MOEA/D for a Tri-objective Vehicle Routing Problem

Andreas Konstantinidis, Savvas Pericleous, Christoforos Charalambous

HAL Id: hal-01459605
https://hal.inria.fr/hal-01459605

Submitted on 7 Feb 2017

MOEA/D for a Tri-Objective Vehicle Routing Problem

Andreas Konstantinidis, Savvas Pericleous and Christoforos Charalambous

Department of Computer Science and Engineering, Frederick University, Nicosia, Cyprus

Abstract. This work examines the Capacitated Vehicle Routing Problem with Balanced Routes and Time Windows (CVRPBRTW). The problem aims at optimizing the total distance cost, the number of vehicles used, and the route balancing, under the existence of time windows and other constraints. The problem is formulated as a Multi-Objective Optimization Problem where all objectives are tackled simultaneously, so as to effect a better solution space coverage. A Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), hybridized with local search elements, is proposed. The application of local search heuristics is not uniform but depends on specific objective preferences and instance requirements of the decomposed subproblems. To test the efficacy of the proposed solutions, extensive experiments were conducted on well known benchmark problem instances and results were compared with other MOEAs.

1 Introduction and Related Work

The Vehicle Routing Problem (VRP) refers to a family of problems in which a set of routes for a fleet of vehicles based at one (or several) depot(s) must be determined for a number of geographically dispersed customers. The goal is to deliver goods to the customers with known demands under several objectives and constraints by originating and terminating at a depot.

The problem has received extensive attention in the literature [1] due to its association with important real-world problems. Several versions and variations of the VRP exist that are mainly classified based on their objectives and constraints [2]. The classic version of the Capacitated VRP (***CVRP***) [1, 3] considers a collection of routes, where each vehicle is associated with one route, each customer is visited only once and aims at minimizing the total distance cost of a solution using the minimum number of vehicles while ensuring that the total demand per route does not exceed the vehicle capacity. The extended *CVRP with Balanced Routes (**CVRPBR**)* [4] introduces the objective of route balancing in order to bring an element of fairness into the solutions. The *CVRP with time windows (**CVRPTW**)* [5] does not include any additional objective but involves the additional constraint that each customer should be served within specific time windows.

CVRP and its variants are proven *NP-hard* [6]. Optimal solutions for small instances can be obtained using exact methods [2], but the computation time increases exponentially for larger instances. Thus, several heuristic and optimization methods [1] are proposed. More recently, metaheuristic approaches are used to tackle harder CVRP instances including Genetic Algorithms [7] and hybrid approaches [3]. Hybrid approaches, which often include combinations of different heuristic and metaheuristic methods such as the hybridization of Evolutionary Algorithms (EAs) with local search

(aka *Hybrid* or *Memetic Algorithms*), have been more effective in dealing with hard scheduling and routing problems [3] than conventional approaches in the past.

When real-life cases are considered, it is common to examine the problem under multiple objectives as decision makers rarely take decisions examining objectives in isolation. Therefore, proposed solutions often attack the various objectives in a single run. This can be done by tackling the objectives individually and sequentially [4], or by optimizing one objective while constraining the others [8] or by aggregating all objectives into one single objective function [9] usually via a weighted summation. Such approaches often lose "better" solutions, as objectives often conflict with each other and the trade-off can only be assessed by the decision maker. Therefore, the context of Multi-Objective Optimization (MOO) is much more suited for such problems.

A *Multi-objective Optimization Problem (MOP)* [10] can be mathematically formulated as follows:

$$\text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^T, \quad \text{subject to } x \in \Omega \tag{1}$$

where Ω is the decision space and $x \in \Omega$ is a decision vector. $F(x)$ consists of m objective functions $f_i : \Omega \to \Re, i = 1, \ldots, m$, and $\Re^m$ is the objective space.

The objectives in (1) often conflict with each other and an improvement on one objective may lead to the deterioration of another. In that case, the best trade-off solutions, called the set of Pareto optimal (or non-dominated) solutions, is often required by a decision maker. The Pareto optimality concept is formally defined as,

Definition 1. A vector $u = (u_1, \ldots, u_m)^T$ is said to dominate another vector $v = (v_1, \ldots, v_m)^T$, denoted as $u \prec v$, iff $\forall i \in \{1, \ldots, m\}$, $u_i \leq v_i$ and $u \neq v$.

Definition 2. A feasible solution $x^* \in \Omega$ of problem (1) is called *Pareto optimal solution*, iff $\nexists y \in \Omega$ such that $F(y) \prec F(x^*)$. The set of all Pareto optimal solutions is called the Pareto Set (PS) and the image of the PS in the objective space is called the Pareto Front (PF).

Multi-Objective Evolutionary Algorithms (MOEAs) [11] are proven efficient and effective in dealing with MOPs. This is due to their population-based nature that allows them to obtain a well-diversified approximation of the PF. That is, minimize the distance between the generated solutions and the true PF as well as maximize the diversity (i.e. the coverage of the PF in the objective space). In order to do that, MOEAs are often combined with various niching mechanisms such as crowding distance estimation [12] to improve diversity, and/or local search methods [13] to improve convergence.

In the literature there are several studies that utilized generic or hybrid Pareto-dominance based MOEAs to tackle Multi-Objective CVRPs and variants [14]. For example, Jozefowiez et al. [15] proposed a bi-objective CVRPBR with the goal to optimize both the total route length and routes balancing. In [16], the authors proposed a hybridization of a conventional MOEA with multiple LS approaches that were selected randomly every 50 generations to locally optimize each individual in the population and tackle a bi-objective CVRPTW. In [17], Geiger have tackled several variations of the CVRPTW by optimizing pairs of the different objectives. Over the past decade numerous variants of the investigated problem have been addressed under a MOP setting, involving different combinations of objectives and different search hybridization elements. For the interested reader, indicative examples include ([18], [19]).

Even though the objectives and constraints presented are all important, challenging, and by nature conflicting with each other, to the best of our knowledge no research work has ever dealt with the minimization of the total distance cost, the number of vehicles and the route balancing objectives as a MOP trying to satisfy all side-constraints, simultaneously. Moreover in all the above studies, MOEAs based on Pareto Dominance (such as NSGA-II [12]) are hybridized either with a single local search approach [18, 19] or with multiple local search heuristics with one being selected randomly [16] each time a solution was about to be optimized locally.

In this paper, we investigate the **CVRPBRTW**, formulated as a MOP composed of three objectives (minimize the total distance cost, minimize the number of vehicles and balance the routes of the vehicles) and all relevant constraints aiming at increasing its practical impact by making it closer to real-life cases. Solutions are obtained through a hybrid MOEA/D [20] approach that decomposes the proposed MOP into a set of scalar subproblems, which are solved simultaneously using neighborhood information and local search methods each time a new solution is generated. Specifically, the MOEA/D is hybridized with multiple local search heuristics that are adaptively selected and locally applied to a subproblem's solution based on specific objective preferences and instant requirements. We examine our proposition on Solomon's benchmark problem instances [5] against several other MOEA/Ds.

2 Multi-Objective Problem Definition and Formulation

The elementary version of the CVRP [1, 3] is often modelled as a complete graph $G(V, E)$, where the set of vertices V is composed of a unique depot $u_0 = o$ and l distinct customers, each based at a prespecified location. The Euclidean distance between any pair of customers is associated with the corresponding edge in E. Each customer must be served a quantity of goods (customer's demand) that requires a predefined service time. To deliver those goods, K identical vehicles are available, which are associated with a maximal capacity of goods that they can transport. Vehicles traverse a unit distance in unit time and time is measured as time elapsed from commencing operations. A solution of the CVRP is a collection of routes, where each route is a sequence of vertices starting and ending at the depot and served by a single vehicle, each customer is visited only once and the total amount of goods transported per route is at most the vehicle's capacity. The CVRP aims at a *minimal total distance cost* of a solution, using *minimum number of vehicles*. In the investigated problem a third objective, that of *route balancing*, is also examined. The balancing objective, which is defined as the difference between the maximum distance traveled by a vehicle and the mean traveled distance of all vehicles [4], brings an element of fairness in solutions. Finally, the well known 'time windows' constraint is imposed. This constraint requires the vehicle serving each customer to arrive within specific time windows.

Note that in the problem variant investigated in this work, if a vehicle arrives at a customer before the earliest arrival time it is allowed to wait until that time is reached, resulting in additional route traveled time. Time windows are treated as a hard constraint in the sense that if the vehicle arrives at a customer after the latest arrival time the solution is considered infeasible.

Therefore, the proposed *CVRP with Balanced Routes and Time Windows (**CVRP-BRTW**)* can be mathematically formulated as follows:

Given,

V	— the set of $l+1$ vertices (customers) composed of a depot o and for $i = 1, ..., l$ vertices u_i located at coordinates (x_i, y_i).
E	— the set of edges (u_i, u_j) for each pair of vertices in $u_i, u_j \in V$ associated with their Euclidean distance $dist(u_i, u_j)$.
$[e_u, e'_u]$	— the time window of customer $u, \forall u \in V$.
q_u	— the quantity demand of customer $u, \forall u \in V$; in particular, $q_o = 0$.
t^s_u	— the service time of customer $u, \forall u \in V$; in particular, $t^s_o = 0$.
K	— the maximum number of vehicles to be used (at most l).
c	— the capacity of each vehicle z.
R^m	— the *route* followed by the mth vehicle used in the solution. The route is defined as a sequence of customer vertices (excluding the depot vertex).
X	— a collection of k routes $X = \{R^1, R^2, ..., R^k\}$ where k is at most K.
$suc(u)$	— given $u \in R^m$, $suc(u)$ is the vertex immediately following u in R^m, if it exists (i.e., u is not the last vertex in R^m), otherwise the depot o.
$pre(u)$	— given $u \in R^m$, $pre(u)$ is the vertex immediately preceding u in R^m, if it exists (i.e., u is not the first vertex in R^m), otherwise the depot o.
$init(R^m)$	— the initial vertex in R^m
t^a_u	— the vehicle arrival time at vertex $u \in V \setminus \{o\}$ which can be calculated by the function $\max\{e_u, t^a_{pre(u)} + t^s_{pre(u)} + dist\big((pre(u), u\big)\}$, with $t^a_o = 0$.
$D^m(X)$	— the total distance covered by the vehicle serving route R^m in solution X obtained by $dist(o, init(R^m)) + \sum_{\forall u \in R^m} dist\big(u, suc(u)\big)$

$$\min F(X) = (D(X), B(X), N(X)) \qquad (2)$$

$$D(X) = \sum_{m=1}^{k} D^m(X) \qquad (3)$$

$$N(X) = k + \left(\min_{1 \le m \le k} \left(\frac{|R^m|}{l} \right) \right) \qquad (4)$$

$$B(X) = \left(\max_{1 \le m \le k} \{D^m(X)\} \right) - \frac{1}{k} D(X) \qquad (5)$$

subject to

$$\sum_{\forall u \in R^m} q_u \ \le c, \ \ \forall m = 1, ..., k \qquad (6)$$

$$e_u \le t^a_u \ \le e'_u \ \ \forall u V \setminus \{o\} \qquad (7)$$

$$\{u\} \cap \bigcup_{m=1,...,k} R^m = \{u\} \ \forall u \in V \setminus \{o\} \qquad (8)$$

$$\sum_{m=1,...,k} |R^m| \ = l \qquad (9)$$

Equation (2) specifies the multi-objective function we wish to minimize, comprising the total distance cost, defined in (3), route balancing, defined in (5), and the number of routes, thus vehicles, used, $k = |X|$. Note that instead of $|X|$, the auxiliary function $N(X)$ defined in (4) is used, as it gives a bias towards solutions with the least customers in the smallest route.

Constraints (6) ensure that the total quantity of goods transported in a route does not exceed the vehicle's capacity, whereas constraints (7) require that the arrival time at all customers is within the corresponding time window. The combination of constraints (8) and (9) guarantee that all customers are served exactly once; constraints (8) ensure that each customer vertex is visited by at least one route, and constraint (9) that the total number of vertices visited is equal to the number of customers.

3   The proposed Hybrid MOEA/D

3.1   Preliminaries

The problem is tackled by a decomposed MOEA. Before explaining the algorithm, the encoding representation used and the solution evaluation algorithm will be explained.

Encoding Representation: In VRP, solutions are often represented by a variable length vector of size greater than l, which consist of all l customers exactly once and the depot, o, one or more times signifying when each vehicle starts and ends its route. Under such a representation, the solution's phenotype (the suggested routes) can readily be obtained, although several issues of infeasibility arise. In this work however, a candidate solution X is a fixed length vector of size l, composed of all customers only. This solution encoding X is translated to the actual solution using the following algorithm. An empty route R_1 is initially created. The customers are inserted in R_1 one by one in the same order as they appear in solution X. A customer u_j that violates any of the constraints of Section 2 is directly inserted in a newly created route R_2. In the case where more than one route is available, and for the remaining customers, a competitive process starts, in which the next customer u_{j+1} in X is allowed to be inserted in any available route that does not violate a constraint. When more than one such routes exist, the one with the shortest distance to the last customer en route is preferred. If a customer violates a constraint in all available routes, a newly created route is initiated. Note that this process guarantees feasibility irrespective of the actual sequence.

Decomposition: In MOEA/D, the original MOP needs to be decomposed into a number of M scalar subproblems. Any mathematical aggregation approach can serve for this purpose. In this article, the Tchebycheff approach is employed as originally proposed in [20].

Let $F(x) = (f_1, ..., f_m)$ be the objective vector, $\{w_1, ..., w_m\}$ a set of evenly spread weight vectors, which remain fixed for each subproblem for the whole evolution, and z^* the reference point. Then, the objective function of a subproblem i is stated as:

$$g^i(X^i|w^i, z^*) = min\{\sum_{j=1}^{m}(w_j^i \hat{f}_j(X) - z_j^*)\}$$

where $w^i = (w^i_1, ..., w^i_m)$ represents the objective weight vector for the specific decomposed problem i, $\hat{f}$ denotes the min-max normalization of f and $z^* = (z_1, ..., z_m)$ is a vector equal to all best values z_j found so far for each objective f_j. MOEA/D minimizes all these objective functions simultaneously in a single run. As stated in [20], one of the major contributions of MOEA/D is that the optimal solution of subproblem i should be close to that of k if w^i and w^k are close to each other in the weight space. Therefore, any information about these g^ks with weight vectors close to w^i should be helpful for optimizing $g^i(X^i|w^i, z^*)$. This observation will be later utilized for improving the efficiency and the adaptiveness of the newly proposed local search heuristic.

Neighborhoods: In MOEA/D, a neighborhood N^i is maintained for each subproblem i of weight vector w^i. Particularly, N^i is composed of the T subproblems of which the weight vectors are closest to w^i, including i itself. T is a parameter of the algorithm. The Euclidean distance is used to measure the closeness between two weight vectors.

3.2   The Evolutionary Algorithm

The algorithm commences by creating an initial population, named Internal Population (IP) of generation $\gamma = 0, IP_0 = \{X^1, ..., X^M\}$. The initial solutions are randomly generated and each individual is evaluated using the process described earlier.

At each step of MOEA/D, for each subproblem i a new solution Y^i is generated through the use of genetic operators. Specifically, using the Neighborhood Tournament Selection (NTS) operator [21], two parent solutions, Pr^1 and Pr^2, are selected from N^i. The two parent solutions are then recombined with a probability rate c_r using the well-known Partially Mapped Crossover (PMX) operator [22] to produce an offspring solution O. Finally, a random mutation operator is utilized to modify each element of solution O with a mutation rate m_r and generate solution Y^i.

After the new solution Y^i is generated for a given subproblem, an attempt is made to improve it through the use of local search. Specifically, one local search heuristic is applied on Y^i, yielding a new solution Z^i. The local search (LS) heuristic used is selected from the following pool [23]:

- *Double Shift (DS):* is a combination of the Backward and Forward Shifts. That is, it initially takes a customer from its current position u_{j_1} and inserts it before a customer u_{k_1}, where $j_1 > k_1$. Then it takes a customer from its current position u_{j_2} and inserts it after a customer u_{k_2}, where $j_2 < k_2$.
- *Lambda Interchange (LI):* First, two routes A and B are chosen. The heuristic starts by scanning through nodes in route A and moves a feasible node into route B. The procedure repeats until a predefined number of nodes are shifted or the scanning ends at the last node of route A.
- *Shortest Path (SP):* attempts to rearrange the order of nodes in a particular route such that the node with the shortest distance from the incumbent is selected.

Central to the proposed approach is the way the local search heuristic (LS) is selected for application each time a new solution is generated. Specifically, in our work the LS is selected based on a weighted probability that is not static among all subproblems but is based on the objective weights each subproblem i holds. Through extensive

experimentation on random solution instances we have established an affinity of each LS described above with an objective function and adopt an association between objective and LS. The associations applied are: SP for distance cost, DS for number of vehicles, and LI for route balancing. As a result, we use the weight value w_j^i for subproblem i and objective j as the weighted probability of selecting the associated LS on that subproblem.

Once all solutions Z^i are constructed, the population is updated as follows. Firstly, solution Z^i replaced the incumbent solution X^i for subproblem i iff it achieves a better value for the specific objective function of that subproblem. Subsequently, in an attempt to propagate good characteristics, Z^i is evaluated against the incumbent solutions X^ks of the T closest neighbors of i. For each of these subproblems, if $g^k(Z^i|w^k, z^*) < g^k(X^k|w^k, z^*)$ then Z^i becomes also the incumbent for subproblem k. Finally, a test is made to check whether Z^i is dominated by any solution in the maintained Pareto Front, and if not, it is added to PF. The aforementioned process is repeated for a prespecified number of generations g_m.

4 Experimental Studies

4.1 Experimental Setup and Performance Measures

The experiments were carried out on the well-known Solomon's instances (100-customer problem sets). These instances are categorized into six classes: C1, C2, R1, R2, RC1 and RC2. Category C problems represent clustered data, which means the customers are clustered either geographically or in terms of the time windows. Category R problems represent uniformly randomly distributed data and RC are combinations of the other two classes. Classes C1, R1 and RC1 consider customers with narrower time windows. The algorithmic settings used are as follows: $c_r = 0.9, m_r = 0.01, T = 10, M = 630$ and $g_m = 3000$. Due to the limited space we present results on a subset of instances.

The performance of an MOEA is usually evaluated from two perspectives: the obtained non-dominated set should be (i) as close to the true Pareto Front as possible, and (ii) distributed as diversely and uniformly as possible. No single metric can reflect both of these aspects and often a number of metrics are used [24]. In this study, we use the **Coverage** C [24] and **distance for reference set** I_D [25] metrics:

$$C(A,B) = \frac{|\{x \in B | \exists y \in A : y \prec x\}|}{|B|}; \quad I_D(A) = \frac{\sum_{y \in R}\{min_{x \in A}\{d(x,y)\}\}}{|R|}.$$

Coverage is a commonly used metric for comparing two sets of non-dominated solutions A and B. The $C(A,B)$ metric calculates the ratio of solutions in B dominated by solutions in A, divided by the total number of solutions in B. Therefore, $C(A,B) = 1$ means that all solutions in B are dominated by the solutions in A. Note that $C(A,B) \neq 1 - C(B,A)$.

The distance from reference set is defined by Czyzzak et al. in [25]. This shows the average distance from a solution in the reference set R to the closest solution in A. The smaller the value of I_D, the closer the set A is to R. In the absence of the real

(a) Conventional MOEA/D (M) vs. MOEA/D with adaptive LS (M-aLS) in C101

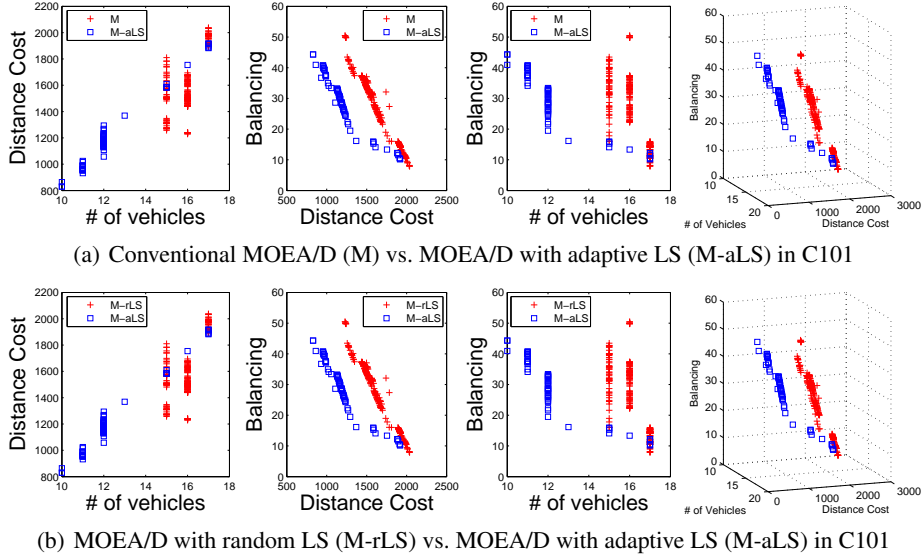(b) MOEA/D with random LS (M-rLS) vs. MOEA/D with adaptive LS (M-aLS) in C101

Fig. 1. Evaluation of the proposed MOEA/D with the adaptive Local Search (M-aLS) with respect to the conventional MOEA/D (M) and the MOEA/D with random local search selection (M-rLS).

reference set (i.e., Pareto Front), we calculate the average distance of each single point to the nadir point since we consider minimization objectives.

4.2 Experimental Results

The proposed MOEA/D-aLS (M-aLS) is evaluated with respect to the conventional MOEA/D as proposed by Zhang and Li in [20] and a MOEA/D with a random local search (M-rLS) selection mechanism. To increase the fidelity of our experimental studies we have repeated each experiment of each algorithm for 30 independent runs, having the same number of function evaluations for fairness.

Figure 1 shows that the hybrid M-aLS improves the performance of the conventional MOEA/D and outperforms the M-rLS in test instance C101 in terms of both convergence and diversity. In particular, the M-aLS has obtained a PF that dominates most of the non-dominated solutions obtained by the other MOEA/Ds providing a better approximation towards the nadir point as well. Note that similar results were obtained in most test instances. This is more evident in Table 1 that summarizes the statistical performance of M-aLS and M-rLS in terms of the Coverage (C) and the Distance to the reference set (I_D). The results show that the non-dominated solutions obtained by the M-aLS dominate most (on average 75%) of the non-dominated solutions obtained by M-rLS and performs no worse on average than M-rLS in terms of I_D. Finally, the results in Table 2, which summarize the best objective values obtained by each approach during the evolution, clearly show that the performance of the proposed M-aLS is better than its competitors.

Table 1. MOEA/D with proposed adaptive Local Search (M-aLS) is compared to MOEA/D with random LS selection (M-rLS) based on C and I_D metrics. Best results are denoted in bold.

Test Inst.	C(M-rLS,M-aLS)	C(M-aLS,M-rLS)	I_D(M-rLS)	I_D(M-aLS)
C101:	0.03	**0.74**	43.61	**31.99**
C201:	0	**1**	33.68	**20.14**
R101:	0	**0.86**	**19.16**	22.68
R201:	0.33	**0.36**	**29.04**	36.78
RC101:	0.1	**0.66**	**15.11**	33.47
RC201:	0	**0.88**	42.25	**20.3**

Table 2. M-aLS is compared with conventional MOEA/D and M-rLS in terms of best solutions found for each objective. Best results are denoted in bold.

Test Inst.	M			M-rLS			M-aLS		
	V	D	B	V	D	B	V	D	B
C101:	15.0	1229.150	7.913	**12.0**	1023.474	9.802	**12.0**	**933.462**	**6.824**
C201:	9.0	1153.066	4.605	5.0	706.534	3.459	**3.0**	**625.197**	**1.302**
R101:	23.0	1934.568	12.642	22.0	1906.092	10.979	**21.0**	**1823.122**	**10.835**
R201:	10.0	1699.647	6.133	**8.0**	1351.344	**4.469**	**8.0**	**1350.925**	4.894
RC101:	21.0	2039.398	7.646	**18.0**	1891.283	9.779	**18.0**	**1849.835**	**6.545**
RC201:	10.0	1843.807	9.295	**8.0**	1606.290	**4.767**	**8.0**	**1533.477**	5.311

5 Conclusions and Future Work

The Tri-Objective Capacitated Vehicle Routing Problem with Balanced Routes and Time Windows is proposed and tackled with a Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) hybridized with local search. The MOEAD-aLS decomposes the proposed MOP into a set of scalar subproblems which are solved simultaneously using at each generation multiple LSs adaptively selected based on objective preferences and instant requirements. We evaluate our proposition on a subset of the standard benchmark problem instances. The results show that the MOEA/D-aLS clearly improves the performance of the MOEA/D in all cases and of MOEA/D-rLS in most cases. In the future, we aim at incorporating learning for the selection of a local search approach to further improve the performance of the MOEA/D.

References

1. Goel, A., Gruhn, V.: A general vehicle routing problem. European Journal of Operational Research **191**(3) (2008) 650 – 660
2. Laporte, G.: Fifty years of vehicle routing. Transportation Science (2009)
3. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research **31** (2004) 1985–2002
4. Lee, T.R., Ueng, J.H.: A study of vehicle routing problems with load-balancing. International Journal of Physical Distribution & Logistics Management **29**(10) (1999) 646–657
5. Solomon, M.M.: Algorithms for the vehicle routing problem with time windows. Transportation Science **29**(2) (1995) 156–166

6. Lenstra, J.K., Rinnooy Kan, A.H.G.: Complexity of vehicle routing and scheduling problems. Networks **11**(2) (1981) 221–227

7. Reeves, C.R.: A genetic algorithm for flowshop sequencing. Special issue on genetic algorithms in Computers and Operations Research **22**(1) (1995) 5–13

8. Chen, J., Chen, S.: Optimization of vehicle routing problem with load balancing and time windows in distribution. In: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Co. (2008)

9. Ombuki, B., J.Ross, B., Hanshar, F.: Multi-objective genetic algorithms for vehicle routing problem with time windows. Applied Intelligence **24** (February 2006) 17–30

10. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley & Sons (2002)

11. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, Elsevier **1**(1) (2011) 32–49

12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA II. IEEE Transactions on Evolutionary Computation **6**(2) (2002) 182–197

13. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation **7**(2) (2003) 204–223

14. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. European Journal of Operational Research (2008) 293–309

15. Jozefowiez, N., Semet, F., Talbi, E.G.: An evolutionary algorithm for the vehicle routing problem with route balancing. European Journal of Operational Research (2009) 761–769

16. Tan, K.C., Chew, Y.H., Lee, L.H.: A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. Comput. Optim. Appl. **34** (May 2006) 115–151

17. Geiger, M.J.: A computational study of genetic crossover operators for multi-objective vehicle routing problem with soft time windows. CoRR (2008)

18. Tan, K., Cheong, C., Goh, C.: Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. European Journal of Operational Research **177**(2) (2007) 813–839

19. Ghoseiri, K., Ghannadpour, S.F.: Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Appl. Soft Comput. (2010) 1096–1107

20. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6) (2007) 712–731

21. Konstantinidis, A., Yang, K., Zhang, Q., Zeinalipour-Yazti, D.: A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. New Network Paradigms, Elsevier Computer Networks **54** (2010) 960–976

22. Goldberg, D.E., R, L.: Alleles, loci, and the traveling salesman problem. In Grefenstette, J.J., ed.: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Publishers (1985)

23. Tan, K.C., Chew, Y.H., Lee, L.H.: A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. European Journal of Operational Research (2006) 855–885

24. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evolutionary Computation (1999) 257–271

25. Czyzak, P., Jaszkiewicz, A.: Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis **7**(1) (1998) 34–47