Chapter 11
A Hybrid Tabu Search for the m-Peripatetic Vehicle Routing Problem

Sandra Ulrich Ngueveu, Christian Prins, and Roberto Wolfler Calvo

Abstract This chapter 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 the 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 on classical VRP instances and TSPLIP instances 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.

11.1 Introduction

The m-Peripatetic Vehicle Routing Problem (m-PVRP), introduced for the first time in [12], 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 van or truck routes and schedule. For security reasons, peripatetic and capacity constraints

Sandra Ulrich Ngueveu · Christian Prins · Roberto Wolfler Calvo
Institut Charles Delaunay – LOSI, Universitè de Technologie de Troyes (UTT), Troyes, France
e-mail: {ngueveus,christian.prins,roberto.wolfler_calvo}@utt.fr

V. Maniezzo et al., (eds.), *Matheuristics*, Annals of Information Systems 10,
DOI 10.1007/978-1-4419-1306-7_11,

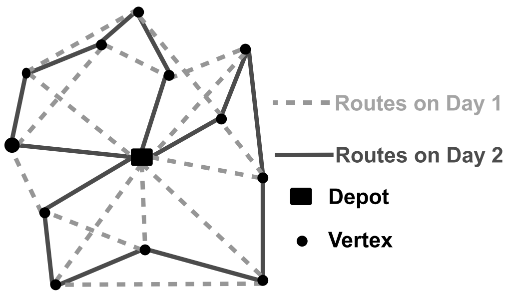

Fig. 11.1 Example of a solution for a 2-PVRP.

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 the 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 11.1 shows an example of a feasible solution 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 Clarke-Wright heuristic [4] and the second from 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 widely studied in the literature with heuristics, metaheuristics and exact methods. The m-PSP, e.g., was introduced by Krarup [11] and mainly studied in [6, 8, 16]. Amongst the numerous publications concerning the VRP, we can cite Toth and Vigo [14], a recent survey of the most effective metaheuristics for VRPs [5], or an effective exact algorithm based on q-route relaxation [2].

In this chapter 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 11.2 presents the tabu components, while Section 11.3 focuses on the hybridization with a b-matching. Finally, the computational evaluation is presented in Section 11.4, before the conclusion.

11.2 Tabu Search

Tabu search [10] is a method that explores the solution 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 solutions 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.

11.2.1 Initial Solution Heuristic and Neighborhood Structure

Inspired by the idea of Krarup for the m-PSP [11], the procedure of Clarke and Wright [4] 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 of any of them, unless there is no other alternative. This procedure will be referred to as *Heuristic*.

To explore the solution space, we try to introduce into the current solution edges that are not currently used during the m periods. Figure 11.2 illustrates the eight different ways, derived from classical 2-opt moves, to introduce an edge [A, B] within a period. There are consequently $8m$ potential insertion moves per edge. Moves involving two routes are authorized only if the 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 11.2) or merge two routes if an edge inserted connects the extremities of two routes.

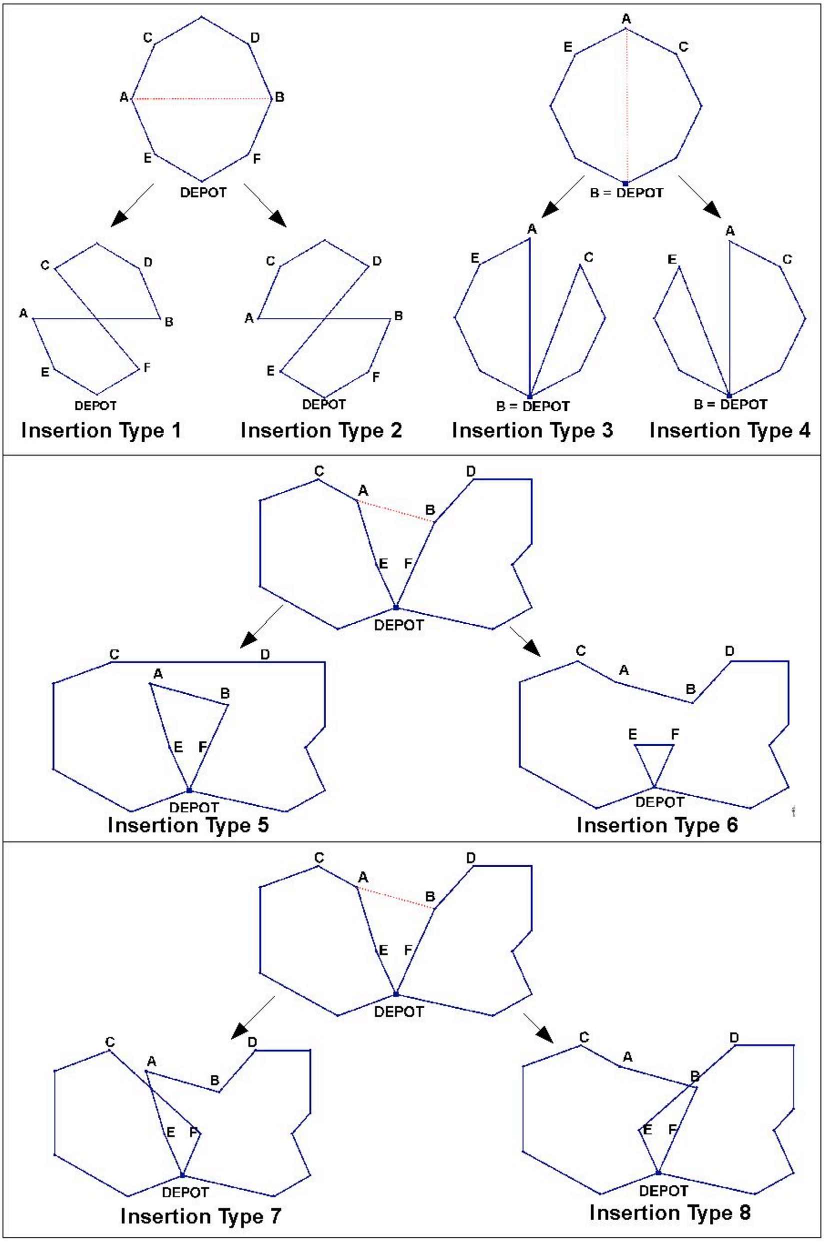

Fig. 11.2 Neighborhood definition: eight ways to insert edge [A,B] during a period.

11.2.2 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 or more different periods. Within the hybrid tabu search, we set α to $2\bar{c}_{max}$ where $\bar{c}_{max}$ is the cost of the most expensive edge of the graph.

To avoid going back to already visited solutions, 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 be already used in a period of the solution, in which case it will be penalized as explained above. The "partial tabu" algorithm obtained in this way is not very sensitive to the value of τ while it avoids cycling. We also applied an aspiration criterion, which consists in authorizing a tabu move when the solution obtained is the best found so far.

11.3 Hybridization with *b*-Matching and Diversification

Hybridization can in our context consist either in using information provided by an exact method to guide the metaheuristic, or in combining the features of two metaheuristics to obtain a more efficient procedure. The hybridization of b-matching with tabu search, as explained in Section 11.3.2, and the diversification procedure, detailed in Section 11.3.3, both improved the speed and efficiency of the tabu search designed for the m-PVRP.

11.3.1 b-Matching

The b-matching problem, also known as the b-directed flow problem, was introduced by Edmonds [9] 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 $\lambda = \left\lceil \frac{1}{Q} \sum_{i \in V} d_i \right\rceil$. The mathematical formulation of the b-matching obtained after relaxing the capacity constraints of the m-PVRP is as follows:

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

s. t.

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

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

A solution to this problem can be easily computed with a linear programming solver. Preliminary results from [12] suggested that the value obtained may be on average about 10% less than 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 tabu search algorithm: the result of the exact method guides the metaheuristic in the solution space.

11.3.2 Hybridization

Granularity is a concept introduced in [15], based on the idea of using restricted neighborhoods. It allows only moves that, based on some criterion, are more likely to produce good feasible solutions. Its 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 solution space. The resulting algorithm is a granular tabu search that uses as candidate list the unused edges that are in the b-matching solution; these edges have a higher probability of being 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, a small number of edges tends to be selected (e.g. 10% for instance B-n45-k7 for the 2-PVRP). This leads to a very small candidate list, which induces a small neighborhood size, 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: this increases the numbers of edges selected by edges that still have a higher probability than others to be in an optimal 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 that have a cost not greater than $\mu\bar{c}$ and are currently unused; $\bar{c}$ is the average cost of edges used within the initial solution and μ is a parameter. The penalty applied to infeasible solutions (see Section 11.2.2) has been included in the computation of $\bar{c}$. 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 plus short-edge subset) is applied every time the best 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: $g = false$).

11.3.3 Diversification Procedure

Diversification ensures that the search process is not restricted to a limited portion of the search space. An example of implementation, as explained in [13], penalizes edge costs depending on their frequency of use during the search. For the m-PVRP, we do not want to penalize edges used very often because they might be required to reach an optimal solution. Instead, 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 the previous subsection).
2. The previous move applied was not an improving move.

The diversification component applied in this way does not disturb the b-matching granularity, but gives a helpful "kick" when necessary. Let $f(S)$ be the total cost of solution S, algorithm 1 summarizes the hybrid tabu search with diversification designed for the m-PVRP.

11.4 Computational Analysis

A computational evaluation was first performed on classical VRP and m-PSP benchmark problems to compare our results with the literature; next we applied our algorithms to 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 [1] contain 27, 23 and 23 instances, respectively, of 19 to 101 nodes. From class vrpnc [3] we selected the seven instances with 50-199 nodes and

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
        γ := γ + 1
        if g = true then
            k := k + 1
            if k > GTSmaxk then
                g := false
                k := 1; γ := 1
            end if
        end if
        if γ > Max_γ and Dec = −1 then
            Diversify(S', Freq)
        end if
    end if
    UpdateTabuList(TL, τ)
until t > maxt
```

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 five Euclidian instances from TSPLIB (`http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`) with 17 to 29 nodes that were already used for the m-PSP in [7].

The experiments were performed on an Intel Core 2 Duo personal computer at 1.80 GHz with 2 GB of RAM running 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 compare four variants of our algorithms, the basic tabu search algorithm (TS), the tabu search algorithm with the diversification component (TS+D), the tabu search algorithm hybridized with b-matching (HTS) and the latter further enhanced by the diversification component (HTS+D). These algorithms are tested on the VRP, the m-PSP and the m-PVRP with $m > 1$.

Some preliminary experiments were made to tune the parameters of the upper bounding procedures. Preliminary results led 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:

Table 11.1 Parameter Settings

Algo	Param	Description	Value
(H)TS(+D)	α	Penalization	$2\bar{c}_{max}$
(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)	$HTSmaxk_1$	Max it before granularity is removed (using Setting 1)	$2n/3$
HTS(+D)	$HTSmaxk_2$	Max it before granularity is removed (using Setting 2)	$2n$
(H)TS+D	Max_γ	Max it before diversification	$2n$

1. VRP and m-PSP with $m = 2, 3, 5, 6, 7$: Setting 1 (using $HTSmaxk_1$)
2. 4-PVRP: Setting 2 (using $HTSmaxk_2$)

As listed in Table 11.1, Settings 1 and 2 only differ in the value of the parameter $HTSmaxk$ while all other parameters remain at a fixed value. Once set up as previously explained, each algorithm is run only once per instance. All algorithms are deterministic, but the results presented in the subsequent sections are aggregated per instance class to avoid extensive tables of results.

11.4.1 VRP and m-PSP

Table 11.2 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 three of four instance classes and the hybridization lowers the average gap to optimality. $HTS + D$ results on the VRP can be further 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%).

Table 11.3 shows the results of our algorithms for the m-PSP on Euclidean TSPLIB instances already used for assessing m-PSP algorithms in [7]. Our metaheuristics perform well on this problem because average gaps remain lower than 0.10%. HTS is the best algorithm, better than $HTS + D$, which means that our diversification procedure is used here too soon. The b-matching selects on average 15% of the edges.

Table 11.2 Results for the VRP (1-PVRP); m is the number of periods; NbI is the number of instances available; LB^* is the ratio between the best known lower and upper bounds, which is equal to 1 if both are optimal; Δ (resp. δ) is the average percentage deviation from the optimal solution value (resp. best known upper bound) for each instance class; σ is the standard deviation of Δ (resp. δ); s is the average duration in seconds to reach the best solution found; sBM is the average computing time of the linear b-matching, in seconds, to obtain the first set of edges for the b-matching granularity; and $Bm = NBm/TNe$ is the 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, TNe = total number of edges of the initial graph).

instance class	m	NbI	LB^*	TS Δ	TS σ	TS s	$TS+D$ Δ	$TS+D$ σ	$TS+D$ s	HTS Δ	HTS σ	HTS s	$HTS+D$ Δ	$HTS+D$ σ	$HTS+D$ s
A	1	27	1	0.56	0.76	3.28	**0.53**	0.73	2.80	**0.54**	0.50	2.88	**0.54**	0.57	3.43
B	1	23	1	**0.84**	1.47	1.97	0.95	1.49	2.46	0.96	1.50	3.29	0.93	1.45	3.77
P	1	23	1	0.50	0.56	3.63	0.56	0.61	2.91	0.47	0.54	3.04	**0.41**	0.53	3.45
vrpnc	1	7	-	1.49	1.71	12.69	**1.22**	1.52	25.02	**1.23**	1.37	26.02	1.26	1.85	17.76
Average		80	1	0.85	1.12	5.39	0.81	1.10	8.30	0.80	0.98	8.81	**0.78**	1.10	7.10

	m	Bm	sBm
A	1	0.05	0.07
B	1	0.05	0.08
P	1	0.06	0.09
vrpnc	1	0.02	0.88
Average		0.04	0.28

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

Tables 11.4 to 11.7 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

Table 11.3 Results for the m-PSP; for an explanation of the table entries, we refer to the caption of Table 11.2.

instance class	m	NbI	$LB*$	TS Δ	TS s	$TS+D$ Δ	$TS+D$ s	HTS Δ	HTS s	$HTS+D$ Δ	$HTS+D$ s	sBm	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
σ				0.12		0.10		0.08		0.12			

Table 11.4 Results for the 2-PVRP; for an explanation of the table entries, we refer to the caption of Table 11.2.

instance class	m	NbI	LB^*	TS δ	σ	s	$TS+D$ δ	σ	s	HTS δ	σ	s	$HTS+D$ δ	σ	s
A	2	26	0.97	1.11	0.63	9.37	0.92	0.60	9.53	**0.81**	0.56	8.55	1.15	0.85	7.76
B	2	23	0.98	1.10	1.14	10.27	0.71	0.56	11.39	**0.68**	0.73	8.53	0.72	0.66	8.86
P	2	19	0.98	1.08	0.83	10.28	**1.03**	0.61	7.78	1.16	0.86	8.44	**1.02**	0.74	10.09
vrpnc	2	7	0.95	1.48	0.67	39.87	1.28	0.64	55.50	1.25	0.76	47.23	**1.03**	0.78	68.44
Average		75	0.97	1.20	0.82	17.45	**0.98**	0.60	21.05	**0.97**	0.73	18.20	**0.98**	0.76	23.79

	m	Bm	sBm
A	2	0.10	0.15
B	2	0.10	0.14
P	2	0.11	0.16
vrpnc	2	0.04	1.67
Average		0.09	0.53

available decreases 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. These are not proven to be optimal but LB^* gives an idea of their quality. LB^*=0.99% suggests that the best upper bound is very close to the optimal value. $LB* = 0.95\%$ means there is a 5% gap between the best known upper and lower bounds.

Figure 11.3 shows the evolution of the percentage deviation from the best known solutions over time for vrpnc instances. It suggests that the dominance of $HTS + D$ over the three other algorithms is reinforced when m increases. This remark is confirmed by most tables of results: $HTS(+D)$ is the best performing of the algorithms since it has the lowest gap from the best known upper bounds on most instances, except for those of 4-PVRP. The relaxed b-matching necessary is still computed very fast (four seconds 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 a specific setting of Max_γ is applied: e.g., 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 $3n/2$.

11.5 Conclusion

The partial tabu algorithm we designed gives good results not only on the m-Peripatetic Vehicle Routing Problem, but also on two well-known special 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 adequately combined with the diversification procedure.

Table 11.5 Results for the 3-PVRP; for an explanation of the table entries, we refer to the caption of Table 11.2.

instance class	m	NbI	LB^*	TS δ	TS σ	TS s	$TS+D$ δ	$TS+D$ σ	$TS+D$ s	HTS δ	HTS σ	HTS s	$HTS+D$ δ	$HTS+D$ σ	$HTS+D$ s
A	3	25	0.98	1.28	1.20	14.14	1.05	0.79	12.97	1.07	0.88	11.30	**0.84**	0.85	13.77
B	3	22	0.98	1.94	1.91	15.51	1.12	0.91	17.27	1.47	1.35	14.51	**1.02**	0.98	14.86
P	3	14	0.99	0.97	0.45	14.23	**0.75**	0.33	13.13	0.88	0.49	0.17	**0.76**	0.36	14.08
vrpnc	3	7	0.95	1.06	0.54	69.59	**0.97**	0.76	85.80	1.13	0.76	67.50	**0.98**	0.68	47.44
Average		68	0.97	1.31	1.02	28.37	0.97	0.70	32.29	1.14	0.87	23.37	**0.90**	0.72	22.54

	m	Bm	sBm
A	3	0.15	0.20
B	3	0.15	0.20
P	3	0.17	0.23
vrpnc	3	0.07	2.44
Average		0.13	0.77

Table 11.6 Results for the 4-PVRP, for an explanation of the table entries, we refer to the caption of Table 11.2.

instance class	m	NbI	LB^*	TS δ	TS σ	TS s	$TS+D$ δ	$TS+D$ σ	$TS+D$ s	HTS δ	HTS σ	HTS s	$HTS+D$ δ	$HTS+D$ σ	$HTS+D$ s
P	4	8	0.99	**0.32**	0.24	19.28	0.37	0.33	11.32	0.43	0.19	12.33	0.45	0.20	4.75
vrpnc	4	6	0.96	0.69	0.31	111.30	0.56	0.30	104.92	**0.51**	0.34	158.50	**0.53**	0.29	54.92
Average		14	0.97	0.50	0.27	65.29	**0.46**	0.31	58.12	**0.47**	0.26	85.41	0.49	0.24	29.83

	m	Bm	sBm
P	4	0.28	0.32
vrpnc	4	0.09	3.65
Average		0.18	1.98

Table 11.7 Results for the m-PVRP with $m = 5, 6, 7$; for an explanation of the table entries, we refer to the caption of Table 11.2.

instance class	m	NbI	LB^*	TS δ	TS σ	TS s	$TS+D$ δ	$TS+D$ σ	$TS+D$ s	HTS δ	HTS σ	HTS s	$HTS+D$ δ	$HTS+D$ σ	$HTS+D$ s
P	5,6,7	11	0.99	0.40	0.24	63.84	0.44	0.31	55.04	**0.34**	0.24	61.18	0.40	0.24	37.53
vrpnc	5,6,7	10	0.96	1.08	0.92	208.73	0.76	0.43	187.46	0.57	0.47	235.14	**0.50**	0.46	181.40
Average		21	0.97	0.69	0.58	168.20	0.60	0.37	121.25	**0.45**	0.35	148.16	**0.45**	0.35	109.40

	m	Bm	sBm
P	4	0.22	0.94
vrpnc	4	0.11	4.17
Average		0.16	2.55

References

1. P. Augerat. *Approche polyédrale du problème de tournées de véhicules.* PhD thesis, Institut National Polytechnique de Grenoble, France, 1995.
2. R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008.
3. N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and L. Sandi, editors, *Combinatorial Optimization*, pages 315–338. Wiley, Chichester, UK, 1979.

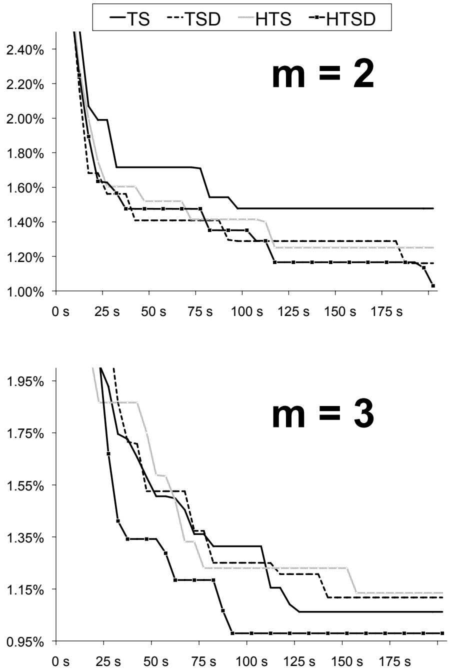

Fig. 11.3 Evolution of δ over time (in seconds) on the vrpnc instances.

4. G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.
5. J.F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.S. Sormany. New heuristics for the vehicle routing problem. In A. Langevin and D. Riopel, editors, *Logistics systems: design and optimization*, pages 279–298. Wiley, 2005.
6. J.B.J.M. De Kort and A. Volgenant. On the generalized peripatetic salesman problem. *European Journal of Operational Research*, 73:175–180, 1994.
7. E. Duchenne, G. Laporte, and F. Semet. Branch and cut algorithms for the undirected m-peripatetic salesman problem. *European Journal of Operational Research*, 162:700–712, 2005.
8. E. Duchenne, G. Laporte, and F. Semet. The undirected m-peripatetic salesman problem: Polyhedral results and new algorithms. *Operations Research*, 55(5):949–965, 2007.
9. J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
10. F. Glover and M. Laguna. Tabu search. In C.R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 70–150. Blackwell, Oxford, UK, 1993.
11. J. Krarup. The peripatetic salesman and some related unsolved problems. In B. Roy, editor, *Combinatorial Programming Methods and Applications*, pages 173–178. Reidel, Dordrecht, 1975.
12. S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. Bornes supérieures et inférieures pour le problème de tournées de véhicules m-péripatétiques. In *Actes de la 7ème conférence internationale de Modélisation et Simulation (MOSIM)*, volume 3, pages 1617–1625, Paris, France, 2008.

13. E.D. Taillard. Parallel iterative search methods for vehicle routing problems. Technical Report G-2002-15, Les Cahiers du GERAD, Canada, 2002.
14. P. Toth and D. Vigo. *The vehicle routing problem*. SIAM, Philadelphia, 2002.
15. P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.
16. R. Wolfler Calvo and R. Cordone. A heuristic approach to the overnight security service problem. *Computers & Operations Research*, 30:1269–1287, 2003.