ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Operations Research ▮▮▮ (▮▮▮) ▮▮▮–▮▮▮

computers & operations research

www.elsevier.com/locate/cor

The bi-objective covering tour problem

Nicolas Jozefowiez[a,*], Frédéric Semet[b], El-Ghazali Talbi[a]

[a] *Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France*

[b] *Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines, Université de Valenciennes et du Hainaut-Cambrésis, 59313 Valenciennes Cedex 9, France*

Abstract

The paper discusses the definition and solution of a bi-objective routing problem, namely the bi-objective covering tour problem. The bi-objective CTP is a generalization of the covering tour problem, which means that the covering distance and the associated constraints have been replaced by a new objective. We propose a two-phase cooperative strategy that combines a multi-objective evolutionary algorithm with a branch-and-cut algorithm initially designed to solve a single-objective covering tour problem. Experiments were conducted using both randomly generated instances and real data. Optimal Pareto sets were determined using a bi-objective exact method based on an ε-constraint approach with a branch-and-cut algorithm.

Keywords: Routing; Multi-objective optimization; Cooperative approach

1. Introduction

This paper investigates the solution of a multi-objective routing problem, namely the bi-objective covering tour problem (BOCTP), by means of a cooperative strategy involving a multi-objective evolutionary algorithm and a single-objective branch-and-cut algorithm.

The BOCTP aims to determine a minimal length tour for a subset of nodes while also minimizing the greatest distance between the nodes of another set and the nearest visited node. The BOCTP can be formally described as follows: let $G=(V\cup W, E)$ be an undirected graph, where $V\cup W$ is the vertex set,

* Corresponding author. Tel.: +33 3 28 77 85 51; fax: +33 3 28 77 85 37.
E-mail address: jozef@lifl.fr (N. Jozefowiez).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2005.07.022

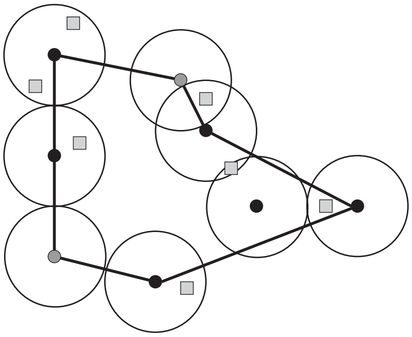

Fig. 1. An example of a solution for the covering tour problem.

and $E = \{(v_i, v_j) | v_i, v_j \in V \cup W, i < j\}$ is the edge set. Vertex v_1 is a depot, V is the set of vertices that *can* be visited, $T \subseteq V$ is the set of vertices that *must* be visited ($v_1 \in T$), and W is the set of vertices that must be *covered*. A distance matrix $C = (c_{ij})$, satisfying triangle inequality, is defined for E. The BOCTP consists of defining a tour for a subset of V, which contains all the vertices from T, while at the same time optimizing the following two objectives: (i) the minimization of the tour length and (ii) the minimization of the cover. The cover of a solution is defined as the greatest distance between a node $w \in W$, and the nearest visited node $v \in V$.

The BOCTP has never been studied in this bi-objective form. However, Current and Schilling [1] have studied a related multi-objective problem, called the maximum covering tour problem (MCTP). This problem was first introduced by Current [2] and was later formulated by Current and Schilling [3]. The MCTP attempts to identify of a tour on p vertices, while minimizing the tour length and the total uncovered demand. Each vertex has an associated demand, and a vertex is said to be covered when it lies within a given distance c from a visited node. p and c are parameters of the problem. The authors proposed a heuristic to generate an approximation of the set of efficient solutions.

The bi-objective model examined in this paper is the generalization of the covering tour problem (CTP) [4] that Boffey [5] described as an *implicit* multi-objective problem. This CTP consists of determining a minimum length tour for a subset of V that contains all the vertices from T, and which covers every vertex w from W that is covered by the tour (i.e. w lies within a distance c from a vertex of the tour, where c is a user-controlled parameter). A feasible solution for a small instance is provided in Fig. 1.

One generic application of the CTP involves designing a tour in a network whose vertices represent points that can be visited, and from which the places that are not on the tour can be easily reached [1]. For instance, Hodgson et al. [6] have used the CTP to model the determination of a tour for a mobile medical facility in an area of Ghana, where every village cannot be reached.

Very few papers on the CTP have been published. Gendreau et al. [4] have proposed a model, a heuristic and a branch-and-cut algorithm for the CTP, and the latter has been applied successfully to the routing of a mobile medical facility in Ghana [6]. In addition, Maniezzo et al. [7] have presented a model and three scatter-search algorithms, and Motta et al. [8] have proposed a GRASP meta-heuristic for a generalized covering tour problem in which the nodes of W can also be visited.

The paper's contribution to the knowledge pool is two-fold. First, it proposes a particularly valuable method for solving a new bi-objective problem, the BOCTP. Indeed, with the addition of a second objective, it becomes possible to define a very general problem rather than working with a family of related problems in which only the covering distance varies. This is useful since it avoids a priori parameterization of the problem. For instance, in the case of the Suhum district in Ghana [6], the maximum covering distance imposed by the Ghanaian Ministry of Health makes covering the total population impossible. Second, this paper introduces a cooperative scheme for solving multi-objective problems using a meta-heuristic and an exact algorithm. Co-operation between meta-heuristics and exact algorithms is frequently used in single-objective optimization, but its use for multi-objective problems is still uncommon and needs to be explored. Note that such a method has to be more efficient than invoking repeatedly the best algorithm for the single-objective case in terms of the quality of solutions produced or of computation time. Thus, this paper investigates a promising avenue for research.

This paper is organized as follows: first, the multi-objective evolutionary algorithm is explained in Section 2, followed by a description of the cooperative strategy in Section 3. Experimental results from both generated data and real data are reported in Section 4, and conclusions are presented in Section 5.

2. A multi-objective evolutionary algorithm for the bi-objective covering tour problem

The multi-objective evolutionary algorithm is a steady-state variant of the NSGA II [9]. The steps of the method that have been modified or that are specific to the BOCTP are described below.

2.1. Solution coding

A CTP solution has two components. The first component, which corresponds to a set covering problem (SCP), provides the vertices that will appear on the tour, and the second, which corresponds to a traveling salesman problem (TSP), indicates the resulting tour of the chosen vertices. Please note that the objective minimizing the cover is not related to the tour, and depends only on the selection of vertices appearing in the solution. We chose to concentrate on the set covering component of the CTP solution for the genetic operators, because designing genetic operators that take the routing aspect of the problem into account is tricky. Indeed, two parents may be so different, in terms of the nodes visited and the edges used, that they cannot pass on enough information to allow the offspring to build a tour without solving the TSP. To avoid this dilemma, the genetic operators in our method choose the visited nodes, and the tour is built with an embedded TSP method. Thus, the solution coding involves a binary table that indicates whether or not each vertex of V was visited. Our implementation uses the GENIUS heuristic for the TSP solution [10]. Briefly, GENIUS contains a tour construction phase, called GENI, and a postoptimization phase, called US. Starting with three arbitrary vertices, GENI inserts, at each iteration, an unrouted vertex between two of its p closest neighbors on the partially constructed tour, where p is a user-controlled parameter. When inserting the vertex, GENI also performs a local reoptimization of the tour. Once a complete tour has been built, the US postoptimization procedure is repeatedly applied to the tour until no further improvement is possible. During this procedure, vertices are successively removed from the tour, and then reinserted, according to the same rules used in the tour construction phase.

Since GENIUS is a heuristic, it will not always provide the optimal TSP solution for the selected nodes. However, we hypothesized that if V_1 and V_2 were two subsets of V, and the optimal tour on V_1 was better than the optimal tour on V_2, then, in most cases, the tour generated by GENIUS on V_1 would be better than the one generated by GENIUS on V_2. Given this hypothesis, we designed the MOEA to identify the covers (and their associated sets of vertices) that would be good candidates for optimal Pareto solutions. In addition, since the method we propose for the BOCTP is cooperative, involving the application of an exact method for the CTP in a second phase, the identified solutions will at least be optimized during this second phase in terms of tour length.

2.2. Initialization of the population

The initial population, which contains $N+1$ individuals, was built as follows: the definition of the cover specifies that for every couple $(v_i, v_j) \in V \times W$, c_{ij} is a candidate cover. However, not every candidate cover corresponds to a feasible solution. Therefore to evaluate a cover's feasibility, the following criterion is employed. Given $v_k \in V$ and $v_l \in W$, we have:

$$c_{kl} \text{ is a feasible cover} \quad \Leftrightarrow \quad \begin{cases} (1) \forall v_i \in T, \quad v_i \neq v_k, \quad c_{il} \geqslant c_{kl}, \\ (2) \forall v_j \in W, \quad v_j \neq v_l, \quad \exists v_i \in V \text{ s.t. } c_{ij} \leqslant c_{kl} \leqslant c_{il}. \end{cases} \tag{1}$$

This criterion is clearly valid since, on one direction, the definition of the cover is implied, and on the other direction, following the rules means building a solution. Thus, using this criterion, all feasible covers were computed.

From these feasible covers, several values were selected as parameters for the CTP-solving heuristic proposed by Gendreau et al. [4] designed to solve CTP. This heuristic gradually extends a tour by means of GENI [10]. Initially, only v_1 is part of the tour. At each step of their heuristic a new vertex v_k is incorporated into the tour, taking into account the increase c_k in tour length and the number of additional vertices b_k covered by that vertex. As in Balas and Ho [11], these two criteria are combined into a single function *f*, defined in one of three different ways: (i) $f(c_k, b_k) = c_k / \log_2 b_k$, (ii) $f(c_k, b_k) = c_k / b_k$, or (iii) $f(c_k, b_k) = c_k$. In their heuristic, Gendreau et al. [4] apply these three criteria in succession. On the test problems solved by these authors, the ratio of the upper bound provided by the heuristic to the optimal solution value is always less than 1.035 and is generally well below 1.01. The selection of the starting cover values provides information concerning the frontier of the problem to be obtained. For example, a preliminary approximation of the Pareto set extremities can be determined given the highest and lowest cover values.

2.3. Population management

At each generation, NSGA II computes two values for each solution *i*: a rank r_i, which corresponds to the quality of the solution in terms of convergence, and a crowding distance metric d_i which corresponds to the quality of the solution in terms of diversification. Then, two parents are selected by means of a tournament favour the solution with the best rank, or in the case of a tie, the solution with the best crowding distance. From the selected parents, an offspring is generated using the genetic operators described in Section 2.4. Multiple occurrences of a solution in the population are not allowed. If the newly generated offspring has already appeared in the population, another new offspring is generated from the same parents, in a process that continues until an offspring not already present in the population is created, or until 50 offspring are unsuccessfully generated. Once an offspring is successfully generated, it is inserted into the population in the place of the solution with the worst rank and the worst diversification; otherwise, the population does not change.

We have added an archive to NSGA II for storing the potentially Pareto optimal solutions to the NSGA II. This archive can also be used to evaluate whether the NSGA II algorithm should continue or be stopped: If the archive is not updated for M generations in a row, the algorithm stops.

2.4. Genetic operators

2.4.1. The crossover operator

Our crossover operator builds a solution by inserting one vertex at a time. Our goal was to minimize the number of vertices in the solution once the crossover has been applied. To achieve this goal, vertices that do not affect the cover value are avoided. Inserting a vertex into a solution can have one of two consequences: either the cover value will decrease, or it will remain unchanged. The first case holds true if the cover value is provided by a couple (v_k, v_l) and a vertex $v_i \in V$, such that $c_{il} < c_{kl}$ is added. With this in mind, the crossover was designed as follows:

Step 1. Set $H \leftarrow T$.

Step 2. Identify the pair (v_k, v_l) that provides the current cover value. Build the set $H' \leftarrow \{v_i \in V \backslash H | c_{il} < c_{kl}\}$.

Step 3. If $H' = \emptyset$, then go to Step 4. If not, choose a node $v' \in H'$ and remove it from H'. Include v' in H with a probability p computed using the parent sets of visited nodes. If v' is included in H, return to Step 2; otherwise, go to Step 3.

Step 4. Build a subset $U \subset H$, so that the cover value is unchanged by removing any element of U from H. If U is empty, stop. Otherwise, select $u \in U$ so that the value of the minimum spanning tree for $H \backslash \{u\}$ is minimal, and set $H \leftarrow H \backslash \{u\}$. Reiterate Step 4.

The probability p in Step 3 is computed using the same rules as those used in the SCP crossover fusion [12]. Let p_1 and p_2 be the two parents, and v be the vertex candidate for inclusion in the offspring. Then p is computed as follows:

(1) If v appears in p_1 and p_2, then $p = 1$.
(2) If v does not appear in any parent, then $p = 0$.
(3) If v appears in only one parent, let $p' = r_{p_2}/(r_{p_1} + r_{p_2})$ if $r_{p_1} \neq r_{p_2}$; otherwise, let $p' = d_{p_1}/(d_{p_1} + d_{p_2})$. If v is used in p_1, then set $p = p'$; otherwise, set $p = 1 - p'$.

2.4.2. The mutation operator

During the mutation phase, the status of each vertex $v \in V \backslash T$ is changed with a probability $1/|V \backslash T|$. Changing the status of a vertex means removing a vertex from the solution, even if it increases the cover value, or adding a vertex to the solution, even if it does not improve the cover value.

3. A cooperative strategy

This section presents a cooperative approach combining the multi-objective evolutionary algorithm for the BOCTP and the branch-and-cut algorithm proposed by Gendreau et al. [4] for the CTP. The latter algorithm has not been modified and can be seen as a black box whose inputs are a subset of V, the set W, and a cover, and whose output is the optimal tour for the CTP.

The branch-and-cut algorithm first relaxes the integrality conditions on the variables and the connectivity constraints of the integer linear programming model. Integrality is then gradually restored by means of a branch-and-bound mechanism. Before initiating branching at any given node of the search tree, a search is conducted for violated constraints, including the initally relaxed connectivity constraints and several other families of valid constraints. Gendreau et al. [4] have considered several classes of valid inequalities such as dominance constraints, covering constraints, subtour elimination constraints, and 2-matching inequalities.

3.1. Principle

In our approach, the MOEA generates potentially Pareto optimal solutions, which are used to build subproblems; these subproblems are then solved using the branch-and-cut algorithm. Subproblem construction is a key point of the cooperative design, given that prohibitive computational times result if the subsets of V are too large. By limiting their size and giving the branch-and-cut algorithm access to the information extracted from the MOEA solutions (i.e. a first approximation of the Pareto optimal solutions), our method makes solving the subproblems relatively easy for the branch-and-cut algorithm.

Two procedures for building the subproblems are proposed below. The purpose of the first procedure is to improve the solutions identified by the genetic algorithm in terms of length, without modifying the cover value. The second procedure aims to build sets of vertices in order to identify potentially Pareto optimal solutions whose cover values were not found by the MOEA. These construction procedures are, respectively, described in the Sections 3.2 and 3.3.

3.2. Construction procedure I

The main purpose of this procedure is to improve the solutions found by the evolutionary algorithm in terms of the tour length objective. It accomplishes this goal by investigating the possibility that some elements of the set of visited vertices $\tilde{V}$ can be replaced by sets of vertices $R \subseteq V \backslash \tilde{V}$ so that the cover value $\tilde{c}$ provided by the couple (v_t, v_c) remains unchanged. A vertex $v_k \in \tilde{V}$ can be replaced by a set R if and only if:

(1) No subset of R can replace v_k.
(2) No vertex from R can provide a better cover:

$$\forall v_i \in R, \quad c_{tc} \leqslant c_{ic}.$$

(3) There must be a vertex from $\tilde{V}$ or from R that can replace v_k for every vertex of W that can be covered by v_k. Therefore, $\forall v_l \in W \backslash \{v_c\}$, such that $c_{kl} \leqslant \tilde{c}$, where the following condition must be true:

$$\exists v_n \in R \cup (\tilde{V} \backslash \{v_k\}), \quad c_{nl} \leqslant \tilde{c}.$$

Replacing a node of $\tilde{V}$ by a subset R tends to become easier as the cardinality of R increases. However, in practice, condition (1) limits the candidate subsets. The larger the R set, the higher the cost of the test. Certainly, if the size of the set used for the branch-and-cut algorithm is very large, the algorithm will require too much computational time. Therefore, in practice, the cardinality of R is limited to one or two elements.

For each solution s of the potentially optimal Pareto set, a problem is built as follows. The set V_{I} of vertices that can be visited is created by the union of $\tilde{V}$ and all subsets of V with a cardinality of 1 or 2 that can replace a vertex of $\tilde{V}$. The set W of vertices that must be covered remains unchanged. Here, the parameter c is equal to the cover of s.

Algorithm 1. Construction of the set V_{II}

$V_{\mathrm{II}} \leftarrow V_A \cup V_B$
for all c so that $c_A < c < c_B$ **do**
  for all $v_l \in W$ **do**
  $V_{\mathrm{II}} \leftarrow V_{\mathrm{II}} \cup \{v_k \in V \setminus V_{\mathrm{II}} | c_{kl} \leqslant c\}$
end for

3.3. Construction procedure II

The cooperative scheme described in Section 3.2 is designed to improve the MOEA solutions in terms of the tour length objective. However, it is unlikely that all the feasible covers corresponding to Pareto optimal solutions will be identified. These unidentified solutions must always be situated between two solutions of the approximation, although not always between the same two solutions. Thus, it is reasonable to assume that new Pareto optimal solutions may be discovered by focusing searches in the area of the objective space between two neighboring solutions.

Let A and B be two neighboring solutions in the approximation sets found by the evolutionary algorithm (i.e. there are no other solutions between A and B). A (respectively, B) is a solution with a cover c_A (respectively, c_B) which visits the vertices of the set V_A (respectively, V_B). Assuming that $c_A < c_B$, the branch-and-cut algorithm can be executed on a set V_{II}, built according to both V_A and V_B, with the first cover $\tilde{c}$ which is strictly smaller than c_B as a parameter. If $\tilde{c}$ is equal to c_A, there is no need to execute the branch-and-cut algorithm.

It appears that neighboring solutions in the Pareto set have a large number of vertices in common. Thus, V_{II} contains V_A and V_B. This inclusion insures that the branch-and-cut algorithm will at least be able to find the solution A, or a solution with the same cover but a better tour length in cases for which the tour on V_A is not optimal. The following process is used to complete V_{II}: For every feasible cover c, so that $c_A < c < c_B$, vertices are added to V_{II} in order to obtain a subset of V_{II} with c as a cover. (Algorithm 1 provides the procedure for constructing the set V_{II}.)

4. Computational results

4.1. Benchmarks

4.1.1. Randomly generated data sets

Experiments were conducted on a series of randomly generated instances. To generate the vertex set, $|V|+|W|$ points were generated in a $[0, 100]\times[0, 100]$ square, with a uniform distribution. The sets T and V

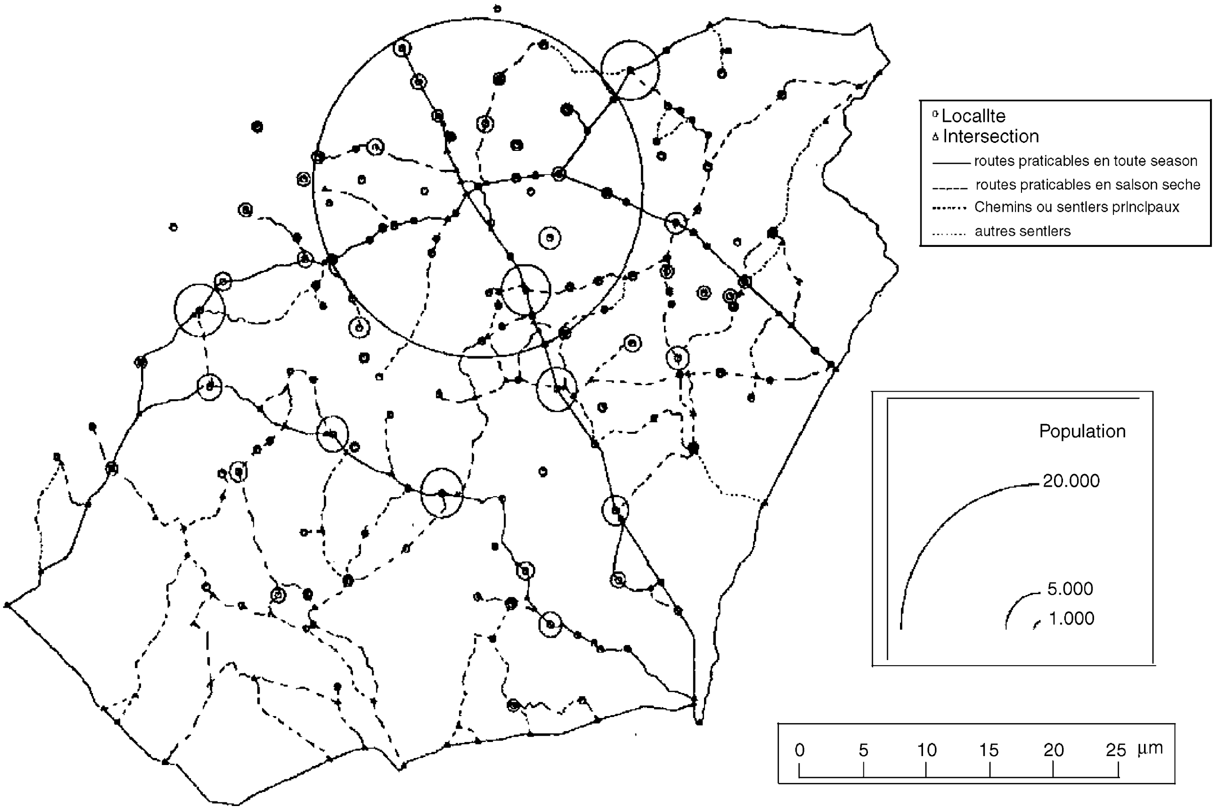

Fig. 2. Roads and villages, Suhum District, East region, Ghana.

were defined as the first $|T|$ and $|V|$ points, respectively, and W was defined as the set of the remaining points. $|V|$ was set to 75, 100, and 120; T to 1, $\lfloor 0.10|V| \rfloor$, $\lfloor 0.20|V| \rfloor$; and $|W|$ to $|V|$, $2|V|$, $3|V|$. For each combination, 5 instances were generated.

4.1.2. Real data

Experiments were also conducted on real data sets taken from the study of the Suhum district in Ghana, described by Hodgson et al. [6]. The purpose of Hodgson et al. was to determine a route for a mobile healthcare facility. However, in the Suhum district, it was not possible to reach every village by a vehicle. Therefore, a route leaving from and returning to Suhum had to be found, one which would go through the accessible villages in such a way that the inhabitants of the unvisited villages would not have to walk more than a given distance to get to a visited village. This problem was solved by Hodgson et al. [6] as a CTP. In our study, we solved it as a BOCTP.

Fig. 2 shows the 148-vertex graph that defines the road network. (Refer to the paper by Hodgson et al. [6] for a complete presentation of the Suhum district data.)

4.2. An exact method for the bi-objective covering tour problem

To assess the efficiency of the proposed cooperative approach, an ε-constraint approach was used to solve the BOCTP exactly. In a bi-objective case, the ε-constraint method adds a new constraint to the problem : $f_i(x) \leqslant \varepsilon$, where f_i is one objective and ε is a given value, and it works to optimize only the second objective. Varying the ε parameter allows different problems with different solutions to be generated. Since the problems are solved with an exact algorithm, the solutions are therefore Pareto optimal solutions.

If the new constraint is built with the covering objective, the problem becomes the CTP defined by Gendreau et al. [4], making it possible to use their branch-and-cut algorithm for the CTP to solve the resulting problems. The algorithm is able to solve instances in which the size of V can be as large as 100, and the size of W as large as 500.

The following procedure is used to choose the ε values so that all optimal Pareto solutions are found without needless computation:

Step 1. Compute the feasible covers according to criterion 1, and sort them into a list $l = (c_0, c_1, \ldots, c_k)$ in order of decreasing value. Set *current_cover* $\leftarrow c_0$.

Step 2. Apply the branch-and-cut algorithm to solve the CTP, using *current_cover* as a parameter.

Step 3. Compute the cover c_s of the solution s provided by the branch-and-cut algorithm. Save s as an optimal Pareto solution. Search $c_i \in l$ so that $c_i = \max_{c_j \in l}(c_j < c_s)$. If such a c_i exists, then set *current_cover* $\leftarrow c_i$ and go to Step 2; otherwise, stop.

4.3. Experimental environment

The MOEA was run 10 times on each instance. The parameters of the MOEA included a population size N of 256, and a stopping criterion parameter M of 5000. The cooperative strategy was tested for $|V| = 100$ and $|V| = 120$.

The MOEA was coded in C. The branch-and-cut algorithm was implemented in C with CPLEX 8.1. Runs were executed on a Pentium IV 2.67 Ghz, 512 Mo of RAM with a Debian Linux 3.0 operating system.

Two metrics were used to assess the efficiency of the heuristic methods. The first examined the ratio of optimal Pareto solutions found when compared to the total number of optimal Pareto solutions (NB), while the second considered the generational distance (GD) [13]. The generational distance is expressed as follows: $\sqrt{\sum_{i=1}^{n} d_i^2}/n$, where n is the number of solutions in the approximation set, and d_i is the Euclidean distance in the objective space between solution i and the nearest solution of the optimal Pareto set. The coordinates of the solutions in the objective space were normalized, and therefore, all values x in the tables related to this metric must be read $x \times 10^{-4}$. The values reported in the tables represent the means for the different runs for each instance size.

4.4. Efficiency of the MOEA

First, the efficiency of the MOEA was assessed. The results were evaluated to determine the quality of the approximations (Ratio and GD) and the magnitude of the computational time (time) in seconds. The results are reported in Table 1.

Table 1
Comparisons between the bi-objective exact method and the multi-objective evolutionary algorithm

$\|V\|$	$\|T\|$	$\|W\|$	ε-const. meth.		MOEA		
			NB	Time	GD	Ratio	Time
75	1	75	84.8	661.0	3.18	0.82	205.4
75	1	150	86.8	1424.6	2.33	0.75	281.0
75	1	225	100.4	2654.4	1.33	0.73	310.8
75	7	75	65	1126.8	1.65	0.78	375.6
75	7	150	59.4	235.0	0.79	0.82	171.0
75	7	225	56	583.2	3.07	0.79	253.0
75	15	75	62.6	276.0	1.23	0.84	180.4
75	15	150	28.4	80.0	0.95	0.90	163.2
75	15	225	34.2	42.4	2.75	0.88	218.6
100	1	100	124.8	19622.2	4.04	0.65	481.9
100	1	200	131.4	21976.8	2.31	0.66	668.1
100	1	300	122.4	21823.8	2.14	0.63	770.7
100	10	100	61.4	951.8	2.66	0.74	439.5
100	10	200	56.6	936.0	2.80	0.77	416.1
100	10	300	82.4	5779.4	1.61	0.72	854.6
100	20	100	31.4	135.6	1.31	0.69	448.6
100	20	200	44.2	477.4	3.39	0.69	570.4
100	20	300	34.4	293.0	4.59	0.61	560.3
120	12	120	52.2	5864.8	2.00	0.80	541.8
120	12	240	83.2	11824.2	2.78	0.69	1119.0
120	12	360	93.6	15481.4	3.19	0.51	1286.5
120	24	120	35.2	334.4	1.73	0.68	529.2
120	24	240	52.6	3062.0	3.73	0.64	863.8
120	24	360	50	1063.8	2.52	0.76	871.6

Approximation quality: The MOEA was able to generate good-quality approximations, which frequently included a significant part (0.73 on average) of the optimal Pareto sets. Furthermore, the generational distance values indicate that the non-optimal solutions found are not far from the optimal solutions. When the size of V increases, the approximations appear to remain of good quality. The sizes of T and W do not seem to have a significant impact on the quality of the approximations.

Computational time: Two things must be kept in mind when considering computational times. First, the exact method only runs as many times as there are optimal Pareto solutions, and on average, their number is not high. The MOEA, on the other hand, must run at least 5000 generations before stopping. Furthermore, in the exact approach, the cover is fixed for each execution of the branch-and-cut algorithm execution, so that rules can be applied to simplify the problem [4]. For example, these rules make solving the problem using the branch-and-cut algorithm more simple when the value of $|T|$ is high. However, since the goal of the MOEA is to generate the complete optimal Pareto set, the cover is not fixed, and the problem must be solved without reduction.

For a small V, the ε-constraint method tends to be faster. This is particularly true when the cardinality of T is large due to the simplification rules. However, these rules are less and less able to restrain the increase in computational time when the cardinality of V increases. For instance, the computational results for $|V| = 120$ and $|T| = 1$ are not reported due to the prohibitive computational time required by the ε-constraint method. The computational times for the MOEA are moderate, and as the cardinality of V increases, the MOEA becomes significantly faster than the exact approach. As a rule, the MOEA turns out to be 9.27 times faster than the ε-constraint approach for the complete set of benchmarks.

4.5. Contribution of the cooperation

Next, the contribution of each cooperation scheme was evaluated. Table 2 reports the average ratio (Ratio), GD, computational times (time) for the MOEA and for both cooperation schemes (Cooperation I and Cooperation II). The cumulated average computational times (TT) are also reported: TT is equal to the time needed for the MOEA to generate a first approximation and then to apply a cooperative scheme. Cooperative scheme I appears to produce interesting results. Overall, it is able to identify an average of five percent of new optimal Pareto solutions. In addition, it decreases the generational distance between the generated set and the optimal Pareto set, without sending computational times skyrocketing. Similar conclusions can be drawn for the other cooperative scheme: In a reasonable amount of time, scheme II is able to improve the approximation quality in terms of the number of optimal Pareto solutions found and the values of the generational distance, although to a lesser extent than scheme I.

Finally, the cumulated contribution of the two cooperative schemes was evaluated (Table 2; heading Cooperation I + II). It appears that the quality of the results using both schemes is better than if only one is used, both in terms of the number of optimal Pareto solutions found (a improvement ratio of 7.2 on average) and in terms of the generational distance values. Furthermore, the additional cumulated computational time is moderate.

4.6. Experiments on the Suhum district data

Experiments were also conducted using the data from the Suhum district in Ghana. Given the dry season data, the MOEA was able to produce a nearly complete Pareto optimal set in considerably less time than the exact algorithm, and solving the problem for the rainy season was even easier: almost all the solutions were identified by the heuristic during the initialization phase. This can be explained by the fact that, because one of the unaccessible villages is very far from the nearest accessible village, the solution providing the best cover has the vehicle visiting only three villages. Indeed, the best cover value is approximately 9 km during the dry season and roughly 25 km during the rainy season. In the study by Hodgson et al. [6], the Ghanian Health Ministry set the maximal cover value at 8 km. In order to solve the problem with this cover, the authors had to remove some villages which meant covering only 98.7% of the villages during the dry season and 89.9% during the rainy season. (Table 3).

Actually, there are solutions that could be interesting for the Suhum district, even though they are not Pareto optimal. For instance, it might be interesting to increase the length of the tour by inserting new vertices without modifying the cover, so that the total distance between the vertices of V and those of

Table 2
Contribution of the cooperative schemes

$\|V\|\backslash\|T\|\backslash\|W\|$	MOEA			Cooperation I				Cooperation II				Cooperation I+II			
	Ratio	GD	Time	Ratio	GD	Time	TT	Ratio	GD	Time	TT	Ratio	GD	Time	TT
100\1\100	0.65	4.04	481.9	0.67	2.90	80.7	562.6	0.65	3.14	34.9	516.8	0.67	2.79	597.5	597.5
100\1\200	0.66	2.31	668.1	0.70	2.05	102.7	770.8	0.69	1.97	120.5	788.6	0.73	1.97	891.3	891.3
100\1\300	0.63	2.14	770.7	0.69	2.03	131.5	902.2	0.65	1.62	113.2	883.9	0.70	1.62	1015.4	1015.4
100\10\100	0.74	2.66	439.5	0.80	2.14	30.5	470.0	0.77	2.25	7.5	447.9	0.82	1.95	477.7	477.7
100\10\200	0.77	2.80	416.1	0.82	2.54	33.1	449.2	0.78	2.71	8.8	424.9	0.83	2.48	458.0	458.0
100\10\300	0.72	1.61	854.6	0.76	1.43	60.6	915.2	0.76	1.53	23.5	878.1	0.78	1.37	938.7	938.7
100\20\100	0.69	1.31	448.6	0.76	0.77	18.6	467.2	0.73	1.21	3.5	452.1	0.78	0.74	470.7	470.7
100\20\200	0.69	3.39	570.4	0.76	2.59	30.9	604.1	0.74	2.67	10.0	580.4	0.79	1.95	614.1	614.1
100\20\300	0.61	4.59	560.3	0.68	4.10	20.6	580.9	0.65	3.56	7.6	567.9	0.73	3.24	588.5	588.5
120\12\120	0.80	2.00	541.8	0.84	1.16	55.7	597.5	0.82	1.93	8.2	550.0	0.86	1.06	605.7	605.7
120\12\240	0.69	2.78	1119.0	0.74	2.52	97.7	1206.7	0.73	2.74	24.2	1243.2	0.77	2.51	1230.9	1230.9
120\12\360	0.51	3.19	1286.5	0.55	2.70	117.6	1404.1	0.54	2.87	35.2	1321.7	0.57	2.55	1439.3	1439.3
120\24\120	0.68	1.73	529.2	0.75	0.34	37.9	567.1	0.69	1.53	6.8	536.0	0.76	0.32	573.9	573.9
120\24\240	0.64	3.73	863.8	0.69	3.21	75.7	939.5	0.68	3.32	13.3	877.1	0.72	2.89	952.8	952.8
120\24\360	0.76	2.52	871.6	0.78	2.49	53.8	925.4	0.80	2.41	18.4	890.0	0.81	2.39	943.8	943.8

Table 3
Results for the Suhum district data.

Season	Exact algorithm		MOEA		
	NB	Time	Ratio	GD	Time
Dry	48	5577	0.96	0.65	108.8
Rainy	19	36	1.00	0.00	5.2

W is decreased. A sample objective able to deal with this would be the minimization of the sum of the distances between each node of *W* and its nearest visited node of *V*.

5. Conclusions

In this paper, we considered the bi-objective covering tour problem, which is a generalization of the covering tour problem. In this generalization, constraints linked to coverage are replaced by a second objective. To solve the problem, we proposed a multi-objective evolutionary algorithm. We also explored the use of cooperation via a two-phase strategy based on the multi-objective evolutionary algorithm for the bi-objective covering tour problem and a branch-and-cut algorithm for the covering tour problem.

Experiments were conducted using generated data sets. Optimal Pareto sets were determined for the generated data sets, using the bi-objective ε-constraint method for the bi-objective covering tour problem. When compared to the optimal Pareto sets, our results show that the multi-objective evolutionary algorithm is able to generate good-quality approximations of the optimal Pareto sets within reasonable computational times. Our evaluation of the contribution of the cooperative approach indicates that the proposed cooperative scheme is able to improve the results of the multi-objective algorithm without requiring great amounts of time. Experiments were also conducted on real data. It appears that for such complex data, the MOEA is more efficient in terms of computational time than the bi-objective exact method. However, in order to solve the real-life problem represented by the data as a multi-objective problem (i.e. considering both the tour length and the coverage), the model needs to be refined.

Acknowledgements

This work was partially supported by the Nord-Pas-de-Calais region. This support is gratefully acknowledged. Thanks are also due to the referees for their valuable comments.

References

[1] Current JR, Schilling DA. The median tour and maximal covering problems. European Journal of Operational Research 1994;73:114–26.
[2] Current JR. Multiobjective design of transportation networks. PhD thesis, Department of Geography and Environmental Engineering, John Hopkins University, 1981.
[3] Current JR, Schilling DA. The covering salesman problem. Transportation Science 1989;23:208–13.
[4] Gendreau M, Laporte G, Semet F. The covering tour problem. Operations Research 1997;45:568–76.

[5] Boffey B. Multiobjective routing problems. Top 1995;3:167–220.
[6] 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.
[7] Maniezzo V, Baldacci R, Boschetti M, Zamboni M, Scatter search methods for the covering tour problem. Technical Report, Science dell'Informazione, University of Bologna, Italy, June 1999.
[8] Motta L, Ochi LS, Martinhon C. Grasp metaheuristics for the generalized covering tour problem. In: MIC'2001—fourth metaheuristics international conference. Porto, Portugal, 2001. p. 387–91.
[9] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 2002;6:182–97.
[10] Gendreau M, Hertz A, Laporte G. New insertion and postoptimization procedures for the traveling salesman problem. Operations Research 1992;40:1086–94.
[11] Balas E, Ho A. Set covering algorithm using cutting planes, heuristics, and subgradient optimization: A computational study. Mathematical Programming 1980;12:37–60.
[12] Beasley JE, Chu PC. A genetic algorithm for the set covering problem. European Journal of Operational Research 1996;94:392–404.
[13] Van Veldhuizen DA. Multiobjective evolutionary algorithms: classifications, analysis, and new innovations. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright–Paterson AFB, Ohio, May 1999.