} + \max\{0, \frac{ur(S')}{m_r} - 1\}$ does not exceed a given limit μ, where $f(S)$ denotes the objective function of S, and $ut(S)$ and $ur(S)$ the number of trucks and trailers used in S. At each call of VNS, μ is initialized at μ_{max}, and it is updated at each iteration with $\mu \leftarrow \mu - \frac{\mu_{max}}{ni}$.

4.2 Overview of the method

Algorithm 1 presents the structure of the heuristic column generation. The GRASP/VNS cycle (lines 2-31) is repeated ns times. At each main iteration, an initial solution for VNS is generated with the route-first cluster-second procedure (lines 3-4) followed by a VND improvement (line 5). The routes of this initial solution are added to the pool $\overline{\Omega}$ (line 7). Then, the VNS cycle (lines 13-30) improves the solution using a perturbation procedure (line 14) followed by VND (line 16). The pool is also updated with the routes of the local optimum reached after VND (line 17), even if it is infeasible. Each time a route is inserted in the pool we check that no duplicates are created. More complex domination rules could be implemented. However, the presolver of the optimizer will automatically reduce the size of the pool before solving the set-partitioning problem.

Within the VNS phase, the search alternates between solutions and giant tours; the perturbation is performed on giant tours, while the VND operates over TTRP solutions, therefore procedure *Concat* (line 20) creates a giant tour from a solution by concatenating its routes in a single string.

After the GRASP/VNS main cycle, the post-optimization phase solves the SPP over the routes in the pool $\overline{\Omega}$ (line 32). The columns of the optimal solution are used to reconstruct a possibly better solution. Since a given column does not specify the structure of the route it represents (to build a solution from the set of optimal columns), it is necessary to record the route associated with each column in the pool.

Algorithm 1 Heuristic column generation for the TTRP

```
Require: $ns$, $\kappa$, $ni$, $p_{max}$, $\mu_{max}$
Ensure: $S^*$
 1: $f^* := \infty$, $\overline{\Omega} = \emptyset$
 2: for $i = 1$ to $ns$ do
 3:   $T := RandomizedNearestNeighbor(N, \kappa)$
 4:   $S := Split(T)$
 5:   $S := VND(S)$
 6:   $T := Concat(S)$
 7:   Update $\overline{\Omega}$ with the routes of $S$
 8:   if $f(S) < f^*$ and $\Phi(S) = 0$ then
 9:     $S^* := S$
10:     $f^* := f(S)$
11:   end if
12:   $p := 1$, $\mu := \mu_{max}$
13:   for $j = 1$ to $ni$ do
14:     $T' := perturb(T, p)$
15:     $S' := Split(T')$
16:     $S' := VND(S')$
17:     Update $\overline{\Omega}$ with the routes of $S'$
18:     if $f(S') < f(S)$ and $\Phi(S') \leq \mu$ then
19:       $S := S'$
20:       $T := Concat(S)$
21:     end if
22:     if $f(S') < f^*$ and $\Phi(S') = 0$ then
23:       $S^* := S'$
24:       $f^* := f(S')$
25:       $p := 1$
26:     else
27:       $p := \min\{p + 1, p_{max}\}$
28:     end if
29:     $\mu = \mu - \frac{\mu_{max}}{ni}$
30:   end for
31: end for
32: $S^*$ :=Solve SPP over the pool $\overline{\Omega}$
33: return $S^*$
```

5 Computational experiment

The heuristic column generation was implemented in Java and compiled using Eclipse JDT 3.5.1. For the solution of the set-partitioning problem we use Gurobi version 3.0 [9]. All the experiments were ran on a computer with an Intel Xeon running at 2.67 GHz (clock speed) under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. For the sake of fairness, we used the same parameters as Villegas et al. [18] for the GRASP/VNS: $ns = 60$, $\kappa = 2$, $ni = 200$, $\mu_{max} = 0.25$, $p_{max} = 6$. We evaluated the performance of the proposed method on the test bed described by Chao [7], which comprises 21 problems ranging from 50 to 199 customers and three different fractions of truck customers for each size.

Table 1
Results of the heuristic column generation on the test bed of Chao [7]

Problem					Heuristic column generation				
Number	n	$\|Nt\|$	Previous BKS	New BKS	Best	Gap(%)	Avg.	Gap(%)	Time (min)
1	50	12	564.68[a,b,d,e]	-	**564.68**	0.00	564.68	0.00	1.35
2	50	15	611.53[b,d,e]	-	**611.53**	0.00	611.94	0.07	1.16
3	50	37	618.04[a,b,d,e]	-	**618.04**	0.00	618.04	0.00	0.91
4	75	18	798.53[a,b,d,e]	-	**798.53**	0.00	798.53	0.00	2.37
5	75	37	839.62[a,b,d,e]	-	**839.62**	0.00	839.62	0.00	2.25
6	75	56	930.64[b]	-	943.39	1.37	948.61	1.93	2.24
7	100	25	830.48[a,b,d,e]	-	**830.48**	0.00	830.48	0.00	11.10
8	100	50	872.56[b,d]	**870.94**	**870.94**	-0.19	872.42	-0.02	5.80
9	100	75	912.02[b]	-	914.23	0.24	914.23	0.24	6.75
10	150	37	1039.07[a,b]	**1036.96**	**1036.96**	-0.20	1038.37	-0.07	40.96
11	150	75	1093.37[d]	**1091.91**	**1091.91**	-0.13	1092.64	-0.07	17.31
12	150	112	1152.32[d]	**1149.41**	**1149.41**	-0.25	1150.26	-0.18	24.60
13	199	49	1287.18[a,b]	**1284.71**	**1284.71**	-0.19	1288.50	0.10	177.72
14	199	99	1339.36[d]	**1333.66**	**1333.66**	-0.43	1335.84	-0.26	22.55
15	199	149	1420.72[d]	**1416.51**	**1416.51**	-0.30	1420.07	-0.05	28.95
16	120	30	1002.49[a,b,d]	-	**1002.49**	0.00	1002.81	0.03	10.35
17	120	60	1026.20[b]	-	1042.35	1.57	1042.57	1.60	9.93
18	120	90	1098.15[b]	-	1112.68	1.32	1115.00	1.53	9.08
19	100	25	813.30[b]	-	813.50	0.02	814.48	0.14	4.10
20	100	50	848.93[a,c]	-	849.34	0.05	850.48	0.18	4.51
21	100	75	909.06[a,c,d,e]	-	**909.06**	0.00	909.06	0.00	5.02

a. Solution found by Scheuerer [14]
b. Solution found by Lin et al. [11]
c. Solution found by Caramia and Guerriero [6]
d. Solution found by Villegas et al. [18]
e. Solution found by heuristic column generation

Table 1 shows the average and best results of the heuristic column generation over ten runs for each problem in the test bed. The gap to the previous best-known solution (BKS) is reported in the table. Negative gaps indicate that the heuristic column generation improved the BKS; and values in bold indicate that the BKS has been found by the proposed method. Table 1 also includes the average running time in minutes.

In summary, the proposed heuristic column generation has a good performance for the solution of the TTRP, improving 7 out of 21 BKS. Remarkably, all these new BKS were found with a single set of parameters. On the contrary, most of the previous methods discovered the BKS during sensibility analysis with different sets of parameters or after extensive computational experiments [11,14]. Moreover, the difference between the average and best results of the proposed matheuristic is very narrow proving the robustness of the method.

In terms of computing time, all but two instances have been solved in less than 30 minutes; being the exceptions problems 10 and 13, that take 40 minutes and almost 3 hours, respectively. It seems that problems with a smaller fraction of truck customers (problems 7, 10 and 13) are more difficult for the SPP post-optimization, note that within the same problem size these instances take in general longer running times. Moreover, for the same problem size the running time of the heuristic column generation can be highly variable when the fraction of truck customers varies.

To evaluate the contribution of the post-optimization phase it is necessary to compare the results of the heuristic column generation with the results obtained with a simple GRASP/VNS and other methods from the literature. Table 2 presents the results of GRASP/VNS without any post-optimization phase, the GRASP/VNS with evolutionary path relinking (EvPR) proposed by Villegas et al. [18], and the simulated annealing (SA) of Lin et al. [11] (SA). Table 2 reports the best (*Best*) and average (*Avg.*) cost and the average time in minutes (*Time*) over ten runs for each problem and method. The last rows of the table summarize the average gaps above BKS, the number of times that each method found the best-known solution (NBKS) and the average running time. As it can be seen, the proposed heuristic column generation outperforms the previous methods from the literature, with a very small average gap to BKS of 0.33% that is almost one third of the gap of EvPR (the second best method). Values in bold indicate that the BKS has been found by a given method. The heuristic column generation approach found 15 out of 21 BKS compared to 8 found by EvPR, and 5 found by SA. Moreover, except for the BKS of problem 9, the proposed matheuristic also found the BKS retrieved by the other two methods.

The comparison with GRASP/VNS reveals the effectiveness of the post-optimization phase. Without such a phase, GRASP/VNS is not competitive, but the post-optimization phase increases the average running time by a factor of 1.49. Moreover, the average running time of the heuristic column generation (18.52 min.) is greater than the average running time of EvPR (14.58 min.). Nonetheless, this value is highly biased by the running time of problem 13 (177.72 min.). If this value is not taken into account the average running time decreases to 10.58 min. It is also important to note that the running time of the heuristic column generation is shorter than the running time of EvPR in 17 out of 21 problems.

6 Conclusions and future work

In this work we presented a set-partitioning formulation of the truck and trailer routing problem that is used in a two-phase column generation heuristic. First, routes are picked from the local optima found by a GRASP/VNS. Second, these routes are used as columns of the set-partitioning formulation that is solved using a commercial mixed-integer programming optimizer. With this post-optimization approach the performance of GRASP/VNS is clearly improved giving rise to a very effective matheuristic combining GRASP and column generation. The proposed heuristic column generation improved the best-known solutions for 7 out of 21 test problems and outperforms all the other previous methods from the literature. Nonetheless, there is still room for improvement for the proposed method, mainly to reduce and stabilize its running time. Future research directions include the use of specialized methods to solve the set-partitioning problem and the implementation of more elaborated pool-management strategies to reduce the number of columns of the set-partitioning problem.

Acknowledgments

This research was partially supported by the Champagne-Ardenne Regional Council (France) and by Colciencias (Colombia).

References

[1] Cristiano Arbex Valle, Leonardo Conegundes Martinez, Alexandre Salles da Cunha, and Geraldo R. Mateus. Heuristic and exact algorithms for a min-max selective vehicle routing problem. *Computers & Operations Research*, October 2010.

[2] Claudia Archetti, Maria Grazia Speranza, and Martin W. P. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1):22–31, 2008.

[3] Michael O. Ball. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21–38, 2011.

[4] Guilherme Bastos Alvarenga, Geraldo Robson Mateus, and G. de Tomi. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584, 2007.

[5] John E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11:403–408, 1983.

[6] Massimiliano Caramia and Francesca Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61:1168–1180, 2010.

[7] I-Ming Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[8] Chris Groër, Bruce Golden, and Edward Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2(2):79–101, 2010.

[9] Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, April 2010. http://www.gurobi.com/doc/30/refman/node201.html.

[10] Laetitia Jourdan, Matthieu Basseur, and El-Ghazali Talbi:. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, 199(3):620–629, 2009.

[11] Shih-Wei Lin, Vincent F. Yu, and Shuo-Yan Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5):1683–1692, 2009.

[12] Jakob Puchinger and Günther R. Raidl. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In José Mira and José R. Álvarez, editors, *IWINAC*, volume 3562 of *Lecture Notes in Computer Science*, pages 41–53, 2005.

[13] Yves Rochat and Eric D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[14] Stephan Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[15] Eric D. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33(1):1–14, 1999.

[16] Juan G. Villegas, Christian Prins, Caroline Prodhon, Andrés L. Medaglia, and Nubia Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

[17] Juan G. Villegas, Christian Prins, Caroline Prodhon, Andrés L. Medaglia, and Nubia Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, Tromsø, Norway, June 20-25 2010.

[18] Juan G. Villegas, Christian Prins, Caroline Prodhon, Andrés L. Medaglia, and Nubia Velasco. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9):1319–1334, 2011.

Table 2. Comparison of the heuristic column generation with other methods

Problem			GRASP/VNS			EvPR			SA			Heuristic column generation		
Number	n	BKS	Best	Avg.	Time[1]	Best	Avg.	Time[1]	Best	Avg.	Time[2]	Best	Avg.	Time[1]
1	50	564.68	**564.68**	568.31	0.91	**564.68**	565.99	1.17	566.82	568.86	6.8	**564.68**	564.68	1.35
2	50	611.53	614.27	617.53	1.00	**611.53**	614.23	1.29	612.75	617.48	6.67	**611.53**	611.94	1.16
3	50	618.04	**618.04**	619.07	0.86	**618.04**	618.04	1.05	**618.04**	620.50	5.59	**618.04**	618.04	0.91
4	75	798.53	802.41	815.16	1.86	**798.53**	803.51	2.69	808.84	817.71	16.32	**798.53**	798.53	2.37
5	75	839.62	841.81	857.79	1.99	**839.62**	841.63	2.82	**839.62**	858.95	14.42	**839.62**	839.62	2.25
6	75	930.64	989.71	1040.19	1.95	940.59	961.47	2.89	934.11	942.60	13.65	943.39	948.61	2.24
7	100	830.48	830.62	832.27	4.11	**830.48**	830.48	6.05	**830.48**	838.50	24.96	**830.48**	830.48	11.10
8	100	870.94	881.53	885.01	4.35	872.56	876.21	6.96	875.76	882.70	24.03	**870.94**	872.42	5.80
9	100	912.02	916.63	930.55	5.67	914.23	918.45	8.38	**912.64**	921.97	21.75	914.23	914.23	6.75
10	150	1036.96	1050.76	1062.03	9.95	1046.71	1050.11	18.84	1053.90	1074.38	63.61	**1036.96**	1038.37	40.96
11	150	1091.91	1114.64	1122.43	11.02	1093.37	1100.95	21.20	1093.57	1108.88	60.33	**1091.91**	1092.64	17.31
12	150	1149.41	1159.88	1174.89	14.00	1152.32	1158.88	25.78	1155.44	1166.59	51.7	**1149.41**	1150.26	24.60
13	199	1284.71	1319.38	1332.55	18.71	1298.89	1305.83	43.94	1320.21	1340.98	119.56	**1284.71**	1288.50	177.72
14	199	1333.66	1380.86	1395.50	20.07	1339.36	1354.04	45.57	1351.54	1367.91	113.75	**1333.66**	1335.84	22.55
15	199	1416.51	1454.10	1462.23	25.14	1423.91	1437.52	59.83	1436.78	1454.91	93.87	**1416.51**	1420.07	28.95
16	120	1002.49	1003.99	1005.88	8.13	**1002.49**	1003.07	14.73	1004.47	1007.26	41.46	**1002.49**	1002.81	10.35
17	120	1026.20	1045.08	1050.86	8.09	1042.46	1042.61	13.17	1026.88	1035.23	38.81	1042.35	1042.57	9.93
18	120	1098.15	1121.07	1128.51	7.73	1113.07	1118.63	12.69	1099.09	1110.13	31.34	1112.68	1115.00	9.08
19	100	813.30	817.11	820.94	3.82	813.50	819.81	5.21	814.07	823.01	29.58	813.50	814.48	4.10
20	100	848.93	860.12	861.34	4.21	860.12	860.12	5.62	855.14	859.06	28.47	849.34	850.48	4.51
21	100	909.06	912.35	913.62	4.59	**909.06**	909.06	6.31	**909.06**	915.38	24.03	**909.06**	909.06	5.02
Avg. gap above BKS			1.37%	2.34%		0.44%	0.92%		0.57%	1.59%		0.22%	0.33%	
NBKS			2			8			5			15		
Avg. time (min)					7.53			14.58			39.56			18.52

[1]Average time in minutes on a 2.7 GHz Intel Xeon

[2]Average time in minutes on a 1.5 GHz Pentium IV