CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Tabu Search Heuristic for the Vehicle Routing Problem with Private Fleet and Common Carrier

Jean-François Côté
Jean-Yves Potvin

May 2007

CIRRELT-2007-08

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Tabu Search Heuristic for the Vehicle Routing Problem with Private Fleet and Common Carrier

Jean-François Côté[1], Jean-Yves Potvin[1*]

[1.] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7

Abstract. This paper describes a tabu search heuristic for a vehicle routing problem where the owner of a private fleet can either visit a customer with one of his vehicles or assign the customer to a common carrier. The owner's objective is to minimize the variable and fixed costs for operating his fleet plus the total costs charged by the common carrier. The proposed tabu search is shown to outperform the best approach reported in the literature on 34 benchmark instances with homogeneous fleet.

Keywords. Vehicle routing, private fleet, common carrier, tabu search.

Acknowledgements. Financial support for this work was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.

---

* Corresponding author: potvin@crt.umontreal.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2007

1 Introduction

In this work, we consider the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC), where each customer is served by one of the vehicles of an internal fleet or by an external common carrier. When a customer is assigned to the common carrier, a penalty cost is incurred that represents all costs associated with this assignment (since the specific route followed by the common carrier is not known). The objective of the private fleet's owner is to minimize the sum of fixed vehicle costs, travel costs and common carrier costs. Clearly, customers must be referred to the common carrier if the total demand from the customers exceeds the internal fleet capacity. However, it can also be more economical to do so.

A single-vehicle version of this problem is first mentioned in [17] and later solved to optimality with a branch-and-bound algorithm in [9] for problems with up to 200 customers. In [2], the problem is introduced as a special case of the Prize Collecting TSP (PCTSP), which is defined in [1] as follows: "A salesman who travels between pairs of cities at a cost depending only on the pair, gets a prize in every city that he visits and pays a penalty to every city that he fails to visit, wishes to minimize his travel costs and penalties, while visiting enough to collect a prescribed amount of prize money". When there is no prize associated with the vertices or, alternatively, when the prescribed amount is null, the single-vehicle version of the problem is obtained.

The multi-vehicle version is first reported and solved in [6] with a savings-based construction heuristic, followed by intra-route and inter-route customer exchanges. The authors in [3] do better by using two different initial solutions that are then improved with more sophisticated customer exchanges. Finally, even better results are reported in [4] on a set of benchmark instances with vehicle fleets that are either homogeneous or heterogeneous. In the latter case, a perturbation metaheuristic combines a local descent based on different neighborhood structures with two diversification strategies, namely a randomized construction procedure and a perturbation mechanism where a number of pairs of customers are swapped.

A variant of the problem is finally addressed in [13] in a context where the customers are partitioned into sectors and the private and common carrier fleet size must be determined for each sector.

In this paper, we present a new metaheuristic approach, based on tabu search, for solving the VRPPC with a homogeneous fleet of vehicles. It is empirically demonstrated on a set of benchmark instances that this metaheuristic outperforms the best known approach reported in the literature with regard to solution quality and computation time.

The remainder of the paper is organized as follows. In Section 2, the problem is formally introduced. Then, our problem-solving methodology is described in section 3. Computational results are finally reported in section 4.

2 Problem Formulation

The problem can be stated as follows. Let $G = (V, A)$ be a directed graph, with $V = \{1, 2, ..., n\}$ the vertex set and A the arc set. Vertex 1 is the depot and the other vertices are customers to be visited. A travel cost c_{ij} is defined between each pair of vertices $i, j \in V,\ i \neq j$. Each customer $i \in V\backslash\{1\}$ is characterized by a demand q_i and a penalty cost e_i when it is assigned to the common carrier. A fixed homogeneous private fleet of m vehicles, initially located at the depot, is also available to serve the customers, where each vehicle has capacity Q. Also, a fixed cost f is incurred each time a vehicle from the private fleet is used. The goal of the VRPPC is to design a set of at most m routes for the private fleet such that:

- a vehicle from the private fleet serves a single route that starts and ends at the depot;
- every customer is visited exactly once either by a vehicle of the private fleet or by the common carrier;
- the total demand on each vehicle route of the private fleet should not exceed the capacity Q;
- the sum of travel costs, fixed vehicle costs and common carrier costs is minimized.

The problem can be mathematically formulated as follows [4]. Let:

- x_{ij}^k be 1 if vehicle k visits vertex j immediately after vertex i, 0 otherwise, $i, j = 1, ..., n,\ i \neq j;\ k = 1, ..., m$;
- y_i^k be 1 if vehicle k visits vertex i, 0 otherwise, $i = 1, ..., n;\ k = 1, ..., m$;
- z_i be 1 if customer i is assigned to the common carrier, 0 otherwise, $i = 2, ..., n;\ k = 1, ..., m$;
- $u_i^k \geq 0$ be an upper bound on the load of vehicle k upon leaving customer i, $i = 2, ...n;\ k = 1, ...m$.

We then have:

$$\min \sum_{k=1}^{m} f_k y_1^k + \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} c_{ij} x_{ij}^k + \sum_{i=2}^{n} e_i z_i \tag{1}$$

subject to

$$\sum_{k=1}^{m}\sum_{j=2}^{n} x_{1j}^k = \sum_{k=1}^{m}\sum_{i=2}^{n} x_{i1}^k \leq m, \tag{2}$$

$$\sum_{\substack{j=1\\ j\neq h}}^{n} x_{hj}^k = \sum_{\substack{i=1\\ i\neq h}}^{n} x_{ih}^k, \quad h = 1, ...n;\ k = 1, ...m, \tag{3}$$

$$\sum_{k=1}^{m} y_i^k + z_i = 1, \quad i = 2, ..., n, \tag{4}$$

$$\sum_{i=2}^{n} q_i y_i^k \leq Q, \quad k = 1, ..., m, \tag{5}$$

$$u_i^k - u_j^k + Q x_{ij}^k \leq Q - q_j, \quad i, j = 2, ...n,\ i \neq j;\ k = 1, ..., m. \tag{6}$$

In this formulation, the objective function (1) minimizes the sum of vehicle fixed costs, travel costs and common carrier costs (note that y_1^k is 1 when vehicle k is used, 0 otherwise). Constraints (2) state that at most m vehicles from the private fleet can be used. Constraints (3) specify that the same vehicle must enter and leave a given vertex. Constraints (4) assign each customer either to a vehicle from the private fleet or to the common carrier. Constraints (5) ensure that the vehicle capacity is not exceeded. Finally, constraints (6) eliminate subtours that do not include the depot.

In the following section, a tabu search heuristic for solving this problem is introduced.

3 Tabu search

The problem is solved with a tabu search heuristic based on the unified tabu search framework of Cordeau et al. [7, 8]. The main components of this framework will now be described. We assume that the reader is already familiar with the tabu search metaheuristic, otherwise a detailed presentation is provided in [11]. To simplify the notation, the "vehicle" $m + 1$ stands for the common carrier in the following.

3.1 Search space

During the search, infeasible solutions are allowed. Accordingly, let us define $g(s) = f(s) + \alpha q(s)$, where $f(s)$ is the original objective value of solution s, as defined in equation (1), α is a positive parameter and $q(s)$ is a penalty term for exceeding vehicle capacity, that is

$$q(s) = \sum_{k=1}^{m} \left[\sum_{i=2}^{n} q_i y_i^k - Q \right]^+,$$

with $[x]^+ = \max\{0, x\}$. Clearly, when s is feasible, both $f(s)$ and $g(s)$ coincide, otherwise a positive penalty is incurred.

At each iteration, the value of α is modified by a factor $1 + \delta > 1$. When the current solution is feasible, the value of α is divided by $1 + \delta$, otherwise it is multiplied by $1 + \delta$.

3.2 Initial solution

First, a number of customers are assigned to the common carrier, if the total demand exceeds the capacity of the private fleet. Then, routes are created for the private fleet with the remaining customers. More precisely:

1. If $\sum_{i=2}^{n} q_i > mQ$ then
   1.1 Reindex the customers in increasing order of ratio $\frac{e_i}{q_i}$, $i = 2, ..., n$.
   1.2 Assign the first h customers to the common carrier where
   $$\sum_{i=2}^{h} q_i \geq \sum_{i=2}^{n} q_i - mQ \geq \sum_{i=2}^{h-1} q_i$$
2. Create vehicle routes for the remaining customers through a least-cost insertion heuristic based on the c_{ij}'s (travel costs). This is the starting solution for the improvement step that follows.
3. While there is an improvement do:
   3.1 Apply a first-improvement local descent based on the unified tabu search neighborhood structure, as described in subsection 3.4.
   3.2 Apply a first-improvement local descent based on 4-opt* exchanges to each vehicle route.

Note that for a given selection of arcs, the 4-opt* in step 3.2 implements only 8 of the 48 potential 4-opt exchanges [14]. Furthermore, the selection of arcs is biased toward those that are likely to yield an improvement (see [15], for details).

3.3 Attribute set of a solution

With each solution s is associated an attribute set $B(s) = \{(i,k) \mid i = 2,...,n;\ k = 1,...,m+1$ and customer i is served by vehicle $k\}$. The transition from the current solution s to a solution in the neighborhood then corresponds to the removal and addition of attributes in set $B(s)$.

3.4 Neighborhood

The union of two different neighborhoods is considered. These neighborhoods are:

1. Remove customer $i = 2,...,n$ from vehicle route $k = 1,...,m+1$ and insert it into another route.
2. Exchange customers i in vehicle route k and i' in vehicle route k', with $i,i' = 2,...,n;\ k,\ k' = 1,...,m+1,\ k \neq k'$. That is, customers i and i' are first removed from routes k and k' and inserted at least cost in routes k' and k, respectively.

The second neighborhood is essential to the success of our algorithm. In fact, it is almost impossible to add a customer to another vehicle route through the first neighborhood (apart from the common carrier) when the capacity constraints are tight.

3.5 Tabu duration

A tabu duration is associated with each attribute. Basically, if customer i is removed from route k, attribute (i,k) is tabu and it is forbidden to reinsert customer i in route k for the next θ iterations. The last iteration for which attribute (i,k) is tabu is noted τ_i^k.

3.6 Aspiration criterion

The tabu status of an attribute can be revoked if a transformation leads to a feasible solution which is better than the best feasible solution found with this attribute. The aspiration level of attribute (i,k) is noted σ_i^k.

3.7 Diversification

If a solution in the neighborhood is worse than the current solution, its evaluation is biased to penalize the most frequently added attributes. Let ρ_i^k be the number of times attribute (i, k) has been added to the current solution. The penalty is then proportional to the product of $f(s)$, ρ_i^k and $\sqrt{nm}$, as suggested in [16]. Note that the frequency values ρ_i^k tend to diminish with problem size and the factor $\sqrt{nm}$ compensates for it. A parameter γ is used to adjust the intensity of the diversification.

3.8 Stopping criterion

The search is stopped after a fixed number of iterations.

3.9 Overall algorithm

A step-by-step description of our unified tabu search is provided below. The notation used is summarized in Table 1.

Notation	*Interpretation*
$B(s)$	Attribute set of solution s
$f(s)$	Cost of solution s
$q(s)$	Penalty cost of solution s for overcapacity
$g(s)$	$f(s) + \alpha q(s)$
$N(s)$	Neighborhood of solution s
$M(s)$	Subset of $N(s)$
s	Current solution
$\overline{s}$	Solution in $M(s)$
s'	Best solution in $M(s)$
s^*	Best solution found
α	Penalty parameter for overcapacity
γ	Adjustment parameter for diversification intensity
δ	Update parameter for α
θ	Tabu duration
λ^{max}	Total number of iterations
λ	Iteration counter
ρ_i^k	Number of times attribute (i, k) has been introduced into the solution
σ_i^k	Aspiration level of attribute (i, k)
τ_i^k	Last iteration with tabu status for attribute (i, k)

Table 1: Notation

Pseudo-code

0. Create an initial solution s.
1. If s is feasible then set $s^* \leftarrow s$.
2. Set $\alpha \leftarrow 1$ and $\lambda \leftarrow 0$.
3. For every attribute (i, k) do
   - 3.1 Set $\rho_i^k \leftarrow 0$ and $\tau_i^k \leftarrow 0$.
   - 3.2 If $(i, k) \in B(s)$ and s is feasible then set $\sigma_i^k \leftarrow f(s)$ else set $\sigma_i^k \leftarrow \infty$.
4. While $\lambda \leq \lambda^{max}$ do
   - 4.1 Set $M(s) \leftarrow \emptyset$.
   - 4.2 For each $\overline{s} \in N(s)$ do
     - If $(\tau_i^k < \lambda)$ or ($\overline{s}$ is feasible and $f(\overline{s}) < \sigma_i^k$) for all $(i, k) \in B(\overline{s}) \backslash B(s)$ then set $M(s) \leftarrow M(s) \cup \{\overline{s}\}$.
   - 4.3 For each $\overline{s} \in M(s)$ do
     - If $g(\overline{s}) \geq g(s)$ then
       - Set $\rho^{min} \leftarrow \min_{(i,k) \in B(\overline{s}) \backslash B(s)} \rho_i^k$.
       - Set $h(\overline{s}) \leftarrow g(\overline{s}) + \gamma\sqrt{nm} f(s) \ \frac{\rho^{min}}{\lambda}$.

       else set $h(\overline{s}) \leftarrow g(\overline{s})$.
   - 4.4 Identify a solution $s' \in M(s)$ that minimizes $h(\overline{s})$.
   - 4.5 For each attribute $(i, k) \in B(s) \backslash B(s')$ do
     - Remove customer i in route k using the GENI heuristic.
     - Set $\tau_i^k \leftarrow \lambda + \theta$.
   - 4.6 For each attribute $(i, k) \in B(s') \backslash B(s)$ do
     - Insert customer i in route k using the GENI heuristic.
     - Set $\rho_i^k \leftarrow \rho_i^k + 1$.
   - 4.7 Apply a first-improvement local descent based on 4-opt* exchanges to each modified route in s' to obtain s''.
   - 4.8 If s'' is feasible then
     - If $f(s'') < f(s^*)$ then set $s^* \leftarrow s''$;
     - For each $(i, k) \in B(s'')$, set $\sigma_i^k \leftarrow \min\{\sigma_i^k, f(s'')\}$;
     - Set $\alpha \leftarrow \frac{\alpha}{1+\delta}$;

     else set $\alpha \leftarrow \alpha(1 + \delta)$.
   - 4.9 Set $\lambda \leftarrow \lambda + 1$.

The GENeralized Insertion heuristic (GENI) used in steps 4.5 and 4.6 is described in [10]. Basically, a vertex i is inserted between two of its p closest vertices i' and i'' in a vehicle route ($p = 3$ in our experiments). Although i' and i'' are not necessarily consecutive in the route before the insertion, the sequence i' - i - i'' is part of the route after the insertion, due to a reoptimization based on arc exchanges around i' and i''. The reverse of this procedure is applied when a vertex is removed from a route.

4 Computational Results

The experiments have been performed on benchmark instances with homogeneous fleet produced by Bolduc et al. [4] based on the 14 VRP instances of Christofides and Eilon [5] and the 20 VRP instances of Golden et al. [12]. In these instances, the travel cost between two vertices corresponds to the Euclidean distance. In all cases, the customer coordinates, demands and vehicle capacity are those found in the original instances. The number of vehicles is set to $\frac{0.8q}{Q}$, where q is the total customer demand and Q is the vehicle capacity. The fixed vehicle cost f is set to the average route length in the best known solution for the original instance, rounded to the nearest multiple of 20. The common carrier cost for customer i is set to $\frac{f}{\overline{n}} + \mu_i c_{1i}$ where $\overline{n}$ is the average number of customers per route in the best known solution for the original instance and μ_i equals 1, 1.5 or 2, depending if the demand of customer i is in the lowest, medium or highest customer demands in the instance [4].

All tests were run on a workstation with a 2.2 GHz DualCore Opteron AMD Processor 275. The specifications of this processor are similar to those of the 3.6 GHz Xeon used by Bolduc et al. in [4] for running their RIP heuristic. The latter is compared with our unified tabu search in section 4.3.

4.1 Benchmark instances

The characteristics of the various test instances are shown in Table 2. In this table, $n - 1$ is the number of customers, m is the number of vehicles, and Q and f are the capacity and fixed cost of each vehicle, respectively. The names of the instances begin with CE or G, depending if they come from Christofides and Eilon [5] or Golden et al. [12].

4.2 Parameter values

Preliminary experiments, as well as some guidelines in previous studies on unified tabu search, led to the following parameter values:

Instance	$n-1$	m	Q	f
CE-01	50	4	160	120
CE-02	75	9	140	100
CE-03	100	6	200	140
CE-04	150	9	200	120
CE-05	199	13	200	100
CE-06	50	4	160	140
CE-07	75	9	140	120
CE-08	100	6	200	160
CE-09	150	10	200	120
CE-10	199	13	200	120
CE-11	120	6	200	180
CE-12	100	8	200	120
CE-13	120	6	200	260
CE-14	100	7	200	140
G-01	240	7	550	820
G-02	320	8	700	1060
G-03	400	8	900	1380
G-04	480	8	1000	1720
G-05	200	4	900	1620
G-06	280	5	900	1700
G-07	360	7	900	1460
G-08	440	8	900	1480
G-09	255	11	1000	60
G-10	323	13	1000	60
G-11	399	14	1000	80
G-12	483	15	1000	80
G-13	252	21	1000	60
G-14	320	23	1000	60
G-15	396	26	1000	60
G-16	480	29	1000	60
G-17	240	18	200	40
G-18	300	22	200	60
G-19	360	26	200	60
G-20	420	31	2000	60

Table 2: Benchmark instances

- tabu duration θ: random value in the interval [7, 14];
- adjustment for diversification intensity γ: 0.01;
- update parameter δ (for self-adjusting overcapacity parameter α): 1.25.

4.3 Experiments

The results of the tabu search based on 10,000 and 25,000 iterations are shown in Table 3, along with the results reported in [4] with the RIP heuristic. The latter is a Randomized construction, Improvement and Perturbation heuristic which has produced the best known results on these benchmark instances. For the unified tabu search, we indicate the best and average solution value as well as the average computation time in seconds over 10 independent runs. For RIP, we show the solution value and computation time reported in [4], presumably obtained after a single run. The last row, called *Avg.*, summarizes the results over the 34 instances. In the case of solution values, this overall average is expressed as the average gap in percent over the best known solution, as it seems to be more meaningful than average solution values. The best known solutions correspond to the bold entries in Table 4.

We can see that the tabu search with 10,000 and 25,000 iterations is on average 1.05% and 0.75%, respectively, over the best known solutions. This is to be compared with 1.63% for RIP. Furthermore, the computation time of the tabu search is significantly lower, even when 25,000 iterations are performed.

In [4], the authors also report the best solutions obtained over the course of their experiments. Although, we did not put much effort in this regard, Table 4 reports the best solutions ever produced with our unified tabu search, using different parameter settings. Over the 34 instances, 26 best solutions were produced with our algorithm, as compared with 12 for RIP, including 4 ties.

The performance of our algorithm, with regard to both solution quality and computation time, comes from the exploitation of the unified tabu search framework. Apart from its relative simplicity and the small number of parameters that needs to be tuned, this framework is known to be both effective and efficient. The second neighborhood type where customers are exchanged between vehicle routes is also very important, when capacity constraints are tight.

Instance	*Tabu with 10,000 iterations*			*Tabu with 25,000 iterations*			*RIP*	
	Best sol.	*Avg. sol.*	*Avg. CPU*	*Best sol.*	*Avg. sol.*	*Avg. CPU*	*Sol*	*CPU*
CE-01	1119.47	1119.47	3.4s	1119.47	1119.47	8.5s	1132.91	25s
CE-02	1814.52	1821.38	5.0s	1814.52	1816.07	12.5s	1835.76	73s
CE-03	1926.99	1933.55	14.1s	1924.99	1930.28	34.7s	1959.65	107s
CE-04	2520.63	2531.11	33.4s	2515.50	2526.41	83.3s	2545.72	250s
CE-05	3115.50	3124.79	51.2s	3097.99	3112.25	128.3s	3172.22	474s
CE-06	1207.47	1207.47	4.0s	1207.47	1207.47	9.9s	1208.33	25s
CE-07	2004.53	2011.37	5.6s	2006.52	2010.96	14.0s	2006.52	71s
CE-08	2052.05	2063.06	14.6s	2055.64	2063.06	36.9s	2082.75	110s
CE-09	2429.19	2440.02	32.6s	2429.19	2433.86	83.3s	2443.94	260s
CE-10	3403.94	3412.45	52.9s	3393.41	3402.72	129.6s	3464.90	478s
CE-11	2332.26	2335.75	21.3s	2330.94	2336.59	54.6s	2333.03	195s
CE-12	1955.10	1968.85	9.8s	1952.86	1961.49	24.2s	1953.55	128s
CE-13	2859.12	2864.22	21.3s	2859.12	2863.96	53.7s	2864.21	188s
CE-14	2214.14	2237.45	10.1s	2214.14	2220.23	24.8s	2224.63	110s
G-01	14272.21	14321.55	93.0s	14214.44	14264.33	237.4s	14388.58	651s
G-02	19782.38	19830.36	197.6s	19609.62	19731.28	488.6s	19505.00	1178s
G-03	25427.29	25502.81	372.0s	25271.50	25405.71	942.2s	24978.17	2061s
G-04	35302.68	35404.61	603.9s	35068.47	35182.19	1488.2s	34957.98	3027s
G-05	14614.69	14669.09	134.0s	14486.14	14604.19	341.6s	14683.03	589s
G-06	21860.63	21908.15	227.9s	21690.25	21788.71	551.6s	22260.19	1021s
G-07	24079.87	24239.44	334.1s	24112.79	24169.11	829.5s	23963.36	1628s
G-08	30604.67	30664.12	458.0s	30466.04	30535.30	1131.0s	30496.18	2419s
G-09	1325.33	1328.33	110.5s	1323.57	1324.87	274.7s	1341.17	832s
G-10	1599.51	1605.64	188.2s	1592.93	1598.14	471.2s	1612.09	1294s
G-11	2179.59	2188.48	290.0s	2166.66	2174.45	720.2s	2198.45	2004s
G-12	2509.12	2514.79	427.1s	2490.01	2499.80	1071.4s	2521.79	2900s
G-13	2270.39	2275.63	63.1s	2271.29	2274.13	157.6s	2286.91	802s
G-14	2699.37	2702.48	100.7s	2693.35	2702.50	248.4s	2750.75	1251s
G-15	3163.60	3170.12	152.7s	3157.31	3162.85	377.1s	3216.99	1862s
G-16	3640.69	3650.27	223.6s	3637.52	3645.80	556.4s	3693.62	2778s
G-17	1632.83	1639.66	64.4s	1631.49	1636.84	159.8s	1701.58	806s
G-18	2691.61	2705.40	96.3s	2692.49	2697.16	240.3s	2765.92	1303s
G-19	3457.28	3469.15	136.2s	3452.00	3458.47	340.3s	3576.92	1903s
G-20	4287.89	4297.97	183.4s	4272.98	4279.74	453.8s	4378.13	2800s
Avg.	0.71%	1.05%	139.3s	0.46%	0.75%	346.5s	1.63%	1047s

Table 3: Comparative results

Instance	*Tabu*	*RIP*
CE-01	**1119.47**	**1119.47**
CE-02	**1814.52**	**1814.52**
CE-03	**1920.90**	1937.23
CE-04	**2512.64**	2528.36
CE-05	**3097.67**	3107.04
CE-06	**1207.47**	**1207.47**
CE-07	**2004.53**	2006.52
CE-08	**2052.05**	**2052.05**
CE-09	**2425.26**	2436.02
CE-10	**3386.08**	3407.13
CE-11	**2330.94**	2332.21
CE-12	**1952.86**	1953.55
CE-13	2859.12	**2858.94**
CE-14	**2213.02**	2216.68
G-01	14168.07	**14160.77**
G-02	19479.41	**19234.03**
G-03	24985.36	**24646.79**
G-04	34615.23	**34607.12**
G-05	14409.28	**14249.82**
G-06	**21555.94**	21703.54
G-07	24007.92	**23549.53**
G-08	**30059.25**	30173.53
G-09	**1321.73**	1336.91
G-10	**1583.82**	1598.76
G-11	**2163.34**	2179.71
G-12	**2479.62**	2503.71
G-13	2268.70	**2268.32**
G-14	**2692.46**	2704.01
G-15	**3153.55**	3171.20
G-16	**3631.47**	3654.20
G-17	**1632.83**	1677.22
G-18	**2691.61**	2742.72
G-19	**3442.16**	3528.36
G-20	**4265.11**	4352.95

Table 4: Best solutions

5 Conclusion

This paper has described a unified tabu search for solving a vehicle routing problem with a private fleet and common carrier. The results show that the tabu search performs well and can reach solutions within 1% of the best known solutions in reasonable computation times (i.e., a few minutes). The next step would be to address problems with an heterogeneous fleet of vehicles. However, the heterogeneity issue seriously impacts the solution approach and a distinctive heuristic or metaheuristic scheme needs to be developed to address this variant.

Acknowledgments. Financial support for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC). This support is gratefully acknowledged.

References

[1] Balas E., "The Prize Collecting Traveling Salesman Problem", *Networks 19*, 621-636, 1989.

[2] Bienstock D., M.X. Goemans, D. Simchi-Levi, D. Williamson, "A Note on the Price Collecting Traveling Salesman Problem", *Mathematical Programming 59*, 413–420, 1993.

[3] Bolduc M.-C., J. Renaud, F. Boctor, "A Heuristic for the Routing and Carrier Selection Problem", Technical Report 2006-006, Faculté des Sciences de l'Administration, Université Laval, Québec, Canada, 2006.

[4] Bolduc M.-C., J. Renaud, F. Boctor, G. Laporte "A Perturbation Metaheuristic for the Vehicle Routing Problem with Private Fleet and Common Carriers", Technical Report 2006-014, Faculté des Sciences de l'Administration, Université Laval, Québec, Canada, 2006.

[5] Christofides N. and S. Eilon, "An Algorithm for the Vehicle-Dispatching Problem", *Operations Research 20*, 309–318, 1969.

[6] Chu C.-W., "A Heuristic Algorithm for the Truckload and Less-Than-Truckload Problem", *European Journal of Operational Research 165*, 657-667, 2005.

[7] Cordeau J.-F., M. Gendreau, G. Laporte, "A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems", *Networks 30*, 105-119, 1997.

[8] Cordeau J.-F., G. Laporte, A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows", *Journal of the Operational Research Society 52*, 928-936, 2001.

[9] Diabi M. and R. Ramesh, "The Distribution Problem with Carrier Service: A Dual Based Penalty Approach", *ORSA Journal on Computing 7*, 24-35, 1995.

[10] Gendreau M., A. Hertz, G. Laporte, "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem", *Operations Research 40*, 1086-1094, 1994.

[11] Glover F. and M. Laguna, Tabu Search, Kluwer, Boston, 1997.

[12] Golden B.L., E.A. Wasil, J.P. Kelly, I.-M. Chao, "The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Tests and Computational Results", in T.G. Crainic, G. Laporte (Eds.), *Fleet Management and Logistics*, Kluwer, Boston, 33-56, 1998.

[13] Klincewicz J.G., H. Luss, M.G. Pilcher, "Fleet Size Planning when Outside Carrier Services are Available", *Transportation Science 24*, 169–182, 1990.

[14] Lin S., "Computer Solutions of the Traveling Salesman Problem", *Bell System Technical Journal 44*, 2245–2269, 1965.

[15] Renaud J., F.F. Boctor, G. Laporte, "A Fast Composite Heuristic for the Symmetric Traveling Salesman Problem", *ORSA Journal on Computing 8*, 134–143, 1996.

[16] Taillard É.D., "Parallel Iterative Search Methods for Vehicle Routing Problems", *Networks 23*, 661–673, 1993.

[17] Volgenant T. and R. Jonker, "On Some Generalizations of the Travelling-Salesman Problem", *Journal of the Operational Research Society 38*, 1073–1079, 1987.