Adaptive Evolutionary Algorithm for a Multi-Objective VRP

3 authors:


Andreas Konstantinidis
Frederick University


Savvas Pericleous
Frederick University


Christoforos Charalambous
Frederick University

Adaptive Evolutionary Algorithm for a Multi-Objective VRP

Andreas Konstantinidis*, Savvas Pericleous*, 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 subject to 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 Hybrid Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) with local search heuristics is proposed. The application of local search heuristics is not uniform but depends on specific objective preferences and instance requirements. To test the efficacy of the proposed approach, extensive experiments were conducted on well known benchmark problem instances and results were compared with other MOEAs.**

***Index Terms*—multi-objective optimization, evolutionary algorithms, decomposition, adaptive local search, vehicle routing problem**

I. Introduction

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] also includes 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:

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

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. 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], [12] 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 [13] to improve diversity, and/or local search methods [14] 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 [15]. For example, Jozefowiez et al. [16] proposed a bi-objective CVRPBR with the goal to optimize both the total route length and routes balancing. In [17], 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 [18], 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 [19] and [20].

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 of the CVRP, CVRPTW and CVRPBR, simultaneously. For the remaining of this article we will refer to this MOP as the CVRPBRTW (CVRP with Balanced Routes and Time Windows.)

Moreover in all the above studies, MOEAs based on Pareto Dominance (such as NSGA-II [13]) are hybridized either with a single local search approach [19], [20] or with multiple local search heuristics with one being selected randomly [17] each time a solution was about to be optimized locally.

In this paper, **CVRPBRTW** is investigated and 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 Multi-Objective Evolutionary Algorithm based on decomposition (MOEA/D) [21]. In this approach, the proposed MOP is decomposed into a set of scalar subproblems, which are solved simultaneously using neighborhood information and local search methods each time a new solution is generated. In particular, 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.

This work is an extension of our preliminary work in [22], in which common local search (LS) heuristics [23] were employed (Double Shift, Lambda Interchange and Shortest Path) and combined with MOEA/D. Through extensive experimentation on random solution instances we established an affinity of each LS with an objective function and adopted an association between objectives and LS's.

In this work, we are able to obtain improved results by replacing the above pool of LS heuristics with three newly designed ones, having the property that each one exhibits preference on a different objective.

Major contributions of this paper include the following:

- Define and formulate as MOP a Tri-Objective Capacitated Vehicle Routing Problem with Balanced Routes and Time Windows (CVRPBRTW).
- Propose a Multi-Objective Evolutionary Algorithm based on Decomposition hybridized with an adaptive local search mechanism (MOEA/D-aLS). An important element of the proposed method is the way the newly designed LS heuristics are selected for application each time a new solution is generated: this is done based on a weighted probability which depends on the objective weights each subproblem holds.
- Results show that the MOEA/D-aLS consistently improves the performance of the conventional MOEA/D and MOEA/D-rLS (MOEA/D with random application of LS heuristics). Furthermore, in many cases the solutions obtained by MOEA/D-aLS, improve the best component-wise solutions found by MOEA/D-iLS (MOEA/D with individual application of LS heuristic, i.e., always applying one of the three LS).

The remaining of this paper is as follows. Related work on variants of the Capacitated Vehicle Routing Problem and Multi-Objective Evolutionary algorithms employed to solve such problems are presented next. The proposed Multi-Objective problem definition and formulation, namely CVRPBRTW, is described in Section III. In Section IV, an MOEA/D (Multi-Objective Evolutionary Algorithm based on decompositions) approach, combined with an adaptive strategy of applying local search heuristics is proposed for tackling the proposed problem. In Section V, the performance of our proposed method is evaluated on the well studied Solomon's benchmark problem instances [5] and compared against several other MOEA/D variants. Finally, Section VI concludes the paper and provides insights for possible future directions.

II. Related Work

In this section, we introduce and discuss the most prominent and relevant research work on CVRP and MOO.

A. *Capacitated Vehicle Routing Problem (CVRP) & variants*

In the literature, several studies have tried to extend classic VRPs to improve their practical applications [4], [24], [16], to generalize them [25], [18], [19] and to study real-life cases [26], [23], [27]. In order to do that multiple objectives are often identified (mainly from various Single Objective Optimization (SOO) variants) and tackled at the same time. For example, Lee and Ueng [4] have developed an integer linear programming model of the CVRPBR to firstly minimize the total distance and secondly balance the workload among employees by using a hybrid GA. Ombuki et al. [9] have tackled the number of vehicles and total distance cost objectives of the CVRPTW by aggregating them into a single objective function using the weighted sum approach. Furthermore, Chand et al. [28] have tackled the traditional CVRP by aggregating the number of vehicles and distance cost minimization objectives into a single objective function. In 2010, Kritikos and Ioannou [29] have formulated a challenging multi-objective CVRPTW including three objectives, i.e. the distance cost, the number of vehicles and the routes balance, which were tackled as an aggregated single objective function using the weighted sum approach. Similarly, Chen and Chen [8] have proposed a similar MOP but instead of aggregating the objective functions using weights, the authors have tackled the distance cost objective individually and constrained the number of vehicles as well as the balancing objectives to some pre-defined values.

Other variants of the CVRP include the Multiple Depots VRP (MDVRP) [30] that aims at initially assigning customers to depots and a fleet of vehicles is based at each depot. Each vehicle originate from one depot, service the customers assigned to that depot, and return to the same depot. The Periodic VRP (PVRP) [31], [32] that is generalized by extending the planning period from a single day to several days. Split Delivery VRP (SDVRP) relaxes the original VRP by allowing customers to be served by different vehicles if the overall cost is reduced. The VRP with Pick-ups and Deliveries (VRPPD) [33] that includes pick-ups in addition to deliveries during the route, therefore a solution should also consider the possibility that the customers may also return some goods and try to fit them in the vehicles. The VRP with Backhauls (VRPB) [34] is similar to VRPPD with the main difference that in VRPB all deliveries of goods must be completed before any pick-ups are made. Effectual surveys that include several variants as well as the methodologies used to tackle them can be found in [2], [35].

B. CVRP & Multi-Objective Optimization (MOO)

Here it is important to notice that all latter studies consider the multiple objectives individually and sequentially [4], or by optimizing one and constraining the others [8] or by aggregating all objectives into one single objective function [9], [28], [29]. This often results in losing "better" solutions, since multiple objectives often conflict with each other and an optimal trade-off is required by the decision maker. Some research studies have dealt with a CVRP and its variants from a MOO point of view and focused at obtaining a set of near-optimal solutions. For example in [24], the authors used an ant colony optimization technique to tackle a Dynamic CVRP aiming at minimizing the total mean transit time and total variance in transit time. Similarly, Murata and Itai [36] have defined a MOO CVRP that aimed at minimizing both the number of vehicles and the maximum routing time of those vehicles. Jozefowiez et al. [16] proposed a bi-objective CVRPBR with the goal to optimize both the total route length and routes balancing. Hong and Park [25] formulated a bi-objective CVRPTW having as major goal to minimize the total route transit time and the total customer waiting time. In [18], Geiger have tackled several variations of the CVRPTW by optimizing pairs of the following objectives, minimize the distance cost, minimize the travel time, minimize the number of vehicles and maximize the service. i.e. minimize the time windows violations. Furthermore, Baran and Schaerer [26] have tackled a tri-objective optimization CVRPTW dealing with minimizing the number of vehicles, the total travel time and the total delivery time. Similarly, Tan et al. [19] have tackled a tri-objective CVRP including minimizing the travel distance, the driver remuneration (i.e. the driver's cost per hour) and the number of vehicles. Recently in 2011, Najera and Bullinaria [37] have tackled a multi-objective CVRPTW by tackling the number of vehicles, total travel distance and total travel time objectives, simultaneously. Please refer to Jozefowiez et al. [15] for a detailed survey on Multi-Objective VRPs. Even though the objectives and constraints of the CVRP, CVRPTW and CVRPBR are all important, challenging and by nature conflicting with each other, no research work has ever dealt with them simultaneously.

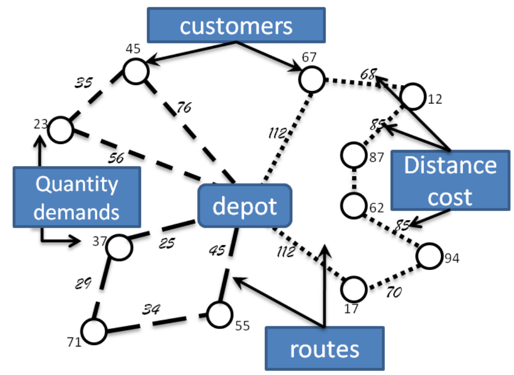


Fig. 1. The Capacitated Vehicle Routing Problem (CVRP)

III. 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 $\{u_1, \ldots, u_l\}$, with customer u_i based at location (x_i, y_i) and the Euclidean distance $dist(u_i, u_j)$ between customers u_i and u_j associated with each edge $(u_i, u_j) \in E$. Each customer $u \in V$ must be served a quantity q_u (also known as customer's demand) of goods that requires a pre-defined service time t^s_u. We denote by t^a_u the arrival time at customer u, assuming that unit distance is traversed in unit time and that time is measured as time elapsed from commencing operations. To deliver those goods, K identical (i.e. of same type, capacity etc.) vehicles are available, which are associated with a maximal capacity c of goods that they can transport. A solution of the CVRP is a collection of routes $X = \{R^1, ..., R^k\}$, where each route R^m is a sequence of vertices starting and ending at depot o and served by a single vehicle, each customer u is visited only once and the total amount of goods transported per route is at most c. The CVRP aims at a minimal total distance cost $D(X)$ of a solution, using minimum number of vehicles k.

In Figure 1, the customers u are indicated with empty circles; the required quantity demands q_u are shown just next

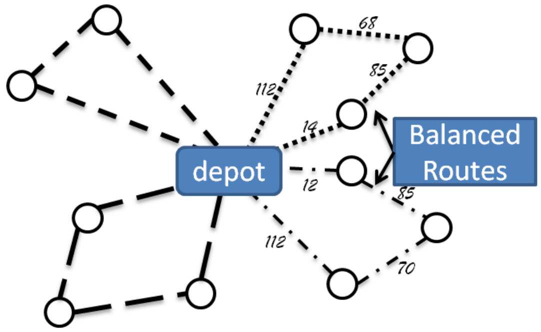


Fig. 2. The CVRP with Balanced Routes (CVRPBR)

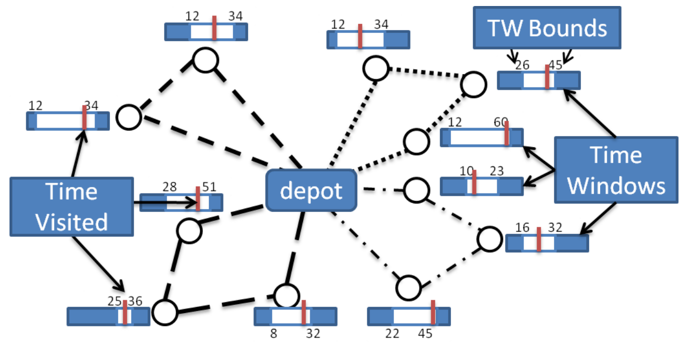


Fig. 3. The CVRP with Time Windows (CVRPTW)

to them. Four independent routes are denoted with different types of lines, with the distance cost between customers covered by each vehicle shown in the middle of each edge.

The *CVRP with Balanced Routes (**CVRPBR**)* [4] extends the elementary CVRP by introducing the objective of routes balancing in order to bring an element of fairness into the solutions. The CVRPBR aims at also minimizing the routes balancing objective $B(X)$ which we define as the difference between the maximum distance traveled by a vehicle and the mean distance travelled by all vehicles in X.

In Figure 2, two balanced routes are shown on the right hand side of the figure, denoted with different types of lines. In this example, the distance covered by corresponding vehicles associated to each route is the same and sums up to 279m.

The *CVRP with time windows (**CVRPTW**)* [5] does not include any additional objective, but it involves an additional 'time windows' constraint: the vehicle serving each customer u should arrive within specific time windows $[e_u, e'_u]$. The depot is also associated with a time window $[e_o, e'_o]$, which constrains the total travel time of a vehicle from departure to arrival. Note that in the problem variant investigated in this work, if a vehicle arrives at a customer u before the earliest arrival time e_u, 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 u after the latest arrival time e'_u, then the solution is considered infeasible.

In Figure 3, each customer is associated with a time window indicated by a rectangle next to it. With each rectangle two integer numbers representing the lower and upper bounds of the time window are shown. The vertical line in the rectangle indicates a feasible visit (arrival time) of the vehicle at the customer en route.

The proposed *CVRP with Balanced Routes and Time Windows (**CVRPBRTW**)* aims at optimizing all required objectives and satisfying all constraints without violating the time windows; mathematically it can be formulated as follows:

Given:

- V = set of $l+1$ vertices composed of a depot o and customers u_i located at coordinates (x_i, y_i) for $i = 1, ..., l$.
- E = set of edges (u_i, u_j) for each pair of vertices in $u_i, u_j \in V$ associated with their Euclidean distance $dist(u_i, u_j)$.
- $[e_u, e'_u]$ = time window of customer $u, \forall u \in V$.
- q_u = quantity demand of customer $u, \forall u \in V$; $q_o = 0$.
- t^s_u = service time of customer $u, \forall u \in V$; $t^s_o = 0$.
- K = max number of vehicles to be utilized (at most l).
- c = capacity of each vehicle.
- R^m = *route* followed by the m^{th} 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.
- $suc(u)$ = the vertex immediately following u in R^m, for some customer $u \in R^m$, if it exists (i.e., u is not the last vertex in R^m), otherwise the depot o.
- $pre(u)$ = the vertex immediately preceding u in R^m, for some customer $u \in R^m$, 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 $u \in V \setminus \{o\}$; taking $t^a_o = 0$, this can be calculated by the function
$$\max\{e_u, t^a_{pre(u)} + t^s_{pre(u)} + dist\big((pre(u), u)\big\}.$$

Let $D^m(X)$ denote the total distance covered by vehicle serving route R^m in solution X:

$$D^m(X) = dist(o, init(R^m)) + \sum_{\forall u \in R^m} dist\big(u, suc(u)\big)$$

Problem Objectives

$$\min\ F(X) = (N(X), D(X), B(X)) \tag{2}$$

where,

$$N(X) = k + \left(\min_{1 \le m \le k}\left(\frac{|R^m|}{l}\right)\right) \tag{3}$$

$$D(X) = \sum_{m=1}^{k} D^m(X) \tag{4}$$

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

subject to

$$\sum_{\forall u \in R^m} q_u \quad \leq c, \qquad \forall m = 1, ..., k \tag{6}$$

$$e_u \leq t_u^a \quad \leq e'_u \qquad \forall u V \setminus \{o\} \tag{7}$$

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

$$\sum_{m=1,...,k} |R^m| \quad = l \tag{9}$$

Equation (2) specifies the multi-objective function we wish to minimize, comprising the total distance cost, defined in (4), 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 (3) 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.

IV. The proposed Hybrid MOEA/D

In this section, the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) framework is initially introduced and its major operators are discussed. Then the proposed algorithm, namely MOEA/D-aLS: a hybridized MOEA/D with an adaptive local search mechanism, is described. Finally, this section briefly discuss other implemented MOEA/D's, which will later form the basis for evaluating the proposed method in Section V.

A. MOEA/D framework

Our proposed method follows the general MOEA/D framework described in Algorithm 1. Further details on the various steps of the algorithm are provided below.

Step 0: Pre-processing

MOEA/D requires some pre-processing steps before initiating the main part of the algorithm. These steps are briefly summarized and discussed next.

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 III is directly inserted in a newly created route R_2.

Algorithm 1 MOEA based on Decomposition

Input:
- a multi-objective CVRPRBTW (xxx) instance;
- M : population size and number of decomposed subproblems;
- T: neighborhood size of each subproblem;
- uniform spread of weight vectors $(w^1, ..., w^M)$;
- the maximum number of generations, γ_m;
- the tournament size, τ;
- the crossover and mutation rates, c_r, m_r;

Output: the external population, EP.

Step 0-Pre-processing:

Decomposition: Decompose the original multi-objective CVRPRBTW into a set of M single-objective CVRPBRTW subproblems $\{g^1, ..., g^M\}$ having weights $(w^1, ..., w^M)$;
Neighborhoods: Compute the Euclidean distance between each pair of weight vectors. Then set a neighborhood N^i for each g^i that includes the T closest weight vectors of w^i.
Setup: Set $EP := \emptyset$; $\gamma := 0$; $IP_\gamma := \emptyset$;

Step 1-Initialization: Uniformly randomly generate and evaluate an initial internal population $IP_0 = \{X^1, \cdots, X^M\}$;

Step 2:For $i = 1, \ldots M$ **do**

Step 2.1-Genetic Operators: Generate a new solution Y^I using the genetic operators.
Step 2.2 (Optional)-Local Search: Apply a local search heuristic on Y^i to produce Z^i.
Step 2.3-Update: Update z^* and use Z^i to update IP_γ, EP and the neighborhood N^i of the T closest neighbor solutions of Z^i.

Step 3-Stopping criterion: If stopping criterion is satisfied, i.e. $\gamma = \gamma_m$, **then** stop and output EP, **otherwise** $\gamma = \gamma + 1$, go to Step 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.

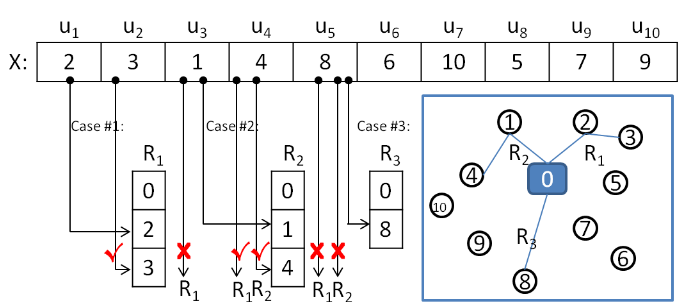


Fig. 4. Encoding Representation

Figure 4 illustrates an example of a CVRP instance with $l = 10$ customers and a candidate solution X. Before evaluation, solution X is used to create some feasible routes as explained above. In Case #1 of this example, customer $u_1 = 2$ will be served first as it appears first in X. Therefore, empty route R_1 is created and customer 2 is inserted. Then, we assume that customer 3, which follows, satisfies all constraints (denoted as bold tick in Figure 4) and it is inserted in R_1 after customer 2. Customer 1, however, does not satisfy the constraints when added after customer 3 in R_1 (denoted as bold **X** in Figure 4) and therefore, a new route R_2 is created to serve customer 1. In Case #2, it is assumed that customer 4 satisfies the constraints of both routes R_1 and R_2 and it is inserted in R_2 since customer 1 of route R_2 is closer to customer 4, compared to customer 3 of R_1. Finally in Case #3, it is assumed that customer 8 does not satisfy the constraints neither when it is inserted after customer 4 of R_2, nor when it is inserted after customer 3 of R_1. Therefore, a new route R_3 is created to serve customer 8. This continues until all customers $6, 10, 5, 7.9$ are served and every vehicle returns back to depot o. The topology on the right-hand side of the figure illustrates the solution from a CVRP point of view.

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 [21].

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:

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

where $w^i = (w^i_1, ..., w^i_m)$ represents the objective weight vector for the specific decomposed problem i with each $w^i_j \in [0, 1]$, $\hat{f}$ 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 [21], 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^k's 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 T: 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.

Step 1: Initialization

An initial population $IP_0 = \{X^1, ..., X^M\}$ is created, named Internal Population of generation $\gamma = 0$. The initialization process is random and the feasibility of the candidate solutions is maintained as discussed earlier in the pre-processing step. Each time a solution X^i is created, it is added in IP_0. The process continues until M solutions are created, one for each subproblem g^i.

Step 2.1: Genetic Operation

At each step of MOEA/D, a new solution X^i is generated for each subproblem i using the genetic operators (i.e. selection, crossover, mutation) as follows.

Selection: In this paper, a Neighborhood Tournament Selection (NTS) operator [38] is used for selecting two parent solutions, Pr^1, Pr^2 for each subproblem g^i from population IP^γ and forward them to the crossover operator for recombination. The NTS operator works as follows: the first parent solution Pr^1 is always selected to be the best known solution X^i found so far for subproblem g^i. Then, a tournament is created by uniformly randomly selecting τ neighbor solutions from neighborhood N^i, where $\tau \leq T$. The second parent Pr^2 is selected to be the neighbor solution $Pr \in N^i$ with the best $g^i(Pr|w^i, z^*)$. Then, the two parent solutions, Pr^1 and Pr^2 are forwarded to the crossover operator for recombination. The insight behind the Neighborhood Tournament Selection operator is that neighbor solutions of subproblem i in the weight space is more likely to have good information for optimizing g^i as discussed earlier in this section.

Crossover: The two parent solutions Pr^1 and Pr^2 are then

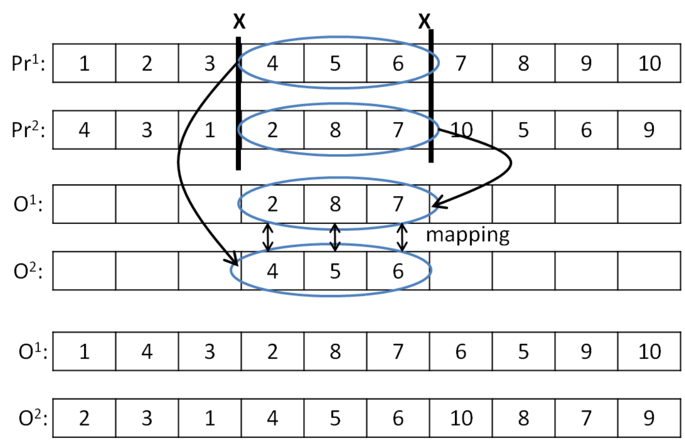


Fig. 5. The Partially Mapped Crossover (PMX)

recombined with a probability rate c_r using the well-known Partially Mapped Crossover (PMX) operator, originally proposed by Goldberg and Lingle in [39], to produce an offspring solution O. The PMX works as follows: First, two random cut points are uniformly randomly selected along Pr^1. The indexes falling between the cut points are called the mapping sections. For example, in Figure 5, let's assume that the two parent solutions Pr^1 and Pr^2 are of $l = 10$ customers length, the two cut point are denoted with bold **X**s and the mapping section is composed by 4-2,5-8,6-7. Now the mapping section of the first parent Pr^1 is copied into the second offspring O^2 and the mapping section of the second parent Pr^2 is copied into the first offspring O^1. Then offspring O^1 is filled up by copying the elements of Pr^1 and O^2 by Pr^2 . In the case that an index appears in an offspring twice, then it is replaced according to the mapping. For example, the second element of O^1 was 2 that is already copied in O^1 from Pr^2. Hence, because of the mapping 4-2 we set the second element of O^1 to be 4. The first, third, ninth and tenth elements of O^1 are taken from Pr^1 and the remaining are filled up from the mapping as well. Therefore, O^1=(1,4,3,2,8,7,6,5,9,10) and similarly O^2=(2,3,1,4,5,6,10,8,7,9). Then, $O = O^1$ if $g^i(O^1|w^i, z^*) < g^i(O^2|w^i, z^*)$ and $O = O^2$ otherwise. Finally, the offspring solution O is forwarded to the mutation operator to be slightly modified.

Mutation: A random mutation operator is utilized to modify each element of solution O with a mutation rate m_r and generate solution Y^i. Particularly, a customer u_j is mutated by uniformly randomly changing its position j in O.

Step 2.2: Local Search

In the local search step of MOEA/D in Algorithm 1, the generated solution Y^i for a given subproblem is locally improved by using a problem specific local search heuristic [40] to generate solution Z^i. Note that in the original MOEA/D scheme, the local search step was introduced as optional and has not been utilized by the authors in [21].

In this paper, the local search step is utilized in various ways as described next. The improved solution Z^i is finally used to update the populations; note that in the case when the local search step is not utilized, we set $Z^i = Y^i$. The LS heuristic used at each subproblem and at each iteration, is selected from the following pool of LS heuristics; the following three LSs were designed so that each one exhibits preference in a different objective (number of vehicles, total distance and balancing, respectively).

1) *LS favouring Number of Vehicles (LS_V):*
WHILE constraints are satisfied for each solution Y^i REPEAT at most a pre-defined number N_{LS} of iterations
   - Find route $R_{MinCust}$ with least number of customers (break ties using least amount of vehicle's used capacity).
   - Pick randomly any customer u from this route.
   - Pick randomly any other route R_V and try to move chosen customer u to this route provided:
     - no constraints are violated, and
     - number of customers in route R_V is at least a given threshold γ_V, set equal to
     $MinSize + (AvgSize - MinSize) * tol_V$,
     where $AvgSize$ and $MinSize$ are the average and minimum number of customers in constructed routes, respectively, and tol_V a given tolerance value.
2) *LS favouring Total Distance (LS_D):*
WHILE constraints are satisfied for each solution Y^i REPEAT a pre-defined number N_{LS} of iterations
   - Find randomly a customer u in the sequence of customers $u_1, ..., u_l$ and with a given probability tol_D select this customer or not. If customer u is selected then check if u is in the worst position, meaning that the sum $dist(pre(u), u) + dist(u, suc(u)$ of distances from its two neighbouring customers in the route containing u is the largest found so far.
   - Given u, find its closest neighbour v among the set of all customers and try to add u in the route R_D containing v (before of after v), provided no constraints are violated. In case some constraints are violated, the next closest neighbour of u is selected.
3) *LS favouring Balancing (LS_B):*
WHILE constraints are satisfied for each solution Y^i REPEAT a pre-defined number N_{LS} of iterations
   - Find route $R_{MaxDist}$ with maximum total distance (break ties using least amount of vehicle's used capacity).
   - Pick randomly any customer u from this route.
   - Pick randomly any other route R_B and try to move chosen customer to this route provided:
     - no constraints are violated, and
     - number of customers in route R_B is at most a given threshold γ_B, set equal to
     $MaxDist - (MaxDist - AvgDist) * tol_B$,
     where $MaxDist$ and $AvgDist$ are the maximum and average route distance in constructed routes, respectively, and tol_B a given tolerance value.

It is noted that the parameters tol_V, tol_D and tol_B employed in the LSs above might be allowed to take different values according to the stage of execution of the algorithm (number of generations) so as to facilitate search diversity and/or exhaustiveness.

Step 2.3: Update of populations

The Internal Population (IP_γ) that keeps the best solutions found so far for each subproblem, the external population (EP) that keeps the non-dominated solutions and the neighborhood T of each subproblem g^i are updated using the constucted solutions Z^i as follows:

1 The (IP_γ) update phase:
Firstly, solution Z^i replace the incumbent solution X^i for subproblem i iff it achieves a better value for the specific objective function of that subproblem; in other words, if $g^i(Z^i|w^i, z^*) < g^i(X^i|w^i, z^*)$ then $IP_\gamma \cup \{Z^i\}$ and $IP_\gamma \setminus \{X^i\}$, otherwise X^i is not replaced in IP_γ.

2 The neighborhood T update phase:
Subsequently, in an attempt to propagate good characteristics, Z^i is evaluated against the incumbent solutions X^js of the T closest neighbors of i; in other words, for all T closest neighbor solutions $X^j \in IP_\gamma$, and for $j = 1, \ldots, T$, if $g^j(Z^i|w^j, z^*) < g^j(X^j|w^j, z^*)$ then, $IP_\gamma \cup \{Z^i\}$ and $IP_\gamma \setminus \{X^j\}$, otherwise, X^j is not replaced in IP_γ.

3 The (EP) update phase:
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; in other words, if there is no solution $X^j \in EP$ such that $X^j \prec Z^i$ then $EP = EP \cup \{Z^i\}$ and for any $X^j \in EP$, if $Z^i \prec X^j$ then $EP = EP \setminus \{X^j\}$.

Step 3: Termination Criterion

At the end of each iteration if the maximum number of generations γ_m is reached, the search terminates.

B. MOEA/D-aLS: Proposed hybrid MOEA with adaptive LS

A central aspect of the proposed **MOEA/D-aLS** (Multi-Objective Evolutionary Algorithm based on Decomposition hybridized with an adaptive local search mechanism) is the way the LSs are selected for application each time a new solution is generated: this is done based on a weighted probability which depends on the objective weights each subproblem holds. As a result this probability is not static among subproblems. In the proposed aLS heuristic, a local search approach is probabilistically selected and applied to a solution based on each subproblem's weight vector that shows its objective preference and a uniformly randomly generated number. The adaptive LS strategy was designed to assign higher probability to all subproblems i that favor the number of vehicles objective (i.e., high w_i^1) to be locally optimized with LS_V, those that favor the total distance cost objective (i.e., high w_i^2) with LS_D and those that favor the balancing objective (i.e., high w_i^3), with LS_B. Note that this approach cannot be utilized by any non-decompositional MOEA.

The aLS proceeds as follows:

For each subproblem g^i with a weight vector (w_1^i, w_2^i, w_3^i) and generated solution Y^i **do**:

Uniformly randomly generate number $rand \in [0, 1]$.
If $0 <= rand <= w_1^i$ then apply LS_V on Y^i to obtain Z^i.
else if $w_1^i < rand <= (w_1^i + w_2^i)$ then apply LS_D on Y^i to obtain Z^i
else apply LS_B on Y^i to obtain Z^i.

C. Other MOEA/D variants considered

The following variants of MOEA/D have also been implemented and used as benchmarks for evaluating the performance of our proposed adaptive method in Section V. They all differ in Step 2.2 (Local Search) of the MOEA/D framework in Algorithm 1:

MOEA/D: conventional MOEA/D as proposed by Zhang and Li in [21]. No local search heuristics are applied.

MOEA/D-rLS: MOEA/D with random application of a heuristic from the given LS pool $\{LS_V, LS_D, LS_B\}$ at each step (i.e., each LS has equal probability of being employed).

MOEA/D-iLS: MOEA/D with application of individual heuristic from the given LS pool $\{LS_V, LS_D, LS_B\}$ at each step (i.e., always applying the same LS).

V. Experimental Studies

This section introduces our experimental setup by briefly explaining the Solomon's test instances and the performance metrics used in our experimental studies to evaluate the performance of the MOEA/D variants, followed by several experimental studies.

A. Experimental Setup

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 common algorithmic settings used are as follows: $c_r = 0.9, m_r = 0.01, T = 5, \tau = 30, M = 630$ and $g_m = 5000$. In addition, for the hybrid MOEA/Ds, the following parameters for the LS step were set: $N_{LS} = 50$, $tol_D = 0.5\ tol_V$, and tol_B increase from 0 to 0.8 by 0.2 every 500 generations and then remain fixed. Due to the limited space we present results on a subset of instances.

B. Performance Measures

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 (PF) 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 [41]. In this study, we use the **Coverage** C [41] and **distance for reference set** I_D [42] metrics:

$$C(A, B) = \frac{|\{x \in B | \exists y \in A : y \prec x\}|}{|B|},$$

$$I_D(A) = \frac{\sum_{y \in R}\{min_{x \in A}\{d(x, y)\}\}}{|R|}.$$

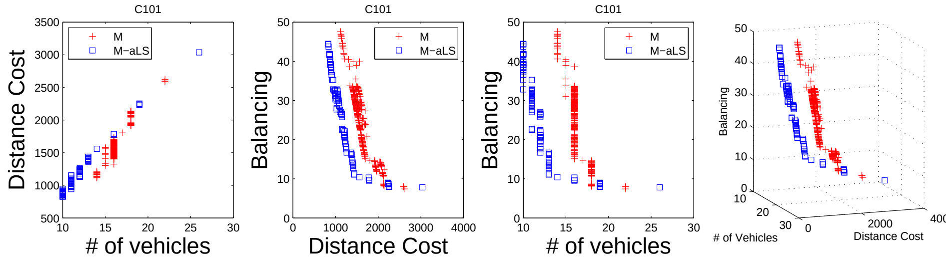
Coverage is a commonly used metric for comparing two sets of non-dominated solutions A and B. The $C(A, B)$ metric calculates the ratio of solutions in B dominated by solutions in A, divided by the total number of solutions in B. Therefore, $C(A, B) = 1$ means that all solutions in B are dominated by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

The distance I_D from reference set is defined by Czyzzak et al. in [42]. This shows the average distance from a solution in the reference set R to the closest solution in A. The smaller the value of $I_D(A)$, the closer the set A is to R. In the absence of the real reference set (i.e., Pareto Front), we calculate the average distance of each single point to the nadir point since we consider minimization objectives.

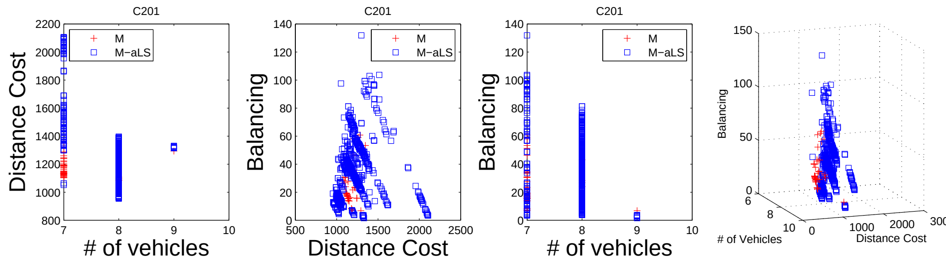
C. Experimental Results

The proposed MOEA/D-aLS (M-aLS) is evaluated with respect to the conventional MOEA/D as proposed by Zhang and Li in [21], a MOEA/D with a random local search (M-rLS) selection mechanism and a MOEA/D-iLS with individual LS heuristics (M-iLS) as described in subsection IV-C. The MOEA/D variants are compared both visually as well as in terms of the performance metrics of subsection V-B. 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.

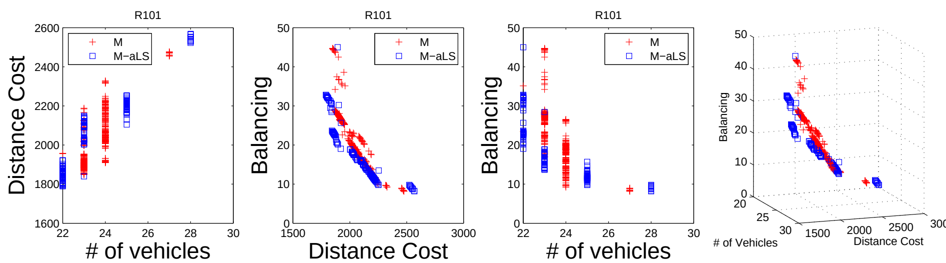
Experimental Series 1 - MOEA/D-aLS vs. MOEA/D: Figure 6 shows that the MOEA/D-aLS improves the performance



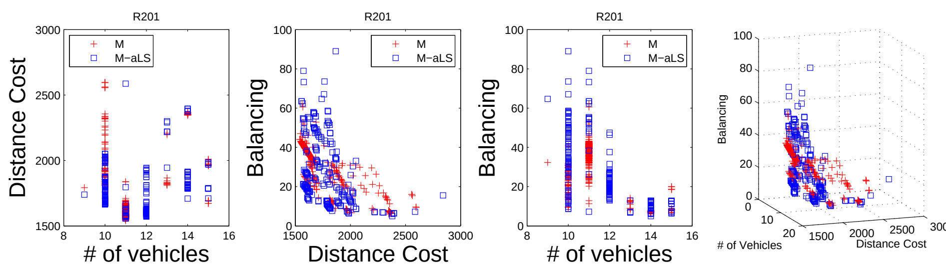
(a) C101



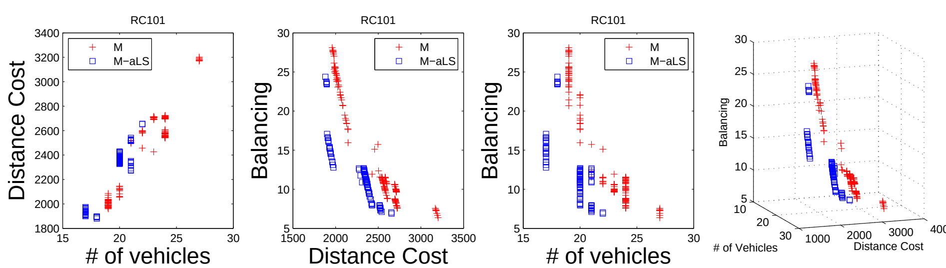
(b) C201



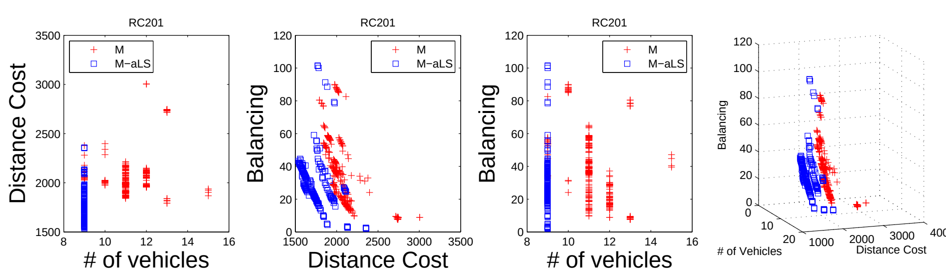
(c) R101



(d) R201



(e) RC101



(f) RC201

Fig. 6. Comparison between the proposed MOEA/D with adaptive Local Search (MOEA/D-aLS) and the conventional MOEA/D.

TABLE I
PROPOSED MOEA/D WITH ADAPTIVE LS (M-ALS) COMPARED TO CONVENTIONAL MOEA/D (M) IN TERMS OF C AND I_D METRICS.

Test Inst.	C(M,M-aLS)	C(M-aLS,M)	I_D(M)	I_D(M-aLS)
C101:	0.0352	0.9254	50.98	45
C201:	0.01	0.2	33.6	13.7
R101:	0.01	0.6	14.8	2.6
R201:	0.05	0.3	17.6	2.0
RC101:	0	0.1	55.6	42.8
RC201:	0	0.82	33.5	18.6

of the conventional MOEA/D in terms of both convergence and diversity. In particular, the MOEA/D-aLS obtained a PF that dominates the non-dominated solutions obtained by the conventional MOEA/Ds providing a better approximation towards the optimal point. This is more evident in Table I that summarizes the statistical performance of MOEA/D-aLS and MOEA/D 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 MOEA/D-aLS dominate most (on average 75%) of the non-dominated solutions obtained by MOEA/D and performs better in terms of I_D. An example of the best solution in terms of number of vehicles obtained by MOEA/D-aLS is illustrated in Figure 7.

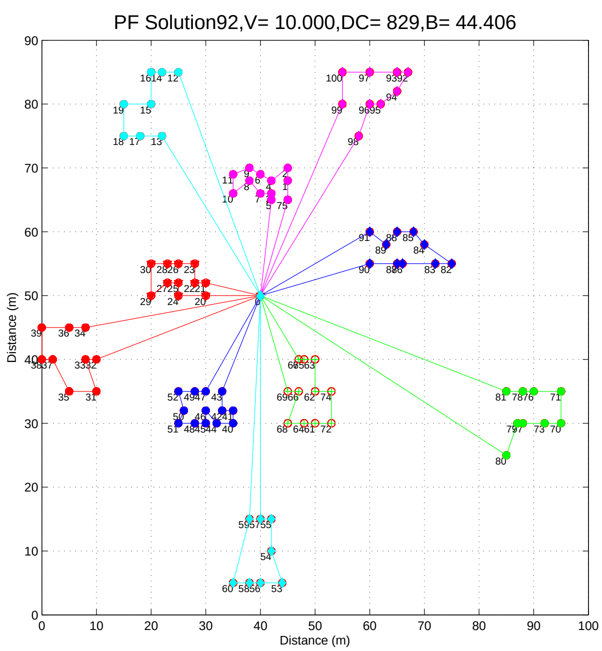


Fig. 7. Routes corresponding to best solution for C101 using MOEA/D-aLS

Experimental Series 2 - MOEA/D-aLS vs. MOEA/D-rLS: Figure 8 shows that the performance of MOEA/D-aLS is better than MOEA/D-rLS in terms of both convergence and diversity, as well. In particular, the MOEA/D-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. This is more evident in Table II that summarizes the statistical performance of MOEA/D-aLS and MOEA/D 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 MOEA/D-aLS dominate most of the non-dominated solutions obtained by MOEA/D-rLS and performs better in terms of I_D.

TABLE II
PROPOSED MOEA/D WITH ADAPTIVE LS (M-ALS) COMPARED TO MOEA/D WITH RANDOM LS (M-RLS) BASED ON C AND I_D METRICS.

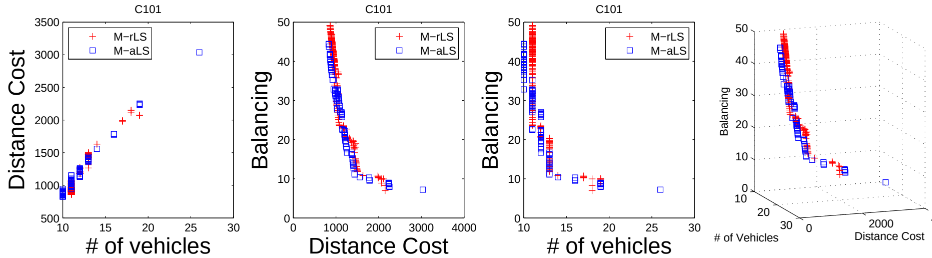
Test Inst.	C(M-rLS, M-aLS)	C(M-aLS, M-rLS)	I_D(M-rLS)	I_D(M-aLS)
C101:	0.51	0.57	45	45
C201:	0.45	0.66	14.87	13.7
R101:	0.2	0.6	31.7	2.6
R201:	0.2	0.6	2.2	2.0
RC101:	0.4	0.5	26.6	42.8
RC201:	0	0.1	37.3	18.6

TABLE III
MOEA/D-ALS COMPARED WITH CONVENTIONAL MOEA/D AND MOEA/D-RLS IN TERMS OF BEST SOLUTIONS OBJECTIVE-WISE (V = NO. OF VEHICLES, D = DISTANCE COST, B = BALANCING).

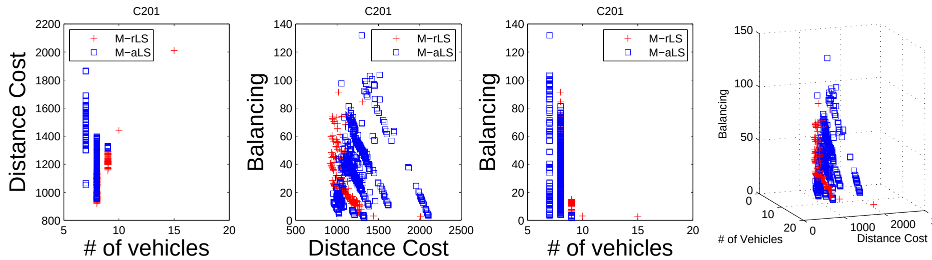
Test Inst.	MOEA/D			MOEA/D-rLS			MOEA/D-aLS		
	V	D	B	V	D	B	V	D	B
C101:	14	1116	7.4	11	860.2	**6.9**	**10**	**828.9**	7.2
C201:	**7**	1085	6.8	8	**917**	2.5	**7**	954	**1.5**
R101:	**22**	1848	8.2	**22**	1874	8.4	**22**	**1789**	**8.0**
R201:	**9**	1540	6.0	**9**	1555	5.0	**9**	**1538**	**5.0**
RC101:	19	1960	**6.4**	18	1899	8.0	**17**	**1879**	**6.4**
RC201:	**9**	1792	7.6	**9**	1726	6.1	**9**	**1535**	**2.0**

Finally, Table III shows the best solutions obtained for each objective function by each MOEA/D variant for all the examined Solomon test instances. For any such test instance, the best solution found for each objective is shown in bold. The MOEA/D-aLS in almost all test instances and for each objective function obtained better or equal near-optimal solutions than its competitors. In particular, MOEA/D-aLS found the best distance cost D in five out of six cases and came second in C201 behind MOEA/D-aLS, the least number of vehicles V in two out of six test instances and the same one as other method(s) in the remaining cases and the best balancing B in four out of six test instances (for the remaining two test instances MOEA/D-aLS obtained the same balancing as MOEA/D-rLS in R201 and worse balancing in C101.) We think that this is mainly due to the fact that balancing is an objective function that can be optimized with less effort than the other two objectives and sometimes this is in favour of the more stochastic methods.

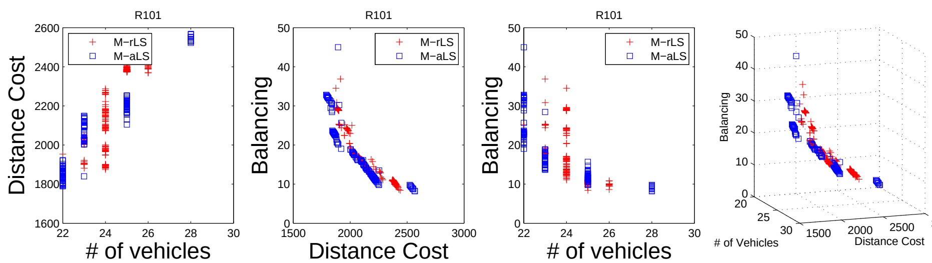
Experimental Series 3- MOEA/D-aLS vs. MOEA/D-iLS: In this experimental series, we compared the best solutions for the considered Solomon test instances obtained by MOEA/D-aLS with respect to solutions from MOEA/D hybridized with each single heuristic individually. We denote by MOEA/D-iLS$_V$, MOEA/D-iLS$_D$ and MOEA/D-iLS$_B$ the MOEA/D hybridized with the LS that favours the number of vehicles objective function (LS_V), the distance cost objective function (LS_D), and the balancing objective function (LS_B), respectively. For any test instance, the best solution found for each objective



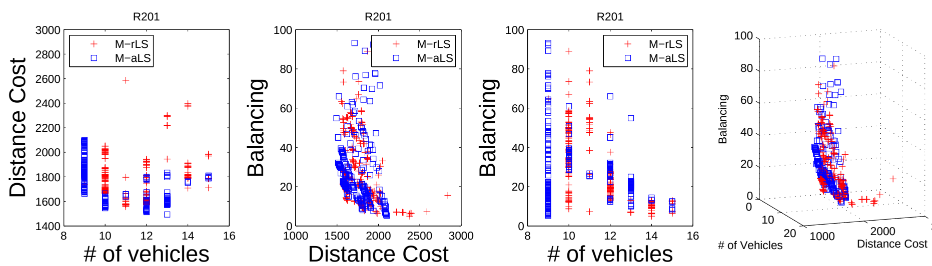
(a) C101



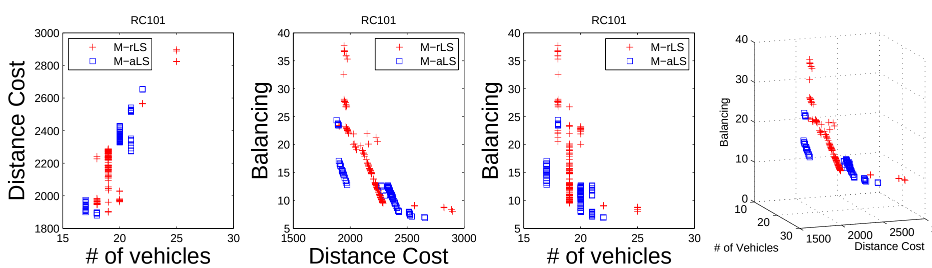
(b) C201



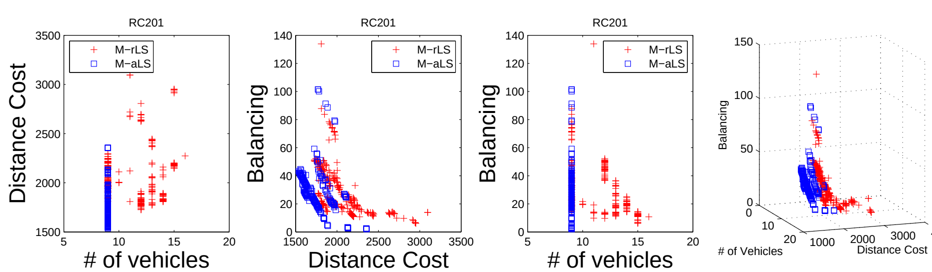
(c) R101



(d) R201



(e) RC101



(f) RC201

Fig. 8. Comparison between the proposed MOEA/D with the adaptive LS (MOEA/D-aLS) and MOEA/D with random selection of LS (MOEA/D-rLS).

TABLE IV
M-aLS IS COMPARED WITH MOEA/D WITH INDIVIDUAL HEURISTICS IN TERMS OF BEST SOLUTIONS FOUND FOR EACH OBJECTIVE (V = NO. OF VEHICLES, D = DISTANCE COST, B = BALANCING).

Test Inst.	MOEAD-iLS$_V$			MOEAD-iLS$_D$			MOEAD-iLS$_B$			MOEAD-aLS		
	V	D	B	V	D	B	V	D	B	V	D	B
C101:	11	894	9.4	13	1110	**6.6**	13	977	9.0	**10**	**828.9**	7.2
C201:	**5**	**856**	4.4	8	1064	5.2	11	1125	6.3	7	954	**1.5**
R101:	**21**	1814	**7.8**	22	1885	9.0	23	1836	11.9	22	**1789**	8.0
R201:	**8**	1512	5.0	9	**1421**	**2.9**	10	1497	3.4	9	1538	5.0
RC101:	18	**1873**	9.1	19	1941	8.0	18	1906	7.3	**17**	1879	**6.4**
RC201:	9	1729	6.4	**8**	1699	3.6	10	1794	7.2	9	**1535**	**2.0**

is shown in bold. Among the three individualistic hybrid evolutionary algorithms, MOEA/D-iLS$_V$ seems to be performing better (in relation to the values of its associated objective), while MOEA/D-iLS$_B$ fails to produce any solutions in bold. The overall picture of Table IV though, is that, in general, the individualistic LSs when applied with MOEA/D do not perform as well as our proposed method: combining the LSs and adaptively hybridizing the MOEA/D, can provide a powerful tool with near-optimal performance in many cases. Therefore, it is correct to argue that our suggested approach, i.e., adaptively selecting problem-specific LSs based on the weight coefficient of each subproblem, is efficient and effective compared to either using a single heuristic or randomly selecting a LS from a pool of LSs.

VI. 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 (LS). 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 traditional MOEA/D in all cases and in most cases of the MOEA/D hybridized with randomly selected LS; in addition, in many cases, it is able to improve on the best component-wise solutions obtained by variants of MOEA/D hybridized with the same individual LS method.

Directions for future work include the investigation of the possibility of improving various components of the Evolutionary Algorithm as well as of incorporating learning for the selection of a local search approach to further improve the performance of the MOEA/D.

References

[1] A. Goel and V. Gruhn, "A general vehicle routing problem," *European Journal of Operational Research*, vol. 191, no. 3, pp. 650 – 660, 2008.

[2] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, 2009.

[3] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, pp. 1985–2002, 2004.

[4] T.-R. Lee and J.-H. Ueng, "A study of vehicle routing problems with load-balancing," *International Journal of Physical Distribution & Logistics Management*, vol. 29, no. 10, pp. 646–657, 1999.

[5] M. M. Solomon, "Algorithms for the vehicle routing problem with time windows," *Transportation Science*, vol. 29, no. 2, pp. 156–166, 1995.

[6] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[7] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Special issue on genetic algorithms in Computers and Operations Research*, vol. 22, no. 1, pp. 5–13, 1995.

[8] J. Chen and S. Chen, "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] B. Ombuki, B. J.Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, pp. 17–30, February 2006.

[10] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley & Sons, 2002.

[11] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation, Elsevier*, vol. 1, no. 1, pp. 32–49, 2011.

[12] A. Konstantinidis, H. Haralambous, A. Agapitos, and H. Papadopoulos, "A gp-moea/d approach for modelling total electron content over cyprus," *CoRR*, vol. abs/1111.5720, 2011.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[14] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.

[15] N. Jozefowiez, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems." *European Journal of Operational Research*, pp. 293–309, 2008.

[16] ——, "An evolutionary algorithm for the vehicle routing problem with route balancing." *European Journal of Operational Research*, pp. 761–769, 2009.

[17] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *Comput. Optim. Appl.*, vol. 34, pp. 115–151, May 2006.

[18] M. J. Geiger, "A computational study of genetic crossover operators for multi-objective vehicle routing problem with soft time windows," *CoRR*, 2008.

[19] K. Tan, C. Cheong, and C. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, no. 2, pp. 813–839, 2007.

[20] K. Ghoseiri and S. F. Ghannadpour, "Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm." *Appl. Soft Comput.*, pp. 1096–1107, 2010.

[21] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[22] A. Konstantinidis, S. Pericleous, and C. Charalambous, "MOEA/D for a tri-objective vehicle routing problem," in *Artificial Intelligence Applications and Innovations*, ser. IFIP Advances in Info. and Communication Technology. Springer Boston, 2013, vol. 412, pp. 131–140.

[23] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing

problems." *European Journal of Operational Research*, pp. 855–885, 2006.

[24] D. Chitty and M. Hernandez, "A hybrid ant colony optimisation technique for dynamic vehicle routing," in *Genetic and Evolutionary Computation GECCO 2004*, ser. Lecture Notes in Computer Science, K. Deb, Ed. Springer Berlin / Heidelberg, 2004, vol. 3102, pp. 48–59.

[25] S.-C. Hong and Y.-B. Park, "A heuristic for bi-objective vehicle routing with time window constraints," *International Journal of Production Economics*, vol. 62, no. 3, pp. 249 – 258, 1999.

[26] B. Barn and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows." in *Applied Informatics'03*, 2003, pp. 97–102.

[27] K. G. Zografos and K. N. Androutsopoulos, "A heuristic algorithm for solving hazardous materials distribution problems," *European Journal Of Operational Research*, vol. 152, no. 2, pp. 507–519, 2004. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0377221703000419

[28] P. Chand, B. S. P. Mishra, and S. Dehuri, "A multi objective genetic algorithm for solving vehicle routing problem," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 503–506, July-December 2010.

[29] M. N. Kritikos and G. Ioannou, "The balanced cargo vehicle routing problem with time windows," *International Journal of Production Economics*, vol. 123, no. 1, pp. 42–51, 2010.

[30] C. Hjorring, *The Vehicle Routing Problem and Local Search Metaheuristics*. Chapter 2, PhD thesis,Department of Engineering Science, The University of Auckland, 1995.

[31] E. Angelelli and M. Speranza, "The periodic vehicle routing problem with intermediate facilities," *European Journal of Operational Research, Elsevier*, vol. 137, pp. 233–247, 2002.

[32] S. Baptista, R. Oliveira, and E. Zquete, "A period vehicle routing case study," *European Journal of Operational Research, Elsevier*, vol. 139, pp. 220–229, 2002.

[33] G. Righini, "Approximation algorithms for the vehicle routing problem with pick-up and delivery," *Note del Polo - Ricerca 33, Polo Didattico e di Ricerca di Crema, Universita degli Studi di Milano, Luglio*, 2000.

[34] A. Wade and S. Salhi, *An ant system algorithm for the mixed vehicle routing problem with backhauls*. Norwell, MA, USA: Kluwer Academic Publishers, 2004, pp. 699–719.

[35] L. C. Yeun, W. R. Ismail, K. Omar, and M. Zirour, "Vehicle routing problem: models and solutions," *Journal of Quality Measurement and Analysis*, vol. 4, no. 1, pp. 205–218, 2008.

[36] T. Murata and R. Itai, "Multi-objective vehicle routing problems using twofold emo algorithms to enhance solution similarity on non-dominated solutions," in *In Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 885–896.

[37] A. Garcia-Najera and J. A. Bullinaria, "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 38, pp. 287–300, January 2011.

[38] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," *New Network Paradigms, Elsevier Computer Networks*, vol. 54, pp. 960–976, 2010.

[39] D. E. Goldberg and L. R, "Alleles, loci, and the traveling salesman problem," in *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, J. J. Grefenstette, Ed. Lawrence Erlbaum Associates, Publishers, 1985.

[40] A. Konstantinidis and K. Yang, "Multi-objective energy-efficient dense deployment in wireless sensor networks using a hybrid problem-specific MOEA/D," *Applied Soft Computing*, vol. 11, no. 6, pp. 4117–4134, 2011.

[41] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach." *IEEE Trans. Evolutionary Computation*, pp. 257–271, 1999.

[42] P. Czyzak and A. Jaszkiewicz, "Pareto simulated annealing - a meta-heuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.