PERGAMON

Computers & Operations Research 27 (2000) 29–42

computers & operations research

www.elsevier.com/locate/orms

Heuristics for the multi-vehicle covering tour problem

Mondher Hachicha[a], M. John Hodgson[b], Gilbert Laporte[a,*], Frédéric Semet[a,c,d]

[a] *Centre de recherche sur les transports, Université de Montréal, Case postale 6128, succursale "Centre-ville", Montreal, Canada H3C 3J7*

[b] *Department of Earth and Atmospheric Sciences, University of Alberta, Edmonton, Canada T6G 2E3*

[c] *GRIS, Départment d'administration de la santé, Université de Montréal, Case postale 6128, succursale "Centre-ville", Montreal, Canada H3C 3J7*

[d] *Laboratoire d'Informatique et de Mathématiques Appliquées, Université de Valenciennes et du Hainaut-Cambresis, BP 311, Le Mont Houy, 59304-Valenciennes Cedex, France*

Received April 1998; received in revised form September 1998

Abstract

The *multi-vehicle covering tour problem* is defined on a graph $G = (V \cup W, E)$, where W is a set of vertices that must collectively be covered by up to m vehicles. The problem consists of determining a set of total minimum length vehicle routes on a subset of V, subject to side constraints, such that every vertex of W is within a prespecified distance from a route. Three heuristics are developed for this problem and tested on randomly generated and real data.

Scope and purpose

In the problem considered in this article, we are given two sets of locations. The first set, V, consists of potential locations at which some vehicles may stop, and the second set, W, are locations not actually on vehicle routes, but within an acceptable distance of a vehicle route. The problem is to construct several vehicle routes through a subset of V, all starting and ending at the same locations, subject to some side constraints, having a total minimum length, and such that every location of W is within a reasonable distance of a route. A common application of this problem arises in the delivery of health care facilities by mobile units in developing countries. Here, vehicles travel through a limited number of villages, and every location that is not visited must be within walking distance of a visited location. This article proposes three heuristics capable of solving instances of realistic size within reasonable computing times.

Keywords: Covering tour; Vehicle routing; Savings algorithm; Sweep algorithm; Route-first/cluster-second

*Corresponding author. Tel.: + 1-514-343-6143; fax: + 1-514-343-7121.

E-mail address: gilbert@crt.umontreal.ca (G. Laporte)

0305-0548/99/$ - see front matter
PII: S 0 3 0 5 - 0 5 4 8 (9 9) 0 0 0 0 6 - 4

1. Introduction

The purpose of this paper is to present and compare three heuristics for the *multi-vehicle covering tour problem* (m-CTP) defined as follows. Let $G = (V \cup W, E)$ be a complete undirected graph where $V \cup W$ is the vertex set, $V = \{v_0, \dots, v_n\}$, and $E = \{(v_i, v_j): v_i, v_j \in V \cup W, i < j\}$ is the edge set. Vertex v_0 is a depot at which are based m identical vehicles. The set V is made up of vertices that *can be visited* and includes a subset T of vertices that *must be visited* ($v_0 \in T$); W is a set of vertices that *must be covered*. A distance or travel time matrix $C = (c_{ij})$ satisfying the triangle inequality is defined on E. In what follows, c_{ji} must be interpreted as c_{ij} if $j > i$. The m-CTP consists of designing a set of m vehicle routes of minimum total length satisfying the following constraints:

1. there are at most m vehicle routes, and each of them starts and ends at the depot;
2. each vertex of V belongs to at most one route while each vertex of T belongs to exactly one route;
3. each vertex of W must be covered by a route in the sense that it must lie within a preset distance c of a vertex of V belonging to a route (we assume that v_0 does not cover all vertices of W);
4. the number of vertices on any route (excluding the depot) cannot exceed a preset value p;
5. the length of each route cannot exceed a preset value q.

Applications of the m-CTP arise in a number of settings. An example is the post-box location problem [1] where one must simultaneously locate post-boxes in a set V and construct optimal collection routes. Another example is the design of routes for mobile health care delivery teams in developing countries where services are rendered at a selected number of locations by medical teams, and the population living outside these locations must travel on foot to reach them (see, e.g., [2–5]. Similar problems are encountered in several Western countries by health care prevention teams [6], in the dairy industry [7], and by mobile library or banking systems.

The m-CTP with $T = V$ reduces to a *vehicle routing problem* (VRP) with unit demands (see, e.g., [8, 9] for recent surveys on the VRP). The 1-CTP without side constraints 4 and 5 was recently solved exactly by branch-and-cut for $|V| \leqslant 100$ and $|W| \leqslant 500$ [9]. This reference also contains a heuristic capable of producing solutions within 3% of optimality on most instances.

The methodology developed in [10] does not extend easily to the m-CTP with side constraints as this problem appears to be more difficult than the standard VRP which itself can rarely be solved exactly when $|V| \geqslant 50$. Heuristics therefore appear to be the only practical solution approach for the m-CTP. Our aim is to develop three such heuristics. The first two extend the savings [11] and the sweep [12] heuristics for the VRP. The third is based on the route-first/cluster-second heuristic developed in [13] for the 1-CTP.

The remainder of this paper is organized as follows. An integer linear programming formulation for the m-CTP is presented in Section 2. The three heuristics are described in Section 3. These are compared on randomly generated data and on a real-life example in Section 4. The conclusion follows in Section 5.

2. Formulation

To better focus on the m-CTP, we formulate it as an integer linear program. For each vertex $v_l \in W$, define its covering set $S_l = \{v_h \in V: c_{hl} \leqslant c\}$. Let y_{hk} be a binary variable equal to 1 if and only if vertex $v_h \in V$ is visited by vehicle k in the solution. Also, let x_{ijk} $(i < j)$ be an integer variable represented by the number of times vehicle k uses edge (v_i, v_j). If $i = 0$, this variable takes the values 0, 1 or 2 (in the case of a return trip). If $i > 0$, x_{ijk} is binary. The problem is then (m-CTP)

$$\text{minimize} \sum_{k=1}^{m} \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_{ijk} \tag{1}$$

subject to

$$\sum_{k=1}^{m} \sum_{v_h \in S_l} y_{hk} \geqslant 1 \quad (v_l \in W), \tag{2}$$

$$\sum_{k=1}^{m} y_{hk} \leqslant 1 \quad (v_h \in V \backslash \{v_0\}), \tag{3}$$

$$\sum_{i=0}^{h-1} x_{ihk} + \sum_{j=h+1}^{n} x_{hjk} = 2y_{hk} \quad (v_h \in V \backslash \{v_0\},\ k = 1, \dots, m), \tag{4}$$

$$\sum_{k=1}^{m} \sum_{\substack{v_i \in S,\, v_j \in V \backslash S \\ \text{or} \\ v_j \in S,\, v_i \in V \backslash S}} x_{ijk} \geqslant 2 \sum_{i=1}^{m} y_{hk} \quad (S \subset V,\ T \backslash S \neq \emptyset,\ v_h \in S), \tag{5}$$

$$\sum_{j=1}^{n} x_{0jk} \leqslant 2 \quad (k = 1, \dots, m), \tag{6}$$

$$\sum_{h=1}^{n} y_{hk} \leqslant p \quad (k = 1, \dots, m), \tag{7}$$

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_{ijk} \leqslant q \quad (k = 1, \dots, m), \tag{8}$$

$$y_{hk} = 1 \quad (v_h \in T,\ k = 1, \dots, m), \tag{9}$$

$$y_{hk} = 0 \text{ or } 1 \quad (v_h \in V \backslash T,\ k = 1, \dots, m), \tag{10}$$

$$x_{0jk} = 0, 1 \text{ or } 2 \quad (v_j \in V \backslash \{v_0\},\ k = 1, \dots, m), \tag{11}$$

$$x_{ijk} = 0 \text{ or } 1 \quad (v_i, v_j \in V \backslash \{v_0\},\ i < j,\ k = 1, \dots, m). \tag{12}$$

In this formulation, constraints (2) ensure that each vertex v_l of W is covered at least once, while constraints (3) specify that each vertex v_k of V appears at most once in the solution. Constraints (4) ensure that the solution will contain two edges used by vehicle k and incident to vertex v_h, or none at all. Constraints (5) are connectivity constraints. If vertex v_h does not appear in the solution, then the right-hand side is equal to zero and the constraint is redundant. Otherwise, at least two edges will connect S and its complement $V \backslash S$; this is valid since a visited vertex v_h belongs to S and there exists vertices of T outside S that must be visited by a vehicle. Constraints (6) ensure that at most m vehicles enter and leave the depot, constraints (7) state that no route contains more than p vertices of V, and constraints (8) impose the maximal route length requirement. The remaining constraints correspond to the standard integrality conditions.

3. Heuristics

We have developed three heuristics for the m-CTP. The first two extend the savings heuristic [11] and the sweep heuristic [12] for the VRP. The third one is a route-first, cluster-second approach of the type suggested by Beasley [13]. Each of these uses H-1-CTP, a heuristic for the 1-CTP, and 2-opt*, a modification of the standard 2-opt edge exchange heuristic for the *traveling salesman problem* (TSP) [14]. We first describe these two procedures, followed by the three m-CTP heuristics.

3.1. Description of H-1-CTP

This heuristic described in [10] combines the Balas and Ho [15] PRIMAL1 heuristic for the set covering problem with the GENIUS composite heuristic for the TSP [16]. Given a set covering problem of the form

$$\text{minimize} \sum_{v_h \in V} c_h y_h \tag{13}$$

subject to

$$\sum_{v_h \in S_l} y_h \geqslant 1 \quad (v_l \in W), \tag{14}$$

$$y_h = 1 \quad (v_h \in T), \tag{15}$$

$$y_h = 0 \text{ or } 1 \quad (v_h \in V \backslash T), \tag{16}$$

PRIMAL1 constructs a solution in a greedy fashion by first setting $y_h := 1$ for all $v_h \in T$, and selecting at each subsequent step a variable y_h minimizing a function $f(c_h, b_h)$, where b_h is the number of vertices $v_l \in W$ satisfying $c_{hl} \leqslant c$ and not yet covered. Three definitions of f are used: (i) $f(c_h, b_h) = c_h/\log_2 b_h$; (ii) $f(c_h, b_h) = c_h/b_h$; (iii) $f(c_h, b_h) = c_h$. PRIMAL1 considers in two different orders the three definitions of f. It constructs a set covering solution using the first definition. All vertices of V covering a vertex of W at least twice are then removed from the cover and the next definition of f is used in the following round. In H-1-CTP, we apply in turn the two sequences (i)–(ii)–(iii) and (i)–(iii)–(ii).

GENIUS first constructs a TSP tour starting with three arbitrarily chosen vertices, and inserting at each iteration a vertex between two of its closest neighbours in the partial tour, while performing a local reoptimization of the tour. When all vertices have been included in this fashion, an attempt to find a better solution is made by removing and reinserting in turn each vertex of the tour while performing local reoptimizations. A full description of this algorithm is provided in [16].

Heuristic H-1-CTP constructs an initial tour over the set T of compulsory vertices by means of GENIUS. It then gradually adds vertices to it by using in turn each of the three definitions of f used in PRIMAL1, until all vertices of W are covered by the vertices on the tour. When a feasible solution has been obtained with a given f, all vertices of the current tour associated with overcovered vertices of W are removed and the next definition of f is used. Whenever a vertex is added to the current tour, this is done using a GENIUS type insertion. The step-by-step description of H-1-CTP follows.

Step 1 (Initialization). Set $H := T$, $\bar{z} := \infty$. The current definition of f is (i).

Step 2 (Tour construction). Construct a Hamiltonian tour of length z over all vertices of H, using GENIUS.

Step 3 (Termination rules). If at least one vertex of W is not covered by a vertex of H, go to Step 4. If $z \leqslant \bar{z}$, set $\bar{z} := z$ and $\bar{H} := H$. If the current definition of f is the last one, stop with a local optimum of cost $\bar{z}$. Otherwise, remove from H all vertices associated with overcovered vertices of W and go to Step 2 with the next definition of f.

Step 4 (Vertex selection). Compute for every vertex $v_h \in V \backslash H$ a coefficient c_h equal to the cost of its cheapest insertion in the current tour on H. Determine the best vertex v_{h^*} to include on H according to the current definition of f. Set $H := H \cup \{v_{h^*}\}$ and go to Step 2.

3.2. Description of 2-opt*

Given a feasible m-CTP solution currently containing μ vehicle routes ($\mu \leqslant m$), heuristic 2-opt* attempts to determine a better feasible solution using at most μ vehicles. This is done by executing the following steps.

Step 1 (Depot replication). Create a single tour by replacing the depot with μ copies as is currently done for the multiple TSP [17] (see Figs. 1 and 2).

Consider in turn all combinations of edge pairs in the current tour.

Step 2 (Edge removal). Given the edge combination $\{(v_r, v_s), (v_t, v_u)\}$, remove (v_r, v_s), (v_t, v_u) from the tour and successively consider the two edge creation the two edge creation options: (i) (v_r, v_t) and (v_s, v_u); (ii) (v_r, v_u) and (v_s, v_t). This creates one or two cycles (see Fig. 3). Such a cycle is feasible if (i) it contains at least one copy of the depot and (ii) all chains starting and ending at a depot satisfy constraints (7) and (8).

- If both options yield infeasible solutions or fail to improve upon the best known solution, repeat Step 2 with the next edge combination, or stop if all have been considered.
- Otherwise implement the best feasible option and go to Step 3.

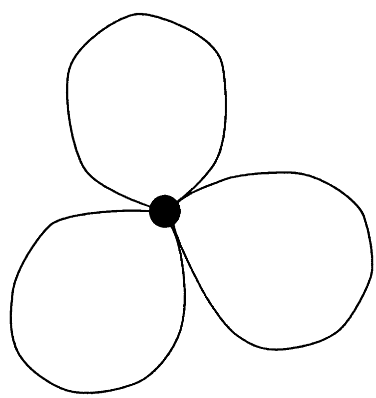

Fig. 1. Initial solution with $\mu = 3$ (0 = depot).

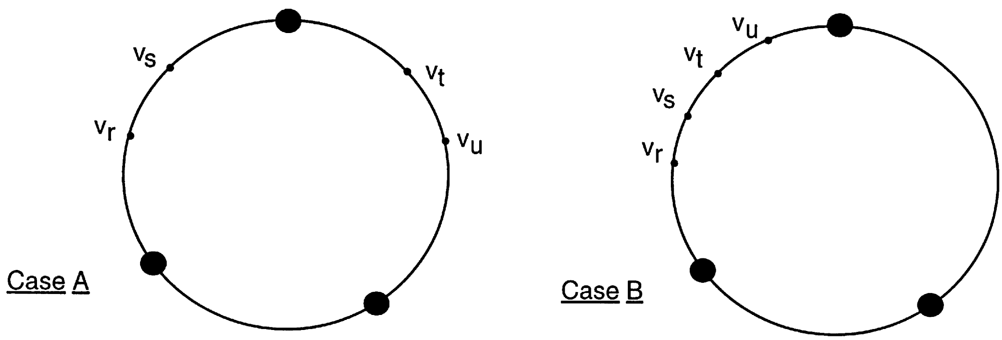

Fig. 2. Single tour obtained at the end of Step 1 by replicating the depot (large dots).

Step 3 (Tour patching). Construct an m-CTP solution using a single depot, where each chain linking two depots or the same depot defines a route. Denote by μ' the number of vehicle routes in this solution. Note that μ' may be less than μ if the solution at the end of Step 2 contains an edge linking two copies of the depot. Set $\mu := \mu'$ and go to Step 1.

In Fig. 3, we illustrate the effect of removing two edges and of reconnecting the chains. Four cases are obtained according to whether (v_r, v_s) and (v_t, v_u) are removed from the same inter-depot chain or not, and according to the reconnection option. In cases A1 and A2, two new vehicle routes are obtained by recombining portions of old ones. As can be seen from case A1, two disconnected tours may be obtained. In a TSP context, this type of reconnection is infeasible. In the m-CTP, such solutions are acceptable as long as they do not violate constraints (7) and (8). Case B1 is clearly infeasible since one of the cycles does not contain a copy of the depot. Case B2 corresponds to rearranging a single vehicle route.

3.3. Description of the modified savings algorithm

This procedure determines in a first step a feasible m-CTP solution by applying the parallel route construction version of the classical Clarke and Wright [11] algorithm for the VRP. In a second step, using a greedy criterion, it removes from the solution vertices of V without leaving any vertex of W uncovered. In a third step, it sequentially considers each route k, the subset W_k of W covered by vertices of route k, and V_k, the vertices of V covering W_k. It then applies heuristic H-1-CTP to the subproblem induced by $V_k \cup W_k$. If at the end of this step a vertex $v \in V$ belongs to several routes, it is kept in only one route, where it is the most economical. The second step is then reapplied and, finally, the solution is post-optimized by means of 2-opt*. The steps of this heuristic can now be summarized.

Step 1 (VRP solution on V). Construct a VRP solution on the graph induced by V by means of the parallel version of the Clarke and Wright algorithm, i.e., vehicle routes are gradually augmented according to the largest saving.

Step 2 (Vertex removal). Compute for each vertex of V included in a vehicle route the saving obtained by removing it from the solution and reconnecting by an edge its predecessor and

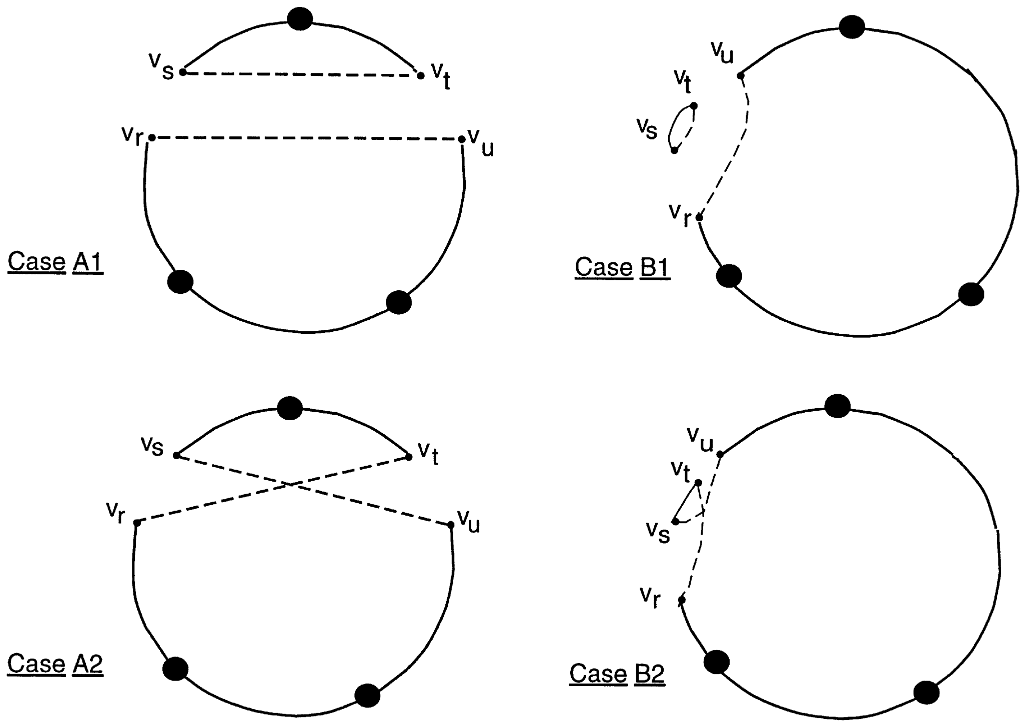

Fig. 3. Two ways of reconnecting the tour for each of Case A and Case B. Large dots represent copies of the depot.

successor on the route. Sort these vertices in a list by decreasing order of saving. Consider in turn each vertex of the list and remove it if this does not cause a vertex of W to be uncovered.

Step 3 (H-1-CTP). Consider in turn all vehicle routes k. Determine W_k, the set of vertices of W covered by a vertex of route k, and W_k, the vertices of V covering W_k. Apply heuristic H-1-CTP to the subproblem induced by $V_k \cup W_k$. If any vertex of v belongs to several routes, it is only kept in one route, where it is the most economical. Let k' be the new route obtained in this manner from route k. If k' is feasible and has a cost lower than that of route k, then substitute route k with route k'. Otherwise, route k is unchanged.

Step 4 (Vertex removal). Reapply Step 2.

Step 5 (2-opt).* Post-optimize the solution by means of 2-opt*.

3.4. Description of the modified sweep algorithm

This algorithm first constructs a feasible m-CTP solution using the basic idea of the classical sweep algorithm of Gillett and Miller [12]. The sweeping process is applied to the vertices of $T \cup W$. Once a feasible solution is known, it is improved as in the modified savings algorithm. Several solutions can be generated. The modified sweep algorithm can be described as follows.

Step 1 (Initialization). Determine the vertex v of $(T \cup W)\backslash\{v_0\}$ having the smallest ordinate and consider the half-line having an extremity at the depot and passing through v. Relabel all vertices v_h of $T \cup W$ in increasing order of the angle $(v\ \hat{v}_0 v_h)$.

Apply Steps 2 *and* 3 *for each angle* θ *determined in Step* $1(\theta = 1, \ldots, \bar{\theta})$ *and select the best overall solution.* (The value of $\bar{\theta}$ is the angle of the last vertex of the first route constructed in Step 2).

Step 2 (Sweep). Starting with the half-line $v_0 v$, rotate a radius having a fixed point at v_0 in the counterclockwise direction until a vertex v_k of $(T \cup W)\backslash\{v_0\}$ is reached.

- If $v_h \in T$, include it in the current route if this creates no infeasibility (i.e., the number of vertices of V in the current route does not exceed p, and the length of the current route, including c_{0h} does not exceed q), and continue the sweeping process. If v_h cannot feasibly be included into the current route, complete the current route by linking its last vertex to v_0, initialize a new route starting from v_h, and continue the sweeping process.
- If $v_h \in W$, determine the vertex v_r of S_h covering the largest number of vertices of W and include it in the current route if this is feasible. Otherwise complete the current route by linking its last vertex to v_0, initialize a new route starting from v_r, and continue the sweeping process.

Step 3 (Solution improvement). Attempt to improve the current m-CTP solution by applying Steps 3–5 of the modified savings algorithm.

3.5. Description of the route-first/cluster-second algorithm

This algorithm constructs a feasible m-CTP solution in two steps. First, an unconstrained 1-CTP solution is determined by means of the H-1-CTP heuristic. This tour is then cut into smaller feasible routes, starting at an arbitrary vertex. The solution is then post-optimized as in the previous two heuristics. Several solutions can be generated by using different starting points to cut the initial tour. The steps of this algorithm are as follows.

Step 1 (Route-first). Construct an initial 1-CTP current tour by means of the H-1-CTP heuristic. *Repeat Steps* 2 *and* 3 *starting with each vertex* v *included in the first route built in Step* 2 *and select the best overall solution.*

Step 2 (Cluster-second). Starting from an arbitrary vertex v, construct a route by following the current tour counterclockwise until it becomes impossible to include an additional vertex without creating an unfeasibility. Consider the next vertex v and repeat this step until all vertices have been included in a route.

Step 3 (Solution improvement). Attempt to improve the current m-CTP solution by applying Steps 3–5 of the modified savings algorithm.

4. Computational results

The heuristics just described were coded in Pascal and tested on some randomly generated instances and on real-life data. For the random instances, we generated $|V| + |W|$ points in the $[0, 100] \times [0, 100]$ square according to a continuous uniform distribution, with the restriction that the depot was generated in the $[25, 75] \times [25, 75]$ square. The sets T and V were defined by taking the first $|T|$ and $|V|$ points, respectively. All remaining points were assigned to W. The

c_{ij} coefficients were taken as Euclidean distances between pairs of vertices. The value of c was defined as $c = \max\{\max_{v_h \in V \setminus T} \min_{v_l \in W}\{c_{lh}\}, \max_{v_l \in W}\{c_{l,h(l)}\}\}$, where $h(l)$ is the index of the vertex of $V \setminus T$ that is the second closest to v_l. With this rule, each vertex of $V \setminus T$ covers at least one vertex of W and each vertex of W is covered by at least two vertices of $V \setminus T$.

A covering matrix was then constructed by defining in row l $(v_l \in W)$ and column h $(v_h \in V \setminus T)$ a binary coefficient δ_{lh} equal to 1 if and only if $c_{lh} \leqslant c$. We then made the following simplifications: (i) each row l with $\delta_{lh} = 1$ for all h was eliminated; (ii) only one of several identical rows was kept; (iii) dominated rows were eliminated (row l dominates row l' if $\delta_{lh} \leqslant \delta_{l'h}$ for all h); (iv) if a column h covers no row of the matrix due to previous reductions, it is then eliminated.

Tests were run for various combinations of $|V|$, $|W|$, $|T|$, p and q. More specifically, we tested the following values: $|V| = 50, 100, 200$, $|W| = 50, 100, 200, 400$, $|T| = 1, \lceil |V|/4 \rceil, \lceil |V|/2 \rceil$, $p = 2, 4, 8, \infty$, $q = 200, 250, 300, \infty$. In addition, we also solved a real-life problem associated with the planning of mobile health care facilities in Suhum District, Ghana. In this problem, $|W| = 148$, $|V| = 103$ in the dry season, and $|W| = 148$, $|V| = 51$ in the rainy season. For this data set, we executed tests with $p = 5, 10$, $q = \infty$ and $c = 3, 4, 5, 6, 7, 8$ (km).

Our results are summarized in Tables 1–4. In Tables 1 and 2, $|V| = 100$, $|W| = 100$, and the impact of p and q is assessed for each of the three heuristics. In Table 3, $p = 4$, $q = 200$ and the impact of $|V|$ and $|W|$ is assessed for the best two heuristics identified in Tables 1 and 2. Table 4 summarizes results for the Suhum data. In Tables 1–3, all statistics are averages over 10 instances. In Table 4, only one instance is tested. Most headings are self-explanatory, except perhaps for the following:

Ratio : value obtained using a given heuristic divided by the best value produced by all heuristics compared in the table;
Routes : number of vehicle routes in the solution;
Seconds : number of CPU seconds on a SUN Sparc 1000 station;
Length : route length (km) for the Suhum data (Table 4).

Table 1
Average results for $|V| = 100$, $|W| = 100$, $q = \infty$

$\|T\|$	p	Modified savings			Modified sweep			Route-first/cluster-second		
		Ratio	Routes	Seconds	Ratio	Routes	Seconds	Ratio	Routes	Seconds
1	2	1.047	8.2	1	1.017	8.1	8	1.031	8.2	6
1	4	1.082	4.4	1	1.006	4.1	49	1.027	4.3	6
1	8	1.095	2.5	1	1.013	2.4	114	1.007	2.4	7
26	2	1.024	19.9	2	1.006	19.6	24	1.034	20.1	19
26	4	1.035	10.3	1	1.014	10.0	98	1.025	10.2	20
26	8	1.054	5.3	1	1.010	5.4	210	1.021	5.3	22
51	2	1.028	30.3	2	1.001	30.0	62	1.014	30.1	32
51	4	1.025	15.4	2	1.007	15.3	168	1.011	15.5	32
51	8	1.044	8.0	2	1.003	7.9	297	1.023	7.9	33

Table 2
Average results for $|V| = 100$, $|W| = 100$, $p = \infty$

$\|T\|$	q	Modified savings			Modified sweep			Route-first/cluster-second		
		Ratio	Routes	Seconds	Ratio	Routes	Seconds	Ratio	Routes	Seconds
1	200	1.104	2.8	1	1.009	2.7	43	1.015	2.7	7
1	250	1.115	2.1	1	1.019	2.0	81	1.007	2.1	7
1	300	1.166	2.0	1	1.023	1.7	128	1.007	1.8	8
26	200	1.064	3.9	2	1.006	3.8	132	1.021	3.9	23
26	250	1.125	3.2	2	1.019	2.8	234	1.015	2.4	23
26	300	1.126	2.4	2	1.009	2.0	357	1.007	1.9	26
51	200	1.090	4.8	3	1.023	4.9	272	1.022	4.6	35
51	250	1.071	3.9	3	1.013	3.2	417	1.012	3.5	37
51	300	1.086	3.2	3	1.018	3.0	593	1.017	2.9	38

Table 3
Average results for $p = 4$, $q = 200$

$\|T\|$	$\|V\|$	$\|W\|$	Modified sweep			Route-first/cluster-second		
			Ratio	Routes	Seconds	Ratio	Routes	Seconds
1	50	50	1.015	2.7	6	1.017	2.8	1
1	50	100	1.005	2.7	8	1.020	2.6	2
1	100	100	1.010	4.2	36	1.033	4.2	6
1	100	200	1.009	4.2	73	1.039	4.5	7
1	200	200	1.003	6.9	243	1.077	7.5	27
1	200	400	1.004	7.4	532	1.042	8.1	39
13	50	50	1.008	5.2	14	1.017	5.5	3
13	50	100	1.007	5.2	19	1.034	5.2	4
26	100	100	1.011	5.2	84	1.031	10.2	19
26	100	200	1.013	5.2	109	1.017	10.3	21
51	200	200	1.019	5.2	611	1.006	18.2	94
51	200	400	1.008	5.2	803	1.009	18.3	114
26	50	50	1.008	7.9	26	1.009	7.9	8
26	50	100	1.013	7.9	26	1.010	7.9	6
51	100	100	1.018	15.6	154	1.009	15.5	30
51	100	200	1.011	15.6	167	1.008	15.6	36
101	200	200	1.014	29.2	1128	1.000	29.4	194
101	200	400	1.015	28.9	1355	1.000	29.3	206

Results presented in Tables 1 and 2 indicate that the best two heuristics modified and route-first/cluster-second produce the best results, but modified savings is the fastest. When $q = \infty$, irrespective of p, modified sweep produces the best results. When $p = \infty$, the top two algorithms produce similar results, but route-first/cluster-second is faster than modified sweep.

Table 4
Suhum problem results

Season (p)	Covering distance (c) (km)	Modified sweep			Route-first/cluster-second		
		Length (km)	Routes	Seconds	Length (km)	Routes	Seconds
Dry (5)	8	262.3	4	67	258.3	4	2
	7	297.2	5	63	307.3	5	2
	6	342.4	6	82	345.3	5	2
	5	450.6	7	124	435.9	7	2
	4	494.1	8	128	498.3	8	2
	3	582.3	10	209	593.4	10	3
Rainy (5)	8	187.6	3	23	189.3	3	1
	7	206.3	4	17	205.3	4	1
	6	220.9	4	20	231.2	4	1
	5	288.5	5	28	289.3	5	1
	4	285.0	5	29	303.1	6	1
	3	278.4	6	41	291.4	6	1
Dry (10)	8	202.0	3	150	217.8	2	3
	7	237.3	3	155	237.8	3	3
	6	267.6	3	183	259.5	3	3
	5	332.5	4	226	329.1	4	4
	4	375.3	4	249	355.6	4	4
	3	430.1	5	432	436.7	5	5
Rainy (10)	8	141.7	2	37	142.7	2	1
	7	156.7	2	31	156.6	2	1
	6	175.0	2	38	175.0	2	1
	5	203.6	3	48	203.5	3	1
	4	205.0	3	49	209.9	3	1
	3	221.1	3	77	218.3	3	2

Table 3 compares the modified sweep and route-first/cluster-second heuristics for several combinations of $|T|$, $|V|$ and $|W|$. One interesting observation is that modified sweep produces better results for small values of $|T|$ and worse results for large large values of $|T|$. In terms of computation times, route-first/cluster-second remains the best choice. Overall, computation times grow with $|V|$, but seem unaffected by the size of $|W|$.

The Suhum problem was also solved only by modified sweep and route-first/cluster-second. Results presented in Table 4 indicate that both heuristics can solve this real-life problem within reasonable time. In terms of solution quality, modified sweep is slightly better, but route-first/cluster-second is much faster. To illustrate, we depict in Figs. 4 and 5 the solutions obtained by means of the route-first/cluster-second algorithm for the dry season, $p = 5$, and $c = 5$ and 8. It can seen that more vehicles are required in the first case and these travel longer distances in order to reach population centers.

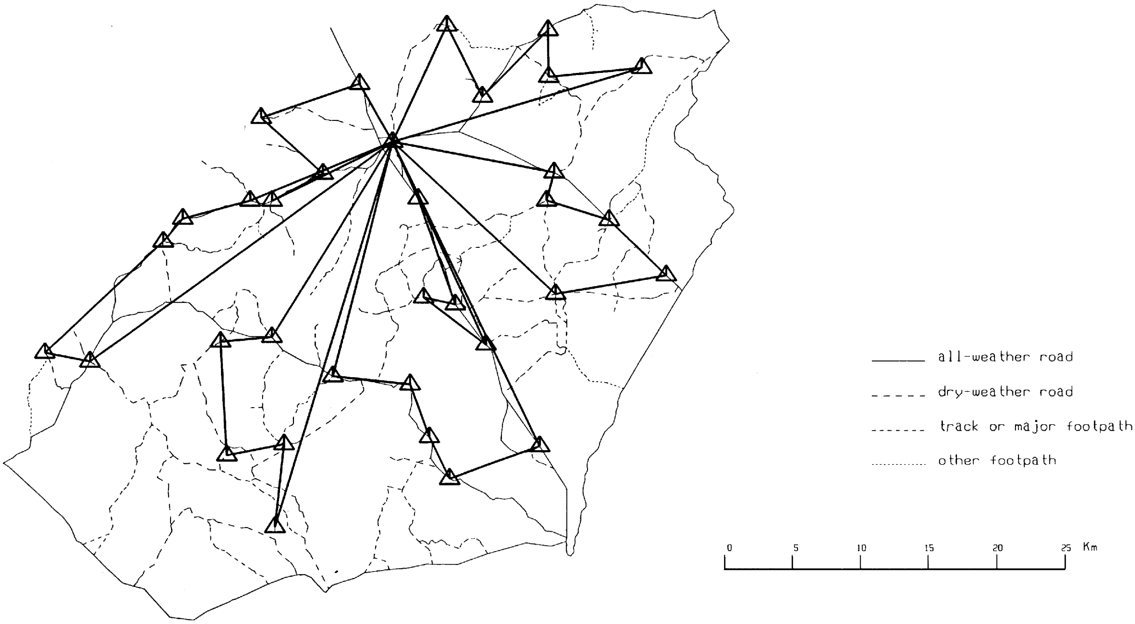

Fig. 4. Solution obtained for the Suhum district problem ($c = 5$).

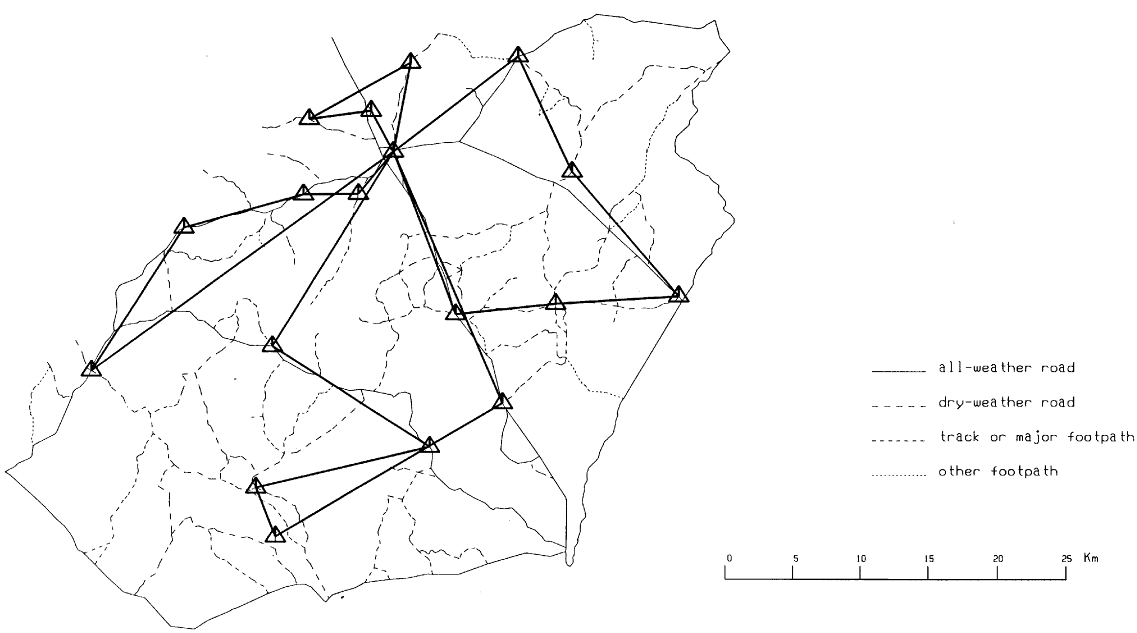

Fig. 5. Solution obtained for the Suhum district problem ($c = 8$).

5. Conclusion

We have developed three heuristics for the multi-vehicle covering tour problem, a location-routing problem with several applications, namely in the delivery of medical services in rural areas. The three heuristics, called modified savings, modified sweep and route-first/cluster-second, are partly based on corresponding methods for the standard VRP. Extensive tests show that these heuristics can solve instances of realistic size within reasonable computing times. The modified savings heuristics is the fastest but, in terms of solution quality, modified sweep and route-first/cluster-second are better by about 10%.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council (grants OGP 0005392, 0039682 and 0184123), by the Canadian Social Sciences and Humanities Research Council (grant SSHRC 410 89 0090), by the Canadian International Development Agency and by the CETAI at the École des Hautes Études Commerciales de Montréal. This support is gratefully acknowledged. Thanks are due to two anonymous referees for their valuable comments.

References

[1] Labbé M, Laporte G. Maximizing user convenience and postal service efficiency in post box location. Belgian Journal of Operations Research, Statistics and Computer Science 1986;26:21–35.
[2] Foord F. Gambia: evaluation of the mobile health care service in West Kiang district. World Health Statistics Quarterly 1995;48:18–22.
[3] Hodgson MJ, Laporte G, Semet F. A covering tour model for planning mobile health care facilities in Suhum district, Ghana. Journal of Regional Science 1998;38:621–38.
[4] Oppong JR, Hodgson MJ. Spatial accessibility to health care facilities in Suhum district, Ghana. The Professional Geographer 1994;46:199–209.
[5] Swaddiwudhipong W, Chaovakiratipong C, Nguntra P, Lerdlukanavonge P, Koonchote S. Effect of a mobile unit on changes in knowledge and use of cervical cancer screening among rural Thai women. International Journal of Epidemiology 1995;24:493–8.
[6] Brown ML, Fintor L. U.S. Screening mammography services with mobile units: results from the National Survey of Mammography Facilities. Radiology 1995;195:529–32.
[7] Simms JC. Fixed and mobile facilities in dairy practice. Veterinary Clinics of North America – Food Animal Practice 1989;5:591–601.
[8] Fisher ML. Vehicle routing. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Network routing, Handbooks in Operations Research and Management Science, Amsterdam: North-Holland, 1995:1–33.
[9] Laporte G. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 1992;59:345–58.
[10] Gendreau M, Laporte G, Semet F. The covering tour problem. Operations Research 1997;45:568–76.
[11] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568–81.
[12] Gillett B, Miller L. A heuristic algorithm for the vehicle dispatch problem. Operations Research 1974;22:340–9.
[13] Beasley JE. Route first-cluster second methods for vehicles routing. Omega 1983;11:403–8.

[14] Croes GA. A method for solving traveling-salesman problems. Operations Research 1958;6:791–812.
[15] Balas E, Ho A. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. Mathematical Programming Study 1980;12:37–60.
[16] Gendreau M, Hertz A, Laporte G. New insertion and postoptimization heuristics for the travelling salesman problem. Operations Research 1992;40:1086–94.
[17] Lenstra JK, Rinnooy Kan AHG. Some simple applications of the traveling salesman problem. Operational Research Quarterly 1975;26:717–33.

Mondher Hachicha holds an M.Sc. in Modeling and Decision from the École des Hautes Études Commerciales de Montréal.

John Hodgson is professor of Earth and Atmospheric Sciences at the University of Alberta, Canada. His research focuses on spatial analysis, location modelling and geographical information systems.

Gilbert Laporte is professor at the École des Hautes Études Commerciales de Montréal and member of the Center for Research on Transportation and of the GERAD. His research interests include vehicle routing, location and scheduling.

Frédéric Semet is professor at the Laboratoire d'Informatique et de Mathématiques Appliquées of Université de Valenciennes, France. His research interests include vehicle routing, location, and health related problems.