GRASP/Evolutionary Local Search Hybrids for a Truck and Trailer Routing Problem

Juan G. Villegas*† Andrés L. Medaglia† Christian Prins* Caroline Prodhon*
Nubia Velasco†

*Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay,
Université de Technologie de Troyes
BP 2060, 10010 Troyes Cedex, France
{juan_guillermo.villegas, christian.prins, caroline.prodhon}@utt.fr

†Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería
Industrial, Universidad de los Andes
A.A. 4976, Bogotá D.C., Colombia
{jg.villegas64, amedagli, nvelasco}@uniandes.edu.co

1 Introduction

The vehicle routing problem (VRP) aims to find a set of routes of minimum total length to serve the demand of a set of customers using a fleet of capacitated vehicles based at a central depot [6]. In this work we present a new extension of the VRP, the single truck and trailer routing problem with satellite depots (STTRPSD). To solve the STTRPSD we developed a hybrid metaheuristic based on greedy randomized adaptive search procedures (GRASP) and evolutionary local search (ELS). The remainder of the paper is organized as follows. Section 2 explains the STTRPSD and gives a brief literature review of related problems. The elements of the hybrid GRASP/ELS are described in Section 3. Section 4 presents a computational evaluation of the proposed method. Finally, some conclusions are given in Section 5.

2 Problem statement and literature review

In the STTRPSD, a single vehicle (a truck with a detachable trailer) based at a main depot serves the demand of a set of customers, reachable only by the truck without the trailer. Therefore, there is a set of parking locations (called trailer points or satellite depots) where it is possible to detach the trailer and to transfer products between the truck and the trailer. The STTRPSD is defined on a graph $G = (V, A)$. $V = \{0\} \cup V_D \cup V_C$ is the node set, the main depot is node 0, $V_D = \{1, 2, \ldots, p\}$ is the set of trailer points, and $V_C = \{p + 1, p + 2, \ldots, p + n\}$ is the set of customers with known demands q_i ($i \in V_C$); A is the arc set, with costs c_{ij} ($i, j \in V$) satisfying the triangle inequality.

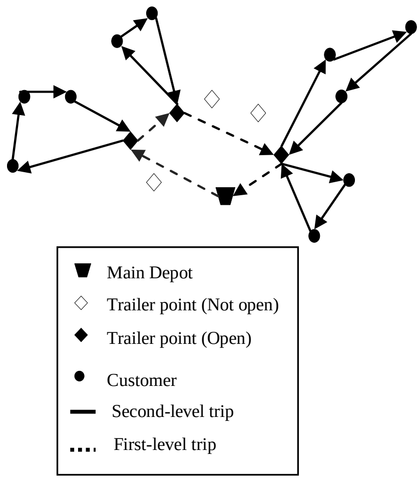


Figure 1: STTRPSD feasible solution.

The parameters Q_V and Q_T are the capacities of the truck and the trailer, respectively. To ensure feasibility with one vehicle, the sum of demands does not exceed $Q_V + Q_T$.

In feasible solutions of the STTRPSD, each client $i \in V_C$ is assigned to one trailer point. Consequently, trailer points with assigned customers are said to be open. The first-level trip departing from the main depot is performed by the truck with the trailer and visits the subset of open trailer points. Each customer must be serviced by exactly one second-level trip (performed by the truck alone), starting and ending at the trailer point to which it is assigned. The total load in a second-level trip should not exceed Q_V. The goal is to minimize the total cost of the trips. The STTRPSD is NP-hard since it includes the VRP (one satellite) and the Multi-Depot VRP (MDVRP, null cost between any two depots) as particular cases. Note that in the STTRPSD along with the routing, there is a location decision related to the selection of trailer points. Figure 1 depicts a feasible solution of the STTRPSD.

Applications of the STTRPSD appear, for instance, in milk collection [5], where customers are often served by a single tanker with a detachable tank trailer. Trailer points are in general parking locations on main roads, while farms are located on narrow roads inaccessible with the trailer. The arc routing counterpart of the STTRPSD appears in the design of park-and-loop routes for postal delivery [7], where the postman drives a vehicle from the postal facility to a parking location, loads his sack, and delivers mail by walking the streets forming a loop. Then, he returns to the vehicle, loads his sack again, and delivers by foot in a second loop. Once he has served all the walking loops nearby, he drives the vehicle to the next parking location. At the end of the day the postman returns to the postal facility.

The STTRPSD is closely related with the truck and trailer routing problem (TTRP), another variant of the VRP in which a heterogeneous fleet comprised of m trucks and b trailers ($b < m$) is used to satisfy the demand of a set of customers partitioned in *vehicle* and *truck customers*, where the latter are only accessible by truck. The TTRP has been tackled using tabu search [1, 15] and simulated annealing [8]. The STTRPSD differs from the TTRP in the use of a single vehicle and the definition of trailer points independent from customer locations. When the STTRPSD is extended to include several vehicles and capacitated depots it transforms into the seldomly studied two-echelon capacitated vehicle routing problem (2E-CVRP) introduced in [3].

3 Solution methods for the STTRPSD

3.1 Hybrid GRASP/ELS

GRASP [2] is a memory-less multi-start method in which local search is applied to initial solutions obtained with a greedy randomized heuristic. Recently, GRASP has been used to develop competitive hybrid metaheuristics [13]. On the other hand, Evolutionary Local Search (ELS) [16] could be seen as an evolutionary extension of Iterated Local Search (ILS) [9], in which a single solution is mutated to obtain λ children that are further improved using local search. Following a $(1 + \lambda)$ selection paradigm, the solution of the next generation is the best among the parent and its children. A hybrid GRASP/ELS is derived by replacing the local search of GRASP by ELS. Despite its simplicity a hybrid GRASP/ELS is currently one of the best metaheuristics for the VRP [12]. This fact motivated its use in the solution of the STTRPSD. The elements of the proposed hybrid metaheuristic are described in the following sections.

3.2 Greedy randomized construction

Route-first, cluster-second based metaheuristics have shown to be a good alternative to solve capacitated routing problems [10]. Following this approach, we generate initial solutions by means of a tour splitting procedure (hereafter *Split*) that obtains a solution S of the STTRPSD from a giant tour T visiting all the customers, where $T = (t_1, t_2, \ldots, t_j, \ldots, t_n)$ and t_j represents the customer in the j-th position. Giant tours are constructed with a randomized nearest neighbor method with a restricted candidate list (RCL) of size r, that ignores capacity constraints and trailer-point selection.

It is worth mentioning that the splitting procedure for the STTRPSD is more complicated than that of the VRP. The reason being that the distance between trailer points is included in the objective function, making the selection and routing of trailer points an important decision. For the STTRPSD, *Split* partitions T (optimally subject to its sequence) into second-level trips, inserts the trailer points, and gives the corresponding first-level trip. *Split* is a dynamic programming method, in which state $[l, j]$ represents an optimal splitting of $(t_1, t_2, \ldots, t_j)$ with trailer point l for the last trip, and $[0, 0]$ denotes the initial state. Let F_{lj} denote the cost of the state $[l, j]$ and θ_{ijk} the cost of a trip visiting customers $(t_i, t_{i+1}, \ldots, t_j)$ from trailer point k. We have the following recurrence relations for any customer t_j and any trailer point l:

$$F_{lj} = \begin{cases} 0, & \text{if } l = 0 \text{ and } j = 0 \\ min\left\{F_{ki} + c_{kl} + \theta_{i+1,j,l}\right\}, & i < j : \sum_{u=i+1}^{j} q(t_u) \leq Q_V, k = 0 \text{ if } i = 0 \text{ else } k \in V_D \end{cases}$$

Note that states $[l, n]$ $(l \in V_D)$ do not include the return cost to the main depot. Hence, the cost of an optimal splitting is then $z = min_{l \in V_D} \{F_{ln} + c_{l0}\}$.

The dynamic programming method can be viewed as a shortest path problem in a state graph $H = (X, U, W)$. The node set X contains $np + 2$ nodes: two copies of the main depot (α and β), acting as input and output nodes for H, and np nodes for the states $[l, j]$. The arc set U has three types of arcs: (i) the outgoing arcs of α $((\alpha, [l, j]);\ l \in V_D,\ j \in V_C)$, that represent second-level trips performed just after leaving the main depot, their costs include the distance from the main

Arc	Trip	Inter-depot distance	Trip distance	Arc Cost
(0,[1,4])	(1,4,1)	$c_{01} = 1.41$	$\theta_{111} = c_{14} + c_{41} = 4.47$	5.89
(0,[2,6])	(2,4,6,2)	$c_{02} = 3.16$	$\theta_{122} = c_{24} + c_{46} + c_{62} = 6$	9.16
([1,4],[1,6])	(1,6,1)	$c_{11} = 0$	$\theta_{221} = c_{16} + c_{61} = 4.47$	4.47
([1,6],[1,5])	(1,3,5,1)	$c_{11} = 0$	$\theta_{341} = c_{13} + c_{35} + c_{51} = 6.58$	6.58
([1,6],[2,5])	(2,3,5,2)	$c_{12} = 2.83$	$\theta_{342} = c_{23} + c_{35} + c_{52} = 4.83$	7.66
([1,5],0)	-	$c_{10} = 1.41$	-	1.41
([2,5],0)	-	$c_{20} = 3.16$	-	3.16

Table 1: Example of the calculation of the cost of the arcs of Figure 2(c).

depot to the first selected trailer point; (ii) the arcs between internal nodes of the state graph $(([k,i],[l,j]);\ k,l \in V_D;\ i,j \in V_C)$, that represent second-level trips, the cost of the arcs includes the distance between the trailer point of the previous trip and that of the current trip; and finally (iii) the incoming arcs of β $(([l,n],\beta);\ l \in V_D)$, that represent the return to the main depot from the last trailer point. W is a mapping defining the cost of each arc. *Split* can be implemented in $O(nbp^2)$, where b is the average number of customers per trip, and without the need of generating H explicitly.

As an example, Figure 2(a) illustrates a simple instance of the STTRPSD with $Q_V = 3$, $p = 2$ trailer points (1 and 2), and $n = 4$ customers (3, 4, 5 and 6) with demands 2, 1, 1 and 1 respectively. The length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean. Figure 2(b) shows the results of splitting the giant tour $T = (4, 6, 3, 5)$. Figure 2(c) shows the corresponding state graph, where the arcs of the optimal split are in bold. Table 1 illustrates the calculation of the cost of the arcs of Figure 2(c).

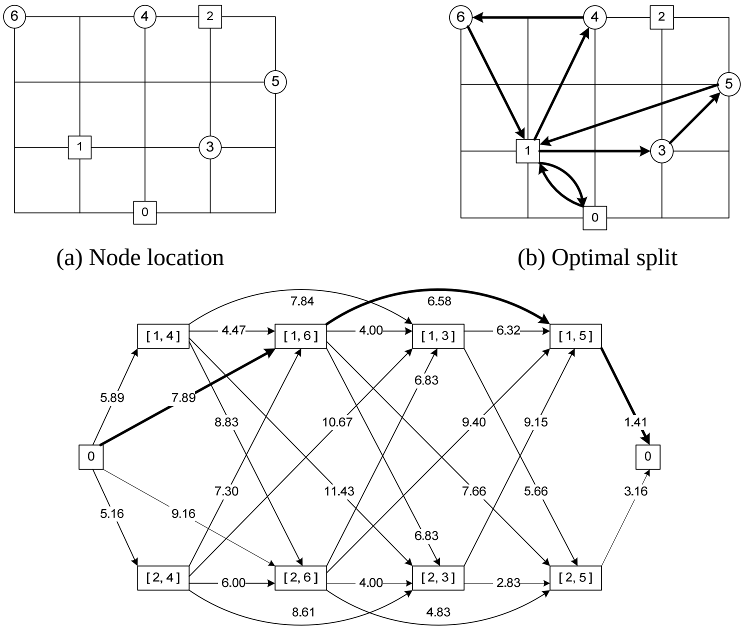
In general, each trailer point may have several trips. Consider two capacity-feasible sequences of customers close to one trailer point k but separated in the giant tour T by other customers. Since S follows the order of T, it will contain two non-consecutive trips from trailer point k, so k is visited twice in the first-level trip. Such a solution is called weak splitting. Since the triangle inequality holds, after *Split*, the length of the first-level trip is further reduced by making adjacent in T non-consecutive trips with common trailer points, thus eliminating the weak splitting.

3.3 Local search

The improvement of solutions of the hybrid GRASP/ELS is done using a variable neighborhood descent (VND) [4] with a first-improvement strategy. In the VND there are two types of moves: (i) moves 1, 2 and 3 aimed to improve the routing within the solution; and (ii) moves 4 and 5 that modify the set of open trailer points. Moreover, when applied to second-level trips moves 1-3 only explore feasible solutions that meet the capacity constraint. A brief description of the five moves follows:

Move 1. One customer or trailer point is removed from its current position and reinserted elsewhere. When applied to a customer the new position could be in the same second-level trip or in another second-level trip; whereas moves of trailer points are made within the first-level trip.

Move 2. Two customers or two trailer points are exchanged. The two customers may belong to



(a) Node location (b) Optimal split

(c) State graph

Figure 2: Example of *Split* for the STTRPSD, for $T = (4, 6, 3, 5)$.

different second-level trips.

Move 3. A modified 2-opt move in which two edges are exchanged. If the edges belong to the same trip the move reduces to a simple 2-opt for the traveling salesman problem; whereas if the edges belong to a pair of second-level trips with different trailer points, the new trips are assigned to the trailer point from the original trips with the smallest connecting cost. This move is inspired by a neighborhood for the capacitated location-routing problem [11].

Move 4. Each open trailer point is considered for closure. If the trailer point is closed, its trips are relocated to the remaining trailer points. When relocating a second-level trip the trailer point is inserted in the position of the trip with the smallest routing cost.

Move 5. In contrast to move 4, each closed trailer point is considered for opening, only if there exist second-level trips whose cost decrease when relocated to the new trailer point. Again, when relocating a second-level trip the new trailer point is inserted in the best position.

3.4 Mutation

As shown in Algorithm 1, the hybrid GRASP/ELS alternates between solutions and giant tours. The algorithm begins with a giant tour T (obtained with the randomized nearest neighbor procedure), then *Split* is used to obtain a solution S that is further improved with VND. A new giant tour is obtained from the improved solution with a simple procedure (*Concat*) that creates chains of customers by concatenating the second-level trips departing from each open trailer point. To obtain a giant tour, these chains are again concatenated using the order of the visits to the trailer points in the first-level trip. Within the ELS, the mutation operator (*Mutate*) exchanges p pairs of customers in T, the value of p is controlled dynamically in the algorithm and varies from 1 to p_{max}.

3.5 Algorithm overview

Algorithm 1 presents the pseudocode of the proposed hybrid GRASP/ELS. Depending on the values of the parameters ns (number of initial solutions), ni (number of iterations of ELS) and nc (number of children) it is possible to obtain pure GRASP ($ns > 1$, $ni = nc = 0$), ELS ($ns = 1$, $ni > 1$, $nc > 1$) and ILS ($ns = 1$, $ni > 1$, $nc = 1$), or hybrids GRASP/ILS ($ns > 1$, $ni > 1$, $nc = 1$) and GRASP/ELS ($ns > 1$, $ni > 1$, $nc > 1$).

Algorithm 1 GRASP/ELS General Structure

Parameters: ns, ni, nc, p_{max}, r
Output: S^*

$f^* := \infty$
$p := 1$
for $i := 1$ to ns **do**
  $T := GreedyRandomizedNearestNeighbor(V_C, r)$
  $S := Split(T)$
  $S := VND(S)$
  $T := Concat(S)$
  for $j := 1$ to ni **do**
    $\widehat{f} := f(S)$
    for $k := 1$ to nc **do**
      $T' := Mutate(T, p)$
      $S' := Split(T')$
      $S' := VND(S')$
      if $f(S') < \widehat{f}$ **then**
        $\widehat{f} := f(S')$
        $\widehat{S} := S'$
      end if
    end for
    if $\widehat{f} < f(S)$ **then**
      $S := \widehat{S}$
      $T := Concat(S)$
      $p := 1$
    else
      $p := min(p + 1, p_{max})$
    end if
  end for
  if $f(S) < f^*$ **then**
    $f^* = f(S)$
    $S^* := S$
  end if
end for

4 Computational results

The proposed methods were tested on a set of randomly generated Euclidean instances with the following characteristics: $n \in \{25, 50, 100\}$, $p \in \{5, 10, 20\}$, and $Q_V \in \{1000, 2000, 4000\}$. The coordinates for trailer points and customers are generated randomly in a square of size 100×100. Each instance is named with the convention $STTRPSD$-n-p-$(0.001 \times Q_V)$-s, where s defines the strategy used to generate the location of customers and trailer points: c, for clustered; and r, for randomly distributed. The hybrid GRASP/ELS (Algorithm 1) was implemented in Java and the experiments described in this section were performed on a computer with an Intel Pentium D 945 running at 3.4 GHz with 1024 MBytes of RAM, on a Windows XP Professional environment.

By properly setting the values of ns, ni, and nc in Algorithm 1, it is possible to replicate the behavior of different metaheuristics. Consequently, after some preliminary experiments we decided to allocate a "budget"of 2500 calls of the VND to each method, distributing them as shown in Table 2. The idea is to see which metaheuristic makes the best use of the local search. All the methods share $p_{max} = 4$; and $r = 2$ for the multi-start methods (i.e., GRASP, GRASP/ILS and GRASP/ELS), and $r = 1$ for the methods with a single initial solution (i.e., ELS and ILS).

Table 2: Parameters of each metaheuristic.

Method	ns	ni	nc
GRASP	2500	0	0
ILS	1	2500	1
ELS	1	250	10
GRASP/ILS	50	50	1
GRASP/ELS	5	50	10

We used two heuristics as benchmark to compare the performance of the different metaheuristics. The first one is a simple cluster-first, route-second (CFRS) approach in which each customer is assigned to its nearest trailer point, then the customers assigned to each open trailer point are routed using an insertion heuristic (i.e., solve one VRP for each open trailer point), and finally, the first-level trip that visits the set of open trailer points is derived again from an insertion heuristic. The second heuristic is to apply VND (with a best-improvement strategy) to solutions constructed with the cluster-first, route-second heuristic (CFRS+VND).

Table 3 presents the performance of the heuristics and metaheuristics in a set of 36 test problems. Columns 2-6 represent the average cost of the solutions over 10 runs of each metaheuristic. Columns 7-8 report the cost of the solution found by the benchmark heuristics. The last column reports the cost of the best solution found for each problem. The last three rows report the number of times each method found the best known solution (NBKS), the average deviation (in %) above the best known solution (BKS), and the average running times (in seconds).

Table 3: Results for each instance.

Instance	GRASP	ILS	ELS	GRASP/ ILS	GRASP/ ELS	CFRS	CFRS+VND	Best
STTRPSD-25-5-1-c	405.5	405.5	405.5	405.5	405.5	444.1	405.5	405.5
STTRPSD-25-5-2-c	391.6	391.6	392.1	391.6	391.6	444.1	391.6	391.6
STTRPSD-25-5-4-c	348.8	348.8	348.8	348.8	348.8	444.1	391.6	348.8
STTRPSD-25-5-1-r	544.9	544.9	544.9	544.9	544.9	640.0	586.0	544.9
STTRPSD-25-5-2-r	463.5	463.5	463.5	463.5	463.5	640.0	526.3	463.5
STTRPSD-25-5-4-r	463.5	463.5	463.5	463.5	463.5	640.0	526.3	463.5
STTRPSD-25-10-1-c	417.7	417.7	417.7	417.7	417.7	477.2	423.8	417.7
STTRPSD-25-10-2-c	386.5	386.5	386.5	386.5	386.5	460.3	386.5	386.5
STTRPSD-25-10-4-c	381.2	381.8	381.8	381.6	381.8	460.3	386.5	381.2
STTRPSD-25-10-1-r	652.6	653.0	653.0	652.6	652.8	838.4	668.1	652.5
STTRPSD-25-10-2-r	555.6	555.6	555.6	555.6	555.6	789.7	600.3	555.6
STTRPSD-25-10-4-r	477.4	477.4	477.4	477.4	477.4	789.7	582.6	477.4
STTRPSD-50-5-1-c	491.8	491.8	491.8	491.8	491.8	574.2	560.2	491.8
STTRPSD-50-5-2-c	431.9	431.9	431.9	431.9	431.9	574.2	560.2	431.9
STTRPSD-50-5-4-c	404.5	404.5	404.5	404.5	404.5	574.2	560.2	404.5
STTRPSD-50-5-1-r	752.7	752.3	752.5	752.5	752.8	1 030.5	776.8	752.3
STTRPSD-50-5-2-r	701.6	701.9	702.0	701.6	701.6	980.6	751.7	701.6
STTRPSD-50-5-4-r	665.5	665.6	665.7	665.5	665.5	980.6	751.7	665.5
STTRPSD-50-10-1-c	387.1	386.1	386.3	386.1	386.1	466.4	393.3	385.5
STTRPSD-50-10-2-c	367.8	368.4	369.9	366.3	366.3	460.4	381.3	366.3
STTRPSD-50-10-4-c	353.6	353.6	353.6	353.6	353.6	460.4	381.3	353.6
STTRPSD-50-10-1-r	777.1	777.6	777.4	777.1	777.2	1013.2	823.7	777.0
STTRPSD-50-10-2-r	686.2	686.2	686.2	686.2	686.2	1013.2	758.9	686.2
STTRPSD-50-10-4-r	622.6	622.6	622.6	622.6	622.6	1013.2	758.9	622.6
STTRPSD-100-10-1-c	536.1	536.1	536.3	536.0	535.6	664.9	541.5	535.3
STTRPSD-100-10-2-c	478.9	479.3	479.0	478.9	478.9	656.0	517.4	478.9
STTRPSD-100-10-4-c	449.3	449.3	449.3	449.3	449.3	656.0	517.4	449.3
STTRPSD-100-10-1-r	1010.7	1011.6	1012.2	1009.4	1011.9	1345.9	1114.3	1006.9
STTRPSD-100-10-2-r	870.1	870.5	870.6	870.1	870.2	1246.7	1101.3	870.1
STTRPSD-100-10-4-r	841.9	842.4	842.6	841.8	841.9	1246.7	1101.3	840.0
STTRPSD-100-20-1-c	648.4	645.8	647.3	645.2	645.3	820.1	661.8	641.3
STTRPSD-100-20-2-c	571.4	571.1	571.7	570.8	571.0	808.6	630.8	569.8
STTRPSD-100-20-4-c	535.9	537.9	537.7	536.0	536.5	808.6	602.4	535.9
STTRPSD-100-20-1-r	1144.0	1133.9	1134.6	1129.8	1128.3	1388.6	1193.2	1118.8
STTRPSD-100-20-2-r	1032.7	1040.0	1038.9	1029.4	1030.1	1342.1	1142.9	1023.0
STTRPSD-100-20-4-r	897.4	904.7	902.8	899.3	902.0	1342.1	1138.8	895.0
NBKS	32	33	31	33	32	0	3	
Avg. Dev. BKS (%)	0.18	0.21	0.23	0.11	0.13	34.05	11.24	
Avg. CPU Time (s)	490.7	226.4	221.9	224.4	213.1	< 1	< 1	

Eventhough the CFRS heuristic is extremely fast, the average deviation above the best solutions is large (34.05%), compared to VND (11.24%) which still reports average running times of less than one second. As expected, better quality results can be obtained with metaheuristics, all reporting average deviations of less than 0.25%. ILS and GRASP/ILS provide only one best solution more than GRASP and GRASP/ELS. Eventhough GRASP is the slowest metaheuristic, it is able to achieve a small deviation (0.18%). One the other hand, ILS and ELS have almost the same results than GRASP, but run approximately two times faster. Finally, GRASP/ILS and GRASP/ELS, report the smallest deviations (0.11% and 0.13%, respectively) with competitive running times. Moreover, the hybrid methods scale much better than GRASP as the number of customers increases, as shown in Table 4.

Table 4: Average running time for different problem sizes

	Average Running Time (s)				
n	GRASP	ILS	ELS	GRASP/ILS	GRASP/ELS
25	17.63	13.22	13.13	12.98	12.82
50	131.41	84.03	83.35	82.50	80.52
100	1323.09	581.91	569.15	577.62	546.04

5 Conclusion

In this paper, we introduced the STTRPSD, a new variant of the VRP arising from practical applications such as milk collection. To solve the STTRPSD we proposed a hybrid metaheuristic that combines the elements of GRASP and ELS in a simple way. With this hybrid method it is possible to obtain better results that those obtained with GRASP, ILS or ELS with shorter running times. The hybrid metaheuristic combines in an effective way the sampling of the solution space of multi-start methods like GRASP, with the distinctive speed of ELS and ILS. Future directions of research include the development of hybrid metaheuristics based on GRASP and Path Relinking [14], and a lower bound to evaluate the proposed metaheuristics.

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

References

[1] I.-M. Chao. A Tabu Search Method for the Truck and Trailer Routing Problem. *Computers & Operations Research*, 29(1):33–51, 2002.

[2] T.A. Feo and M.G.C. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6: 109–133, 1995.

[3] J. Gonzalez-Feliu, G. Perboli, R. Tadei, and D. Vigo. The Two-Echelon Capacitated Vehicle Routing Problem. Working Paper, Control and Computer Engineering Department, Politecnico di Torino, 2007.

[4] P. Hansen and N. Mladenović Variable Neighborhood Search: Principles and Applications. *European Journal of Operational Research* 130(3): 449–467, 2001.

[5] A. Hoff and A. Løkketangen. A Tabu Search Approach for Milk Collection in Western Norway Using Trucks and Trailers. In *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Thailand, June 10-15, 2007.

[6] G. Laporte. What You Should Know About the Vehicle Routing Problem. *Naval Research Logistics*, 54(8): 811–819, 2007.

[7] L. Levy and L. Bodin. Scheduling of Local Delivery Carriers for the United States Postal Service. In M. Dror, editor, *Arc Routing: Theory, Solutions and Applications*, pages 419–442. Kluwer, Boston, 2000.

[8] S.-W. Lin, V. F. Yu, and S.-Y. Chou. Solving the Truck and Trailer Routing Problem Based on a Simulated Annealing Heuristic. *Computers & Operations Research*, 36(5): 1683–1692, 2009.

[9] H.R. Lourenço, O. Martin, and T. Stützle. Iterated Local Search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 321-353. Kluwer, Boston, 2003.

[10] C. Prins. N. Labadi, and M. Reghioui. Tour Splitting Algorithms for Vehicle Routing Problems. *International Journal of Production Research*, 47(2):507–536, 2009.

[11] C. Prins, C. Prodhon, and R. Wolfler-Calvo. Solving the Capacitated Location-Routing Problem by a GRASP Complemented by a Learning Process and a Path Relinking. *4OR - A Quarterly Journal of Operations Research*, 4(3):221–238, 2006.

[12] C. Prins. A GRASP × Evolutionary Local Search Hybrid for the Vehicle Routing Problem. In F. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, Heidelberg, 2009.

[13] M.G.C. Resende. Metaheuristic Hybridization with Greedy Randomized Adaptive Search Procedures. In Z-L. Chen and S. Raghavan, editors, *TutORials in Operations Research*, pages 295–319, INFORMS, 2008.

[14] M.G.C. Resende and C.C. Ribeiro. GRASP with Path-relinking: Recent Advances and Applications. In T. Ibaraki, K. Nonobe and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, pages 29–63, Springer, 2005.

[15] S. Scheuerer. A Tabu Search Heuristic for the Truck and Trailer Routing Problem. *Computers & Operations Research*, 33(4):894–909, 2006.

[16] S. Wolf and P. Merz. Evolutionary Local Search for the Super-peer Selection Problem and the p-hub Median Problem In T. Bartz-Beielstein, M.J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Lecture Notes in Computer Science*, 4771, pages 1-15. Springer, Heidelberg, 2007.