Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 195 (2009) 761–769

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

An evolutionary algorithm for the vehicle routing problem with route balancing

Nicolas Jozefowiez [a,b], 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, 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*

Received 14 January 2007; accepted 16 June 2007
Available online 4 December 2007

Abstract

In this paper, we address a bi-objective vehicle routing problem in which the total length of routes is minimized as well as the balance of routes, i.e. the difference between the maximal route length and the minimal route length. We propose a meta-heuristic method based on an evolutionary algorithm involving classical multi-objective operators. To improve its efficiency, two mechanisms, which favor the diversification of the search, have been added. First, an elitist diversification mechanism is used in cooperation with classical diversification methodologies. Second, a parallel model designed to take into account the elitist diversification is proposed. Our method is tested on standard benchmarks for the vehicle routing problem. The contribution of the introduced mechanisms is evaluated by different performance metrics. All the experimentations indicate a strict improvement of the generated Pareto set.

Keywords: Routing; Multi-objective optimization; Genetic algorithms; Parallel algorithms

1. Introduction

This paper investigates methods for solving multi-objective combinatorial optimization (MOCO) problems. We are specially interested in solving a bi-objective vehicle routing problem (VRP). The elementary version of the vehicle routing problem is the capacitated vehicle routing problem (CVRP). It can be modeled as a problem on a complete graph where the vertices are associated to a unique depot and to m customers. Each customer must be served a quantity q_i of goods ($i = 1, \ldots, m$) 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 [1] and solution methods range from exact methods to specific heuristics, and meta-heuristics [2].

Multi-objective vehicle routing problems have been more and more studied over these past years [3]. These studies are motivated by a need to extend or generalize existing problems or to solve problems clearly defined as multi-objective ones by decision-makers. Among the different objectives considered in the literature, some objectives are designed to even out disparities between the tours. Such objectives are often introduced in order to bring an element of *fairness* into play. To define a *balancing* objective, it is necessary to define the workload for a tour, which can be expressed as the number of customers visited, the quantity of delivered goods, the tour length or the required time, for example. Example of studies solving problems containing balancing objectives can be found in [4–7]. In this paper, we consider an extension of the CVRP, called the vehicle

* Corresponding author. Tel.: +33 3 59577888; fax: +33 3 59577850.
E-mail address: talbi@lifl.fr (E.-G. Talbi).

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2007.06.065

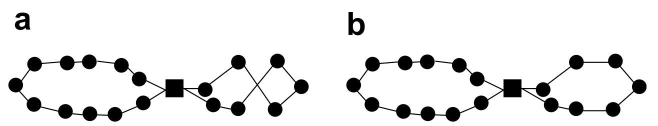


Fig. 1. (a) is better-balanced than (b), but (b) does not artificially improve the balance.

routing problem with route balancing (VRPRB). In the VRPRB, two objectives are optimized:

(1) Minimization of the distance traveled by the vehicles.
(2) Minimization of the difference between the longest route length and the shortest route length.

Note that each tour of the solutions should be optimal in terms of length to avoid artificially balanced solutions (see Fig. 1). Usual methods such as Taburoute [8] or Prins' GA [9], which can be regarded as some of the best algorithms for the CVRP, do not take into account the route balancing objective. As shown in our computational results, the solutions of these methods are of poor quality if the balance is considered.

Our solution is based on a meta-heuristic approach to approximate the Pareto set. Our choice of meta-heuristics is motivated by the difficulty of solving the problem with exact approaches. Since a Pareto set has to be generated, population based methods like multi-objective evolutionary algorithms (MOEA) [10,11] seem well-fitted. To obtain well-diversified approximation of the Pareto set, the authors use diversification mechanisms [12]. To improve the efficiency of MOEA, parallelization of algorithms is proposed by some researchers [13–15]. In this paper, we propose a MOEA for the VRPRB which includes two new mechanisms to improve the quality of the approximation of the Pareto set. The first mechanism, called the elitist diversification, is a diversification technique which is used in addition of a classic diversification mechanism. The elitist diversification naturally induces a co-operative model which is extended into an island model, which is the second mechanism used. The island model is defined to incorporate the main features of the elitist diversification, i.e. the additional archives.

The paper is organized as follows. Section 2 introduces the new mechanisms proposed for the solution of a multi-objective problem. In Section 3, we specify the implementation of both mechanisms and we present the main components of the MOEA devised for the VRPRB. In Section 4, we assess the efficiency of the new mechanisms and of the MOEA with different indicators on a set of standard benchmarks. Conclusions are drawn in Section 5.

2. New mechanisms for MOEA

We have defined two new mechanisms for the solution of a multi-objective problem using a MOEA. A standard iteration of a MOEA usually consists of the following steps:

(1) *Evaluation*: The *fitness* (quality) of the individuals in the population is evaluated. Typically, two kinds of information are computed for each individual. The first one determines its quality in terms of convergence and the second one is related to the diversification.
(2) *Parent selection*: Solutions (*parents*) are selected according to their fitness. Non-dominated solutions kept in an archive are also considered.
(3) *Crossover*: Solutions selected during the previous step are combined to produce solutions (*offspring*).
(4) *Mutation*: The offspring is randomly modified with a given probability.
(5) *Population update*: Parents and offspring are merged in the current population. All the solutions are not necessarily kept. The archive is also updated by removing the solutions dominated by one offspring and including the non-dominated offspring.

Here, we propose two mechanisms. The first one, the *elitist diversification*, modifies the parent selection step as solutions not in the current population or in the non-dominated archive are considered to become parents. The second mechanism, the parallelization, modifies the population update by adding to the archives solutions from archives of other MOEAs. Before describing these mechanisms, we introduce some useful definitions.

2.1. Solution of a multi-objective problem

A multi-objective problem can be stated as follows:

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

with $n \geqslant 2$ 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 points of the objective space, and $y = (y_1, y_2, \ldots, y_n)$, where $y_i = f_i(x)$, is a point of the objective space. The solution of a multi-objective problem (MOP) is the set of the non-dominated solutions called the Pareto set (PS). The dominance is defined as follows:

Definition 2.1. A solution $y = (y_1, y_2, \ldots, y_r)$ dominates (denoted $\prec$) a solution $z = (z_1, z_2, \ldots, z_r)$ if and only if $\forall\, i \in \{1, \ldots, n\},\; f_i(y) \leqslant f_i(z)$ and $\exists\, i \in \{1, \ldots, n\},\; f_i(y) < f_i(z)$.

Definition 2.2. A solution y found by an algorithm A is said to be a potential Pareto optimal solution (PPS), relatively to A, if A does not find a solution z so that z dominates y.

Since solving a MOP is in general NP-hard, exact algorithms tend to be disregarded and methods providing an approximation of the Pareto set are considered. Evolutionary algorithms and local search [16] methods were generally proposed to approximate PS. These heuristics must be designed according to two purposes: (i) the solutions found by the algorithm must converge toward the PS and (ii) the identified solutions should be well diversified along the frontier.

Different mechanisms were proposed to realize these two tasks. For example, to favor the convergence, the solutions can be arranged according to the Pareto dominance. This is what *ranking* methods do [12]. The diversification can be achieved by ecological niche methods like sharing [17], nearest neighbor methods like crowding [18], and histograms [19]. However, for hard problems or large scale benchmarks, these methods may not be sufficient.

In what follows, we describe new mechanisms to perform more efficiently the convergence and diversification tasks. To improve the diversification, we propose the elitist diversification in Section 2.2 and a parallel model in Section 2.3. The parallel model is also meant to enhance the convergence toward the PS.

2.2. The elitist diversification

When a MOEA with a classic diversification mechanism is used for the solution of the VRPRB, the algorithm tends to converge prematurely to specific areas of the objective space. Fig. 2 illustrates the results of preliminary experimentations, for instances, E101 − 08e and E151 − 12c. The implemented sharing method [17] was not able to provide solutions of good quality regarding the total length objective. This is due to the fact that the MOEA is not able to explore the objective space in the direction of the hardest objective to optimize. To preserve the population diversity through the search, we propose to add an *elitist diversification* technique to the sharing method. This technique is inspired from the elitism strategy. The elitism is a way to speed up and improve the convergence toward the Pareto set. It consists in maintaining an archive that contains the PPS encountered during the search. Some solutions of this archive are included into the main population of the MOEA at each generation.

In the elitist diversification, additional archives are considered. They contain the PPS when one objective is maximized instead of being minimized. Let $S(P)$ be the subset of solutions of D found by an algorithm P, and k the index of the objective function component which is maximized. To define new archives, the dominance operator $\prec_k$ is introduced:

$$\forall y,z \in S(P), y \prec_k z \iff (\forall i \in \{1 \dots n\} \setminus \{k\}, f_i(y) \leqslant f_i(z)) \wedge (f_k(y) \geqslant f_k(z)) \wedge ((\exists i \in \{1 \dots n\} \setminus \{k\}, f_i(y) < f_i(z)) \vee (f_k(y) > f_k(z))).$$

Then, we have $P_k = \{s \in S(P) | \forall s' \in S(P), s' \not\prec_k s\}$, with $k = 1, \dots, n$, the archive of PPS associated with the maximization of the kth objective component instead of the minimization. We denote $\prec_0$ the dominance operator given in Definition 2.1, and A_0 the standard archive.

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. Preliminary experiments point out that the improvement is less important when all archives are embedded in the same MOEA. This leads us to distribute the archive among several searches resulting in a co-operative model. In the general case with n objectives, the co-operative model is composed of n islands denoted I_k. Each island I_k has two types of archive: A_0 and A_k. Each $Migration_t$ generation, I_k sends its A_0 archive to its two neighbors I_{k-1} and I_{k+1}. The communication topology is toric, therefore k is computed modulo n. This co-operative model and its communication topology consist in the model described in Fig. 3.

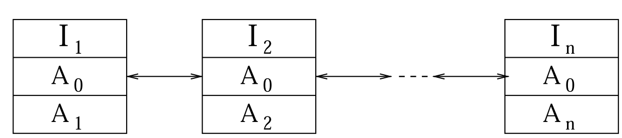


Fig. 3. The basic co-operative model – the toric structure is not shown in order not to obfuscate the figure.

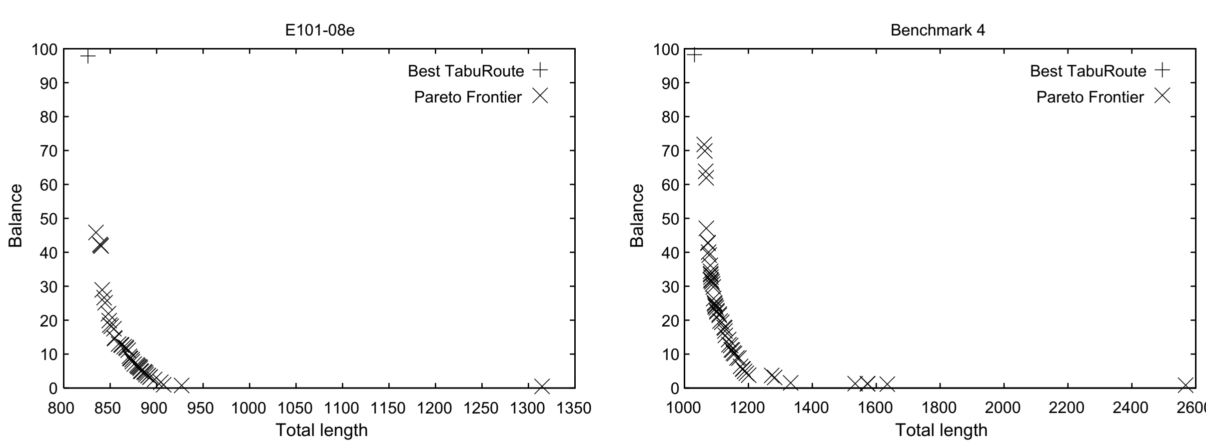


Fig. 2. The Pareto set generated with a standard MOEA and the solution provided by TabuRoute for E101−08e and E151−12c.

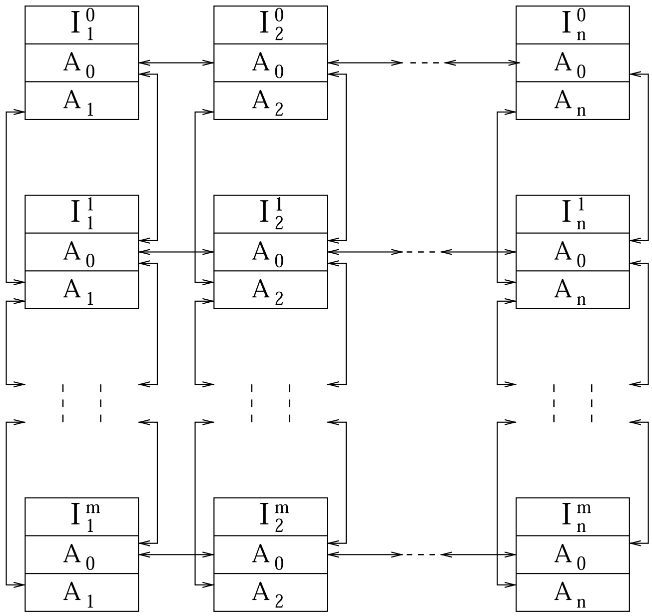


Fig. 4. The complete co-operative model – the toric structure is not shown in order not to obfuscate the figure.

2.3. Parallelization

The co-operative model described previously formed the elementary brick of a more general island model used to favor the convergence and diversification tasks (see Fig. 4).

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 the number of islands. An island is denoted I_j^i. It means it belongs to the ith 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.

3. Implementation

In this section, we first precise the implementation of the island model in Algorithm 1. We notably present the different used multi-objective mechanisms 3.2. Next, we describe the recombination phase (3.3). It includes the VRPRB dedicated genetic operators.

3.1. The main loop of an island

The main loop of an island I_k^i is given in Algorithm 1. It includes three main phases: the selection phase, the recombination phase and the archive updating. Multi-objective mechanisms used in addition to the elitist diversification during the selection phase and the updating of the archives are detailed in Section 3.2. Offspring generation procedure and genetic operators involved in the recombination phase are described in Section 3.3.

Algorithm 1 Main loop of the island I_k^i

$\{N$: size of the population$\}$
$P_0 \leftarrow$ initialize_population()
$A_0 \leftarrow \{s \in P_0 \mid \forall s' \in P_0, s' \nprec s\}$
$A_k \leftarrow \{s \in P_0 \mid \forall s' \in P_0, s' \nprec_k s\}$
$t \leftarrow 0$
while $t < max\_generation$ **do**
  communication(t, A_0, A_k)
  selection(P_t, P_{t+1}, A_0, A_k) {c.f. 3.2}
  recombination(P_{t+1}) {c.f. 3.3}
  update_archives(P_{t+1}, A_0, A_k) {c.f. 3.2}
  $t \leftarrow t+1$
end while

Algorithm 2 communication(t: Integer, A_0, A_k: Archive)

if t mod $Migration_t = 0$ **then**
  communicate$(A_0, I_{k+1}^i, I_k^{i-1}, I_k^{i+1})$
  communicate$(A_k, I_k^{i-1}, I_k^{i+1})$
endif

Algorithm 3 selection(P_t, P_{t+1}: Population, A_0, A_k: archive)

{elitism(A, M): choose M individuals from A)}
ranking(P_t)
sharing(P_t)
compute_fitness(P_t)
sort_by_fitness(P_t)
$P_{t+1} \leftarrow P_t[1, \ldots, N/2 - 2M] \cup$ elitism$(A_0, M) \cup$ elitism(A_k, M)

3.2. Multi-objective mechanisms

The following multi-objective mechanisms are used in our MOEA. We have chosen these mechanisms based on the results obtained by Talbi et al. [20]. They have compared several multi-objective mechanisms on a bi-objective flowshop scheduling problem and it appears from the results that the following mechanisms are the ones which provide the best results.

The ranking function of NSGA [18] is applied. In this ranking mechanism, non-dominated individuals of the population obtain rank 1 and form the subset E_1. Rank k is given to the solutions only dominated by the individuals belonging to the subset $E_1 \cup E_2 \cup \cdots \cup E_{k-1}$. These solutions are included in E_k. The fitness of an individual s with the rank k is then given by the following formula:

Algorithm 4 recombination(P: Population)

$i \leftarrow 0$
while $i < N/2$ **do**
  $parent_1 = \text{binary\_tournament}(P[1 \dots N/2])$
  $parent_2 = \text{binary\_tournament}(P[1 \dots N/2])$
  if rand( ) $< Prob_{rbx}$ **then**
    xover $\leftarrow$ RBX
  else
    xover $\leftarrow$ SPLIT
  end if
  $P[N/2 + i] \leftarrow \text{xover}(P_{t+1}[parent_1], P_{t+1}[parent_2])$
  $P[N/2 + i + 1] \leftarrow \text{xover}(P_{t+1}[parent_2], P_{t+1}[parent_1])$
  if rand( ) $< prob_{\text{mutation}}$ **then**
    or_opt($P[N/2 + i]$)
  end if
  if rand( ) $< prob_{mutation}$ **then**
    or_opt($P[N/2 + i + 1]$)
  end if
  local_search_2opt($P[N/2 + i]$)
  local_search_2opt($P[N/2 + i + 1]$)
  $i \leftarrow i + 2$
end while

Algorithm 5 update_archives(P: Population, A_0, A_k: archive)

$A_0 \leftarrow A_0 \setminus \{s \in A_0 | \exists s' \in P, s' \prec s\} \cup$
$\{s \in P | \forall s' \in A_0, s' \not\prec s\}$
$A_k \leftarrow A_k \setminus \{s \in A_k | \exists s' \in P, s' \prec_k s\} \cup$
$\{s \in P | \forall s' \in A_k, s' \not\prec_k s\}$
if $A_0.size > archive\_max\_size$ **then**
  cluster(A_0)
end if
if $A_k.size > archive\_max\_size$ **then**
  cluster(A_k)
end if

$$\pi(s) = \frac{S(N + 1 - R_k) + R_k - 2}{N(N - 1)} \tag{1}$$

where S is the selection pressure, N the size of the population and

$$R_k = 1 + |E_k| + 2 \times \sum_{i \in \{1 \dots k-1\}} |E_i|. \tag{2}$$

To maintain diversity along the Pareto frontier, we use a sharing technique [17] that tends to spread the population along the Pareto frontier by penalizing individuals that are strongly represented in the population. The sharing is performed in the objective space. This mechanism can be described as follows. The fitness π of individual u is divided by a value called the *niche counter*:

$$\pi'(u) = \frac{\pi(u)}{m(u)}. \tag{3}$$

The niche counter $m(u)$ is computed as $\sum_{v \in pop} sh(d(u, v))$, where sh is the sharing function defined as

$$sh(d(u,v)) = \begin{cases} 1 - \left(\frac{d(u,v)}{\gamma}\right)^{\alpha} & \text{if } d(u,v) < \gamma, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where γ is the size of the niche. As suggested in [21], γ is fixed to $\frac{2}{N}$. Distance $d(u, v)$ is the Manhattan distance between $F(u)$ and $F(v)$.

The elitism was implemented according to the description given in Section 2.3. However, the best solutions according to each objective are always chosen. Archive A_0 is an archive of small cardinality. This leads to reduce the pressure when elitism is applied as shown by Zitzler and Thiele [22]. When the archive is full, its size is reduced by using the clustering algorithm *average linking method* [23]. This algorithm has been proved to work well with data such as non-dominated solution sets [22].

In addition to archives A_0 and A_k, a third archive is maintained to store the complete approximation of the PS. The size of this archive has been fixed to 1000 solutions, which is large enough according to preliminary experiments. This archive is not used in the elitism strategy, it is only used not to lose solutions of the approximation due to the stochasticity of the GA.

3.3. The recombination phase

3.3.1. Offspring generation procedure

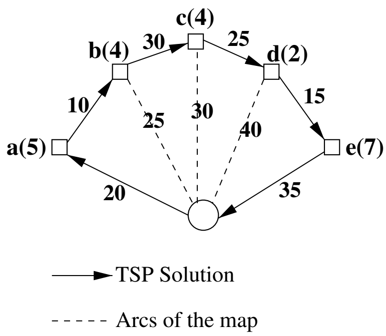
First, two parents are chosen among the intermediate population. To select a parent, a binary tournament is used. A solution u is preferred to a solution v if it has a better rank or if they have the same rank and $m(u)$ is lesser than $m(v)$. Then, two offsprings are created by the application of a crossover. Two crossovers are employed: the RBX crossover and the SPLIT crossover. They are described in 3.3.2 and 3.3.3. Each offspring can be mutated with a probability $prob_{\text{mutation}}$. The mutation operator is the Or-opt and it is explained in 3.3.4. Finally a 2-opt local search [24] is applied to each route of each offspring. The local search is used to improve the length of the routes and to take into account the fact that tours should be optimal in order to avoid artificially balanced solutions. A 2-opt local search has been chosen since it will not be too much time consuming and it should be enough due to the fact that a tour may not contain a large number of customers.

3.3.2. Crossover RBX

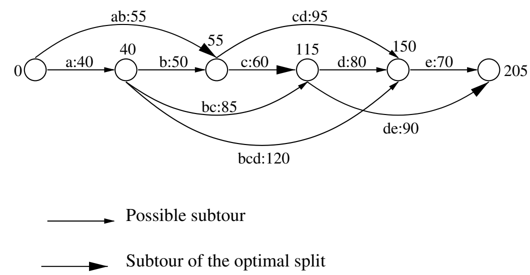
RBX [25], *route based crossover*, can be described as follows. Some randomly chosen routes of the first parent are kept. Then, the offspring is completed with the routes of the second parent. Already served customers are removed from those routes. This crossover has been chosen because it works on the route level and the balance objective is computed at that level.

3.3.3. Crossover SPLIT

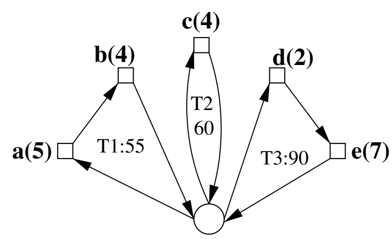
In a recent paper, Prins proposed a GA for the solution of the CVRP [9]. Each solution is coded as a TSP solution,



(a) A TSP solution.



(b) The corresponding DAG ($Q = 10$).



(c) The VRP solution corresponding to the shortest path in the DAG of (b).

Fig. 5. From a TSP solution to a VRP solution.

and is transformed into a VRP solution for its evaluation thanks to a procedure called SPLIT. Following the same lines, we define the SPLIT crossover as follows. Parents are considered as two TSP solutions (i.e. arcs entering or leaving the depot are removed and routes are merged). The crossover OX is applied and a VRP solution is built from the offspring. It is done by building a directed acyclic graph (DAG) $D = (V, A)$ with $V = \{0, 1, \ldots, m\}$ and searching a shortest path in it. Let the permutation σ represent the TSP solution. We always have $\sigma(0) = 0$. The arcs of D represent the different feasible routes (i.e. split). The set A is defined as follows: for all $i, j \in V$ $(i < j)$, the arc (i, j) belongs to A if and only if the total load on the route $0\sigma(i+1)\ldots\sigma(j)0$ is less or equal to the vehicle capacity. The valuation of the arc (i, j) is given by the formula:

$$d(0, \sigma(i+1)) + \sum_{l=0}^{j-1} d(\sigma(i+l), \sigma(i+l+1)) + d(\sigma(j), 0)$$

where $d(c_1, c_2)$ is the distance between customers c_1 and c_2. Finding the optimal partition of the TSP tour is equivalent to solving a shortest path problem on the DAG. An example provided by Prins [9], with $Q = 10$, is given in Fig. 5.

3.3.4. The mutation operator: Or-opt

The Or-opt [26] consists in moving 1 to N consecutive customers from a route to another one. In our implementation N was fixed to 3. This operator has two advantages: (i) it is particularly well adapted to the VRP which is a set of routes and (ii) it allows the creation of new routes.

4. Computational results

The MOEA just described was coded in C/MPI. The computational experiments were conducted on an IBM RS6000/SP with Power4 1.1 GHz CPUs. Eight processors were available. One brick was implemented into one process.

The evaluation was conducted on the seven CVRP benchmarks proposed by Christofides and Eilon [27], and Christofides and al. [28], are considered. Following the naming scheme used in Toth and Vigo [2], 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 [27], $k = c$ to Christofides et al. [28]. Each instance was solved five times. The GA was experimentally tuned. The following parameters were used:

- Size of the population: $N = 256$.
- Maximal size of the archive: $archive\_max\_size = 50$.
- Size of the archive after a clustering phase: $M = 20$.
- Pressure of the sharing: $S = 1.7$.
- Probability to apply the crossover RBX: $prob_{rbx} = 0.5$.
- Probability to mutate the solution: $prob_{\text{mutation}} = 0.4$.
- Number of bricks (parallel implementation): 8.
- Migration frequency: $Migration_t = 250$.

The stopping criterion is linked to the maximum number of generations, $max\_generation$. Three methods have been tested: the MOEA without the elitist diversification (NED), the MOEA with the elitist diversification (ED), and the parallel MOEA with elitist diversification (PAR). NED is composed of a single island while ED is composed of two islands. To ensure that both algorithms generate the same number of solutions, the maximum number of generations was fixed to 100,000 in the case of NED and 50,000 in the case of ED. It was also fixed to 50,000 generations for PAR; in that way an elementary brick of PAR and, thanks to the parallelization, the complete island model should take the same amount of time as the algorithm ED.

As suggested in [29], the S metric [30] was used. The S metric is a good choice when the number of objectives is small as it is the case here and when the optimal Pareto set is unknown. $S(A)$ represents the size of the area dominated by the approximation generated by an algorithm A. It is based on computing the volume (area in the bi-objective case) dominated by a given Pareto-front approximation. The S metric requires a reference point Z_{ref} consisting of a reference value for each of the two objectives. For both objectives and for each instance, we used the worst value found by all methods. We have normalized the S metric. Therefore, if $S(A) - S(B)$ is greater than 0, it means that algorithm A outperforms algorithm B, otherwise algorithm B outperforms algorithm A. We also used the C metric [30] where $C(A, B)$ corresponds to the ratio of the approximation generated by B dominated by the approximation generated by A.

In Tables 2–5, we report the mean, maximal, and minimal results as well as the standard deviation. First, we have evaluated the interest of additional archives. We have used the S metric [30] to assess the contribution of the elitist diversification. We compare the implementations of the MOEA with and without the elitist diversification. Note that both implementations include all the mechanisms and operators described in Section 3. Results are provided in Table 1. The elitist diversification version is always able to improve the quality of the approximation except for some runs on the instance E51 – 05e. Due to the fact that this is the easiest instance, NED was already able to find good bounds for both objectives and notably the optimal solution according to the length criterion. The elitist diversification technique was designed to improve the diversification of the approximation. In order to assess its impact on the convergence of the approximation generated by the meta-heuristic, we have used the C coverage measure [30]. Results reported in Table 2 clearly demonstrate that the elitist diversification improves the convergence.

The same metrics were used to assess the contribution of the parallel model. The results show that it leads to an improvement of the generated approximation (see Tables 3 and 4). For the different implementations, the average computing times in minutes are reported in Table 5. This table shows that the additional archives of the elitist diversification, and the communication of the island model, are not time consuming. Since optimal Pareto frontiers are unknown, we consider the values for the extremities of the approximation. Table 7 reports the best-known value for the total length, the best found value for the total length, and the associated route balance, the best found route balance, and the associated total length. Table 7 also provides the average number (NB) of solutions in the approximation of the PS. Even if the proposed MOEA

Table 1
Contribution of the elitist diversification: $S(ED) - S(NED)$

Instance	Best	Mean	Worst	Standard deviation
E51–05e	0.01597	0.00729	−0.00013	0.00415
E76–10e	0.03301	0.02193	0.00505	0.00710
E101–08e	0.04291	0.03249	0.01506	0.00787
E151–12c	0.09167	0.06603	0.04362	0.01196
E200–17c	0.09281	0.06157	0.03746	0.01311
E121–07c	0.03051	0.02300	0.01588	0.00426
E101–10c	0.05898	0.04126	0.02252	0.00926

Table 2
Results of the coverage measure $C(ED, NED)$ (first entry per cell) and $C(NED, ED)$ (second entry per cell)

Instance	Best	Mean	Worst	Standard deviation
E51–05e	0.59	0.42	0.12	0.12
	0.45	0.15	0.00	0.14
E76–10e	0.89	0.52	0.26	0.17
	0.44	0.21	0.00	0.12
E101–08e	1.00	0.94	0.69	0.09
	0.31	0.06	0.00	0.09
E151–12c	1.00	0.99	0.87	0.02
	0.07	0.01	0.00	0.01
E200–17c	1.00	0.88	0.45	0.18
	0.25	0.04	0.00	0.07
E121–07c	1.00	0.97	0.89	0.03
	0.05	0.01	0.00	0.02
E101–10c	0.97	0.93	0.79	0.04
	0.04	0.01	0.00	0.01

Table 3
Contribution of the parallel model: $S(PAR) - S(ED)$

Instance	Best	Mean	Worst	Standard deviation
E51–05e	0.01045	0.00531	−0.00267	0.00267
E76–10e	0.02071	0.00858	0.00014	0.00553
E101–08e	0.01437	0.00688	0.00114	0.00359
E151–12c	0.03270	0.02061	−0.00241	0.01004
E200–17c	0.05525	0.03123	0.00621	0.01234
E121–07c	0.01399	0.09198	0.00430	0.00256
E101–10c	0.02966	0.01633	0.00421	0.00654

Table 4
Results of the coverage measure $C(PAR, ED)$ (first entry per cell) and $C(ED, PAR)$ (second entry per cell)

Instance	Best	Mean	Worst	Standard deviation
E51–05e	0.66	0.50	0.25	0.10
	0.34	0.07	0.00	0.09
E76–10e	1.00	0.82	0.43	0.15
	0.61	0.19	0.00	0.21
E101–08e	0.97	0.80	0.48	0.13
	0.46	0.16	0.00	0.13
E151–12c	1.00	0.94	0.42	0.12
	0.55	0.04	0.00	0.11
E200–17c	1.00	0.75	0.43	0.27
	0.42	0.17	0.00	0.21
E121–07c	0.91	0.77	0.62	0.07
	0.24	0.11	0.02	0.06
E101–10c	0.86	0.79	0.68	0.05
	0.15	0.05	0.01	0.04

Table 5
Average computing times in minutes

Instance	E51–05e	E76–10e	E101–08e	E151–12c	E200–17c	E121–07c	E101–10c
NED	10.10	25.10	34.37	63.13	81.39	38.57	34.41
ED	10.09	25.24	34.59	63.40	81.46	39.05	34.58
PAR	10.22	25.38	35.22	65.60	82.20	40.30	35.43

Table 6
Objective values for the best found solutions of Taburoute and Prins' GA

Instance	Taburoute		Prins′ GA	
	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

Table 7
Best found solutions by the parallel MOEA for both objectives

Instance	NB	Total length				Route balance	
		Best-known	Best-found	%	Associated route balance	Best found	Associated total length
E51–05e	49.8	524.61	524.61	0.00	20.07	0.24	618.22
E76–10e	108	835.26	835.32	0.01	78.10	0.59	1203.98
E101–08e	154	826.14	827.39	0.15	67.55	0.29	1871.06
E151–12c	242.4	1028.42	1047.35	1.84	74.78	0.80	1484.48
E200–17c	243	1291.45	1352.46	4.72	76.60	1.38	1902.64
E121–07c	741.8	1042.11	1042.11	0.00	146.67	0.10	2388.30
E101–10c	348	819.56	819.56	0.00	93.43	1.15	1429.90

was not specifically designed to provide good solutions on the total length objective, the values of the solutions identified are close to the best-known values. Regarding the balancing objective, no best-known solution has been reported yet, but the best values range from 0.10 to 1.38. Therefore, the best solutions identified according to this criterion seem to be of good quality since an obvious lower bound on this objective is zero. This is also confirmed if we look at the poor quality in terms of balance of two of the most efficient methods for the single-objective CVRP, namely Taburoute [8] and Prins' GA [9], reported in Table 6 Finally, the important number of solutions for each approximation of the PS as well as the difference in the values of the extremities of the approximation tend to indicate that the two objectives are not strongly related.

5. Conclusions

In this paper, we have proposed a multi-objective evolutionary algorithm for a bi-objective vehicle routing problem, called the vehicle routing problem with route balancing, where both the minimization of the total length and the balance of the routes, i.e. the minimization of the difference between the longest route length and the shortest route length, have to be optimized. Preliminary experiments have shown that a multi-objective evolutionary algorithm with a classic diversification mechanism such as the sharing method was not able to provide good quality solutions regarding the total length objective. To improve the diversification of the multi-objective evolutionary algorithm, we have defined a new mechanism, called the elitist diversification, which was used in cooperation with the sharing method. To improve the quality of our results we have also used parallelization by designing an island model which takes into account the elitist diversification. Even if both mechanisms are proposed for the solution of any multi-objective problem, their contribution was evaluated in the case of the solution of the vehicle routing problem with route balancing on a set of standard benchmarks with standard metrics. The positive impact of both mechanisms has been observed through computational experiments. Indeed, computational results show a strict improvement of the generated Pareto sets. Since optimal Pareto sets remain unknown for the problem, the fact that the values found for the total length objective are close to the best-known ones, and that the best values for the other objective are quite small seems to indicate that our generated approximations are of good quality.

Acknowledgements

This work was partially supported by the Nord-Pas-de-Calais Region. This support is gratefully acknowledged. Thanks are also due to C. Prins for providing the best solutions obtained using his GA approach.

References

[1] J.K. Lenstra, A.H.G. Rinnooy Kan, Complexity of vehicle routing and scheduling problem, Networks 11 (1981) 221–227.
[2] P. Toth, D. Vigo (Eds.), The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications, vol. 9, SIAM, 2001.
[3] N. Jozefowiez, F. Semet, E.-G. Talbi, Multi-objective vehicle routing problems, European Journal of Operational Research 189 (2008) 293–309.
[4] T. Lee, J. Ueng, A study of vehicle routing problems with load-balancing, International Journal of Physical Distribution and Logistics Management 29 (1999) 646–658.
[5] R. Ribeiro, H.R. Lourenço, A multi-objective model for a multi-period distribution management problem, in: MIC'2001, 2001, pp. 97–102.
[6] N. El-Sherbeny, Resolution of a vehicle routing problem with multi-objective simulated annealing method, Ph.D. Thesis, Faculté Polytechnique de Mons, 2001.
[7] M. Mourgaya, F. Vanderbeck, Periodic vehicle routing problem: classification and heuristic, RAIRO Operations Research 40 (2006) 169–194. (abstract, working paper version).
[8] M. Gendreau, A. Hertz, G. Laporte, A tabu search heuristic for the vehicle routing problem, Management Science 40 (1994) 1276–1290.
[9] C. Prins, A simple and effective evolutionary algorithm for the vehicle routing problem, Computers and Operations Research 31 (2004) 1985–2002.
[10] K. Deb, Multi-objective Optimization using Evolutionary Algorithms, John Wiley and Sons, 2001.
[11] C.A. Coello Coello, D.A. Van Veldhuizen, G.B. Lamont (Eds.), Evolutionary Algorithms for Solving Multi-objective Problems, Kluwer Academic Press, 2002.
[12] E. Zitzler, M. Laumanns, S. Bleuler, A tutorial on evolutionary multiobjective optimization, in: X. Gandibleux, M. Sevaux, K. Sörensen, V. T'Kindt (Eds.), Metaheuristics for Multiobjective Optimization, Lecture Notes in Economics and Mathematical Systems, vol. 535, Springer, 2004, pp. 3–37.
[13] P. Husbands, Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation, in: T. Fogarty (Ed.), Evolutionary Computing, Lecture Notes in Computer Science, vol. 865, Springer Verlag, 1994, pp. 150–165.
[14] T. Hiroyasu, M. Miki, S. Watanabe, The new model of parallel genetic algorithm in multi-objective optimization problems – divided range multi-objective genetic algorithm, in: 2000 Congress on Evolutionary Computation, vol. 1, IEEE Service Center, 2000, pp. 333–340.
[15] E.-G. Talbi, H. Meunier, Hierarchical parallel approach for GSM mobile network design, Journal of Parallel and Distributed Computing 66 (2006) 274–290.
[16] M. Ehrgott, X. Gandibleux (Eds.), Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, Kluwer Academic Publishing, 2002.
[17] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multi-modal function optimization, in: Second International Conference on Genetic Algorithms, 1987, pp. 41–49.
[18] K. Deb, S. Agrawal, A. Pratab, T. Meyunivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, IEEE Transactions on Evolutionary Computing (2002) 182–197.
[19] J.D. Knowles, Local search and hybrid evolutionary algorithms for pareto optimization, Ph.D. Thesis, University of Reading, January 2002.
[20] E.-G. Talbi, M. Rahoual, M.H. Mabed, C. Dhaenens, A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop, in: E. Zitzler et al. (Eds.), Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, vol. 1993, Springer-Verlag, 2001, pp. 416–428.
[21] M. Basseur, F. Seynhaeve, E.-G. Talbi, Design of multi-objective evolutionary algorithms: application to the flow-shop scheduling problem, in: 2002 Congress on Evolutionary Computation, vol. 2, IEEE Service Center, 2002.
[22] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (1999) 257–271.
[23] J.N. Morse, Reducing the size of non-dominated set: Prunning by clustering, Computers and Operations Research 7 (1980) 55–66.
[24] S. Lin, Computer solutions of the traveling salesman problem, Bell System Technical Journal 44 (1965) 2245–2269.
[25] J.-Y. Potvin, S. Bengio, The vehicle routing problem with time windows part II: Genetic search, INFORMS Journal on Computing 8 (1996) 165–172.
[26] I. Or, Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking, Ph.D. Thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1976.
[27] N. Christofides, S. Eilon, An algorithm for the vehicle dispatching problem, Operational Research Quarterly 20 (1969) 309–318.
[28] N. Christofides, A. Mingozzi, P. Toth, C. Sandi (Eds.), Combinatorial Optimization, John Wiley, Chichester, 1979 (Chapter 11).
[29] J. Knowles, D. Corne, On metrics for comparing nondominated sets, in: Congress on Evolutionary Computation (CEC2002), vol. 1, IEEE Service Center, 2002, pp. 711–726.
[30] E. Zitzler, Evolutionary algorithm for multiobjective optimization: Methods and applications, Ph.D. Thesis, Swiss Federal.