From Single-Objective to Multi-Objective Vehicle Routing Problems: Motivations, Case Studies, and Methods

Nicolas Jozefowiez[1], Frédéric Semet[1], and El-Ghazali Talbi[2]

[1] Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines
Université de Valenciennes et du Hainaut-Cambrésis
Le Mont-Houy, 59313 Valenciennes Cedex 9, France
nicolas.jozefowiez@univ-valenciennes.fr,
frederic.semet@univ-valenciennes.fr

[2] Laboratoire d'Informatique Fondamentale de Lille
Université des Sciences et Technologies de Lille
59655 Villeneuve d'Ascq Cedex, France
talbi@lifl.fr

Summary. Multi-objective optimization knows a fast growing interest for both academic researches and real-life problems. An important domain is the one of vehicle routing problems. In this chapter, we present the possible motivations for applying multi-objective optimization on vehicle routing problems and the potential uses and benefits of doing so. To illustrate this fact, we also describe two problems, namely the vehicle routing problem with route balancing and the bi-objective covering tour problem. We also propose a two-phased approach based on the combination of a multi-objective evolutionary algorithm and single-objective techniques that respectively provide diversification and intensification for the search in the objective space. Examples of implementation of this method are provided on the two problems.

Key words: Vehicle routing problems; multi-objective optimization; cooperation; metaheuristics; parallelization.

1 Introduction

The goal of this chapter is to present an overview of what multi-objective optimization can bring to vehicle routing problems. This is illustrated by two problems representing the two main aspects of multi-objective vehicle routing problems and a general optimization strategy.

Vehicle routing problems, while widely used to deal with real-life cases, are usually optimized on a single objective, which generally aims at optimizing a

B. Golden et al. (eds.), *The Vehicle Routing Problem*,
doi: 10.1007/978-0-387-77778-8_20,

cost (distance, financial). However, many real industrial problems cannot be limited to the aspect of cost and deal with multiple objectives. For instance, even if only cost is considered, it is possible to define several costs for a same problem: financial and time. It is also possible to consider objectives that are not limited to cost objectives, but also objectives that deals with aspects like fairness or lateness. Therefore, there is a real interest in studying multi-objective vehicle routing problems.

The study of these problems falls into the field of multi-objective optimization, which proposes methods to solve problems containing several (and usually conflictive) objectives. This domain finds its roots in the works of Edgeworth [17] and Pareto [44] in the context of economic research in the 19^{th} century. The field knows a growing interest since the mid 80s [52] and a fast expansion since the mid 90s with, notably, the apparition of methods like multi-objective evolutionary algorithms [7, 14].

Formally a multi-objective problem can be stated as follows.

$$(MOP) = \begin{cases} min\ F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) \\ s.t.\ x \in D \end{cases} \tag{1}$$

with $n \geq 2$ being the number of objective functions; $x = (x_1, x_2, \ldots, x_r)$, the decision variable vector; D, the feasible solution space; and $F(x)$, the objective vector. The set $O = F(D)$ corresponds to the feasible solutions in the objective space, and $y = (y_1, y_2, \ldots, y_n)$, where $y_i = f_i(x)$, is a solution. A MOP solution is the set of the non-dominated solutions called the Pareto set (PS). Dominance is defined as follows:

Definition 1.1 *A solution $y = (y_1, y_2, \ldots, y_n)$ dominates (denoted $\prec$) a solution $z = (z_1, z_2, \ldots, z_n)$ if and only if $\forall\ i \in \{1 \ldots n\}$, $y_i \leq z_i$ and $\exists\ i \in \{1 \ldots n\}$, such that $y_i < z_i$.*

Definition 1.2 *A solution y found by an algorithm A is said to be potentially Pareto optimal (PPS), relative to A, if A does not find a solution z, such that z dominates y.*

When solving a multi-objective problem, the proposed method should be able to converge toward the optimal Pareto set while at the same time providing a set of diversified solutions in the objective space. They are the goals of intensification and diversification. These two goals are illustrated in Figure 1. Figure 2 shows an approximation answering both goals, while the approximation in Figure 3 is good in terms of intensification but bad in terms of diversification; the approximation in 4 illustrates the opposite case.

The chapter is organized as follows. Section 2 deals with multi-objective problems in general and presents the main motivations for using multiple objectives as well as several uses made in the literature. Section 3 presents the two bi-objective vehicle routing problems used as illustrative examples. Section 4 explains a two-phase strategy that can be used and shows how the

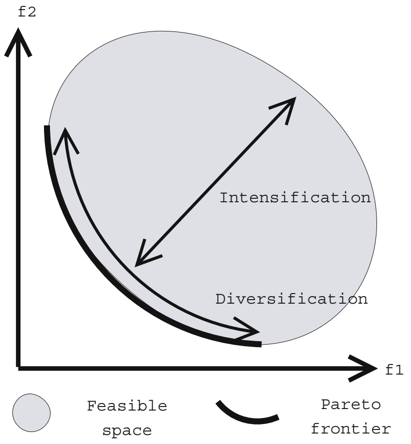

Fig. 1. The two goals of a multi-objective method.

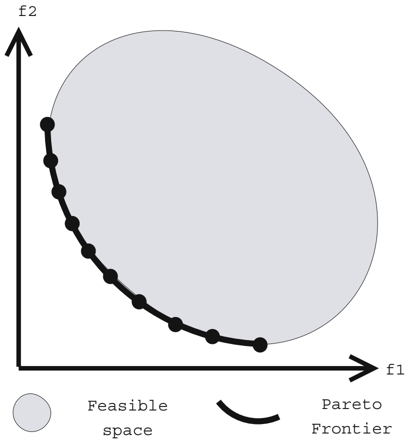

Fig. 2. An ideal solution.

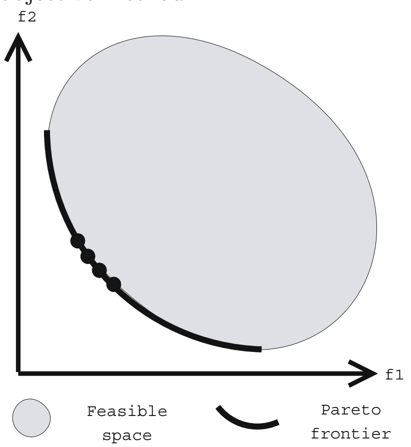

Fig. 3. A good solution in terms of intensification.

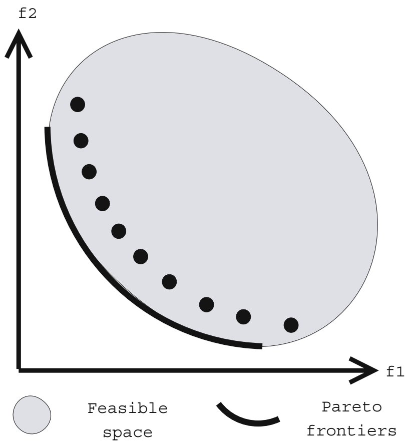

Fig. 4. A good solution in terms of diversification.

strategy was implemented for the two problems presented in the previous section. Conclusions are drawn in Section 5.

2 Multi-objective Vehicle Routing Problems

In this section, we present the possible motivations to use multiple objectives in a vehicle routing problem. Then, the main uses are presented: extension of a classic academic problem, generalization of a classic academic problem, real-life problems. Finally, we also discuss some of the objectives that appear in the literature and have been introduced in the multi-objective studies. A complete survey of the literature can be found in [34].

2.1 Motivation

Academic vehicle routing problems need adaptations for real-life applications. These adaptations are mostly additions of new constraints and/or parameters to a basic problem. For instance, the capacitated vehicle routing problem deals with the construction of a minimal length collection of tours for a fleet of vehicles to serve a set of customers wanting a given amount of goods delivered such that the total weight of products transported by a vehicle does not exceed a given capacity. Several variants of these problems have been defined to consider numerous aspects that can appear in real-life cases [55]. A specific example is the vehicle routing problem with time windows in which a customer must be delivered during a given lapse of time.

Another way to improve the practical aspects of vehicle routing problems is to use several objectives. The objectives that are used may be related to different aspects of vehicle routing problems: tour (cost, profit, makespan, balance ...), nodes/arcs (time windows, customer satisfaction ...), and resources (management of the fleet, specificities of the product to deliver ...).

The introduction of multi-objective routing problems is motivated by three reasons: to extend classic academic problems in order to improve their practical application while never losing sight of the initial objective, to generalize classic problems, and to study real-life cases in which the objectives have been clearly identified by the decision-maker.

2.2 Extension of Classic Academic Problems

Multi-objective optimization can be used as a possibility to study other objectives in addition to the one defined initially, which is often related to a solution cost. In this context, the problem definition remains unchanged, and new objectives are added. The purpose of such extensions is often to enhance the practical applications of the model by recognizing that logistic problems are not simply cost driven. This can be done in order to consider the driver's workload and to try to balance it to bring fairness between the drivers [36,49]. Customer satisfaction can also be an issue and modeled by an objective [51]. Other illustrative extensions can be found in the issues in commercial distribution pointed by Ribeiro and Lourenço [49], in the study of the multi-objective traveling salesman problem [18, 38, 56] or other TSP variations like the bi-objective median tour problem and the maximal covering tour problem [11].

2.3 Generalization of Classic Academic Problems

Another motivation to use multi-objective optimization is to generalize a problem by adding objectives instead of one or several constraints and/or parameters. In the literature, this strategy has notably been applied to the vehicle routing problem with time window constraints where the time windows are replaced by one or several objectives.

Boffey [3] provides a list of routing problems that he classified as problems which are implicitly multi-objective. In those problems, a constraint and/or parameter or a set of constraints and/or parameters is used instead of what can be naturally modeled as an objective. Feillet *et al.* [20] have described a class of problems, called traveling salesman problems with profits (TSPP), which belong to this category. In these problems, a profit, associated with each customer, can be collected when the customer is visited, but it is not compulsory to visit all customers. Two conflicting objectives can be clearly identified:

1. Maximize the profit by visiting the maximum number of customers, thus increasing the length of the solution.
2. Minimize the length of the solution by visiting fewer customers, thus decreasing the profit generated by the solution.

Usually, this problem is solved by defining a single-objective problem that combines the two objectives; or constraints one objective and optimizes the other one. But it is also possible to solve it as a bi-objective problem and to generate a set of non-dominated solutions without advantaging one objective or the other [28]. That way only one problem needs to be optimized.

Other examples of generalization includes the bi-objective covering tour problem (presented below) [30, 33] and the traveling purchaser problem [50].

2.4 Real-Life Applications

Multi-objective routing problems are also specifically studied for real-life situations, in which decision-makers define several clear objectives they want to be optimized. Examples include the case of a Belgian transport company [19], schoolbus routing [4,8,43], urban waste collection [35], healthcare facility tour planning in developing countries [16]. Most of these studies keep the minimization of a cost (mainly the length) but add new objectives according to the needs of the situation. The most illustrative objectives are presented below with their applications. In the case of the transport company, the issues were multiple as eight objectives were defined: balancing, idle times, time window constraint violation for instance. Schoolbus routing presents other challenges: there is a need to position bus stops so that users do not have to walk too much to catch the bus. Then, there is a need to balance the number of users between the selected stops and to distribute the load (*i.e.*, the number of users) between the buses. Makespan-like objectives are also used to ensure that the users do not stay too much time on the bus and that there are no glaring inequities between the first student picked up on the tour and the last one. Makespan is also used in the case of the urban waste collection but for a different reason. Here, there is a need to sort out the waste after the picking. The two tasks (collecting and sorting) are done by the same workers.

Therefore, the tours need to be finished as soon as possible so that the workers can begin to work on sorting out the waste at the factory. In healthcare facility tour planning, the main problem comes from the fact that it is not possible to visit the whole population (*i.e.*, the villages), therefore, in addition to an economic objective, there is a need to facilitate the population access to the facility. This is done by using two additional objectives: maximizing the population in a given radius from the selected stops for the facility and minimizing the total distance that must be walked for the uncovered population to reach a facility stop. Another main real-life application is the case of hazardous product transportation [25, 57]. The non-cost objectives deal with the minimization for the population along the road used and the minimization of the overall probability of an accident to happen as the consequences would be drastic.

2.5 Objectives

The new objectives introduced in the studies of multi-objective vehicle routing problems in the literature can be classified according to the component of the problem they are related to: the tour, the node/arc activity, or the resources.

Objectives on tours include costs (time, distance ...), although it is not always the case as for instance in [8]. Other aspects considered by these objectives include makespan as in the studies by Corberan *et al.* [8], Pacheco and Marti [43], and Lacomme *et al.* [35]. Fairness can also be introduced by means of tour balancing by minimizing the difference between the length of the longest tour and the length of the shortest tour. Problem related objectives like minimizing the risk in the case of hazardous transportation problems [25, 57] or objectives like keeping the tour in a cluster [39, 40] are also proposed in the literature.

In the literature, many of objectives related to nodes/arcs deal with vehicle routing problems with time windows by removing the time windows and adding additional objectives minimizing the lateness or earliness to the bounds of the windows and/or the number of violated constraints [1, 19, 22, 26]. Other objectives are for example linked to assigning priority to arcs or nodes and trying to visit the ones with the greatest priorities [45, 46]. It is also possible to define economic or marketing objectives, such as increasing customer satisfaction [51] or improving the customer-driver relationship [49]. A last family of objectives deals with optimizing the access to the visited nodes by a set of unvisited nodes [4, 11, 30, 33]. An example of uses of this kind of objectives is mobile healthcare facility routing [16].

Objectives related to resources are about managing the fleet: minimizing the size [8, 19, 41–43, 53, 54], optimizing the effectiveness of the vehicle utilization [19, 51]. Objectives can also be related to the transported goods: consideration of the passengers [8], or avoiding the deterioration of perishable products [45, 46].

3 Illustrative Multi-Objective Vehicle Routing Problems

In this section, we present two problems. The vehicle routing problem with route balancing (3.1) (VRPRB) is an extension of the capacitated vehicle routing problem (CVRP), while the bi-objective covering tour problem (3.2) (BOCTP) is a generalization of the covering tour problem (CTP). For each bi-objective problem, we present the single-objective problem from which it is derived and how the single-objective problem is transformed into the bi-objective one. We also provide a bi-objective linear integer program for each bi-objective problem.

3.1 The Vehicle Routing Problem with Route Balancing

The Capacitated Vehicle Routing Problem

The capacitated vehicle routing problem (CVRP) has been introduced by Dantzig and Ramser [13]. It can be modeled as a problem on a complete graph where the vertices are associated to a unique depot and to n customers. Each customer must be served a quantity q_i of goods ($i = 1, \ldots, n$) from the unique depot. To deliver those goods, vehicles are available. With each vehicle is associated a maximal amount Q of goods it can transport. A solution of the CVRP is a collection of routes where each customer is visited only once and the total demand for each route is at most Q. With each arc (i, j) is associated the distance between vertex i and vertex j. The CVRP aims to determine a minimal total length solution. It has been proved NP-hard [37] and solution methods range from exact methods to specific heuristics, and metaheuristics [55].

Extension of the CVRP to the VRPRB

The goal of this extension is to bring *fairness* to the problem without neglecting the economical aspect of the problem through the optimization of the length of the solution. To do so, we add a second objective to balance the length between the tours. This second objective is the minimization of the difference between the length of the longest tour and the length of the shortest tour. That way, we do not break the linearity of the problem. This bi-objective problem is called the vehicle routing problem with route balancing.

In Table 1, the seven CVRP benchmarks proposed by Christofides and Eilon [5], and Christofides *et al.* [6], are considered. Following the naming scheme used in Toth and Vigo [55], the name of each instance has the form $Ei - jk$. E means that the distance metric is Euclidean. i is the number of vertices including the depot vertex. j is the number of vehicles available. k is a character which identifies the paper where the distance data are provided. $k = e$ refers to Christofides and Eilon [5], $k = c$ to Christofides *et al.* [6]. For each instance, we report both objective values associated with the best

Table 1. Objective values for the best found solutions of Taburoute and Prins' GA.

	Taburoute		Prins's GA	
Instance	Distance	Balance	Distance	Balance
E51-05e	524.61	20.07	524.61	20.07
E76-10e	835.32	78.10	835.26	91.08
E101-08e	826.14	97.88	826.14	97.88
E151-12c	1031.17	98.24	1031.63	100.34
E200-17c	1311.35	106.70	1300.23	82.31
E121-07c	1042.11	146.67	1042.11	146.67
E101-10c	819.56	93.43	819.56	93.43

solutions obtained using Taburoute [23] and Prins's GA [48]. These methods, which can be regarded as some of the best algorithms for the CVRP, do not take into account the route balancing objective. This clearly appears in Table 1 where the best solutions are of poor quality regarding the additional objective.

Mixed-Integer Linear Program for the VRPRB

The following IP for the VRPRB is based on the IP for the CVRP proposed by Fisher and Jaikumar [21]. Let x_{ij}^k be a binary variable equal to 1 if the vehicle k visits the customer j after the customer i, 0 otherwise. Let y_i^k be a binary variable equal to 1 if the vehicle k makes the delivery to the customer i, 0 otherwise. The number of vehicles is fixed to m. Let $l_{\min}$ (respectively $l_{\max}$) correspond to the length of the shortest tour (respectively the longest tour). The VRPRB can be modeled as follows.

$$\begin{cases} \min \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k \\ \min l_{\max} - l_{\min} \end{cases} \tag{2}$$

under the constraints :

$$\sum_{i=1}^{n} q_i y_i^k \leq Q \ (k = 1, \ldots, m) \tag{3}$$

$$\sum_{k=1}^{m} y_i^k = \begin{cases} m & (i = 1) \\ 1 & (i = 2, \ldots, n) \end{cases} \tag{4}$$

$$\sum_{i=1}^{n} x_{ij}^k = y_j^k \ (j = 1, \ldots, n; k = 1, \ldots, m) \tag{5}$$

$$\sum_{i=1}^{n} x_{ij}^k = y_i^k \ (i = 1, \ldots, n; k = 1, \ldots, m) \tag{6}$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1 \ (S \subset V; 2 \leq |S| \leq n-2; k = 1, \ldots, m) \tag{7}$$

$$l_{\max} \geq \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k \ (k = 1, \ldots, m) \tag{8}$$

$$l_{\min} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^k \ (k = 1, \ldots, m) \tag{9}$$

$$x_{ij}^k \in \{0, 1\} \ (i, j = 1, \ldots, n; i \neq j; k = 1, \ldots, m) \tag{10}$$

$$y_i^k \in \{0, 1\} \ (i = 1, \ldots, n; k = 1, \ldots, m) \tag{11}$$

$$l_{\max} \geq 0 \tag{12}$$

$$l_{\min} \geq 0 \tag{13}$$

Constraints 3 insure that the capacities of the vehicles are respected. Constraints 4 deal with the fact that m vehicles go through the depot but only one vehicle visits each customer. Constraints 5 and 6 say that a customer visited by a vehicle must be served by this vehicle. Finally, constraints 7 are the subtour elimination constraints proposed by Dantzig *et al.* [12] for the traveling salesman problem.

3.2 The Bi-Objective Covering Tour Problem

The Covering Tour Problem

The covering tour problem (CTP) can be formally described as follows [24]. Let $G = (V \cup W, E)$ be an undirected graph, where $V \cup W$ is the vertex set, 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, the set of vertices that must be *covered*. A distance matrix $C = (c_{ij})$, satisfying triangle inequality, is defined for E. A final parameter is c, the size of the cover. The CTP consists in defining a tour for a subset of V, which contains all the vertices from T, of minimal length such that every node from W is covered by a visited node of V. A node $v_i \in W$ is said to be covered by a node $v_j \in V$ if $c_{ij} \leq c$. A feasible solution for a small instance is provided in Figure 5. The CTP is NP-hard since it can be reduced to the traveling salesman problem when $c = 0$ and $V = W$.

Generalization of the CTP to the BOCTP

The CTP is one of the problems pointed by Boffey as *implicit* multi-objective problems [3]. To transform the CTP into its bi-objective counterpart, the parameters and the constraints asking for the nodes in W to be at less than a given distance c from a visited node in V are removed. The other parameters and constraints of the problem are left unchanged. The objective of the BOCTP is therefore the minimization of:

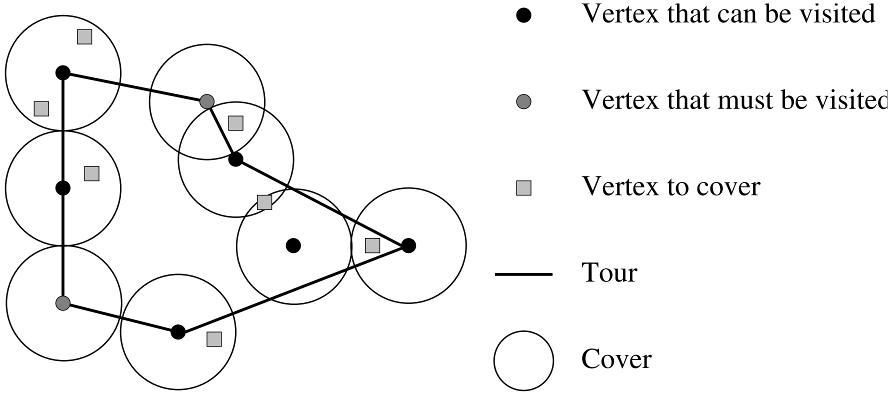

Fig. 5. An example of a solution for the covering tour problem.

1. the length of the tour
2. the cover

The cover of the solution is defined as the greatest distance between a node from $v_l \in W$ and the visited node $v_k \in V$ closest to v_l.

Mixed-Integer Linear Program of the BOCTP

The following bi-objective integer program is based on the integer program for the CTP proposed by Gendreau, Laporte, and Semet [24]. The data for the BOCTP are the same as for the CTP with the exception of the parameter c which no longer exists. The variables and constants of the program are the following ones. For every $v_k \in V$, let y_k be a binary variable equal to one if v_k is visited, 0 otherwise. For every $v_i, v_j \in V$ $(i < j)$, let x_{ij} be a binary variable equal to 1 if and only if the edge (v_i, v_j) belongs to the tour. A vector $d \in \Re^{|W|}$ is introduced. Every component d_l of d corresponds to the smallest distance necessary to cover the node $v_l \in W$. The matrix $S = (s_{lk})$ is also defined. The value of the coefficients belongs to $[1, \ldots, n]$ and the size of the matrix is $|W| \times |V|$. s_{lk} gives the index of the k^{th} closest node in V from the node $v_l \in W$. The function σ is then defined:

$$\begin{aligned} \sigma : W \times V \rightarrow & \quad [1, \ldots, n] \\ (v_l, v_k) \mapsto & \ k' \text{ such that } s_{lk'} = k \end{aligned}$$

A *high value* (HV) constant has been included to some constraints to insure that the right value is obtained. Then, the integer program is as follows:

$$\begin{cases} \min \sum_{i<j} c_{ij} x_{ij} \\ \min c_{max} \end{cases} \tag{14}$$

such that:

$$c_{max} \geq d_l \ (v_l \in W), \tag{15}$$

$$d_l \geq c_{lk} \times y_k - \mathrm{HV} \times (\sum_{i=0}^{\sigma(v_l,v_k)-1} y_{s_{li}}) \ (v_l \in W, v_k \in V), \tag{16}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2y_k \ (v_k \in V), \tag{17}$$

$$\sum_{v_i \in S, v_j \in V \setminus S \text{ or } v_j \in S, v_i \in V \setminus S} x_{ij} \geq 2y_t \ (S \subset V, 2 \leq |S| \leq n-2, T \setminus S \neq \emptyset, v_t \in S), \tag{18}$$

$$x_{ij} \in \{0,1\} \ (1 \leq i < j \leq n), \tag{19}$$

$$y_k = 1 \ (v_k \in T), \tag{20}$$

$$y_k \in \{0,1\} \ (v_k \in V \setminus T). \tag{21}$$

Constraints 15 insure that $c_{\max}$ is the maximum of the minimal values between a node in W and its closest node in V in the tour. Constraints 16 give a lower bound for d_l. Constraints 17 are degree constraints and constraints 18 are connectivity constraints.

4 A Two-Phase Strategy

In this section, we propose methods and a two-phase strategy to consider the two goals in multi-objective optimization: intensification and diversification. We also give implementation of these methods for the VRPRB and the BOCTP.

4.1 Diversification/Intensification

The principle of the strategy in two phases proposed here is a cooperation between a method able to perform a diversified search of the objective space and a method able to perform the intensification task. The strategy works as follows: the former method is first used to generate a most diversified approximation of the optimal Pareto set and then the result of this method is improved by means of the latter method to converge as far as possible toward the optimal Pareto set without losing diversification.

In the first phase, we have investigated the use of a multi-objective evolutionary algorithm (MOEA) [7, 14]. Examples of implementations of the standard MOEA NSGA II for the VRPRB and the BOCTP [15] are explained in subsection 4.2. We have also explained how parallelism can be used to enhance the results of the MOEA as well as a strategy to help diversification in MOEA, namely the elitist diversification (ED) mechanism. The second phase uses either a method to define relevant goals for neighborhood search in the case of the VRPRB or a strategy relying on the definition of subproblems that can be solved efficiently by a branch-and-cut algorithm.

4.2 Phase One: Diversification

NSGA II

NSGA II can be described as follows. Its population R_t, where t is the number of the current generation, is divided into two subpopulations P_t and Q_t. The sizes of P_t and Q_t are equal to N and, therefore, the size of R_t is $2N$. The subpopulation P_t corresponds to the parents and Q_t to the offspring. The four main steps of NSGA II are presented below without going into the details of the mechanisms used such as the ranking and the crowding distance. It is sufficient to recall that a solution i has two fitnesses according to the current population: a rank r_i which represents its quality in terms of convergence toward the optimal Pareto set, and a crowding distance d_i which corresponds to its quality in terms of diversification. A solution of rank i means that it is only dominated by solutions of rank j with $j < i$. It tends to organize the population into layers. The crowding distance is given by an approximation of the size formed by the cuboid formed by the closest solutions with the same rank than the considered solution. The lower the rank and the crowding distance are, the better the solution is. For additional details about NSGA II, the reader is referred to [15]. At generation t, the different steps are:

STEP 1 Combine the parent and offspring populations to create $R_t = P_t \cup Q_t$. Compute the ranks and crowding distances of the solutions in R_t. Sort the solution according to their ranks in an increasing order. Identify the fronts $\mathcal{F}_i$, $i = 1, \ldots, r_{\max}$, where i represents a rank. $\mathcal{F}_i$ is the set of solutions of rank i.

STEP 2 Create a new population $P_{t+1} = \emptyset$. Set $i = 1$. While $|P_{t+1}| + |\mathcal{F}_i| < N$, do $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ and $i = i + 1$.

STEP 3 Sort the solutions of $\mathcal{F}_i$ according to their crowding distance in a decreasing order. The $(N - |P_{t+1}|)$ first solutions of $\mathcal{F}_i$ (*i.e.* the most diversified solutions) are included to P_{t+1}.

STEP 4 Create Q_{t+1} from P_{t+1}.

The solution provided by NSGA II is the set of solutions not dominated in the final population R. However, experiments have shown that the size of the potentially Pareto optimal solution set can be larger than the size of the population. Therefore, we have added an archive to NSGA II whose only purpose is to save the potentially Pareto optimal solutions identified during the search. It prevents such solutions to be lost due to the stochastic behavior of the algorithm and the limited size of the population. This archive can also be used as a stopping criterion: if no new non-dominated solution has been found, *i.e.* no inclusion has been made in the archive, for a given number of generations, the search stops.

Algorithm 1 recombination_phase(P, Q: POPULATION)

```
Q ← ∅
for i ← 1, ..., N do
  pa_1 ← tournament(P ∪ C)
  pa_2 ← tournament(P ∪ C)
  if rand() < 0.5 then
    s ← RBX(pa_1, pa_2)
  else
    s ← SPLIT(pa_1, pa_2)
  end if
  if rand() < 0.4 then
    s ← or_opt(s)
  end if
  2opt_local_search(s)
  Q ← Q ∪ {s}
end for
```

NSGA II for the VRPRB

To implement NSGA II for a given problem, it is necessary to describe the population initialization strategy as well as the recombination phase. For the VRPRB, the starting population was generated by means of a greedy heuristic. The greedy algorithm works by adding the customers in a random order. A new route is created when the previous one is full. When every node is included, a 2-opt local search is applied on every tour.

The recombination phase is described in Algorithm 1. The tournament operator is the binary tournament as described by Deb *et al.*. Two solutions are randomly selected and the solution with the best rank is kept. To break the tie, the solution with the greatest crowding distance is selected. The crossover operators are the route based crossover (RBX) [47] and the SPLIT crossover [27, 29] inspired by Prins's genetic algorithm [48]. When a solution is created, a 2-opt local search is applied on each route in order to avoid artificially balanced solutions [27, 29]. For further details, the reader is referred to [31].

NSGA II for the BOCTP

The starting population is created by solving several CTP for different values of c. These values are chosen so that the points are equally distributed in the objective space between the lowest cover value, which happens when only the nodes in T are visited, and the highest, which can be computed for instance when every node in V is visited. The different CTPs are solved by means of a heuristic for the problem proposed by Gendreau, Laporte, and Semet [24].

During the recombination phase, two parents are selected and one offspring is generated. The purpose of the crossover is to select the nodes visited by the tour without building the tour. The crossover works by adding nodes that appear at least in one of the parents. Only the nodes that can improve the cover are considered at each iteration. An acceptation probability is used to determine if a node is to be included. The probability is computed in a similar way to the crossover fusion for the set covering tour problem of Beasley and Chu [2]. More details on the complete implementation of NSGA II for the BOCTP can be found in [30, 33].

Parallelization

To improve the results obtained by NSGA II, we have implemented it in an island model. The model is built as follows: each island corresponds to one instantiation of NSGA II with its own population. The communication network is a ring, and therefore each island has two neighbors. One island sends information to its neighbors regularly in terms of generations. When the generation corresponds to a communication phase, the recombination (STEP 4) is replaced by the migration between the islands. Due to the fact that the communication network is a ring, an island receives information at the same time it sends information. The computations of a given island do not start again until it has received the information from its two neighbors.

The communication phase runs as follows. An island sends to its two neighbors the $\frac{N}{2}$ best solutions from its population (*i.e.* the $\frac{N}{2}$ first solutions, according to the ranking and crowding distance sorting, of the population after the selection phase (STEP 1 to STEP 3)). Therefore, an island receives $\frac{N}{2}$ solutions twice. These solutions replace those from Q_t since they would have been lost in the case of a standard recombination phase. Figure 6 illustrates the communications in the case of four islands.

The Elitist Diversification Mechanism

In the elitist diversification, additional archives are considered. They contain the potentially optimal Pareto solutions (PPS) when one objective is maximized instead of being minimized. It may be noted that we suppose that every objective is to be minimized. Let $S(A)$ be the subset of solutions of the decision space found by an algorithm A, and k the index of the objective function component which is maximized. To define new archives, the dominance operator $\prec_k$ is introduced :

$$\begin{aligned}\forall y, z \in S(A), y \prec_k z \Longleftrightarrow & (\forall i \in \{1 \ldots n\} \setminus \{k\}, f_i(y) \leq f_i(z)) \\ & \wedge (f_k(y) \geq f_k(z)) \\ & \wedge ((\exists i \in \{1 \ldots n\} \setminus \{k\}, f_i(y) < f_i(z)) \\ & \quad \vee (f_k(y) > f_k(z)))\end{aligned}$$

Then, we have $A_k = \{s \in S(A) | \forall s' \in S(A), s' \not\prec_k s\}$, with $k = 1, \ldots, n$, the archive of PPS associated with the maximization of the k^{th} objective

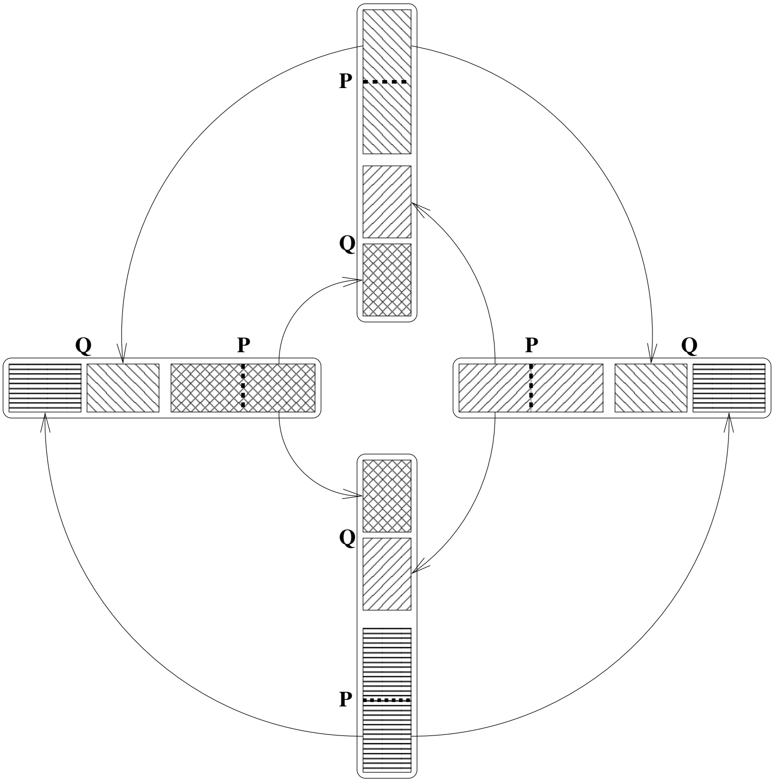

Fig. 6. Extension of NSGA II into an island model.

component instead of the minimization. We denote $\prec_0$ the classical dominance operator *i.e.* a solution x is said to dominate a solution y if x is not worse than y on every objective and there is at least one objective where x is strictly better than y.

Like in the elitism strategy, solutions from the new archives are included into the population of the MOEA at each generation. The role of these solutions is to attract the population to unexplored areas, and so to avoid the premature convergence to a specific area of the objective space. Indeed, using solutions from these archives ensures that an exploration is done while preferring one objective.

Preliminary experiments point out that the improvement is less important when all the archives are embedded in the same MOEA. This leads us to distribute the archives among several searches resulting in a co-operative model. This parallelization is not used in order to speed up the search but to search a larger part of the solution space in a given time. Since every island will be executed at the same time, it will take the same computational time as a single island while the number of solutions created will be multiplied by

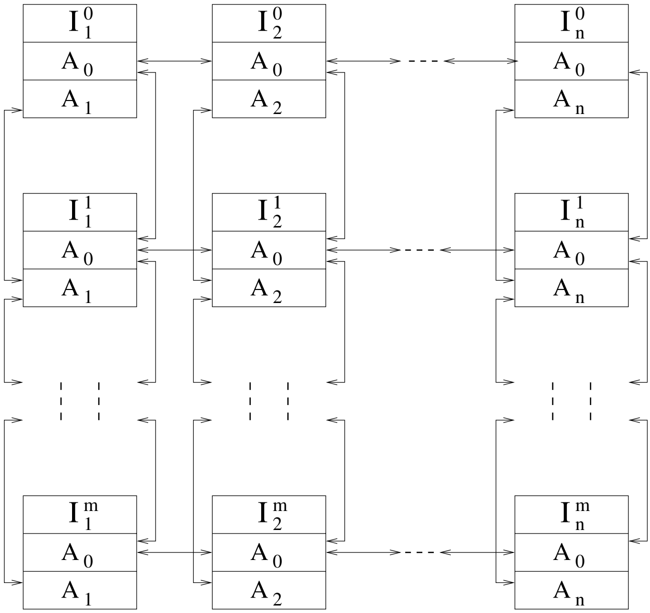

Fig. 7. The complete co-operative model - the toric structure is not shown in order not to obfuscate the figure.

the number of islands. An island is denoted I_j^i. It means it belongs to the i^{th} brick and its additional archive is of A_j type. The island I_j^i sends its A_0 archive to all its neighbors: I_{j-1}^i, I_{j+1}^i, I_j^{i-1}, and I_j^{i+1}. It only communicates its A_j archive to I_j^{i-1} and I_j^{i+1}. Since the communication topology between and within the bricks is toric, the indexes are computed modulo n. The model is illustrated in Figure 7.

This strategy can be used with any MOEA. Its implementation in the case of NSGA II is explained in [31].

4.3 Phase Two: Intensification

During the second phase, we try to improve the intensification aspect of the potentially optimal Pareto sets generated by a MOEA (NSGA II in the case of the studies on the VRPRB and the BOCTP). The fact is that MOEA may not always work well in view of the intensification goal. It may be interesting to use *good* single-objective methods. Indeed, apart from some problems specific to a given real case study, the multi-objective problems are often linked to single-objective problems studied in the literature and for which efficient methods have been proposed. For instance, in the case of the VRPRB, a lot of neighborhood searches, specially Tabu search, exist and for the BOCTP, an exact algorithm have been proposed. The purpose here is to transform the multi-objective problem into a single-objective problem and use good methods for the single-objective problems to improve the potentially optimal Pareto set in terms of intensification. First, we propose a way to build relevant goal points according to the approximation at hand. We also explain how this method was used to post-optimized approximations generated by NSGA II for the VRPRB. In the case of the BOCTP, we illustrate how it can be useful to generate subproblems which can be solved exactly by means of a branch-and-cut algorithm for the CTP.

Target Aiming Pareto Search

We are going to present a strategy called Target Aiming Pareto Search (TAPaS) whose general loop works as follows. First, points are selected from a given potentially Pareto set. Then, for each of these points a goal point is built according to the other selected points. A local search heuristic is then run from each selected point trying to reach the associated goal point. The approximation is updated using the solutions visited by the local search heuristics. The process is iterated until no new non-dominated solutions are found.

More precisely in TAPaS, a local search heuristic l_i is applied to each solution s_i of a potentially Pareto set P. A specific objective function o_i is defined for each local search l_i. The function o_i must take into account the multiplicity of the LS invoked.

Indeed, two LS should not examine the same area of the objective space, and the entire area that dominates P should be explored in order to converge toward the optimal PS. The definition of o_i is based on the partition of O (*i.e.*, the objective space) according to P (Figure 8):

- $A_d = \{s \in O | \exists s' \in P, s' \prec s\}$: the area of the objective space dominated by P.
- $A_{nd} = \{s \in O | \forall s' \in P, (s' \not\prec s) \wedge (s \not\prec s')\}$: the area of the objective space not dominated by P and not dominating any solution from P.
- $A_s = \{s \in O | \exists! s' \in P, s \prec s'\}$: the area of the objective space dominating only one solution of P.
- $A_p = \{s \in O | \exists s_1, s_2 \in P, (s \prec s_1) \wedge (s \prec s_2)\}$: the area of the objective space dominating more than one solution of P.

Each solution $s_i \in P$ is associated with a part A_s^i of A_s. If l_i is able to generate a feasible solution in A_s^i, then the approximation is improved according to the convergence, without impoverishing the diversification. To guide the search, a goal g_i is given to each l_i, with g_i being the point that dominates all points of A_s^i. In cases where certain coordinates of g_i cannot be defined (*e.g.* the extremities of P), a lower bound for the missing coordinates should be used.

Each local search stops when it reaches a solution that dominates g_i or when a stopping criterion specific to the implementation is met. Each local search l_i produces an archive a_i which contains all the current solutions that

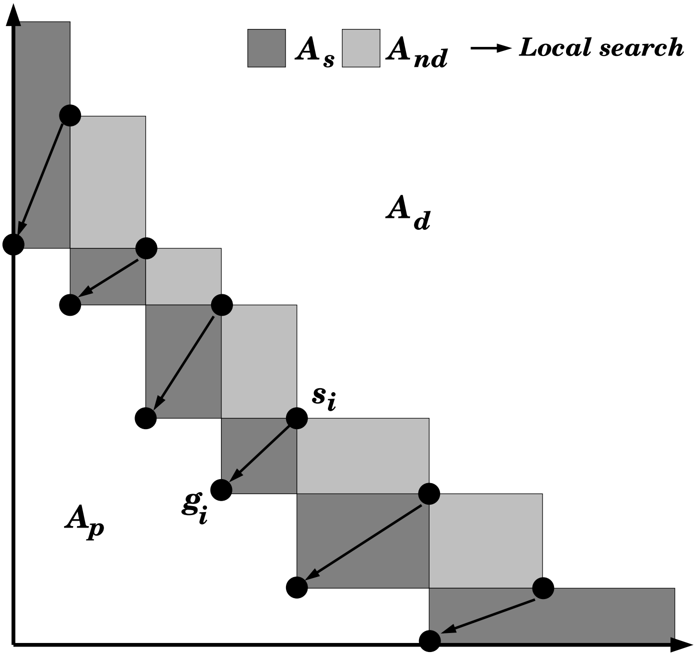

Fig. 8. Partition of O.

are not dominated. P is updated by the Pareto union between P and all the archives a_i. If P is improved, the process is iterated, otherwise it is stopped.

This strategy has been applied to the VRPRB. The starting approximations were generated by means of NSGA II. The local search was a Tabu search : Unified Tabu Search [9, 10]. More details on TAPaS and this implementation are given in [32].

Definition of Subproblems

Another possibility is to define single-objective subproblems that can be solved efficiently by single-objective methods. The interest to define smaller size problems comes from the fact that if one wants to improve a potentially optimal Pareto set by means of single-objective methods, one will need to run the methods several times in order to take into account the spread of the Pareto frontier. Therefore, smaller size subproblems, being easier to solve, enable to iterate several times the single-objective methods without having to pay a prohibitive computational time.

In the case of the BOCTP, we have used the approximation generated by NSGA II by defining several subproblems and solving them by means of a branch-and-cut algorithm for the CTP [24]. Two subset construction procedures have been defined. The first one tends to improve the solutions according

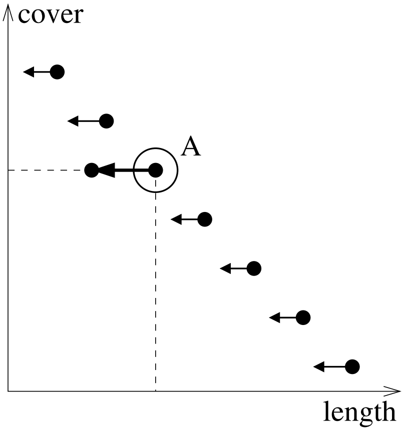

Fig. 9. The first procedure tries to improve a solution length without modifying the cover (*i.e.*, the solution A).

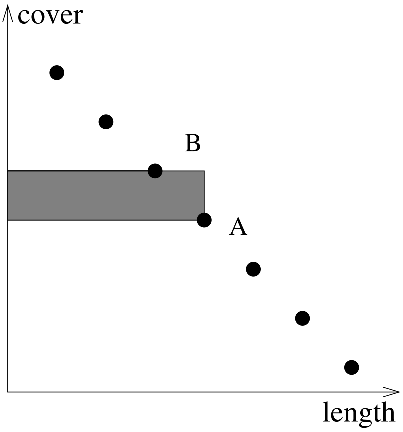

Fig. 10. For a given couple A and B, the second procedure builds a subproblem to which the solution is necessarily in the grey area.

to one objective (minimization of the length) without touching the cover. The second procedure tends to identify potentially Pareto optimal solutions whose cover values were not found by NSGA II.

The first procedure works as follows (see Figure 9). Let consider one solution s found by NSGA II visiting the set $V' \subseteq V$. For every node v_k visited in the solution, we identify a set R_k of nodes that can replace v_k. R_k is said to be able to replace v_k if the removal of v_k and the inclusion of the nodes in R_k do not change the cover value associated to the solution. The size of R_k is bounded by a small number of nodes. Then, the subproblem (*i.e.* the CTP) associated to the solution is the following one. The set of visitable nodes is formed by the union of V' and the subsets R_k formed for every node $v_k \in V'$. The set of nodes to cover is still W and c is fixed to the cover value of s.

The second subproblem construction procedure works as follows (see Figure 10). Let A and B be two neighboring solutions in the approximation sets found by the evolutionary algorithm (*i.e.* there is no other solution 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 as a parameter, the first cover $\tilde{c}$ which is strictly smaller than c_B.

More information on these procedures and their cooperation with NSGA II for the BOCTP can be found in [32].

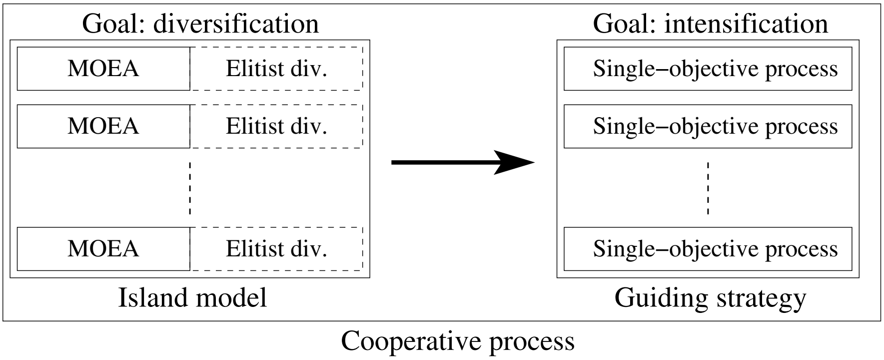

Fig. 11. General optimization strategy.

4.4 General Optimization Strategy

Our general optimization strategy is illustrated in Figure 11. The deepest layer of this strategy is the use of a MOEA (typically in our studies NSGA II) to generate a first approximation. The MOEA can be improved by the addition of the elitist diversification mechanism and/or the use of parallelization through an island model. This general method forms the component dedicated to the diversification of the optimization strategy. Then, the approximation generated is passed to the diversification box, which is composed of one or several intensification procedures based on single-objective methods. These procedures can be combined to ensure an *intelligent* search of the objective space by means of techniques designed to guide the searches like TAPaS or the generation of subproblems to solve them by exact methods in a cooperative scheme.

4.5 Computational Results

Computational Results for the VRPRB

Optimal Pareto sets are not known for the VRPRB. Therefore, we have compared the results of our MOEA with the best-known values on the length objective and with the evident lower bound that is 0 for the balance objective. The different methods reported are a parallel version of NSGA II run on 16 processors, a parallel version of NSGA II with the elitist diversification mechanism run on 8 processors and a cooperative method composed of NSGA II (run on one processor) and TAPaS as a post-optimization process. Table 2 allows to compare our method with the only-known element from the literature: the best-known length. It is interesting to note on the instances E76-10e and E101-08e that if the solutions are near each other in terms of length, there is an important gap concerning the balance. For instance, the best found solution found by TabuRoute and Prins's GA for the instance E101-08e has a balance value of 97.88 for a length of 826.14 whereas the best solution we found has a length of 827.39 for a balance of 67.55. It appears that to gain very little in length (0.15 %), it is necessary to lose a lot in balancing. Therefore, it is possible that there is a great difference between the solutions. As it is also the case for the instances E151-12c and E200-17c, it can explain that the multi-objective metaheuristics have difficulties to reach the best-known solutions in terms of length.

Regarding the balance, all the methods found solutions with values close to 0. They are able to generate very well-balanced solutions. However, there is an important fluctuation concerning the length of the solutions. We also provide the average number of potentially Pareto solutions found by the island model of NSGA II on 16 processors (#PPS). It shows that the Pareto sets are rich in terms of solutions and it indicates experimentally that the study of these two objectives together is interesting.

More computational results, including the evaluation of the contribution of the different mechanisms or the fact that diversity is preserved by the new intensification mechanisms can be found in [27, 29, 31, 32].

Computational Results for the BOCTP

For the BOCTP, it was possible to generate the optimal Pareto set by iterating the branch-and-cut algorithm of Gendreau, Laporte, and Semet [24]. To avoid useless computation, the iteration was done through an ϵ-constraint method insuring that only one run of the branch-and-cut algorithm was done for each solution in the optimal Pareto set [32].

The contribution of each cooperation scheme was evaluated. Table 3 reports the average ratio (Ratio) of optimal Pareto solutions found, computational times in seconds (time) for NSGA II alone 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 NSGA II to generate a first approximation and then to apply a cooperative scheme. Cooperative scheme I seems to produce interesting results. Overall, it is able to identify an average of five percent of new optimal Pareto solutions. 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. Moreover, the cumulated contribution of the two cooperative schemes was evaluated (Table 3; heading Cooperation I+II). It appears that the quality of the results using both schemes is better than if only one was used, both in terms of the number of optimal Pareto solutions found (an improvement ratio of 7.2 on average). Furthermore, the additional cumulated computational time is moderate.

The computational results showed that the cooperative method composed of NSGA II and the resolutions of the subproblems by means of the

Table 2. Best-found bounds for both objectives of the VRPRB for several methods and average number of non dominated solutions found.

Instance	#PPS	Method	Total length				Balance	
			Best-known	Best-found	%	Balance	Best-found	Length
E51-05e	50.4	NSGA II (16 proc.)	524.61	524.61	0.00	20.07	0.18	779.88
		NSGAED (8 proc.)		524.61	0.00	20.07	0.32	645.29
		NSGA II + TAPaS		524.61	0.00	20.07	0.19	633.58
E76-10e	135.2	NSGA II (16 proc.)	835.26	835.32	0.01	78.10	0.48	1153.25
		NSGAED (8 proc.)		835.32	0.01	78.10	0.33	1380.01
		NSGA II + TAPaS		835.32	0.01	78.10	0.65	1665.13
E101-08e	185.8	NSGA II (16 proc.)	826.14	827.39	0.15	67.55	0.27	1132.28
		NSGAED (8 proc.)		827.39	0.15	67.55	0.10	1610.86
		NSGA II + TAPaS		827.39	0.15	67.55	0.11	1665.13
E151-12c	408.5	NSGA II (16 proc.)	1028.42	1032.95	0.44	97.57	0.44	1551.50
		NSGAED (8 proc.)		1034.91	0.63	109.06	0.21	1523.94
		NSGA II + TAPaS		1029.15	0.07	107.39	0.09	2504.54
E200-17c	479.7	NSGA II (16 proc.)	1291.45	1328.44	2.86	96.61	0.86	3490.14
		NSGAED (8 proc.)		1327.89	2.82	97.20	0.58	3746.57
		NSGA II + TAPaS		1303.57	0.93	79.41	0.59	2064.44
E121-07c	921.1	NSGA II (16 proc.)	1042.11	1042.11	0.00	146.67	0.07	2296.43
		NSGAED (8 proc.)		1042.11	0.00	146.67	0.07	1649.76
		NSGA II + TAPaS		1042.11	0.00	146.67	0.03	2207.95
E101-10c	499.6	NSGA II (16 proc.)	819.56	819.56	0.00	93.43	0.21	1324.53
		NSGAED (8 proc.)		819.56	0.00	93.43	0.35	1664.80
		NSGA II + TAPaS		819.56	0.00	93.43	0.24	1233.52

branch-and-cut algorithm are able to generate very good solutions (at least two thirds of the optimal Pareto solutions were found and the approximations were very close to the optimal Pareto sets) while requiring much less time (around 9 times faster than the iteration of the branch-and-cut algorithm on the complete set of benchmark). We also provide the average number of optimal Pareto solutions (NB) on the complete set of instances for each

Table 3. Contribution of the cooperative schemes.

$\|V\|\backslash\|T\|\backslash\|W\|$	NB	MOEA		Cooperation I			Cooperation II			Coop. I+II	
		Ratio	time	Ratio	time	TT	Ratio	time	TT	Ratio	TT
100\1\100	124.8	0.65	481.9	0.67	80.7	562.6	0.65	34.9	516.8	0.67	597.5
100\1\200	131.4	0.66	668.1	0.70	102.7	770.8	0.69	120.5	788.6	0.73	891.3
100\1\300	122.4	0.63	770.7	0.69	131.5	902.2	0.65	113.2	883.9	0.70	1015.4
100\10\100	61.4	0.74	439.5	0.80	30.5	470.0	0.77	7.5	447.9	0.82	477.7
100\10\200	56.7	0.77	416.1	0.82	33.1	449.2	0.78	8.8	424.9	0.83	458.0
100\10\300	82.4	0.72	854.6	0.76	60.6	915.2	0.76	23.5	878.1	0.78	938.7
100\20\100	31.4	0.69	448.6	0.76	18.6	467.2	0.73	3.5	452.1	0.78	470.7
100\20\200	44.2	0.69	570.4	0.76	30.9	604.1	0.74	10.0	580.4	0.79	614.1
100\20\300	34.4	0.61	560.3	0.68	20.6	580.9	0.65	7.6	567.9	0.73	588.5
120\12\120	52.2	0.80	541.8	0.84	55.7	597.5	0.82	8.2	550.0	0.86	605.7
120\12\240	83.2	0.69	1119.0	0.74	97.7	1206.7	0.73	24.2	1243.2	0.77	1230.9
120\12\360	93.6	0.51	1286.5	0.55	117.6	1404.1	0.54	35.2	1321.7	0.57	1439.3
120\24\120	35.2	0.68	529.2	0.75	37.9	567.1	0.69	6.8	536.0	0.76	573.9
120\24\240	52.6	0.64	863.8	0.69	75.7	939.5	0.68	13.3	877.1	0.72	952.8
120\24\360	50	0.76	871.6	0.78	53.8	925.4	0.80	18.4	890.0	0.81	943.8

combination of the sizes of V, T, and W. Additional computational results, including a complete evaluation of the contribution of the cooperation and tests on real-life data, can be found in [33].

5 Conclusions

In this chapter, we have given an overview of what can be done by using multiple objectives for vehicle routing problems as well as how it can be done by providing the keys to a general strategy designed to deal with the challenges encountered in multi-objective optimization. This has been illustrated by the definition of two bi-objective vehicle routing problems and the implementation of the proposed strategy for these problems.

Vehicle routing problems are important as academic problems as well as problems appearing in a lot of real-life situations. It is therefore natural and important to get interested in the definition of multi-objective vehicle routing problems and the application of multi-objective optimization methods to these problems. Both problems studied here are typical of what can, and we believe, what should be done.

In conclusion, multi-objective optimization and multi-objective vehicle routing problems open new horizons of research for studies in vehicle routing. The area knows a fast growing interest. Indeed, more than half of the papers about this subject has been published after 2000. However, even more efforts should be put into these studies. It appears that if the number of studies on multi-objective vehicle routing problems has increased in the recent years, almost each study is made independently from the others. However, some studies could be linked together and, in some cases, different studies deal with the same or almost the same multi-objective problem. Even if the complete problem is not considered, several objectives are shared by different studies. It should therefore be interesting to define *general* multi-objective vehicle routing problems that could be used as starting points for more complex problems. For instance, an analogy with single-objective vehicle routing problems is the traveling salesman problem and the capacitated vehicle routing problem which are the focuses of a lot of academic studies and have a lot of real-life applications, specially through the definition of variants. However, the methods used to solve these variants are based on those obtained for the academic formulations of the problems.

Acknowledgement

This work was partially supported by the Nord-Pas-de-Calais Region and the European funding program FEDER. This support is gratefully acknowledged. Thanks are also due to the referees for their valuable comments.

References

1. B. Baràn and M. Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. In *Proceedings of the twenty-first IASTED International Conference on Applied Informatics*, pages 97–102, 2003.
2. J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering tour problem. *European Journal of Operational Research*, 94:392–404, 1996.
3. B. Boffey. Multiobjective routing problems. *Top*, 3:167–220, 1995.
4. R. Bowerman, B. Hall, and P. Calamai. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A*, 29:107–123, 1995.
5. N. Christofides and S. Eilon. An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, 20:309–318, 1969.

6. N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors. *Combinatorial Optimization*, chapter 11. John Wiley, Chichester, 1979.
7. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, editors. *Evolutionary algorithms for solving multi-objective problems.* Kluwer Academic Press, 2002.
8. A. Corberan, E. Fernandez, M. Laguna, and R. Marti. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53:427–435, 2002.
9. J-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle problems. *Networks*, 30:105–119, 1997.
10. J-F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–932, 2001.
11. J. R. Current and D. A. Schilling. The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, 73:114–126, 1994.
12. G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2:393–410, 1954.
13. G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6:81–91, 1959.
14. K. Deb. *Multi-objective optimization using evolutionary algorithms.* John Wiley and Sons, 2001.
15. K. Deb, A. Pratap, S. Agarwal, and T. Meyarvan. A fast and elitist multiobjective genetic algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
16. K. Doerner, A. Focke, and W. J. Gutjahr. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research*, 179:1078–1096, 2007.
17. F. Y. Edgeworth. *Mathematical physics.* P. Keagan, London, England, 1881.
18. M. Ehrgott. Approximation algorithms for combinatorial multi-criteria problems. *International Transactions in Operations Research*, 7:5–31, 2000.
19. N. El-Sherbeny. *Resolution of a vehicle routing problem with multi-objective simulated annealing method.* PhD thesis, Faculté Polytechnique de Mons, Mons, Belgique, 2001.
20. D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 36:188–205, 2005.
21. M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem. *Networks*, 11:109–124, 1981.
22. M. J. Geiger. *Multi-criteria und Fuzzy Systeme in Theorie und Praxis*, chapter A computational study of genetic crossover operators for multi-objective vehicle routing problem wih soft time windows, pages 191–207. Deutscher Universitats-Verlag, 2003.
23. M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40:1276–1290, 1994.
24. M. Gendreau, G. Laporte, and F. Semet. The covering tour problem. *Operations Research*, 45:568–576, 1997.
25. I. Giannikos. A multiobjective goal programming model for locating treatment sites and routing hazardous wastes. *European Journal of Operational Research*, 104:333–342, 1998.

26. S-C. Hong and Y-B. Park. A heuristic for a bi-objective vehicle routing with time window constraints. *International Journal of Production Economics*, 62:249–258, 1999.
27. N. Jozefowiez. *Modélisation et résolution approchées de problèmes de tournées multi-objectif*. PhD thesis, Laboratoire d'Informatique Fondamentale de Lille, Université des Sciences et Technologies de Lille, Villeneuve d'Ascq, France, December 2004.
28. N. Jozefowiez, M. Laguna, and F. Glover. A multi-objective evolutionary algorithm for the bi-objective traveling salesman problem with profits. In *MOPGP'06: 7th international conference on MultiObjective programming and Goal Programming*, 2006.
29. N. Jozefowiez, F. Semet, and E-G. Talbi. Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In J.J. Merelo Guervos et al., editors, *PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 271–280. Springer-Verlag, September 2002.
30. N. Jozefowiez, F. Semet, and E-G. Talbi. *Applications of multi-objective evolutionary algorithm*, volume 1 of *Advance in Natural Computation*, chapter A multi-objective evolutionary algorithm for the covering tour problem, pages 247–267. World Scientific, 2004.
31. N. Jozefowiez, F. Semet, and E-G. Talbi. Enhancements of NSGA II and its application to the vehicle routing problem with route balancing. In E-G. Talbi, P. Liardet, P. Collet, E. Lutton, and M. Schoenauer, editors, *Artificial Evolution 2005, 7^{th} International Conference (EA'2005)*, volume 3871 of *Lecture Notes in Computer Science*, pages 131–142. Springer, 2006.
32. N. Jozefowiez, F. Semet, and E-G. Talbi. Target Aiming Pareto Search and its application to the vehicle routing problem with route balancing. *Journal of Heuristics*, 2006. (available online).
33. N. Jozefowiez, F. Semet, and E-G. Talbi. The bi-objective covering tour problem. *Computers and Operations Research*, 34:1929–1942, 2007.
34. N. Jozefowiez, F. Semet, and E-G. Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 2007. (available online).
35. P. Lacomme, C. Prins, and M. Sevaux. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers and Operations Research*, 33:3473–3493, 2006.
36. T-R. Lee and J-H. Ueng. A study of vehicle routing problem with load balancing. *International Journal of Physical Distribution and Logistics Management*, 29:646–648, 1998.
37. J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problem. *Networks*, 11:221–227, 1981.
38. W. Li. Finding pareto-optimal set by merging attractors for a bi-objective traveling salesman problem. In C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 797–810, Guanajanto, Mexico, march 2005. Springer.
39. M. Mourgaya. *The periodic vehicle routing problem: planning before routing.* PhD thesis, Laboratoire de Mathématiques Appliquées de Bordeaux, Université de Bordeaux 1, Bordeaux, France, July 2004.
40. M. Mourgaya and F. Vanderbeck. Periodic vehicle routing problem: Classification and heuristic. *RAIRO - Operations Research*, 2005. to appear.

41. T. Murata and R. Itai. Multi-objective vehicle routing problems using two-fold EMO algorithm to enhance solution similiraty on non-dominated set. In C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 885–896, Guanajanto, Mexico, March 2005. Springer.
42. B. Ombuki, B. J. Ross, and F. Hanshar. Multi-objective genetic algorithm for vehicle routing problem with time windows. *Applied Intelligence*, 24:17–30, 2006.
43. J. Pacheco and R. Marti. Tabu search for a multi-objective routing problem. *Journal of the Operational Research Society*, 57:29–37, 2006.
44. V. Pareto. *Cours d'économie politique*. Rouge, Lausanne, Suisse, 1896.
45. Y. Park and C. Koelling. A solution of vehicle routing problems in multiple objective environment. *Engineering Costs and Production Economics*, 10:121–132, 1986.
46. Y. Park and C. Koelling. An interactive computerized algorithm for multicriteria vehicle routing problems. *Computers and Industrial Engineering*, 16:477–490, 1989.
47. J.-Y. Potvin and S. Bengio. The vehicle routing problem with time windows part II: Genetic search. *INFORMS Journal on Computing*, 8:339–370, 1996.
48. C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31:1985–2002, 2004.
49. R. Ribeiro and H. R. Lourenço. A multi-objective model for a multi-period distribution management problem. In *Metaheuristic International Conference 2001 (MIC'2001)*, pages 91–102, 2001.
50. J. Riera-Ledesma and J. J. Salazar-González. The biobjective travelling purchaser problem. *European Journal of Operational Research*, 160:599–613, 2005.
51. W. Sessomboon, K. Watanabe, T. Irohara, and K. Yoshimoto. A study on multi-objective vehicle routing problem considering customer satisfaction with due-time (the creation of Pareto optimal solutions by hybrid genetic algorithm). *Transaction of the Japan Society of Mechanical Engineering*, 1998.
52. R. E. Steuer. *Multiple criteria optimization: Theory, computation, and application*. Krieger Pub Co, 1986.
53. K. C. Tan, Y. H. Chew, and L. H. Lee. A hybrid multi-objective evolutionary algorithm for solving vehicle routing problem with time windows. *European Journal of Operational Research*, 34:115–151, 2006.
54. K.C. Tan, Y.H. Chew, and L.H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172:855–885, 2006.
55. P. Toth and D. Vigo, editors. *The vehicle routing problem*, volume 9 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 2001.
56. Y. Zhenyu, L. Zhang, K. Lishan, and L. Guangming. A new MOEA for multi-objective TSP and its convergence property analysis. In C. M. Fonseca et al., editors, *Evolutionary Multi-criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 342–354. Springer-Verlag, 2003.
57. K. G. Zografos and K. N. Androustsopoulos. A heuristic algorithm for solving hazardous material distribution problems. *European Journal of Operational Research*, 152:507–519, 2004.