A hybrid tabu search for the m-peripatetic vehicle routing problem

Sandra Ulrich NGUEVEU Christian PRINS
Roberto WOLFLER-CALVO
Institut Charles Delaunay - LOSI
Université de Technologie de Troyes (UTT)
12 rue Marie Curie - 10010 Troyes - FRANCE
ngueveus@utt.fr, christian.prins@utt.fr, roberto.wolfler_calvo@utt.fr

29th April 2008

Abstract

This article presents a hybridization of a perfect b-matching within a tabu search framework for the m-Peripatetic Vehicle Routing Problem (m-PVRP). The m-PVRP models for example money transports and cash machines supply where, for security reasons, no path can be used more than once during m periods and the amount of money allowed per vehicle is limited. It consists in finding a set of routes of minimum total cost over m periods from an undirected graph such that each customer is visited exactly once per period and each edge can be used at most once during the m periods. Each route starts and finishes at the depot with a total demand not greater than vehicle capacity. The aim is to minimize the total cost of the routes. The m-PVRP can be considered as a generalization of two well-known NP-hard problems: the vehicle routing problem (VRP or 1-PVRP) and the m-Peripatetic Salesman Problem (m-PSP). Computational results show that the hybrid algorithm obtained improves the tabu search, not only on the m-PVRP in general, but also on the VRP and the m-PSP using classical VRP instances and TSPLIP instances.

KEY WORDS : metaheuristics, tabu search, perfect b-matching, vehicle routing problem, peripatetic salesman problem, peripatetic vehicle routing problem

1 Introduction

The m-Peripatetic Vehicle Routing Problem (m-PVRP), introduced for the first time in (Ngueveu, Prins & Wolfler Calvo 2008), models money collection, transfer and dispatch when it is subcontracted by banks and businesses to specialized companies. These companies need optimized software or applications to organize their vans or trucks routes and schedule. For security reasons, peripatetic and capacity constraints must be satisfied: no path can be used more than once during m periods and the amount of money allowed per vehicle is limited. The m-PVRP is defined on a complete graph $G = (V, E)$ where V is the vertex set and E is edge set. It consists in finding a set of routes of minimum total cost over m periods from an undirected graph such that each customer is visited exactly once per period and each edge can be used at most once during the m periods. Figure 1 shows an example of feasible solution

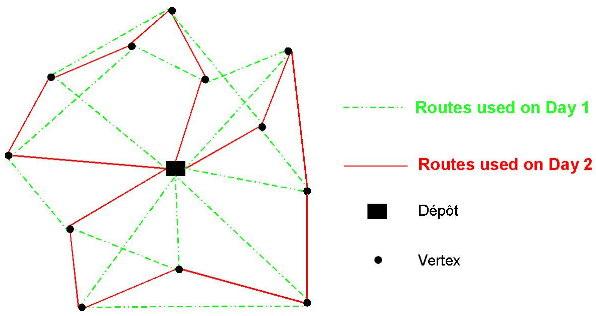

Figure 1: Example of solution for a 2-PVRP

for a 2-PVRP. Ngueveu *et al* introduced the m-PVRP before proposing two lower bounds and two upper bounds. The two lower bounds are based upon k edge-disjoint spanning trees and a perfect b-matching. The first upper bound results from the adaptation of the heuristic from (Clarke & Wright 1964) and the second one is a tabu search with diversification.

The m-PVRP can be considered as a generalization of two well-known NP-hard problems: the vehicle routing problem (VRP) and the m-peripatetic salesman problem (m-PSP). Indeed, the VRP is a particular case of m-PVRP where $m = 1$ since it consists in finding the best routes for one single period. Likewise, any m-PSP is in fact an m-PVRP with an infinite vehicle capacity since the traveling salesman problem (TSP) is a particular case of the VRP with one single vehicle. Both problems were studied in the literature during the last century, with heuristics, metaheuristics and exact methods. The m-PSP for example, introduced in (Krarup 1975), was mainly studied in (De Kort & Volgenant 1994), (Duchenne, Laporte & Semet 2007) and (Wolfler Calvo & Cordone 2003). Amongst the numerous publications concerning the VRP, we can cite the book (Toth & Vigo 2002), a recent survey of the most effective metaheuristics published in (Cordeau, Gendreau, Hertz, Laporte & Sormany 2005), or an effective exact algorithm based on q-route relaxation in (Baldacci, Christofides & Mingozzi 2007).

In this paper we present an efficient algorithm resulting from the hybridization of the perfect b-matching and the tabu search of Ngueveu *et al.* It is designed to solve the m-PVRP. However, due to the lack of publicly available instances for this new problem, the computational analysis was performed using instances of the VRP and the m-PSP to compare with the literature. The remainder of this paper is organized as follows: Section 2 presents the tabu components while Section 3 focusses on the hybridization with a b-matching. Finally, the computational evaluation is presented within Section 4, before the Conclusion.

2 Tabu Search

Tabu search, see for instance (Glover & Laguna 1993), is a method that explores the solutions space by moving from a solution s_t identified at iteration t to the best solution s_{t+1} in the neighborhood $N(s_t)$. Since s_{t+1} may not improve s_t, a tabu mechanism is implemented to prevent the process from cycling over a sequence of solutions. An obvious way to prevent cycles would be to forbid the process from going back to previously encountered solutions, but doing so would typically require excessive bookkeeping. Instead, some attributes of past solutions are recorded and any solution possessing these attributes are discarded for τ iterations. This mechanism is often referred to as short-term memory. Other features like granularity and diversification (long term memory) are often implemented to improve speed and efficiency. The algorithm we designed is stopped after a predefined number of iterations $maxt$ and requires the following components, described hereafter: the initial solution heuristic, the neighborhood structure, the penalization component and the tabu list management.

2.1 Initial solution heuristic and Neighborhood structure

Inspired by the idea of Krarup for the m-PSP (1975), the procedure of Clarke and Wright (1964) is applied m times to obtain at the end an initial m-PVRP solution, and the edges already used are removed from the graph before each iteration. In practice, a penalty is added to the cost of edges already used, forbidding the reuse any of them, unless there is no other alternative. This procedure will be referred to as $Heuristic$.

To explore the solutions space, we try to introduce into the current solution edges that are not currently used during the m periods. Figure 2 illustrates the eight different ways, derived from classical 2-opt moves, to introduce an edge [A, B] within a period. There are consequently $8 \times m$ potential insertion moves per edge. Moves involving two routes are authorized only if capacity constraints are not violated: the total demand on each of the new routes obtained must not exceed the vehicle capacity Q. In addition to the classical 2-opt neighborhood, this neighborhood authorizes moves that split a route in two (see cases 3 and 4 on figure 2) or merge two routes together (if an edge inserted connects the extremities of two routes).

2.2 Penalization and Tabu list management

To allow our algorithm to start from a non-feasible solution, peripatetic constraints are removed and the penalty $\alpha \times max(0, (\sum_{k \in \mathbb{K}} x_e - 1))$ is added to the objective function. Consequently, an edge may be used more than once, during two different periods for example. Within the hybrid tabu search, $\alpha = 2 \times \bar{c}_{max}$ where $\bar{c}_{max}$ is the cost of the most expensive edge of the graph.

To avoid looping back to solutions already visited, after each iteration t, the edges removed from the solution are inserted in the tabu list TL and are declared tabu until iteration $t + \tau$, where τ is the tabu tenure. During each iteration, an unused and non-tabu edge e has to be inserted with the best possible move and the second entering edge e' is free: e' can be tabu or already used in a period of the solution (in this case it will be penalized as explained above). The "partial tabu" algorithm obtained in this way is not very sensitive to parameters settings (value of τ) while preserving the advantages of the tabu list feature (no cycle). We also applied the aspiration criterion, which consists in authorizing a tabu move when the solution obtained is the best found so far.

3 Hybridization with b-matching and Diversification

Define $f(S)$ as the total cost of solution S. Algorithm 1 shows the general structure obtained. The following paragraphs detail the b-matching and the granularity used for its hybridization within the tabu search framework. We also examine the diversification procedure and the way it was interfaced with the granularity. Both components were designed to improve the speed and efficiency of the tabu search.

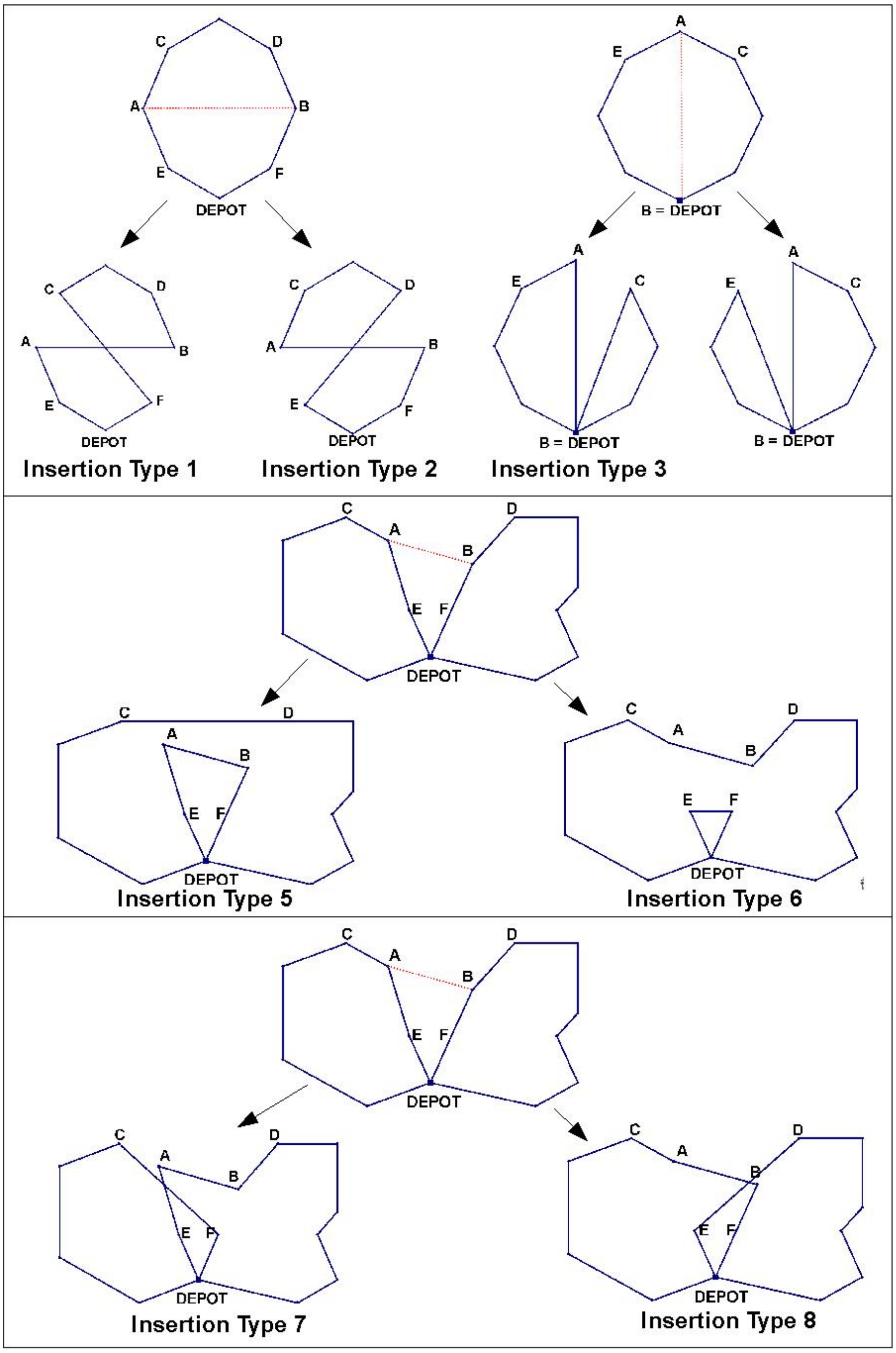

Figure 2: Neighborhood definition: 8 ways to insert edge [A,B] during a period

3.1 b-matching

The b-matching, also known as the b-directed flow problem, was introduced in (Edmonds 1965) within the class of well-solved integer linear problems. Define c_e as the cost of edge e, y_e as the binary variable equal to 1 only if edge e is used, and 0 otherwise. If d_i is the demand of node i and Q is the vehicle capacity, then the minimal number of vehicles per period is $\lambda = \left\lceil \sum_{i \in V} \frac{d_i}{Q} \right\rceil$. The mathematical formulation of the b-matching obtained after relaxing the capacity constraints of the m-PVRP is the following:

$$\min \sum_{e \in E} c_e y_e$$

s. t.

$$\sum_{e \in \delta(i)} y_e = b_i \quad \text{with} \quad b_i = \begin{cases} 2m & \forall i \in \{1...n\} \\ 2m\lambda & \text{if} \quad i = 0 \end{cases}$$

$$y_e \in \{0, 1\}, \qquad \forall e \in E$$

A solution to this problem can be easily computed with a linear programming solver. Preliminary results from (Ngueveu et al. 2008) suggested that the value obtained may be on average less than 10% far from the optimal m-PVRP solution. Therefore, repairing b-matching solutions could lead to potentially good upper bounds. However, extracting an m-PVRP solution from a set of edges is not a straightforward process because it requires to partition the edges between the m periods and the routes. To overcome this difficulty, we hybridize the b-matching with a classical tabu search framework: the result of the exact method guides the metaheuristic in the solutions space.

3.2 Hybridization

Granularity is a concept introduced in (Toth & Vigo 2003), based on the idea of using restricted neighborhoods, allowing only moves that are more likely to produce good feasible solutions. Its classical implementation for the VRP consists in delaying the introduction of long edges into the solution. In our case, the result of the b-matching is used to define the tabu granularity and guides the metaheuristic in the solutions space. The resulting algorithm is a granular tabu search using as candidate list of edges to insert the unused edges that compose the b-matching solution since they have a higher probability of being a part of an optimal solution.

Solving the b-matching produces a set of potentially good edges for the m-PVRP: the cheapest set of edges that satisfy the aggregated degree constraints. However, the small number of edges selected (e.g. 10% for instance B-n45-k7 for the 2-PVRP), leads to a very small candidate list, which induces a tight neighborhood, counter-effective for the metaheuristic efficiency. We found two ways to enlarge this neighborhood without losing the advantage of the b-matching data:

1. Relax the integrality constraints of the b-matching: increasing the number of relevant edges involved, despite the risk of loosing relevance if too many edges are chosen ($y_e > 0$) in the relaxed solution.

2. Complete the b-matching granularity with a short-edge subset: following Toth and Vigo's primary idea, short edges disregarded by the b-matching are added to the candidate list of edges to be inserted into the current solution.

This latter subset is composed of edges with a cost not greater than $\mu \times \bar{c}$ and currently unused, where $\bar{c}$ is the average cost of edges used within the initial solution and μ is a parameter. The penalty applied to infeasible solutions (detailed in section 2.2) has been included in the computation of $\bar{c}_{max}$. The idea behind keeping the penalty in the calculation is that if α was set to 0, the initial infeasible solution may be cheaper than feasible solutions. Therefore, edges included in the candidate list need to be a little more expensive to allow the metaheuristic to find feasible solutions.

The granularity (relaxed b-matching + short-edge subset) is applied every time the best known solution is improved, and removed after $GTSmaxk$ iterations without improving the best solution. During the search, the algorithm oscillates between intensification phases (when granularity is activated: $g = true$) and pseudo-diversification phases (when granularity is removed: $g = false$).

3.3 Diversification procedure

Diversification ensures that the search process is not restricted to a limited portion of the search space. Its classical implementation proposed in (Taillard 2002) penalizes edge costs depending on their frequency of use during the search, since edges used too often retain the algorithm into a region and forbid it to visit the entire solutions space. In spite of that, we do not want to penalize edges used very often because they might be required to reach the optimal solution. Therefore, after a predefined number of iterations, our diversification procedure searches for the best way to insert into the current solution the cheapest edge unused so far. To accommodate this component with the b-matching granularity, the procedure is applied as soon as the following two conditions are satisfied:

1. At least Max_γ iterations have been performed without improving the best solution, since the last removal of the b-matching granularity (described in section 3.2).

2. The previous move applied was not an improving move.

In this way, the diversification component does not disturb the b-matching granularity, but gives a helpful "kick" when necessary.

4 Computational analysis

Computational evaluation was performed on classical VRP and m-PSP benchmark problems to compare with the literature, before being applied on the m-PVRP with $m > 1$. The tests were done on four classes of VRP instances from the literature: A, B, P and vrpnc. Classes A, B and P were proposed in (Augerat 1995) and contain respectively 27, 23 and 23 instances of 19 to 101 nodes. Class vrpnc was proposed in (Christofides, Mingozzi & Toth 1979), and we selected the 7 instances with 50-199 nodes and no additional route length restriction. All VRP instances can be found on the website "http://neo.lcc.uma.es/radi-aeb/WebVRP/". We also used the 5 euclidian TSPLIB instances from (Reinelt 1995) with 17-29 nodes already used for the m-PSP in (Duchenne, Laporte & Semet 2005).

Algorithm 1 Hybrid Tabu Search

```
Heuristic(S)
S' := S
TL := ∅; g := true; t := 1; k := 1; Dec := 1; Freq[e] := 0 ∀e ∈ E
repeat
   FindBestNonTabuSolution(S', TL, g, f(S), Dec)
   if f(S') < f(S) then
      S := S'
      k := 1; γ := 1
      if g = false then
         g := true
      end if
   else
      k := k + 1; γ := γ + 1
      if (k > GTSmaxk) and (g = true) then
         g := false
         k := 1; γ := 1
      end if
      if γ > Max_γ and Dec = −1 then
         Diversify(S', Freq)
      end if
   end if
   UpdateTabuList(TL, τ)
until t > maxt
```

The experiments were performed on an Intel Core 2 Duo personal computer at 1.80 GHz with 2 GB of RAM and running on Windows Vista. Metaheuristics were coded in C, but the linear b-matching solution required for granularity was obtained with the open source software Glpk. The tables of this section report the following columns:

- m: number of periods.
- NbI: number of instances available.
- $LB*$: ratio between the best known lower bound and the best upper bound known, equal to 1 if this upper bound is optimal.
- TS: tabu search with only an aspiration criterion.
- $TS + D$: tabu search with diversification already presented in (Ngueveu et al. 2008).
- HTS: hybrid TS (with b-matching granularity, but without diversification).
- $HTS + D$: hybrid TS including diversification.
- Δ (resp. δ): Average deviation from the optimal solution value (resp. best known upper bound), in percentage.
- s: Average duration in seconds to reach the best solution found.
- sBM: Average computing time of the linear b-matching, in seconds, to obtain the first set of edges for the b-matching granularity.

  If we define TNe as the total number of edges of the initial graph, then:

- $\%Bm = \frac{NBm}{TNe}$: Proportion of edges used by the linear b-matching solution, and used for composing the first set of edges for the granularity (NBm = number of edges used by the linear b-matching solution).

The parameters used for our metaheuristics are the following:

(H)TS(D) Parameters			
Algo	Param	Description	Value
(H)TS(+D)	α	Penalization	$2 \times \bar{c}_{max}$
(H)TS(+D)	$maxt$	Max number of iterations	10000
(H)TS(+D)	τ	Tabu duration	n
HTS(+D)	μ	Proportion of average edge cost	1.30
HTS(+D)	*Setting1*	Max it before granularity is removed	$HTSmaxk = 2n/3$
HTS(+D)	*Setting2*	Max it before granularity is removed	$HTSmaxk = 2n$
(H)TS+D	Max_γ	Max it before diversification	$2n$

We made some preliminary experiments to tune the upper bounds parameters. Preliminary results initially lead to a different HTS setting per problem and per class of instances. To limit the number of settings used, we decided to apply the following HTS settings of the parameters for each problem solved:

- m-PSP, VRP and $2, 3, 5, 6, 7$-PVRP: setting 1
- 4-PVRP: setting 2

In the following section we analyse the performance of our algorithms on the VRP, the m-PSP and the m-PVRP with $m > 1$.

4.1 VRP and m-PSP

Table 1 summarizes the results of our algorithms for the VRP, on the four classes of instances A, B, P and Vrpnc. Computational results show that the metaheuristics designed perform well on this particular problem because average gaps to optimality are around 0.80%. $HTS(+D)$ performs better than $TS(+D)$ on 3 classes of instances out of 4 and the hybridization lowers the average gap to optimality. In addition $HTS + D$ results can be improved if the diversification procedure is activated a bit later on class A or sooner on class B: gap for A = 0.48% if $Max\_\gamma = 3n$ instead of $2n$, and gap for B = 0.89% if $Max\_\gamma = 3n/2$. As expected, the relaxed b-matching is computed very fast (0.28s) and it produces only a small number of edges (4%).

Class of instances	m	NbI	$LB*$	TS		$TS + D$		HTS		$HTS + D$			
				Δ	s	Δ	s	Δ	s	Δ	s	sBm	$\%Bm$
A	1	27	1	0.56	3.28	0.53	2.80	0.54	2.88	0.54	3.43	0.07	0.05
B	1	23	1	0.84	1.97	0.95	2.46	0.96	3.29	0.93	3.77	0.08	0.05
P	1	23	1	0.50	3.63	0.56	2.91	0.47	3.04	0.41	3.45	0.09	0.06
Vrpnc	1	7	-	1.49	12.69	1.22	25.02	1.23	26.02	1.26	17.76	0.88	0.02
Average		80	1	0.85	5.39	0.81	8.30	0.80	8.81	0.78	7.10	0.28	0.04

Table 1. Results for the VRP (1-PVRP)

Table 2 shows the results of our algorithms for the m-PSP, on euclidean TSPLIB instances (Reinelt 1995) already used for assessing m-PSP algorithms in (Duchenne et al. 2005). The metaheuristics designed perform well on this problem because average gaps remain lower than 0.10%. HTS is the best metaheuristic, better than HTS+D, which means that our diversification procedure is happening here too soon. The b-matching selects on average 15% of the edges.

Class of instances	m	NbI	$LB*$	TS Δ	TS s	$TS+D$ Δ	$TS+D$ s	HTS Δ	HTS s	$HTS+D$ Δ	$HTS+D$ s	$HTS+D$ sBm	$HTS+D$ $\%Bm$
bays29	1	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.08
bays29	2	1	1	0.25	0.06	0.25	0.06	0.11	4.31	0.09	0.16	0.14	0.09
fri26	1	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.09	0.10
fri26	2	1	1	0.00	3.28	0.09	0.05	0.00	0.09	0.09	0.05	0.17	0.11
gr17	1	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.13	0.21
gr17	2	1	1	0.08	0.09	0.08	0.20	0.12	0.05	0.08	0.16	0.25	0.21
gr17	3	1	1	0.18	0.12	0.17	0.17	0.18	0.39	0.09	1.25	0.38	0.22
gr17	4	1	1	0.00	1.00	0.00	0.56	0.10	0.17	0.16	0.45	0.50	0.18
gr21	1	1	1	0.00	0.02	0.00	0.02	0.00	0.00	0.00	0.02	0.10	0.12
gr21	2	1	1	0.00	0.41	0.19	1.37	0.00	1.84	0.25	0.06	0.21	0.14
gr21	3	1	1	0.02	2.15	0.02	1.30	0.07	0.20	0.02	2.56	0.30	0.16
gr24	1	1	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.09
gr24	2	1	1	0.00	0.86	0.00	2.93	0.00	0.62	0.00	2.11	0.17	0.13
gr24	3	1	1	0.35	0.39	0.25	0.56	0.27	0.33	0.43	1.47	0.26	0.15
gr24	4	1	1	0.22	0.00	0.22	0.02	0.11	1.53	0.14	1.72	0.35	0.20
Average				0.07	0.56	0.08	0.48	0.06	0.64	0.09	0.67	0.22	0.15
Max				0.35	3.28	0.25	2.93	0.27	4.31	0.43	2.56	0.50	0.22
Min				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.08

Table 2. Results for the m-PSP

4.2 m-PVRP with $2 \leq m \leq 7$

Tables 3 to 6 summarize our results for the m-PVRP with $2 \leq m \leq 7$ on four classes of VRP instances: A, B, P and Vrpnc. Two important preliminary remarks have to be made. First, when m increases, the number of instances available diminishes because there are only n edges connected to the depot, and each route uses two of them. Second, gaps are computed from the best upper bound known, not proven optimal but $LB*$ gives an idea of their efficiency. $LB*$=0.99% suggests that the best upper bound is very close to the optimal value. $LB* = 0.93\%$ means there is a 7% gap between the best upper and lower bounds, but it is not guaranteed that it is the upper bound's fault.

Except for the 4-PVRP, $HTS(+D)$ is the best of the metaheuristic, since it has the lowest gap from best known upper bounds on most problems, except from the 4-PVRP. The relaxed b-matching necessary is still computed very fast (4s for the Vrpnc if $m = 5, 6, 7$) and the percentage of edges used is quite low (14% overall). $HTS + D$ results' can be significantly improved if specific setting of Max_γ is applied: for example, overall average gap of $HTS + D$ on the 2-PVRP can be reduced from 0.98% to 0.91% if the diversification threshold Max_γ is slightly reduced from $2n$ to $3n/2$.

Class of instances	m	NbI	$LB*$	TS Δ	TS s	$TS+D$ Δ	$TS+D$ s	HTS Δ	HTS s	$HTS+D$ Δ	$HTS+D$ s	$HTS+D$ sBm	$HTS+D$ $\%Bm$
A	2	26	0.93	1.11	9.37	0.92	9.53	0.81	8.55	1.15	7.76	0.15	0.10
B	2	23	0.94	1.10	10.27	0.71	11.39	0.68	8.53	0.72	8.86	0.14	0.10
P	2	19	0.97	1.08	10.28	1.03	7.78	1.16	8.44	1.02	10.09	0.16	0.11
Vrpnc	2	7	0.93	1.48	39.87	1.28	55.50	1.25	47.23	1.03	68.44	1.67	0.04
Average		75	0.94	1.20	17.45	0.98	21.05	0.97	18.20	0.98	23.79	0.53	0.09

Table 3. Results for the 2-PVRP

Class of instances	m	NbI	$LB*$	TS		$TS+D$		HTS		$HTS+D$			
				Δ	s	Δ	s	Δ	s	Δ	s	sBm	$\%Bm$
A	3	25	0.97	1.28	14.14	1.05	12.97	1.07	11.30	0.84	13.77	0.20	0.15
B	3	22	0.93	1.94	15.51	1.12	17.27	1.47	14.51	1.02	14.86	0.20	0.15
P	3	14	0.99	0.97	14.23	0.75	13.13	0.88	0.17	0.76	14.08	0.23	0.17
Vrpnc	3	7	0.94	1.06	69.59	0.97	85.80	1.13	67.50	0.98	47.44	2.44	0.07
Average		68	0.96	1.31	28.37	0.97	32.29	1.14	23.37	0.90	22.54	0.77	0.13

Table 4. Results for the 3-PVRP

Class of instances	m	NbI	$LB*$	TS		$TS+D$		HTS		$HTS+D$			
				Δ	s	Δ	s	Δ	s	Δ	s	sBm	$\%Bm$
P	4	8	0.99	0.32	19.28	0.37	11.32	0.43	12.33	0.45	4.75	0.32	0.28
Vrpnc	4	6	0.93	0.69	111.30	0.56	104.92	0.51	158.50	0.53	54.92	3.65	0.09
Average		14	0.96	0.50	65.29	0.46	58.12	0.47	85.41	0.49	29.83	1.98	0.18

Table 5. Results for the 4-PVRP

Class of instances	m	NbI	$LB*$	TS		$TS+D$		HTS		$HTS+D$			
				Δ	s	Δ	s	Δ	s	Δ	s	sBm	$\%Bm$
P	5,6,7	11	0.99	0.40	63.84	0.44	55.04	0.34	61.18	0.40	37.53	0.94	0.22
Vrpnc	5,6,7	10	0.93	1.08	208.73	0.76	187.46	0.57	235.14	0.50	181.40	4.17	0.11
Average		21	0.96	0.69	168.20	0.60	121.25	0.45	148.16	0.45	109.40	2.55	0.16

Table 6. Results for the $5,6,7$-PVRP

5 Conclusion

The partial tabu we designed gives good results not only on the m-Peripatetic Vehicle Routing Problem, but also on two well-known particular cases: the VRP and the m-PSP. Its hybridization with the perfect b-matching through granularity improves significantly the algorithm efficiency, especially when it is adequatly combined with the diversification procedure.

References

Augerat, P. (1995). *Approche polyédrale du problème de tournées de véhicules*, PhD thesis, Institut National Polytechnique de Grenoble, France.

Baldacci, R., Christofides, N. & Mingozzi, A. (2007). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, *Mathematical Programming* (DOI 10.1007/s10107-007-0178-5).

Christofides, N., Mingozzi, A. & Toth, P. (1979). The vehicle routing problem, *in* A. Mingozzi & P. Toth (eds), *Combinatorial Optimization*, Wiley, Wichester, pp. 315–338.

Clarke, G. & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research* **12**: 568–581.

Cordeau, J., Gendreau, M., Hertz, A., Laporte, G. & Sormany, J. (2005). New heuristics for the vehicle routing problem, *in* A. Langevin & D. Riopel (eds), *Logistics systems: design and optimization*, Wiley, pp. 279–298.

De Kort, J. & Volgenant, A. (1994). On the generalized peripatetic salesman problem, *European Journal of Operational Research* **73**: 175–180.

Duchenne, E., Laporte, G. & Semet, F. (2005). Branch and cut algorithms for the undirected m-peripatetic salesman problem, *European Journal of Operational Research* **162**: 700–712.

Duchenne, E., Laporte, G. & Semet, F. (2007). The undirected m-peripatetic salesman problem: Polyhedral results and new algorithms, *Operational Research* **55**(5): 949–965.

Edmonds, J. (1965). Paths, trees, and flowers, *Canadian Journal of Mathematics* **17**: 449–467.

Glover, F. & Laguna, M. (1993). Tabu search, *in* C. R. Reeves (ed.), *Modern heuristic techniques for combinatorial problems*, Blackwell, Oxford, U.K., pp. 70–150.

Krarup, J. (1975). The peripatetic salesman and some related unsolved problems, *in* B. Roy (ed.), *Combinatorial Programming Methods and Applications*, Reidel, Dordrecht, pp. 173–178.

Ngueveu, S., Prins, C. & Wolfler Calvo, R. (2008). Bornes supérieures et inférieures pour le problème de tournées de véhicules m-péripatétiques, *Modélisation, Optimisation et Simulation des Systèmes: Communication, Coopération et Coordination*, Mosim.

Reinelt, G. (1995). http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/.

Taillard, E. (2002). Parallel iterative search methods for vehicle routing problems, *Technical Report G-2002-15*, Les Cahiers du GERAD, Canada.

Toth, P. & Vigo, D. (2002). *The vehicle routing problem*, SIAM, Philadelphia.

Toth, P. & Vigo, D. (2003). The granular tabu search (and its application to the vehicle routing problem), *INFORMS Journal on Computing* **15**(4): 333–346.

Wolfler Calvo, R. & Cordone, R. (2003). A heuristic approach to the overnight security service problem, *Computers and Operations Research* **30**: 1269–1287.