CRT-95-74

AN ADAPTIVE MEMORY HEURISTIC FOR A CLASS OF VEHICLE ROUTING PROBLEMS WITH MINMAX OBJECTIVE

by
Bruce L. Golden[1]
Gilbert Laporte[2]
Éric D. Taillard[2]

Financial support for this research was provided by the Natural Sciences and Engineering Research Council of Canada (N.S.E.R.C.).

[1] Department of Management Science and Statistics, College of Business and Management, University of Maryland, College Park, MD 20742, U.S.A.
[2] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale “Centre-ville”, Montréal, Canada H3C 3J7

Centre de recherche sur les transports – Publication CRT-95-74

November 1995

ABSTRACT

We propose a heuristic for a class of vehicle routing problems (VRPs) with minmax objective. These include the Capacitated VRP, the Capacitated VRP with multiple use of vehicles, and the $m-$Traveling Salesman Problem with multiple use of vehicles. A tabu search based adaptive memory procedure of instances indicate that the method produces very good solutions within reasonable computing times.

Key words : Vehicle routing problem, multiple use of vehicles, minmax objective, tabu search heuristic, adaptive memory procedure.

RÉSUMÉ

On propose une heuristique pour une classe de problèmes de tournées de véhicules (PTV) avec objectif minmax. Parmi ces problèmes se retrouvent les PTV avec capacité, les PTV avec capacité et utilisation multiple de véhicules, ainsi que le problème des m voyageurs de commerce avec utilisation multiple de véhicules. On développe une méthode de recherche avec tabous imbriquée dans une procédure de recherche adaptative. Des tests numériques réalisés sur plusieurs problèmes indiquent que la méthode proposée produit de très bonnes solutions moyennant des temps de calcul raisonnables.

Mots-clefs : Problème de tournées de véhicules, utilisation multiple de véhicules, objectif minmax, heuristique de recherche avec tabous, procédure de mémoire adaptative.

1. Introduction

We propose a heuristic for a class of routing problems with minmax objective. Formally the problems are defined on a directed graph $G = (V, A)$, where $V = \{v_0, v_1, ..., v_n\}$ is the vertex set and $A = \{(v_i, v_j) : i \neq j\}$ is the arc set. Vertex v_0 represents a *depot* at which are based a pre-specified number m of identical vehicles of capacity Q, while the remaining vertices correspond to *customers*. A *distance* or *travel time* matrix $C = (c_{ij})$ is defined on A.

In the classical *Capacitated Vehicle Routing Problem* (CVRP), each customer v_i has a nonnegative demand q_i. The problem consists of determining a set of m vehicle routes of minimum total length, each starting and ending at the depot, such that each customer is visited exactly once and the total demand of any route does not exceed Q. For a recent overview of algorithms for the CVRP, see Laporte [1]. A special case of the CVRP is the $m-$*Traveling Salesman Problem* $(m - \mathrm{TSP})$ in which Q is arbitrarily large. In both these problems, it is implicitly assumed that each vehicle is used for only one route. In some cases, however, multiple use of vehicles is allowed. This gives rise to the *CVRP with Multiple Use of Vehicles* (CVRPM) and to the $m-$*TSP with Multiple Use of Vehicles* $(m - \mathrm{TSPM})$ (see, e.g., Fleischmann [2] and Taillard, Laporte and Gendreau [3]). In both the CVRPM and the $m - \mathrm{TSPM}$, several vehicle routes can be assigned to the same vehicle to form a working day, but the total distance driven by any vehicle may not exceed a preset limit L. Note that if $c_{ij} \leq c_{oi} + c_{jo}$ for all i and j, then the $m - \mathrm{TSPM}$ and the $m - \mathrm{TSP}$ coincide since it is never disadvantageous to combine two routes.

There exist a number of contexts where, for equity reasons, it is important to minimize the length of the *largest distance* traveled by any vehicle. Any of the four routing problems described above can be defined with a minmax objective. The minmax versions of the $m - \mathrm{TSPM}$ and of the $m - \mathrm{CVRPM}$ consist in fact of determining the smallest feasible value of L. Relatively little attention has been paid to minmax routing problems. Some simple heuristics with worst-case behavior ratios of $m + (m/2)\log n$ and $5/2 - 1/m$ are provided by Frederickson, Hecht and Kim [4] in the case of the minmax $m - \mathrm{TSP}$. The same problem is solved by means of a tabu search heuristic and of exact dichotomous search scheme by França et al. [5]. To our knowledge, the minmax CVRP, $m - \mathrm{TSPM}$ and CVRPM have not yet been solved.

Our purpose is to propose a tabu search based heuristic for these minmax routing problems. We will first describe the algorithm for the minmax CVRP, its application to the minmax $m - \mathrm{TSP}$ being straightforward. Then, we will show how this algorithm can be extended to solve the minmax version of the CVRPM. This will be done in Sections 2 and 3, respectively, followed by computational results in Section 4 and by the conclusion in Section 5.

2. Heuristic for the Minmax $m-$TSP and CVRP

Tabu search has been applied successfully by a number of researchers to the solution of the classical CVRP and some of its variants. Some of the best implementations are those of Taillard [6], of Gendreau, Hertz and Laporte [7] and of Rego and Roucairol [8]. Recently Taillard has proposed an *Adaptive Memory Procedure* (AMP) that can be used to enhance the search in several local search algorithms. Basically, AMP operates by first producing several solutions by means of a heuristic, and then combines these solutions to produce new ones, using probabilistic rules. This is similar in some ways to what is done in genetic algorithms, except that offspring can be generated from more than two parents. When used in conjunction with tabu search, AMP produces a strong diversification and intensification effect that enriches the search and directs it towards high quality solutions. An application of this method to the CVRP and to the *Vehicle Routing Problem with Time Windows* is presented in Rochat and Taillard [9].

In order to describe how AMP was applied to the minmax CVRP we need to first summarize Taillard's [6] tabu search heuristic for the vehicle routing problem with capacity and distance restrictions, and some modifications that were made to this algorithm for the minmax CVRP.

The algorithm starts from partial initial solution containing up to m feasible routes and possibly some unserviced customers. In order to favor solutions containing as many routed customers as possible, the solution cost is increased by a penalty term equal to the sum of all lengths of return trips between unrouted customers and the depot. The problem is then decomposed into one or several subproblems for which between three and five vehicle routes should be constructed. A locally optimal solution is then determined for each subproblem using a tabu search mechanism with one of the following four operations: swapping two customers between two different routes; moving a routed customer to a different route; inserting an unrouted customer into a route; removing a customer from an existing route. At all times, feasibility with respect to capacity and distance is maintained. Periodically, a new decomposition into subproblems is attempted and the process ends after a set number of iterations.

The original Taillard algorithm can be adapted to solve the minmax $m - \mathrm{TSP}$ and CVRP. Even if a minmax objective is used, the algorithm attempts to reduce at all times the total route length and records the length $\overline{z}$ of the longest route. The value of $\overline{z}$ is used as a constraint on the length of vehicle routes produced during the course of the algorithm. Initially $\overline{z}$ is set equal to infinity but, as the algorithm may be called several times within the AMP, a finite value of $\overline{z}$ may be known, even at the initialization phase. The modified Taillard algorithm is an improvement mechanism applied to an initial solution. It is similar to the standard Taillard procedure, except for the following. After each decomposition, the value of $\overline{z}$ is updated and acts as a constraint for the creation or modification of routes. Moreover, if at some point the number of routes in a subproblem goes down by 1, then the longest route R in that subproblem is split into two routes of more or less equal lengths. This is achieved by selecting the best of all solutions obtained by removing an arc pair $\{(v_{i_1}, v_{i_2}), (v_{j_1}, v_{j_2})\}$ in the route $R = (v_0, ..., v_{i_1}, v_{i_2}, ..., v_{j_1}, v_{j_2}, ..., v_0)$ and introducing arc (v_{i_1}, v_{j_2}) to create the first route, and arcs (v_0, v_{i_2}), (v_{j_1}, v_0) to create the second route.

To generate an initial solution, we propose the following "modified Clarke and Wright procedure" (see [10] for the original description). The idea is to try and construct m vehicle routes whose maximal length is as close as possible to a known lower bound on the minmax cost. To achieve this, several trials may be necessary. Also, in constrast with the original Clarke and Wright algorithm, more effort is expended on cost minimization while constructing the routes. Ideally, the procedure should end with at most $m+1$ routes. If it contains more than m routes, the first m routes are preserved and the remaining ones are eliminated. Their customers are then inserted into the first m routes during the improvement phase. Here is a description of the proposed modified Clarke and Wright procedure.

<u>Step 1</u> (Initialization). Set $\tilde{z} := \underline{z} = \max\limits_{j} \{c_{oj} + c_{j_0}\}$.

<u>Step 2</u> (Radial routes). Create n return routes between the depot and customers v_j.

<u>Step 3</u> (Route merge). Consider all pairs of routes $R_s = (v_0, v_{i_1}, ..., v_{i_u}, v_0)$ and $R_t = (v_0, ..., v_{j_\ell}, v_{j_{\ell+1}}, ..., v_0)$ and the merged route $R_{st} = (v_0, ..., v_{j_\ell}, v_{i_1}, ..., v_{i_u}, v_{j_{\ell+1}}, ...v_0)$ obtained by inserting R_s between two consecutive customers v_{j_ℓ} and $v_{j_{\ell+1}}$ of R_t. Denoting by $c(R)$ the cost of route R, compute $\max\limits_{s \neq t} \{\theta[c(R_s) + c(R_t) - c(R_{st})]\}$, where θ is a "noise factor" drawn randomly from $[0.9, 1.1]$ according to a continuous uniform distribution for each pair of indices s and t. If no feasible improvement is possible, go to Step 4. (A feasible route must have a demand not exceeding Q and a cost not exceeding $\tilde{z}$). Otherwise implement the best feasible merge, post-optimize the merged route using 2–opt, and repeat this step.

<u>Step 4</u> (Termination check). If the number of routes in the solution does not exceed $m+1$ or if $\tilde{z} > 1.5\,\underline{z}$, stop. Otherwise, set $\tilde{z} := 1.02\,\tilde{z}$ and go to Step 2.

Here, two comments are in order. First, in Step 3, the noise variable θ acts as a diversification factor as it helps generate different merged routes at each application of this step. Second, the termination check $\tilde{z} > 1.5\,\tilde{z}_0$ in Step 4 is only imposed as a precautionary measure. It has never been activated in any of our tests.

The AMP for the minmax $m - \mathrm{TSP}$ or CVRP can now be described. It uses three user controlled parameters p, q and r.

Step 1 (Initial solutions). Set $\overline{z} := \infty$ and repeat the following operations p times:

i) Generate an initial solution as in the modified Clarke and Wright procedure.

ii) If the solution contains at most m routes, let $\overline{z}$ be the minmax cost. If in addition $\overline{z} = \max\limits_{j} \{c_{oj} + c_{jo}\}$, stop with an optimal solution.

iii) Apply the improvement phase of the modified Taillard algorithm with q applications of the decomposition process.

iv) If all customers belong to a route, let $\overline{z}$ be the minmax cost. If in addition $\overline{z} = \max\limits_{j} \{c_{oj} + c_{jo}\}$, stop with an optimal solution.

v) Insert the routes in the adaptive memory.

vi) Remove from the list any route whose length is at least equal to $\overline{z}$.

Step 2 (Generation of new solutions). Repeat the following operations r times. Initially, no route has been selected.

i) If less than m routes have already been selected, randomly select a route from the list by favoring routes having a small label.

ii) Disregard all routes having vertices in common with those already selected; if some routes remain, go to i).

iii) Apply operations ii) to vi) of Step 1.

In our implementation, we used $p = 10$ and $q = 5$. The value of r is problem dependent and is reported in Table 2.

3. Heuristic for the minmax CVRPM

The algorithm we have developed for the minmax CVRPM consists of four steps. The first two steps are almost identical to Step 1 and 2 of the minmax CVRP algorithm except that we use the original, as opposed to the modified, Taillard algorithm. Moreover, $\overline{z}$ does not intervene at this stage since the minmax objective does not apply to individual routes, but to daily vehicle itineraries made up of several routes. In Step 3 we attempt to construct new VRP solutions by recombining the s vehicle routes belonging to the best CVRP solutions at the end of Step 2, where s is an input parameter. Finally, in Step 4, we consider in turn each of these CVRP solutions and attempt a “packing” of the routes into daily vehicle itineraries using a minmax objective. We now proceed to a more detailed description of the algorithm.

Step 1 (Initial CVRP solutions). Repeat the following operations p times:

i) Initially, no customer is serviced.

ii) Apply the improvement phase of the original Taillard algorithm [6] with q applications of the decomposition process. Here we allow as many routes as necessary so that the solution is always feasible.

iii) Insert in a list each route of the solution labeled with its total cost.

Step 2 (Generation of new CVRP solutions). Repeat the following operations r times.

i) Randomly select a route from the list by favoring routes with a small label.

ii) Disregard all routes having vertices in common with those already selected; if some routes remain, go to i).

iii) Apply operations ii) and iii) of Step 1.

Step 3 (Recombination of CVRP solutions). At most s routes are selected for the second part of the algorithm, where s is an input parameter and $s >> m$. Typically, the number of selected routes is s, but there can be fewer if the size of the list is less than s. Routes are selected in non-decreasing order of their labels and inserted in a set J. This selection rule is such that a feasible CVRP solution can always be obtained. Then, within a search tree, all feasible CVRP solutions that can possibly be constructed by combining routes of J are generated. In order to control the growth of the search tree, branching priority is always given to routes containing the largest number of customers. This process ends with a set K of feasible CVRP solutions.

Step 4 (Generation of minmax CVRP solutions). For each CVRP solution k of K, let $f_{k\ell}$ be the length of the ℓ^{th} route, where $\ell = 1, ..., m_k$, and m_k is the number of routes in solution k. Then, for each k, a CVRPM solution is identified by repeatedly assigning each of the m_k routes to the least loaded (in terms of total distance) of m available vehicles. After all routes have been assigned to a vehicle in this fashion, an interchange procedure is applied to improve the minmax objective. The best of all $|K|$ solutions identified is then selected.

Here, we use $p = 20, q = 5, s = 400$. Parameter r is again problem dependent and is reported in Table 4.

4. Computational results

The algorithms just described were tested on instances generated as follows. For the minmax-CVRP, we used instances 1–5 and 11–12 of Christofides, Mingozzi and Toth [11], and instances 11–12 of Fisher. Three values were successively considered for the number of vehicles : $m := \underline{m} = \left\lceil \sum\limits_{i=1}^{n} q_i/Q \right\rceil$, $m := \underline{m} + 1$, and $m := \underline{m} + 2$. However, we did not necessarily use each of these three values for all instances since we stopped increasing m as soon as $\overline{z}/\underline{z} \leq 1.01$, where $\overline{z}$ is the minmax value provided by the heuristic, and $\underline{z}$ is the lower bound $\max\limits_{j} \{c_{oj} + c_{jo}\}$. We used the same generation process and the same initial value of m for the minmax $m - \mathrm{TSP}$, except that we set $Q := \infty$ in subsequent computations. For the minmax CVRPM, we took the same instances as above, and the same values of m as in Taillard, Laporte and Gendreau [3]. The various problem parameters, the values of m that were effectively tested, and the optimal (mincost) CVRP solution values z^* are provided in Table 1. A total of 91 instances are described in Table 1: 22 CVRPs, 17 $m-$TSPs, and 52 CVRPMs.

Table 1 Characteristics of the test problems

Problem number	Source	n	m (CVRP)	m ($m-$ TSP)	m (CVRPM)	z^*
1	CMT-1[1]	50	5,...,7	5,...,7	1,...,4	524.61
2	CMT-2[1]	75	10,...,12	10,...,12	1,...,7	835.26
3	CMT-3[1]	100	8,...,10	8,...,10	1,...,6	826.14
4	CMT-4[1]	150	12,13	12	1,...,8	1028.42
5	CMT-5[1]	199	16,17	16	1,...,10	1291.44
6	CMT-11[1]	120	7,8	7	1,...,5	1042.11
7	CMT-12[1]	100	10,11	10	1,...,6	819.56
8	F-11[2]	71	4,...,6	4,...,6	1,...,3	241.97
9	F-12[2]	134	7,8	7	1,...,3	1162.96

[1] CMT: Christofides, Mingozzi and Toth [11]; [2] Fisher [12].

The computer code was written in Pascal. All CVRP and $m-\mathrm{TSP}$ instances were run on a Sun Sparc 10 workstation, and all CVRPM instances were run on a 100 Mhz Silicon Graphics Indigo machine. Five independent runs of each instance were executed. Computational results for the three problems considered are presented in Tables 2, 3 and 4. The meanings of the column headings are as follows:

n : number of customers;

m : number of vehicles;

r : number solutions generated using the routes in the list adaptive memory;

average (best) : In the minmax-CVRP and the minmax-m-TSP, average (best) value over the five executions of the statistic; 100 $(\overline{z}/\underline{z}-1)$ where again $\overline{z}$ is the heuristic minmax value and $\underline{z}$ is the lower bound $\max_j\{c_{oj}+c_{jo}\}$; a "–" means that no feasible solution could be identified. In the minmax-CVRPM the statistic $100(\overline{z}/\underline{z}-1)$ is meaningless as the same vehicle may be used several times; instead we compute the statistic $100(m\overline{z}/z^*-1)$ which provides a better indication of solution quality. Indeed, z^*/m is the average length of vehicle routes in a good mincost CVRP solution and $\overline{z}$ is the length of the longest route in the minmax CVRPM.

% Increase : This measures the percentage by which the mincost CVRP length z^* increases due to the use of a minmax objective.

Seconds : CPU time in seconds.

Table 2 Computational results for the minmax CVRP

Problem number	n	m	r	Average	Best	%Increase	Seconds
1	50	5	0	27.66	26.75	2.59	37
		5	10	26.75	26.75	2.53	73
		5	30	26.75	26.75	2.53	140
		6	0	15.16	13.98	9.66	45
		6	10	13.38	12.97	10.10	90
		6	30	12.98	12.97	10.46	170
		7	0	5.96	5.00	16.43	44
		7	10	5.30	4.94	18.11	84
		7	30	4.87	4.28	18.92	160
2	75	10	0	8.87	8.00	4.55	38
		10	10	8.31	7.50	5.16	75
		10	30	7.65	7.50	6.41	150
		11	0	5.04	3.60	9.15	45
		11	10	3.20	2.60	11.31	87
		11	30	2.60	2.60	11.77	160
		12	0	1.95	1.79	18.97	40
		12	10	1.79	1.79	19.69	80
		12	30	1.79	1.79	19.69	160
3	100	8	0	16.97	15.61	5.19	130
		8	10	14.91	12.04	5.07	240
		8	30	13.59	12.04	5.61	470
		9	0	10.08	6.42	8.58	150
		9	10	6.70	6.17	10.17	280
		9	30	5.74	5.44	9.47	520
		10	0	3.55	3.42	16.32	130
		10	10	1.94	0.91	16.90	250
		10	30	1.47	0.57	17.59	480
4	150	12	0	3.62	3.15	7.37	190
		12	10	3.09	2.91	8.27	370
		12	30	1.36	0.95	7.60	700
		13	0	1.84	0.94	10.47	210
		13	10	0.76	0.03	12.59	390
		13	30	0.08	0.00	14.68	750
5	199	16	0	—	12.28	—	220
		16	10	—	12.28	—	430
		16	30	23.16	11.29	8.04	830
		17	0	0.84	0.12	8.93	250
		17	10	0.20	0.03	8.96	470
		17	30	0.03	0.00	11.20	900
6	120	7	0	2.52	1.45	19.73	220
		7	10	1.58	1.03	25.07	410
		7	30	1.06	1.01	22.52	790
		8	0	0.73	0.45	28.66	250
		8	10	0.40	0.38	36.83	490
		8	30	0.18	0.07	40.44	950

Table 2 (Continued) Computational results for the minmax CVRP

Problem number	n	m	r	Average	Best	%Increase	Seconds
7	100	10	0	3.50	3.31	14.32	150
		10	10	3.17	3.07	18.78	280
		10	30	3.09	2.98	21.09	530
		11	0	0.99	0.55	22.55	150
		11	10	0.36	0.16	27.85	300
		11	30	0.12	0.08	32.06	580
8	71	4	0	27.85	27.85	6.61	160
		4	10	27.85	27.85	6.61	300
		4	30	27.85	27.85	6.61	560
		5	0	24.87	22.74	17.27	190
		5	10	18.92	18.48	24.66	360
		5	30	18.19	17.55	25.79	690
		6	0	10.80	8.09	30.07	170
		6	10	7.50	5.13	32.05	330
		6	30	4.66	4.42	34.87	610
9	134	7	0	3.54	2.27	15.13	590
		7	10	2.17	2.01	20.20	1100
		7	30	2.01	2.01	24.69	2100
		8	0	0.31	0.31	36.54	600
		8	10	0.31	0.31	36.74	1100
		8	30	0.31	0.31	36.74	2200

Table 3 Computational results for the minmax m-TSP

Problem number	n	m	r	Average	Best	Seconds
1	50	5	0	27.95	27.08	58
		5	10	26.25	25.43	110
		5	30	25.85	25.43	210
		6	0	16.14	15.44	49
		6	10	15.55	13.98	95
		6	30	14.88	12.97	190
		7	0	6.17	6.12	43
		7	10	5.11	4.38	83
		7	30	4.74	4.28	160
2	75	10	0	6.87	6.01	55
		10	10	6.50	5.85	110
		10	30	5.56	5.41	210
		11	0	4.44	3.15	53
		11	10	3.63	2.61	110
		11	30	2.80	2.53	210
		12	0	1.90	1.79	52
		12	10	1.79	1.79	100
		12	30	1.79	1.79	210
3	100	8	0	15.02	13.20	170
		8	10	13.72	11.86	320
		8	30	12.51	11.28	610
		9	0	9.11	5.68	160
		9	10	8.05	5.54	310
		9	30	6.80	5.54	610
		10	0	3.61	3.47	150
		10	10	3.11	2.62	280
		10	30	1.47	0.51	550

Table 3 (Continued) Computational results for the minmax m-TSP

Problem number	n	m	r	Average	Best	Seconds
4	150	12	0	1.65	0.73	280
		12	10	1.17	0.40	540
		12	30	0.41	0.26	1100
5	199	15	0	0	0	51
6	120	7	0	0.84	0.59	380
		7	10	0.58	0.45	730
		7	30	0.47	0.42	1400
7	100	10	0	0	0	7
8	71	4	0	26.85	25.61	250
		4	10	23.70	23.20	510
		4	30	22.95	22.95	960
		5	0	12.52	12.03	210
		5	10	12.29	12.03	400
		5	30	12.15	12.03	790
		6	0	9.32	5.68	190
		6	10	6.61	4.42	370
		6	30	4.43	4.22	700
9	134	7	0	0	0	41

Table 4 Computational results for the minmax CVRPM

Problem number	n	m	r	Average	Best	%Increase	Seconds
1	50	1	30	0.00	0.00	0.00	300
		2		1.65	1.65	1.60	
		3		13.92	13.10	3.25	
		4		16.21	8.11	2.47	
2	75	1	50	0.10	0.00	0.10	420
		2		0.25	0.04	0.14	
		3		0.59	0.22	0.35	
		4		1.28	0.76	0.35	
		5		2.03	2.39	1.08	
		6		9.55	2.23	1.16	
		7		11.84	11.21	2.01	
3	100	1	50	0.28	0.15	0.28	1440
		2		0.50	0.45	0.39	
		3		1.48	1.04	0.91	
		4		2.72	1.93	0.95	
		5		8.83	7.06	3.13	
		6		10.36	7.52	2.25	
4	150	1	90	0.40	0.26	0.40	3060
		2		0.47	0.33	0.43	
		3		0.67	0.36	0.52	
		4		1.08	0.89	0.62	
		5		1.31	0.98	0.78	
		6		4.32	3.22	2.28	
		7		8.10	6.72	3.27	
		8		11.74	8.72	2.70	

Table 4 (Continued) Computational results for the minmax CVRPM

Problem number	n	m	r	Average	Best	%Increase	Seconds
5	199	1	90	1.82	1.54	1.82	3960
		2		1.85	1.57	1.83	
		3		1.96	1.69	1.85	
		4		2.10	1.74	1.89	
		5		2.31	1.91	1.96	
		6		2.50	2.10	2.12	
		7		3.12	2.70	2.30	
		8		3.64	2.91	2.41	
		9		5.18	4.71	2.75	
		10		8.25	6.81	2.66	
6	120	1	70	1.02	0.00	1.02	2700
		2		2.20	1.32	1.69	
		3		2.51	0.60	1.14	
		4		9.57	7.24	1.22	
		5		4.88	2.500	1.28	
7	100	1	50	0.00	0.00	0.00	1380
		2		0.07	0.07	0.00	
		3		2.05	1.70	1.22	
		4		3.23	3.22	0.91	
		5		8.17	7.51	4.56	
		6		10.45	7.43	2.97	
8	71	1	30	0.85	0.00	0.35	1560
		2		5.46	5.18	4.59	
		3		13.43	11.97	7.34	
9	134	1	35	0.42	0.03	0.42	4500
		2		0.76	0.10	0.45	
		3		1.65	0.86	0.72	

The computational results presented in Tables 2 to 4 give rise to a number of comments. In the case of the CVRP, the algorithm typically converges quickly towards a good solution (within a few percent of a lower bound) in most cases, especially when $n \geq 100$. This is shown by the column "Average". A notable exception occurs when $n = 199$ and $m = 16$ (Problem #5). Here the difficulty occurs in the bin packing part of Taillard's algorithm since the average vehicle load is 99.7% when $m = 16$. As soon as m is increased to 17, the algorithm finds an excellent solution very rapidly. The seemingly worse performance of the algorithm when $n < 100$ is probably due to the poor quality of the lower bound. Another indication of the quality of our algorithm is the small difference between the average and the best solution. Finally, it is worth noting that total route length in the minmax solution can be significantly higher than the cost of the minsum solution. The increase can be as large as 40% (see Problem #6). The same comments apply by and large to the $m - \mathrm{TSP}$, except that the convergence rate seems to be slower in this case. This is likely due to the fact that the $m - \mathrm{TSP}$ solution space is much larger than that of the CVRP. It is interesting to note that in the case of Problems #5, 7 and 9, a proven optimal solution was identified by means of the modified Clarke and Wright initialization algorithm. Finally, the performance of the algorithm for the CVRPM is quite variable and also difficult to assess with precision as we did not compare our solution values with a lower bound, but merely with the estimate z^*/m. Using this estimate, it seems that the algorithm produces very good solutions for small values of m, i.e., when several tours can be assigned to the same vehicle. The larger ratios in the column "Best" do not necessarily correspond to poor solutions, but may simply reflect the fact that we were unable to ascertain the quality of our solutions. Here, computing times are significantly larger than for the other two problems. However, a sensitivity analysis performed on r indicates that halving this parameter reduces computing times by more than 50%, but rarely worsens solution quality by more than 2%. We note in closing that contrary to what happens in the case of the minmax CVRP and $m - \mathrm{TSP}$, the cost of the minmax CVRPM solution is very close to that of the minsum CVRPM solution. This is indicated by the column "% Increase" in which the entries are typically less than 3%.

5. Conclusion

We have implemented and tested two Adaptive Memory heuristics for three classes of minmax routing problems. In the case of the minmax $m - \mathrm{TSP}$ the only known computational study appears to be that of França et al. [5] and it only reports results for asymmetric instances. No previous algorithm is known for the minmax CVRP and CVRPM. Our results indicate that the proposed procedures yield high quality solutions within reasonable computing times. This is mostly due to the inherent strength of the AMP which quickly navigates the search space towards its most promising regions by periodically aggregating components of the best known solutions.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grant OGP0039682. Éric Taillard also benefitted from an NSERC International Postdoctoral fellowship and was partially funded by Strategic grant STR0149269. This support is gratefully acknowledged.

References

1. G. Laporte, The vehicle routing problems: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345–358 (1992).

2. B. Fleischmann, The vehicle routing problem with multiple use of vehicles. Working paper, Fachbereich Wirschaftswissenschaften, Universität Hamburg (1990).

3. É.D. Taillard, G. Laporte, M. Gendreau, Vehicle routing with multiple use of vehicles, *Journal of the Operational Research Society* (1996). Forthcoming.

4. G.N. Frederickson, M.S. Hecht and C.E. Kim, Approximation algorithms for some routing problems, *SIAM Journal on Computing* 7, 178–193 (1978).

5. P.M. França, M. Gendreau, G. Laporte and F.M. Müller, The $m-$traveling salesman problem with minmax objective, *Transportation Science* 29, 267–275 (1995).

6. É.D. Taillard, Parallel iterative search methods for vehicle routing problems, *Networks* 23, 661–676 (1993).

7. M. Gendreau, A. Hertz and G. Laporte, A tabu search heuristic for the vehicle routing problem, *Management Science* 40, 1276–1290 (1994).

8. C. Rego and C. Roucairol, An efficient implementation of ejection chain procedures for the vehicle routing problem, Research Report 44, Laboratoire PRISM, Université de Versailles, France (1995).

9. Y. Rochat and É.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics* 1, 147–167 (1995).

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

11. N. Christofides, A. Mingozzi and P. Toth, The vehicle routing problem, in *Combinatorial Optimization* (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds.) pp. 313–338. Wiley, Chichester (1979).

12. M.L. Fisher, Optimal solution of vehicle routing problems using minimum K-trees, *Operations Research* 42, 626–642 (1994).