

Multiobjective vehicle routing problem with fixed delivery and optional collections

Luciana P. Assis · André L. Maravilha ·
Alessandro Vivas · Felipe Campelo ·
Jaime A. Ramírez

Received: 1 October 2011 / Accepted: 1 September 2012 / Published online: 18 September 2012
© Springer-Verlag 2012

Abstract We present an adaption on the formulation for the vehicle routing problem with fixed delivery and optional collections, in which the simultaneous minimization of route costs and of collection demands not fulfilled is considered. We also propose a multiobjective version of the iterated local search (MOILS). The performance of the MOILS is compared with the ϵ -constrained (P_ϵ) ILS, the NSGA-II and the indicator-based multi-objective local search methods in the solution of 14 problem instances containing between 50 and 199 customers plus the depot. The results indicate that the MOILS outperformed the other approaches, obtaining significantly better average values for coverage, hypervolume and cardinality.

Keywords Vehicle routing problem · Selective pickups · Iterated local search · Multiobjective optimization

L. P. Assis · A. Vivas
Departamento de Computação, Universidade Federal dos Vales do Jequitinhonha e Mucuri,
Rod. MGT 367, Km 583, 5000-Alto da Jacuba, 39100-000 Diamantina, MG, Brazil
e-mail: lpassis@ufvjm.edu.br

A. Vivas
e-mail: alessandrovivas@ufvjm.edu.br

A. L. Maravilha · F. Campelo · J. A. Ramírez (✉)
Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais,
Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, MG, Brazil
e-mail: jramirez@ufmg.br

A. L. Maravilha
e-mail: andrems@ufmg.br

F. Campelo
e-mail: fcampelo@ufmg.br

1 Introduction

The vehicle routing problem (VRP) is one of the most important and widely studied within the context of combinatorial optimization. It includes many relevant issues of transportation logistic, e.g., the pickup and delivery of materials, a problem of particular interest for industries concerned with the efficient management of returned goods for recycling, re-manufacturing or reuse.

In some variants of the VRP the delivery of products is mandatory, while the collection can be postponed. Traditionally, the VRP has been defined as mono-objective and solved using exact methods. In [34], a single-VRP with unrestricted backhauls, in which a profit value is associated for each satisfied collect demand, is solved with an exact branch-and-bound algorithm. Later works [14, 15] presented a branch-and-cut algorithm to solve the same problem, and a branch-and-price algorithm to solve five variants of the VRP with delivery and selective pickups with time windows. Recently, mono-objective versions of the VRP have been solved using heuristics, most of them based on local search such as simulated annealing [25] and tabu search [23, 13]. These general procedures explore the space of the solution to find good solutions with a reasonable computational time [12].

Several practical VRPs are multiobjective by nature, due to the inherent necessity of considering aspects other than the minimization of the total cost [28], such as customer satisfaction, labor regulations, or other aspects of the problem [21]. There are however few works that treat the VRP as multiobjective, in particular those problems involving pickup and delivery services [20].

In terms of multiobjective approaches, metaheuristics such as genetic algorithms [8, 30], tabu search [9, 16, 26], memetic algorithms [10] and simulated annealing [5, 31, 36] have been widely used to generate approximately efficient solutions for multiobjective problems. In general, algorithms based on local search perform well in multiobjective combinatorial problems [3, 18, 35] and could be applied to VRPs.

We discuss in this work an adaption on the VRP in order to treat its multiobjective nature with fixed deliveries and optional collections. More specifically, the objectives are defined as the minimization of route costs and of collection demands not fulfilled. We also propose a multiobjective version of the iterated local search (ILS) heuristic. The MOILS approach is applied to a set of instances containing between 50 and 199 customers plus the depot, and is compared to other approaches available in the literature [3, 4, 6, 19].

2 Problem description

The multiobjective VRP with fixed delivery and optional collections consists in defining a set of routes that minimize the transportation cost and the number of collections not carried out, attending all delivery demands. Each customer must be visited by one vehicle, and partial execution of demands is not allowed.

2.1 Mathematical formulation

The mathematical formulation is a generalization of the model proposed in [23] for the VRP with simultaneous pickup and delivery (VRPSPD), obtained by transforming a constraint into an objective.

Let $G = (V, A)$ be a complete graph, with $V = \{0, 1, 2, \dots, n\}$ the set of vertices and $A = \{(i, j) : i, j \in V, i \neq j\}$ the set of edges. The vertex 0 represents the depot and the others represent the customers. Each edge (i, j) has an associated value $c_{ij} \geq 0$ that represents the cost for reaching vertex j from vertex i . Each customer i has a demand d_i for delivery and a demand p_i for collection. There are \bar{k} homogeneous vehicles available, each with capacity Q . The parameter y_{ij} is the sum of the collected load between the depot and the node i (included) driven to node j . The parameter z_{ij} is the sum of the load delivered to customers after node i (excluded) driven to node j . We use the edge variable x_{ij}^k , which is equal to one if edge (i, j) is traveled by vehicle k and zero otherwise, and the choice variable ℓ_j , which is equal to one if pickup demand of customer j is satisfied, and zero otherwise.

The objectives of this problem can be expressed as:

$$\text{minimize } \begin{cases} \sum_{k=1}^{\bar{k}} \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}^k \\ \sum_{j=1}^n p_j (1 - \ell_j) \end{cases} \tag{1}$$

$$\text{Subject to: } \sum_{i=0}^n \sum_{k=1}^{\bar{k}} x_{ij}^k = 1, \quad j = 1, \dots, n \tag{2}$$

$$\sum_{i=0}^n x_{ij}^k - \sum_{i=0}^n x_{ji}^k = 0, \quad j=0, \dots, n \quad \text{and} \quad k=0, \dots, \bar{k} \tag{3}$$

$$\sum_{j=1}^n x_{0j}^k \leq 1, \quad k = 1, \dots, \bar{k} \tag{4}$$

$$\sum_{i=0}^n z_{ij} - \sum_{i=0}^n z_{ji} = d_j, \quad \forall j \neq 0 \tag{5}$$

$$\sum_{i=0}^n y_{ij} - \sum_{i=0}^n y_{ji} = p_j \ell_j, \quad \forall j \neq 0 \tag{6}$$

$$y_{ij} + z_{ij} \leq Q \sum_{k=1}^{\bar{k}} x_{ij}^k, \quad i, j = 0, \dots, n \tag{7}$$

$$x_{ij}, \ell_j \in \{0, 1\}, \quad i, j = 0, \dots, n \tag{8}$$

$$y_{ij}, z_{ij} \geq 0, \quad i, j = 0, \dots, n \tag{9}$$

The two objective functions (1) are the total cost of the routes and the collections not carried out, and have to be minimized. The constraints model the following

requirements: each point of demand must be visited by only one vehicle (2); the same vehicle arrives and departs from each client (3); a maximum of \bar{k} vehicles can be used (4); all delivery demands must be met (5); when variable ℓ_j is equal to one, the pickup demand of customer j must be satisfied (6); all demands should be transported on the arcs included in the solution (7). In addition, (8) represents the integrality restriction and (9) a non-negativity constraint for collection and delivery demands.

2.2 Multiobjective iterated local search

The iterated local search (ILS) algorithm is based in the idea that the local search procedure can be improved from the generation of new solutions for the starting point by perturbing visited local optimal solutions [22].

Some heuristic approaches found in literature to solve the VRP with simultaneous pickup and delivery indicate that the use of ILS is a good alternative [32]. Some of its advantages are: the existence of few adjusting parameters when compared to the evolutionary methods, simplicity, robustness, effectiveness and the ease of implementation [22].

In this section, we present an adaptation of the ILS for the solution of combinatorial optimization multiobjective problems. This approach, called the multiobjective iterated local search (MOILS), is presented in Algorithm 1.

Algorithm 1: Multiobjective Iterated Local Search (MOILS)

Input: $maxIter$, $maxCount$

```

1  $Front \leftarrow generateInitialSolutions();$ 
2  $iter \leftarrow 0;$ 
3 while  $iter < maxIter$  do
4    $s' \leftarrow select(Front);$ 
5    $count \leftarrow 0;$ 
6   while  $count < maxCount$  do
7      $s'' \leftarrow perturb(s');$ 
8      $s'' \leftarrow localSearch(s'');$ 
9      $inserted \leftarrow update(Front, s'');$ 
10    if  $inserted$  then
11       $count \leftarrow 0;$ 
12       $s' \leftarrow s'');$ 
13    else
14       $count \leftarrow count + 1;$ 
15   $iter \leftarrow iter + 1;$ 

```

Output: $Front$

First, the MOILS constructs two initial solutions (line 1) by efficient hybrid algorithm based on ILS and random variable neighborhood descent (RVND), proposed in [27]. These solutions represent the extremes of the front, one being a point of maximum transportation cost and the other a point of minimum transportation cost, where

no collections are performed. When all collection demands should be attended the problem is reduced to the mono-objective VRSPD, and when no collection demand is attended it is reduced to the mono-objective capacitated VRP (CVRP).

The algorithm executes $maxIter$ iterations (lines 3–15), where one non-dominated solution, in the front set, is selected (line 4) to be exploited (lines 6–14). The selection procedure is based on the crowding-distance [6], which allows less explored regions to have higher selection priority. In this work, the crowding-distance is calculated in a different manner, with the value of the extreme solutions equal to twice the distance between this point and the nearest solution. We have proposed this modification in order to guarantee the exploration of the two initial solutions (equivalent to the CVRP and VRSPD solutions) in early iterations and to give advantage to more intermediate solutions after the iterative process starts generating more non-dominated points.

Given a selected solution, the algorithm executes $maxCount$ iterations to explore it. In each iteration, this solution is perturbed and local search is applied, according to the ILS. Here, perturbation is done in two phases. In the first, the collection status (attempt or not attempt) of some customers randomly selected is changed. When the pickup demand ceases to be fulfilled, the respective customer can remain in the same position of the route. Otherwise, the load stored by the vehicle that will satisfy this demand must be verified. If its capacity is exceeded, the customer is inserted in the first viable position found. When no viable position is found, this customer is included in a new route. In the second phase, the algorithm applies one of the following perturbation mechanisms randomly chosen: Multiple Swap that reallocate different customers of the different routes randomly selected; Multiple Shift that exchanges two customers of distinct routes; and Ejection Chain that relocates a customer from route R_1 to route R_2 , a customer from R_2 to R_3 , and so on.

The local search phase explores the neighborhood of a solution with the objective of finding other non-dominated solutions for the problem. We have used the random variable neighborhood search (RVND) refinement method [27]. We implemented 12 types of movements to define the neighborhood, six inter-route and six intra-route. The inter-route movements are: (1) *Shift(1,0)*, (2) *Shift(2,0)*, that relocate a customer or two adjacent customers, (3) *Swap(1,1)*, (4) *Swap(2,1)*, (5) *Swap(2,2)* that interchange two customers, or two adjacent customers with one customer or with others two adjacent customers, and (6) *Crossover* that removes two arcs (i, j) and (i', j') and inserts two new arcs (i, j') and (i', j) . The intra-route movements are: (1) *Or-opt*, (2) *Or-opt 2*, (3) *Or-opt 3* that relocate one customer, or two or three adjacent customers, (4) *2-opt* that removes two non-consecutive arcs (i, j) and (i', j') and replaces by the arcs (i, i') and (j, j') , (5) *Exchange* two customers, and (6) *Reverse* the direction of the route, aimed at reducing the load of the vehicle. Figure 1 shows examples of these operations, in which the highlighted arcs and customers indicate the changes in the original solution.

The best improvement strategy was used in all neighborhood structures. The computational complexity of each one of these moves is $O(n^2)$, except to reverse that is $O(n)$. After RVND executions, the algorithm verifies if it is possible to attempt the pickup demands of some customers. The collection of one customer is carried out only if this demand does not exceed the capacity of the vehicle and if it can remain in the same position of the route.

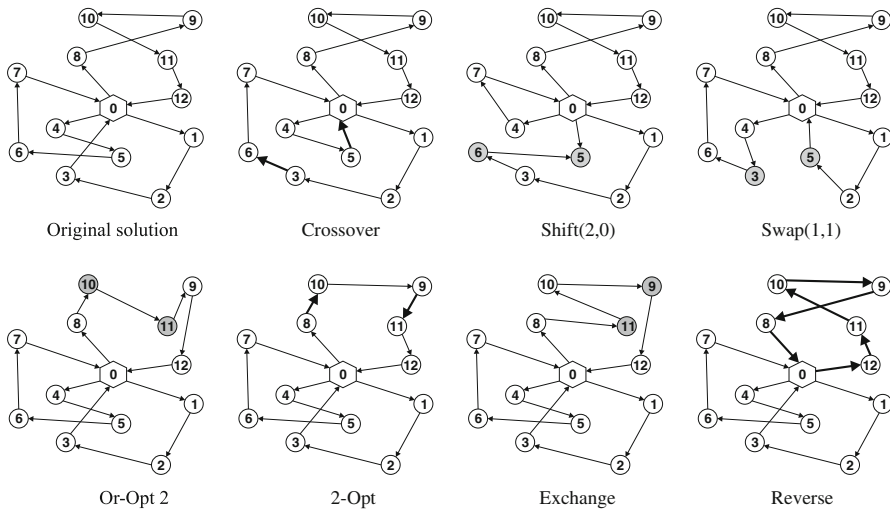


Fig. 1 Neighborhood structures

The solution generated in the perturbation and local search procedures is then included in the front (line 9). The updating phase verifies if a specific solution s is non-dominated by any solution, then it is inserted in the set and the solutions dominated by it are eliminated. The variable *count* represents the number of iterations that one solution is explored without generating a new non-dominated solution. If one solution is inserted in the front then *count* is restarted (line 11), otherwise it is increased by one unit (line 14).

3 Experimental results

3.1 Test problems and performance metrics

To evaluate the performance of the MOILS, an experiment was set up¹ considering 14 problem instances ranging from 50 to 199 customers plus the depot [29]. The travel cost is calculated as the Euclidean distance between the points. Problems 1–5 have randomly placed cities in the plane, while problems 11 and 12 have cities appearing in clusters, in which the depot is not centered.

These instances were solved using the following approaches:

1. the MOILS, as described in Sect. 2.2. After preliminary testing, the constants were set as $maxIter = 100$ and $maxCount = 15$.
2. ϵ -constrained approach (P_ϵ) [4] using the ILS as the base algorithm, in order to compare the performance of our proposed MOILS against that of an adapted

¹ The algorithm implementations, problem instances, and statistical routines used in the analysis of this experiment are available online at the address <http://www.cpdee.ufmg.br/~fcampelo/files/OPTL2012a/>.

- single-objective iterated local search. Preliminary testing suggested the use of 50 as the number of iterations for this method [22];
3. non-dominated Sorting Genetic Algorithm II (NSGA-II) [6], which is a standard evolutionary multiobjective algorithm. The implementation used crossover and mutation operators already adapted for VRPs [19], with a population size of 500 evolving over 250 iterations;
 4. indicator-Based Multi-Objective Local Search (IBMOLS) [3], a local search-based method for multiobjective optimization included to provide a better comparison baseline. The implementation used the epsilon binary indicator, employed the same local search operators as the MOILS and the RM population generation method with the mutation based on perturbation mechanism of the MOILS. Preliminary testing suggested using a scale factor of 0.1, population size of 10 and 50 generation.

To evaluate the relative performance of these approaches, three performance metrics were considered: the hypervolume (S), standardized to the interval (0, 1) independently for each problem; the Coverage of Two Sets [37]; and the cardinality (Card) of the final solution set returned. Both the S and CS-metrics return values representing percentages, while the Cardinality metric provides absolute values.

The free parameters of the algorithms used were adjusted in order to provide each method with approximately the same run time. No significant differences were detected in the runtime of the algorithms tested ($p > 0.05$).

3.2 Statistical design

We have employed statistical tests designed to detect significant differences and to estimate their magnitude for each quality metric. The data used was composed of the metrics calculated for the final Pareto-fronts obtained on eight independent runs of each algorithm on each problem.

For each metric, the experiment was designed as a randomized complete block design (RCBD) with the algorithms as levels of the experimental factor, and the problems as a blocking factor [24]. By treating the problems as blocks, it was possible to model and remove the effects of different instances on the performance of the algorithm, and obtain an overall performance difference across all test instances used. The null hypotheses of absence of differences among the algorithms evaluated over all problems were considered against two-sided alternatives. To avoid the assumptions of the F test, the more versatile Gore test [11], a more robust alternative for the ANOVA, was employed.

After testing for significance, least squares estimators of the block (instance) effects were obtained [24] and subtracted from the samples, thus allowing a problem-independent estimation of the effect size for each algorithm. The estimations of effect size were calculated by means of the Hodges–Lehmann (HL) estimator of the median of differences between two independent samples [17]. A more detailed description of the statistical methods is available in an earlier work [1].

Table 1 Mean values and standard deviations (in the parentheses) for the performance metrics used

Instances (CMT)	Hypervolume				Cardinality			
	MOILS		P _ε		MOILS		P _ε	
	MOILS	NSGA-II	IBMOLS	IBMOLS	MOILS	NSGA-II	IBMOLS	IBMOLS
1X	0.91 (0.16)	0.86 (0.16)	0.97 (0.02)	0.97 (0.02)	3.12 (0.6)	2.12 (0.3)	6.25 (3.6)	2.75 (0.5)
1Y	0.95 (0.02)	0.87 (0.07)	0.91 (0.03)	0.91 (0.03)	11.12 (1.1)	7.5 (1.2)	8.62 (1.5)	7.87 (1.9)
2X	0.88 (0.07)	0.83 (0.10)	0.83 (0.08)	0.83 (0.08)	8.12 (2.8)	4.00 (1.6)	9.50 (2.1)	4.50 (2.1)
2Y	0.93 (0.01)	0.84 (0.06)	0.89 (0.02)	0.89 (0.02)	33.62 (6.5)	31.12 (5.3)	27.25 (4.7)	25.12 (4.3)
3X	0.94 (0.05)	0.91 (0.05)	0.86 (0.07)	0.86 (0.07)	6.37 (0.9)	3.37 (1.1)	8.87 (3.5)	4.37 (1.3)
3Y	0.96 (0.00)	0.86 (0.04)	0.88 (0.06)	0.88 (0.06)	16.63 (2.6)	15.5 (5.1)	13.63 (2.9)	15.38 (4.0)
4X	0.90 (0.04)	0.78 (0.06)	0.73 (0.07)	0.73 (0.07)	15.25 (2.6)	5.0 (2.5)	12.63 (5.0)	6.50 (1.6)
4Y	0.95 (0.02)	0.83 (0.09)	0.92 (0.04)	0.92 (0.04)	42.13 (3.3)	35.5 (8.0)	32.63 (5.5)	30.7 (11.9)
5X	0.91 (0.04)	0.85 (0.06)	0.87 (0.05)	0.87 (0.05)	17.25 (4.2)	8.25 (5.2)	16.25 (4.3)	6.75 (1.8)
5Y	0.94 (0.01)	0.83 (0.03)	0.91 (0.02)	0.91 (0.02)	51.63 (5.9)	59.88 (14.1)	52.0 (10.6)	32.13 (4.8)
11X	0.94 (0.03)	0.87 (0.03)	0.88 (0.02)	0.88 (0.02)	44.75 (5.5)	40.75 (7.1)	42.13 (5.3)	21.25 (3.3)
11Y	0.91 (0.08)	0.70 (0.03)	0.76 (0.09)	0.76 (0.09)	27.63 (5.6)	20.5 (2.8)	20.25 (3.8)	14.75 (3.3)
12X	0.96 (0.02)	0.87 (0.08)	0.88 (0.04)	0.88 (0.04)	14.63 (3.7)	14.38 (3.4)	9.25 (2.2)	11.75 (4.5)
12Y	0.98 (0.01)	0.91 (0.03)	0.90 (0.05)	0.90 (0.05)	32.75 (3.7)	20.00 (5.4)	14.50 (3.6)	13.25 (2.9)
Mean	0.93 (0.04)	0.84 (0.06)	0.87 (0.05)	0.87 (0.05)	23.21 (3.5)	19.13 (4.51)	19.55 (4.19)	14.08 (3.44)

Table 1 continued

Instances (CMT)	Coverage		
	MOILS	P_ϵ	IBMOLS
1X	0.80 (0.29)	0.53 (0.09)	0.65 (0.15)
1Y	0.53 (0.24)	0.63 (0.20)	0.36 (0.12)
2X	0.63 (0.18)	0.54 (0.13)	0.65 (0.30)
2Y	0.67 (0.06)	0.42 (0.09)	0.55 (0.15)
3X	0.59 (0.37)	0.60 (0.14)	0.72 (0.25)
3Y	0.74 (0.08)	0.34 (0.05)	0.56 (0.13)
4X	0.67 (0.17)	0.59 (0.10)	0.48 (0.09)
4Y	0.68 (0.12)	0.52 (0.16)	0.40 (0.12)
5X	0.64 (0.19)	0.59 (0.15)	0.56 (0.18)
5Y	0.73 (0.09)	0.39 (0.06)	0.55 (0.08)
11X	0.72 (0.08)	0.56 (0.06)	0.39 (0.10)
11Y	0.71 (0.14)	0.55 (0.06)	0.41 (0.10)
12X	0.73 (0.11)	0.42 (0.08)	0.58 (0.07)
12Y	0.80 (0.10)	0.31 (0.02)	0.49 (0.09)
Mean	0.69 (0.16)	0.50 (0.10)	0.53 (0.14)

Table 2 Estimated difference in average performance between the row and column algorithms for the performance metrics

	Hypervolume			Coverage			Cardinality		
	P_ϵ	NSGA-II	IBMOLS	P_ϵ	NSGA-II	IBMOLS	P_ϵ	NSGA-II	IBMOLS
MOILS	0.11	0.08	0.06	0.70	0.32	0.13	2.56	4.32	6.06
P_ϵ	–	–0.03	–0.05	–	–0.28	–0.54	–	n.s.	3.50
NSGA-II	–	–	n.s.	–	–	–0.18	–	–	1.73

Only results significant at a 95 % confidence level (adjusted for multiple hypothesis testing using Dunn-Šidák correction [7]) are shown. Positive values indicate higher average value for the algorithm in the row. OBS:

n.s. not statistically significant

3.3 Results and discussion

The results obtained for the experimental comparison are summarized in Table 1, which reports the mean and standard deviation values of the three metrics considered in each problem; and Table 2, where the results of the statistical analysis are summarized and the magnitude of the statistically significant differences are presented.

For the hypervolume, all algorithms presented a relatively solid performance in most problems, as shown in Table 1, with the P_ϵ returning slightly smaller values than the other three methods. The statistical analysis reported in Table 2 confirms this observation, showing that all algorithms presented small but statistically significant differences, with the MOILS showing the best results (11 % superior to the P_ϵ , 8 % better than the NSGA-II and 6 % superior to the IBMOLS). This result indicates that the proposed approach was generally able to obtain better fronts, either by returning a well-spread set of solutions, or points that were closer to the real Pareto-optimal front.

The average differences obtained for the cardinality metric were also relatively modest. Table 1 shows that, for each particular problem, all algorithms returned approximately the same mean number of solutions, with no large trends being easily discernible. Table 2 shows that, when integrated over all problems, small but statistically significant differences were observed, with the MOILS returning on average 2.56 more solutions than the P_ϵ , 4.32 more than the NSGA-II, and 6.06 more than the IBMOLS.

The differences in performance observed for the coverage metric were considerably large. By examining the results in Table 2, it can be easily seen that the MOILS was able to outperform the P_ϵ by an average of 0.7, the NSGA-II by 0.32 and the IBMOLS by 0.13. This is a strong indicator of superiority, since it means that, on average, the fronts found by the MOILS were able to dominate, on average, about 70 % of the ones yielded by the P_ϵ method, 32 % of those obtained by the NSGA-II and 13 % of the fronts returned by the IBMOLS. To simplify the reporting of the relative coverage values, a generalized version of the Coverage of Two Sets, called Coverage of Many Sets [2], was employed to generate the data shown in Table 1. Instead of quantifying how much a given algorithm covers another, this generalized metric instead measures how much a given algorithm covers the *union* of the final fronts returned by all algorithms except itself.

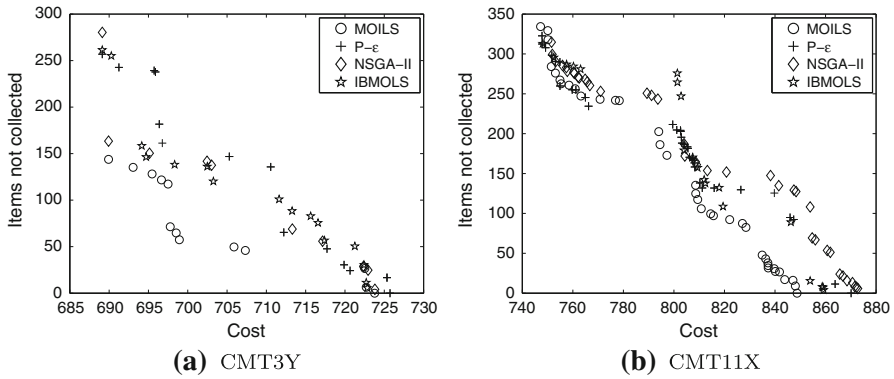


Fig. 2 Typical fronts obtained by the algorithms on selected problems

Table 3 Comparison with best known single-objective results for the VRSPD

	Instances (CMT)						
	1X	1Y	2X	2Y	3X	3Y	4X
MOILS	466.77	466.77	688.05	693.09	721.40	723.28	857.44
Best known	466.77 ^a	466.77 ^a	668.77 ^b	663.25 ^b	721.27 ^a	721.27 ^a	852.46 ^b
Error* (%)	0.00	0.00	2.88	4.50	0.02	0.28	0.58
	Instances (CMT)						
	4Y	5X	5Y	11X	11Y	12X	12Y
MOILS	855.74	1037.25	1032.68	848.72	850.67	665.85	664.00
Best known	852.35 ^b	1029.25 ^b	1029.25 ^b	833.92 ^b	830.39 ^b	644.70 ^b	659.52 ^b
Error* (%)	0.40	0.78	0.33	1.77	2.44	3.28	0.68

* Percent difference between the results obtained by MOILS and the best known

^a Optimal solution obtained by Branch-and-cut [33]

^b Results of the parallel ILS-RVND algorithm reported in the literature [32]

Figure 2 shows two representative examples of fronts obtained by the algorithms. Since the true Pareto-optimal front is not known for the problems considered, it is not possible to objectively evaluate the absolute quality of the fronts obtained. However, the extreme point of the fronts obtained for the case in which all pickups are performed is equivalent to the corresponding VRSPD instance. Table 3 shows a comparison of the extreme point of the non-dominated fronts obtained by the MOILS with the best known results for this problem [32], showing that, at least for the extreme points, the MOILS approach is able to find solutions that are generally very close to the best available in the literature.

4 Conclusions

We proposed an adaptation of the mathematical formulation for the multiobjective VRP with fixed delivery and optional collections, which takes into account the total

cost of the solution and the number of pickups performed. A multiobjective version of the iterated local search algorithm (MOILS) was proposed and compared on a test set of 14 benchmark problems against the P_ϵ , NSGA-II and IBMOLS approaches. The results obtained show that the MOILS was able to significantly outperform the three methods, obtaining superior values for coverage, hypervolume and cardinality across the set of test problems used.

Acknowledgments This work was supported by the following agencies: National Council for Research and Development (CNPq), grants 306910/2006-3 and 472446/2010-0; the Coordination for the Improvement of Higher Education Personnel (CAPES); and the Research Foundation of the State of Minas Gerais (FAPEMIG, Brazil), grants Pronex: TEC 01075/09 and Pronem: CEX APQ-04611-10.

References

- Batista, L.S., Campelo, F., Guimarães, F.G., Ramírez, J.A.: Pareto cone ϵ -dominance: improving convergence and diversity in multiobjective evolutionary algorithms. *Lect. Notes Comput. Sci.* (Springer) **6576**, 76–90 (2011)
- Batista, L.S., Campelo, F., Guimarães, F.G., Ramírez, J.A.: The cone ϵ -dominance: an approach for evolutionary multiobjective optimization. *Evol. Comput.* (submitted)
- Basseur, M., Liefoghe, A., Le, K., Burke, E.K.: The efficiency of indicator-based local search for multi-objective combinatorial optimization problems. *J. Heuristics* **18**(2), 263–296 (2011)
- Chankong, V., Haimes, Y.Y.: *Multiobjective decision making theory and methodology*. Elsevier Science, New York (1983)
- Czyzak, P., Jaszkiwicz, A.: Pareto simulated annealing: a metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Crit. Decis. Anal.* **3**, 83–104 (1998)
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
- Dunn, O.J.: Multiple Comparisons Among Means. *J. Am. Stat. Assoc.* **56**, 52–64 (1961)
- Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.* **3**, 1–16 (1995)
- Gandibleux, X., Mezdaoui, N., Fréville, A.: A tabu search procedure to solve multiobjective combinatorial optimization problems. In: Caballero, R., Steuer, R. (eds.) *Proceedings Volume of MOPGP 96*. Springer, Berlin (1996)
- Goh, C.K., Ong, Y.S., Tan, K.C.: *Multiobjective memetic algorithms*. Springer, Berlin (2009)
- Gore, A.: Some nonparametric tests and selection procedures for main effects in two-way layouts. *Ann. Inst. Stat. Math.* **27**(1), 487–500 (1973)
- Gendreau, M., Potvin, J.Y., Bräysy, O., Hasle, G., Løkketangen, A.: Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: Golden, B., Raghavan, S., Wasil, E. (eds.) *The Vehicle Routing Problem—Latest Advances and New Challenges*. Springer, New York (2008)
- Gribovskaia, I., Laporte, G., Shyshou, A.: The single vehicle routing problem with deliveries and selective pickups. *Comput. Oper. Res.* **35**, 2908–2924 (2008)
- Gutiérrez-Jarpa, G., Marianov, V., Obreque, C.: A single vehicle routing problem with fixed delivery and optional collections. *IIE Trans.* **41**, 1067–1079 (2009)
- Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., Marianov, V.: A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *Eur. J. Oper. Res.* **206**, 341–349 (2010)
- Hansen, M.P.: Tabu search for multiobjective optimization: MOTS. *MCDM Conference* (1997)
- Hodges, J., Lehmann, E.: Estimation of location based on ranks. *Ann. Math. Stat.* **34**, 598–611 (1963)
- Jaszkiwicz, A.: Genetic local search for multi-objective combinatorial optimization. *Eur. J. Oper. Res.* **137**(1), 50–71 (2002)
- Jozefowicz, N., Semet, F., Talbi, E.G.: An evolutionary algorithm for the vehicle routing problem with route balancing. *Eur. J. Oper. Res.* **195**(3), 761–769 (2007)

20. Jozefowicz, N., Smet, F., Talbi, E.G.: Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **189**(2), 293–309 (2008)
21. Lee, T.-R., Ueng, J.-H.: A study of vehicle routing problem with load balancing. *Int. J. Phys. Distrib. Logist. Manag.* **29**, 646–648 (1998)
22. Lourenço H.R., Martin O.C., Stützle T.: Iterated local search. In: Glover F., Kochenberger G.A. (eds.) *Handbook of Metaheuristics*, pp. 321–353. Kluwer Academic Publishers, Boston (2003)
23. Montané, F.A.T., Galvão, R.D.: A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput. Oper. Res.* **33**, 595–619 (2006)
24. Montgomery, D.: *Design and analysis of experiments*, 7th edn. Wiley, New York (2008)
25. Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.* **41**, 421–451 (1993)
26. Pacheco, J., Martí, R.: Tabu search for a multiobjective routing problem. *J. Oper. Res. Soc.* **57**(1), 29–37 (2006)
27. Penna, P., Subramanian, A., Ochi, L.: An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *J. Heuristics* (2011). doi:[10.1007/s10732-011-9186-y](https://doi.org/10.1007/s10732-011-9186-y)
28. Ribeiro, R., Lourenco, H.R.: A multi-objective model for a multi-period distribution management problem. In: *Metaheuristic International Conference (MIC)*, 97–101 (2001)
29. Salhi, S., Nagy, G.: A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J. Oper. Res. Soc.* **50**, 1034–1042 (1999)
30. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: *International Conference on Genetic Algorithm and Their Applications* (1985)
31. Serafini, P.: Simulated annealing for multi-objective optimization problems. In: Tzeng, G.H., Wang, H.F., Wen, V.P., Yu, P.L. (eds.) *Multiple Criteria Decision Making. Expand and Enrich the Domains of Thinking and Application*, pp. 283–292. Springer, Berlin (1994)
32. Subramanian, A., Drummond, L., Bentes, C., Ochi, L., Farias, R.: A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.* **37**(11), 1899–1911 (2010)
33. Subramanian, A., Uchoa, E., Pessoa, A.A., Ochi, L.S.: Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery. *Oper. Res. Lett.* **39**(5), 338–341 (2011)
34. Süral, H., Bookbinder, J.H.: The single-vehicle routing problem with unrestricted backhauls. *Networks* **41**(3), 127–136 (2003)
35. Ulungu, E.L., Teghem, J.: Multi-objective combinatorial optimization problems: a survey. *J. Multi-Crit. Decis. Anal.* **3**, 83–104 (1994)
36. Ulungu, E.L., Teghem, J., Fortemps, Ph: Tuytens: MOSA method: a tool for solving multiobjective combinatorial optimization problems. *J. Multi-Crit. Decis. Anal.* **8**, 221–236 (1999)
37. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**, 117–132 (2003)