Enhancements of NSGA II and Its Application to the Vehicle Routing Problem with Route Balancing

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

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

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

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. For this problem, we propose an implementation of the standard multi-objective evolutionary algorithm NSGA II. To improve its efficiency, two mechanisms have been added. First, a parallelization of NSGA II by means of an island model is proposed. Second, an elitist diversification mechanism is adapted to be used with NSGA II. 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.

1 Introduction

This paper investigates the use of two variants of NSGA II to solve a bi-objective vehicle routing problem. 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 ($i = 1, \dots, m$) from the unique depot. To deliver these 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-heuristic approaches [2].

E. Talbi et al. (Eds.): EA 2005, LNCS 3871, pp. 131–142, 2006.

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

	Taburoute		Prins' 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

Another natural objective to consider in addition to the minimization of the total length is the balance of the routes. Route balancing can be expressed in several ways. In [3], the authors balance the time needed for each trip. It is computed as the sum of the differences between each route length and the shortest route length. Route balancing is also an objective in [4] which addresses a three objective multi-period vehicle routing problem. In this paper the balance is measured by the standard deviation and the load of a route consists in the number of visited customers. In [5], the minimization of the time spent on a bus, which has some common points with the route balancing, is considered. In [6], the authors take into account 8 objectives in the context of a real-life VRP faced by a Belgian transportation firm. One of them is identical to our second objective; i.e. the minimization of the difference between the maximal route length and the minimal route length.

In this paper, we address a variant of the CVRP: the vehicle routing problem with route balancing (VRPRB). The following two objectives are considered:

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

In Table 1, the seven CVRP benchmarks proposed by Christofides and Eilon [7], and Christofides and al. [8], 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 available vehicles. k is a character which identifies the paper where the distance data are provided. $k = e$ refers to Christofides and Eilon [7], $k = c$ to Christofides et al. [8]. For each instance, we report both objective values associated with the best solutions obtained using Taburoute [9] and Prins' GA [10]. 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.

Our solution to generate the Pareto set is based on the standard multi-objective evolutionary algorithm (MOEA) NSGA II proposed by Deb et al. [11]. 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, a population based method like NSGA II seems well-fitted. To improve the results of NSGA II on the VRPRB, we propose a parallelization of the problem. To obtain well-diversified approximations of the Pareto set, we have adpated the elitist diversification mechanism initially proposed in [12, 13] for NSGA II.

The paper is organized as follows. Section 2 presents our implementation of NSGA II for the VRPRB and its parallelization into an island model. In section 3, we specify the adaptation of the elitist diversification mechanism for NSGA II. In section 4, we assess the efficiency of the new mechanisms on a set of standard benchmarks. Conclusions are drawn in section 5.

2 NSGA II for the Vehicle Routing Problem with Route Balancing

We first describe the general framework of NSGA II in subsection 2.1. Then, the recombination phase (*i.e.* STEP 4) is given in the subsection 2.2 since it is the only step which needs to be adapted for the VRPRB. Finally, an improvement of NSGA II by means of an island model is proposed in subsection 2.3.

2.1 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. The lower the rank and the crowding distance are, the better the solution is. For additional details about NSGA II, the reader is refered to [11]. 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 $\mathcal{F}_i$, $i = 1, \ldots, r$, where i represents a rank.

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

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
```

the potentially Pareto optimal solution set can be very large for the VRPRB. 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.

2.2 The Recombination Phase

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) [14] and the SPLIT crossover [12, 13] inspired by Prins' genetic algorithm [10]. When a solution is created, a 2-opt local search is applied on each route in order to avoid artificially balanced solutions [12, 13].

2.3 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, which is performed instead of recombination (STEP 4). 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 begin 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

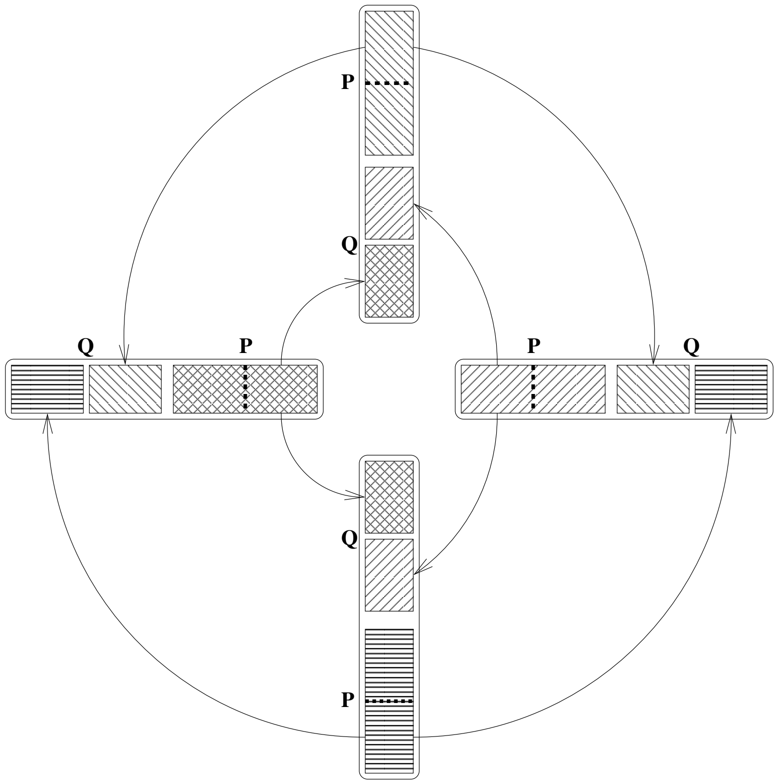

Fig. 1. Extension of NSGA II into an island model

to the ranking and crowding distance sort, 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 1 illustrates the communications in the case of four islands.

3 Using the Elitist Diversification Mechanism in NSGA II

In this section, we propose the enhancement of NSGA II by means of a diversification mechanism called the elitist diversification mechanism initially proposed in [12, 13]. First, the mechanism is presented. Then, the general parallel model is described as well as its use in the case of NSGA II.

3.1 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

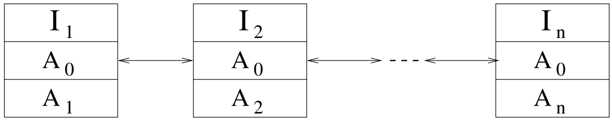

Fig. 2. The basic co-operative model - the toric structure is not shown in order not to obfuscate the figure

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:

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

Then, we have $A_k = \{s \in S(A) | \forall s' \in S(A), s' \not\prec_k s\}$, with $k = 1, \dots, n$, the archive of PPS associated with the maximization of the k^{th} objective 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 favorising one objective. 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 archives 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. At 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 Figure 2.

3.2 Parallel Extension of the Elitist Diversification Mechanism

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 Figure 3). 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

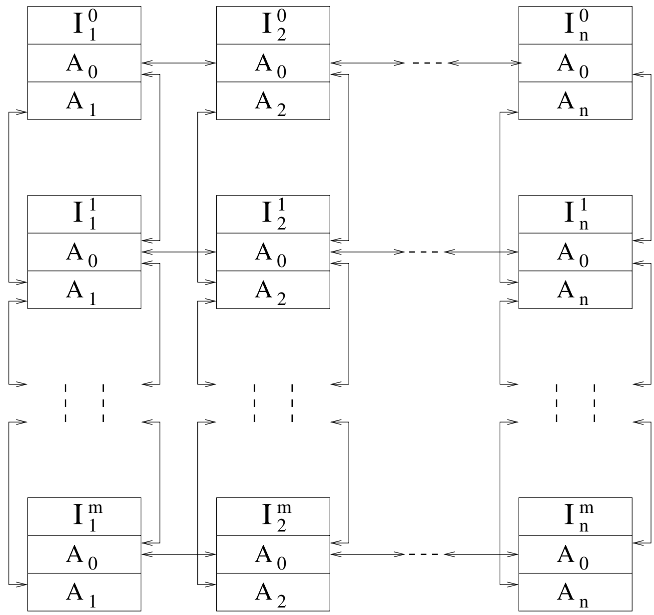

Fig. 3. The complete co-operative model - the toric structure is not shown in order not to obfuscate the figure

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.

3.3 Inclusion of the Elitist Diversification Mechanism in NSGA II

The goal is to add the management of the additional archives in NSGA II. It must be noted that NSGA II initially used no archive and the main population plays the role of the A_0 archive, *i.e.* it saves the non-dominated solutions found during the search. Therefore, each island of the parallel model described before corresponds to one instantiation of NSGA II to which one additional archive has been added. This archive is used during the recombination phase: k individuals are chosen among those belonging to the additional archive and form the set C_t. Then, the recombination phase is the same as the standard one except that the tournament used to select the parents is modified as follows. Two solutions are selected randomly in $P_t \cup C_t$. If pa_1 or pa_2 belongs to C_t, the solution from C_t wins the tournament. If both solutions belong to C_t, one is chosen randomly. Finally, if both solutions come from P_t, the standard binary tournament of NSGA II is applied. The additional archive is updated after each recombination phase ; we try to include the solutions generated during the phase.

The exchange strategy between the islands is the same as the one used in 2.3. However, since an island has four neighbors in this model, it communicates only the $\frac{N}{4}$ best solutions from its population after the selection phase. Therefore, an island receives four times $\frac{N}{4}$ solutions which replace those from Q_t. However, there are two special cases. First, in the bi-objective case, an elementary brick is formed of only two genetic algorithms. Then, an island receives twice the $\frac{N}{4}$ best solutions from the other genetic algorithm of the brick. It is not relevant and, in this case, the strategy is modified for the algorithms from a same brick to exchange $\frac{N}{2}$ solutions between them. The same difficulty occurs when there are only two elementary bricks and can be solved in a similar way.

4 Computational Results

4.1 Protocol

NSGA II for the VRPRB, the parallel model pNSGA II, the variant with elitist diversification NSGAED, and the parallel variant with elitist diversification pNSGAED have been coded in C. MPI has been used for the parallel aspect of the implementation. Experiments have been realised on an IBM RS6000/SP equipped with Power4 1.1 Ghz processors.

Evaluations have been made on the benchmark by Christofides et al. [7] for the capacitated vehicle routing problem. Each instance has been solved 10 times by each method.

The parameterization of the methods has been set experimentally. For the population of NSGA II, N has been fixed to 128. NSGA II and pNSGA II stopped after 100000 generations while NSGAED and pNSGAED stopped after 50000. Thus, we insure that each process generates the same number of solutions. For the elitist diversification, 15 solutions were used from each archive.

As suggested in [15], the S metric [16] was used. $S(A)$ gives the size of the area dominated by the approximation generated by A. The values of the objectives were normalized according to the reference point used in the S metric.

4.2 Contribution of the Parallelization

We have tested the contribution of the parallelization scheme when 1, 4, 8, and 16 processors were used. Table 2 reports the mean values and the standard deviations of the $\mathcal{S}$ metric for the different cases. As it can be expected, the results are improved with the number of processors used. However, the impact of more than 4 processors is less significant than the difference between the sequential version and the one with 4 processors. According to the behavior of the standard deviation, it seems that increasing in the number of processors contributes to improve the robustness of the method.

The impact of communications on computational times have also been assessed. The average computational times in seconds according to the number of processors are reported in Table 3. It seems that communication times do not play a significant role.

Table 2. Mean values and standard deviations of the $\mathcal{S}$ metric for NSGA II according to the number of used processors

Instance		1 proc.	4 proc.	8 proc.	16 proc.
E51-05e	Mean	0.511232	0.527863	0.527733	0.530235
	standard deviation	0.006132	0.001838	0.004329	0.001987
E76-10e	Mean	0.414035	0.420253	0.425498	0.426979
	standard deviation	0.002988	0.001714	0.002892	0.002052
E101-08e	Mean	0.570935	0.576901	0.577431	0.579026
	standard deviation	0.001779	0.001638	0.000724	0.000418
E151-12c	Mean	0.618357	0.631726	0.634581	0.637956
	standard deviation	0.006315	0.001637	0.003460	0.001426
E200-17c	Mean	0.607886	0.628112	0.632612	0.639964
	standard deviation	0.014343	0.005537	0.008276	0.002474
E121-07c	Mean	0.516538	0.526248	0.527154	0.527934
	standard deviation	0.007145	0.001480	0.001405	0.000637
E101-10c	Mean	0.584904	0.620338	0.627408	0.629321
	standard deviation	0.018182	0.004675	0.003061	0.002398

Table 3. Average computation times of NSGA II according to the number of processors

Instance	E51-05e	E76-10e	E101-08e	E151-12c	E200-17c	E121-07c	E101-10c
4 proc.	993.4	1453.4	2451.3	4082.1	4996.3	4615.1	2640.1
8 proc.	937.8	1300.8	2406.1	3621.9	4463.7	4791.1	2425.3
16 proc.	1080.7	1329.0	2289.6	3794.5	4677.7	5171.1	2451.6

Table 4. Mean values and standard deviation of the $\mathcal{S}$ metric for NSGA II without and with the elitist diversification mecanism

		1 processor		8 processors	
Instance		NSGA II	NSGAED	pNSGA II	pNSGAED
E51-05e	Mean	0.511232	0.521232	0.527733	0.529467
	standard deviation	0.006132	0.004139	0.004329	0.001282
E76-10e	Mean	0.414035	0.415599	0.425498	0.425809
	standard deviation	0.002988	0.003651	0.002892	0.002992
E101-08e	Mean	0.570935	0.573612	0.577431	0.577501
	standard deviation	0.001779	0.001800	0.000724	0.001430
E151-12c	Mean	0.618357	0.619450	0.634581	0.635170
	standard deviation	0.006315	0.007012	0.003460	0.003016
E200-17c	Mean	0.607886	0.617594	0.632612	0.643165
	standard deviation	0.014343	0.006185	0.008276	0.004848
E121-07c	Mean	0.516538	0.518553	0.527154	0.527442
	standard deviation	0.007145	0.007998	0.001405	0.000478
E101-10c	Mean	0.584904	0.602430	0.627408	0.629226
	standard deviation	0.018182	0.020408	0.003061	0.003343

Table 5. Best solutions found for each objective with the associated values of the other objective for the different implementations of NSGA II

	NSGA II								NSGAED			
Instance	1 proc.		4 proc.		8 proc.		16 proc.		1 proc.		8 proc.	
E51-05e	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07	524.61	20.07
	0.51	694.79	0.17	648.80	0.23	643.89	0.18	779.88	0.48	1093.31	0.32	645.29
	44.8		44.1		46.5		50.4		45.8		47.6	
E76-10 e	835.32	78.10	842.06	67.55	835.32	78.10	835.32	78.10	835.89	85.60	835.32	78.10
	1.01	1268.02	1.10	1117.29	0.64	997.97	0.48	1153.25	0.85	1110.78	0.33	1380.01
	109.7		124.7		108.8		135.2		115.9		127.00	
E101-08e	827.39	67.55	827.39	67.55	827.39	67.55	827.39	67.55	827.39	67.55	827.29	67.55
	0.64	1596.59	0.09	1695.55	0.35	984.76	0.27	1132.28	0.43	1129.10	0.10	1610.86
	129.4		161.1		175.6		185.8		130.2		171.9	
E151-12c	1047.02	87.10	1044.69	91.14	1043.83	77.89	1032.95	97.57	1043.71	87.67	1034.91	109.06
	1.22	2290.53	0.56	1443.31	0.51	1382.28	0.44	1551.50	0.66	2523.00	0.21	1523.94
	204.1		310.10		332.9		408.5		200.1		322.1	
E200-17c	1358.08	68.77	1349.48	79.14	1332.82	87.50	1328.44	96.61	1353.17	87.84	1327.89	97.20
	3.04	1838.32	2.17	3006.26	1.85	1976.10	0.86	3490.14	2.18	3436.43	0.58	3746.57
	206.6		316.4		373.6		479.7		192.1		354.8	
E121-07c	1043.78	146.67	1042.11	146.67	1042.11	146.67	1042.11	146.67	1043.11	147.27	1042.11	146.67
	0.15	2258.21	0.06	2314.61	0.06	2348.74	0.07	2296.43	0.06	2285.22	0.07	1649.76
	441.5		847.40		903.1		921.1		452.1		943.2	
E101-10c	819.56	93.43	819.56	93.43	819.56	93.43	819.56	93.43	819.56	93.43	819.56	94.43
	1.16	2035.80	0.77	2107.14	0.65	1425.43	0.21	1324.53	1.10	1303.67	0.35	1664.80
	269.1		441.4		487.7		499.6		278.6		490.4	

4.3 Contribution of the Elitist Diversification Mechanism

We have evaluated the performance of NSGAED, pNSGAED with 8 bricks compared to the performance of NSGA II and pNSGA II with 8 processors. The mean values and the standard deviation of the $\mathcal{S}$ metric are reported in Table 4.

It appears that the elitist diversification is always able to improve the results of NSGA II when only one processor is used. The improvement is more important for large instances such as E200-17c. We have also evaluated the contribution when eight processors are used. The contribution is less important on the smallest instances since the parallelization without the elitist diversification was already able to improve the results significantly. However, the contribution is still important on the largest instances.

4.4 Global Efficiency of NSGA II for the Vehicle Routing Problem with Route Balancing

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. We have also reported the number of potentially Pareto optimal solutions in Table 5 as follows: for each entry, the first line corresponds to the best found length with its associated balance, the second line to the best found balance with its associated length, and the third line to the average number of solutions in the approximations. It appears that the elitist diversification is able to improve the results toward the best-known values for the total length objective. Since the best balance is very close to 0, we may assume that very well-balanced solutions are obtained.

5 Conclusions

In this paper, we have described an implementation of NSGA II 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. Two enhancements of NSGA II have been proposed. The first one is the parallelization of NSGA II by means of an island model. The second one is the use of the elitist diversification mechanism, which aims to improve the diversification in NSGA II. Their contributions were evaluated on a set of standard benchmarks with standard metrics. The positive impact of both mechanisms has been observed through computational experiments. 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 route balancing objective are quite small tends to indicate that our generated approximations are of good quality.

References

1. Lenstra, J.K., Rinnooy Kan, A.H.G.: Complexity of vehicle routing and scheduling problem. Networks **11** (1981) 221–227
2. Toth, P., Vigo, D., eds.: The vehicle routing problem. Volume 9 of SIAM Monographs on Discrete Mathematics and Applications. SIAM (2001)
3. Lee, T., Ueng, J.: A study of vehicle routing problems with load-balancing. International Journal of Physical Distribution and Logistics Management **29** (1999) 646–658
4. Ribeiro, R., Lourenço, H.R.: A multi-objective model for a multi period distribution management problem. In: MIC'2001. (2001) 97–102
5. Corberan, A., Fernandez, E., Laguna, M., Marti, R.: Heuristic solutions to the problem of routing school buses with multiple objectives. Journal of the Operational Research Society **53** (2002) 427–435
6. El-Sherbeny, N.: Resolution of a vehicle routing problem with multi-objective simulated annealing method. PhD thesis, Faculté Polytechnique de Mons (2001)
7. Christofides, N., Eilon, S.: An algorithm for the vehicle dispatching problem. Operational Research Quarterly **20** (1969) 309–318
8. Christofides, N., Mingozzi, A., Toth, P., Sandi, C., eds.: 11. In: Combinatorial Optimization. John Wiley, Chichester (1979)
9. Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the vehicle routing problem. Management Science **40** (1994) 1276–1290
10. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Computers and Operations Research (2004) (Article in Press).
11. Deb, K., Agrawal, S., Pratab, A., Meyunivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. IEEE Transactions on Evolutionary Computing (2002) 182–197
12. Jozefowiez, N., Semet, F., Talbi, E.G.: Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In Guervos, J.M., et al., eds.: PPSN VII. Volume 2439 of Lecture Notes in Computer Science., Springer-Verlag (2002) 271–280
13. Jozefowiez, N.: 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 (2004)
14. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows part ii: genetic search. INFORMS Journal on Computing 8 **8** (1996)
15. Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: Congress on Evolutionary Computation (CEC'2002). Volume 1., IEEE Service Center (2002) 711–726
16. Zitzler, E.: Evolutionary algorithm for multiobjective optimization: Methods and applications. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (1999)