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*—The Capacitated Vehicle Routing Problem with Balanced Routes and Time Windows (CVRPBRTW) aims at optimizing the total distance cost, the number of vehicles used, and the route balancing subject to time windows and other constraints. Due to the multiple and often conflicting objectives, the proposed problem is formulated and tackled in a Multi-Objective Optimization manner using a Hybrid Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) hybridized with a pool of local search heuristics. 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 in transportation and supply-chain management (e.g., in parcel delivery services, school bus routing, airline schedules and distribution planning for wholesale retailers.)

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 so as to increase the fairness of the produced solution; as a motivating example in [4], a delivery company in Taiwan is considered in which any discrepancies in schedules might lead to drivers' dissatisfaction and as a possible result, reduced work efficiency. The *CVRP with time windows (**CVRPTW**)* [5] also includes an additional constraint which aims to improve customer satisfaction with regard to delivery times: each customer should be served within a specific time window.

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:

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

where Ω is the decision space and $x \in \Omega$ is a decision vector. $F(x)$ consists of m objective functions $f_i : \Omega \rightarrow \Re, i = 1, \dots, m$, 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 $u = (u_1, \dots, u_m)^T$ is said to dominate another vector $v = (v_1, \dots, v_m)^T$, denoted as $u \prec v$, iff $\forall i \in \{1, \dots, m\}$, $u_i \leq v_i$ 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 Local Search (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 tackled several variations of the CVRPTW by optimizing pairs of the different objectives. Over the past decade numerous variants of the investigated problem were 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 remainder 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 test our proposition on all 56 Solomon's benchmark problem instances [5] against 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 heuristic with an objective function and adopted an association between objectives and LS heuristics.

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 displays preference towards a different objective. We have extended our experimental studies in two directions. Firstly, in order to improve the accuracy of our findings, we evaluated the proposed MOEA/D-aLS approach for all 56 Solomon test instances with respect to both the conventional MOEA/D and the MOEA/D-rLS and secondly we justified the use of the main parameters settings via thorough parameter control experiments.

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 of each subproblem.
- 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).

The rest of the paper is organized 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. The general framework of the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [21] is introduced in Section IV. In Section V, an MOEA/D approach, combined with an adaptive strategy of applying local search heuristics is presented for tackling the proposed problem. In Section VI, the performance of the proposed method is evaluated on the well studied Solomon's benchmark problem instances [5] and compared against other MOEA/D variants. Finally, Section VII 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 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] 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] 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] 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] 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] proposed a similar MOP but instead of aggregating the objective functions using weights, the authors 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 originates 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] and [35].

B. CVRP & Multi-Objective Optimization (MOO)

It is important to note that all previously mentioned 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 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

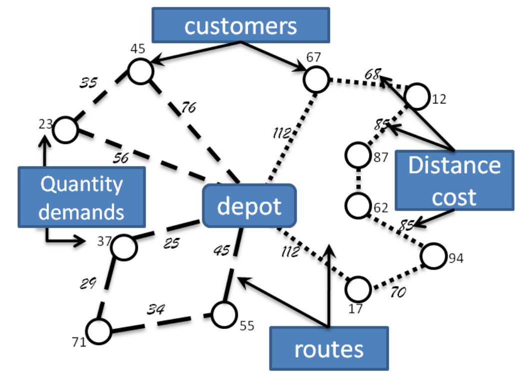


Fig. 1. The Capacitated Vehicle Routing Problem (CVRP)

Itai [36] 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 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] 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] 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] 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.

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 $u_i \in V$ must be served a quantity q_i (also known as customer's demand) of goods that requires a pre-defined service time t_i^s. We denote by t_i^a the arrival time at customer u_i, 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

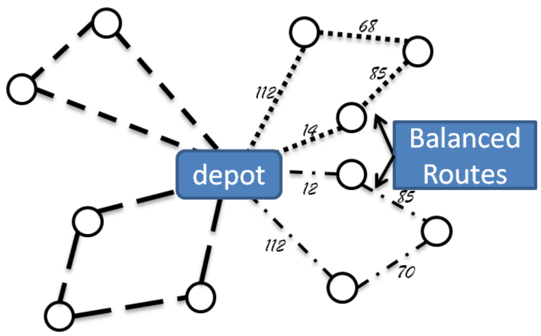


Fig. 2. The CVRP with Balanced Routes (CVRPBR)

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_i 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_i are indicated with empty circles; the required quantity demands q_i are shown just next 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_i should arrive within specific time windows $[e_i, e'_i]$. In particular, the depot is also associated with a time window $[e_0, e'_0]$, which constrains the total travel time of a vehicle from departure to arrival. Note that in the problem variant investigated here, if a vehicle arrives at a customer u_i before the earliest allowed arrival time e_i, it is allowed to wait until that time is reached, resulting in additional route travelled time. Time windows are treated as a hard constraint in the sense that if the vehicle arrives at a customer u_i after the latest allowed arrival time e'_i, 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

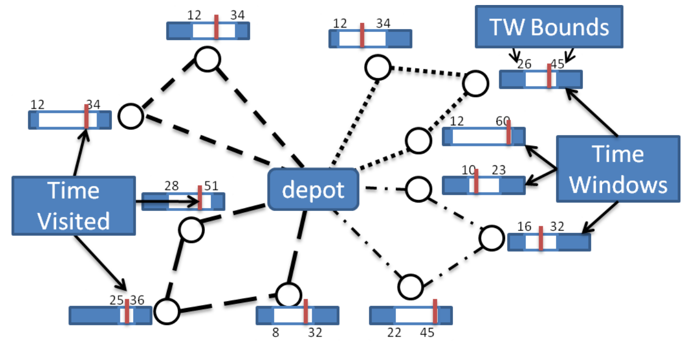


Fig. 3. The CVRP with Time Windows (CVRPTW)

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 vertices composed of a depot $o = u_0$ 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 $u_i, u_j \in V$ associated with their Euclidean distance $dist(u_i, u_j)$.
- $[e_i, e'_i]$ = time window of customer $u_i, \forall i \in \{0, ..., l\}$.
- q_i = quantity demand of customer u_i, $\forall i \in \{1, ..., l\}$; $q_0 = 0$.
- t^s_i = service time of customer $u_i, \forall i \in \{0, ..., l\}$; $t^s_0 = 0$.
- 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_i = the vehicle arrival time at vertex u_i, $\forall i \in \{1, ..., l\}$; taking $t^a_0 = 0$ and assuming 1-1 correspondence between distance and time units, this can be calculated by the function
$$\max\{e_i,\ t^a_{i_p} + t^s_{i_p} + dist(u_{i_p}, u_i)\},$$
where $u_{i_p} = pre(u_i)$ denotes the vertex preceding customer u_i in its route.

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

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

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

$$\sum_{\forall u_i \in R^m} q_i \le c, \quad \forall m \in \{1, ...k\} \tag{6}$$

$$e_i \le t_i^a \le e'_i, \quad \forall i \in \{1, ...l\} \tag{7}$$

$$\{u_i\} \cap \bigcup_{m=1,...,k} R^m = \{u_i\}, \quad \forall i \in \{1, ...l\} \tag{8}$$

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

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 capacity of the vehicle, whereas constraints (7) require that the arrival time at all customers is within their corresponding time window. 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. The combination of constraints (8) and (9) guarantee that all customers are served exactly once.

IV. Background: The MOEA/D framework

In this section, the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [21] general framework is initially introduced in Algorithm 1. Then the MOEA/D's major steps are described and the conventional operators proposed in [21] are introduced.

MOEA/D requires some pre-processing steps before initiating the main part of the algorithm. These steps are briefly summarized and discussed next. The encoding representation is often problem specific and will be discussed in the following section.

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.

Main part of MOEA/D: In the main part of the MOEA/D

Algorithm 1 MOEA based on Decomposition

Input:
- a MOP (e.g., CVRPRBTW in Section III);
- 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.

---

framework, an initial population of size M is uniformly randomly generated. Then the genetic operators are iteratively utilized M times for generating the new and evolved population. In [21], for generating a new solution Y a tournament selection is used for selecting two parent solutions and then a two-point crossover generated an offspring solution that is then modified using a random mutation operator. The newly generated solution Y can be further locally optimized by using any local search heuristic to generated solution Z. Note that the latter step is optional in the original MOEA/D framework.

Update of populations and Termination
Finally, 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 constructed 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 $j = 1, \dots, T$, 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 $EP = EP \setminus \{X^j\}$.

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

V. The proposed Hybrid MOEA/D-aLS

In this section, the proposed algorithm, namely MOEA/D-aLS - a hybridized MOEA/D with an adaptive local search mechanism, is described. Our proposed method follows the general MOEA/D framework described in Algorithm 1 that is carefully customized and extended for efficiently tackling the proposed MOP.

A. Main steps of proposed MOEA/D-aLS approach

Further details on the various steps of the proposed algorithm and the major differences with the general algorithm proposed in [21] are provided below.

Step 0: Pre-processing

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. 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.

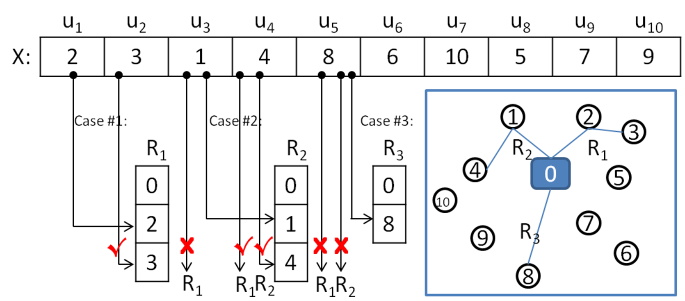


Fig. 4. Encoding Representation

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 customers $6, 10, 5, 7$ and 9 are all 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 and Neighborhoods T: In this article, the Tchebycheff approach is employed and the neighborhoods are calculated as originally proposed in [21] and explained in Section IV.

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

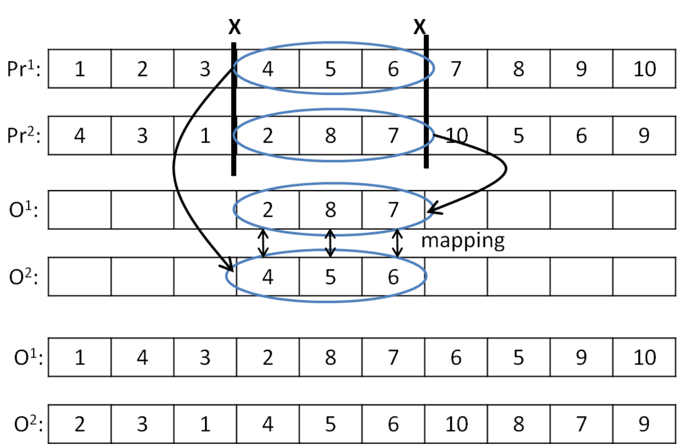


Fig. 5. The Partially Mapped Crossover (PMX)

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 swap mutation operator is utilized to modify each element of solution O with a mutation rate m_r and generate solution Y^i as in [1]. Particularly, a customer u_j is mutated by swapping its position with another customer u_z at position z in O. If either u_j or u_z cannot be feasibly swapped the offspring O is not changed.

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 LS heuristics 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 vehicles' 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 (first try inserting u before v and if any constraints are violated then try 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 vehicles' 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 LS heuristics 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.

Steps 2.3 and 3: Update of populations and Termination The Internal Population (IP_γ), the external population (EP) and the neighborhood T of each subproblem g^i are updated as explained in Section IV. The proposed MOEA/D-aLS terminates after a maximum number of generations γ_m.

B. Adaptive strategy for LS heuristics

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 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. 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.

VI. 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 a series of experiments.

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 R1, C1 and RC1 have a short scheduling horizon and allow only a few customers per route (approximately 5 to 10). In contrast, the sets R2, C2 and RC2 have a long scheduling horizon permitting many customers (more than 30) to be serviced by the same vehicle. In this paper, we examined the entire Solomon dataset that is composed of 56 test instances.

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 Local Search 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. The values of the algorithmic parameters were selected after running several control experiments summarized in Table I for test instance C101. The results are introduced next in Experimental Series 1 of Subsection VI-C.

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 from 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 the 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

In this section, we initially provide some control experiments to discuss the sensitivity of the proposed approach with respect to its major algorithmic parameters. Then the proposed MOEA/D-aLS (M-aLS) is evaluated with respect to the conventional MOEA/D as proposed by Zhang and Li in [21] and a MOEA/D with a random local search (M-rLS) selection mechanism. The MOEA/D variants are compared both visually, as well as, in terms of the performance metrics of subsection VI-B. Note that in all experimental studies we used the same number of function evaluations for fairness.

Experimental Series 1 - Control Experiments:

In this experimental series, we have examined the sensitivity of the major algorithmic parameters with respect to the convergence of each objective function. In each control experiment, summarized in Table I, we vary one algorithmic parameter and we keep the rest of the parameters fixed. For the remaining experimental series, we have used the algorithmic settings of CEx1-Instance 6 (Table I), as already stated in subsection VI-A and explained next.

In Figure 6, we examined the best values of each objective function for each instance of CEx1 that varies the γ_m parameter. The results show that all objective functions sharply converge towards zero for the first 500 generations. The convergence becomes smoother for the next 3000 generations, while it stops for both distance and # of vehicles objective functions after around 3500 generations. The balancing objective keeps converging until the maximum number of 5000 generations, showing that a large number of generations would be preferable. This is due to the fact that the evolutionary algorithm requires a large number of function evaluations in order to find the near-optimum solutions of the proposed MOP since the objective space is large and multi-dimensional.

Figure 7 illustrates the experimental results of control experiment CEx2 of Table I in which the population size is varied keeping all other parameter settings fixed. In this control experiment, all objective functions converge while the population size increases. This behavior changes when the population size increase more than around 630 population size since the objective functions start increasing. This is due to the fact that a very large population size may force the evolutionary algorithm to reach high quality solutions fast and then get trapped into local optima.

Figure 8 summarizes the results of CEx3 that investigates the behavior of the tournament size parameter. The results show that all three objective functions have a preference of a relatively high tournament size (around $\tau = 30$) since this provides the opportunity to the genetic operators to recombine solutions of different neighborhoods and therefore genotypes with high dissimilarity. This results in better exploration in the objective space.

In Figure 9, we examine the sensitivity of the proposed approach with respect to the crossover rate. The results show that the crossover rate influences differently the three objective functions. On the one hand, the # of vehicles and the distance objectives decrease as the crossover rate increase and they are slightly negatively affected for very large values. On the other hand, the balancing objective functions prefers a large crossover rate, since it performs poorly for small values. This is due to the fact that the balancing objective requires more fine-grained global search compared to the two other objectives.

Furthermore, the mutation rate that is varied in control experiment CEx5 of Figure 10 shows that all three objective functions prefer small variations on the evolved solutions and less randomness. This is due to the fact that small mutation rates (around 0.05) perform better compared to large values since on the one hand they vary the existing solutions for

TABLE I
CONTROL EXPERIMENTS

CEx	Instance	γ_m	M	τ	c_r	m_r	T
1	1:	100	630	30	0.9	0.05	5
	2:	500	630	30	0.9	0.05	5
	3:	1000	630	30	0.9	0.05	5
	4:	2000	630	30	0.9	0.05	5
	5:	3500	630	30	0.9	0.05	5
	6:	5000	630	30	0.9	0.05	5
2	7:	500	153	30	0.9	0.05	5
	8:	500	311	30	0.9	0.05	5
	9:	500	630	30	0.9	0.05	5
	10:	500	1275	30	0.9	0.05	5
3	11:	500	630	1	0.9	0.05	5
	12:	500	630	15	0.9	0.05	5
	13:	500	630	30	0.9	0.05	5
	14:	500	630	60	0.9	0.05	5
4	15:	500	630	30	0.1	0.05	5
	16:	500	630	30	0.3	0.05	5
	17:	500	630	30	0.5	0.05	5
	18:	500	630	30	0.7	0.05	5
	19:	500	630	30	0.9	0.05	5
5	20:	500	630	30	0.9	0.01	5
	21:	500	630	30	0.9	0.05	5
	22:	500	630	30	0.9	0.1	5
	23:	500	630	30	0.9	0.2	5
6	24:	500	630	30	0.9	0.05	1
	25:	500	630	30	0.9	0.05	5
	26:	500	630	30	0.9	0.05	10
	27:	500	630	30	0.9	0.05	20
	28:	500	630	30	0.9	0.05	30

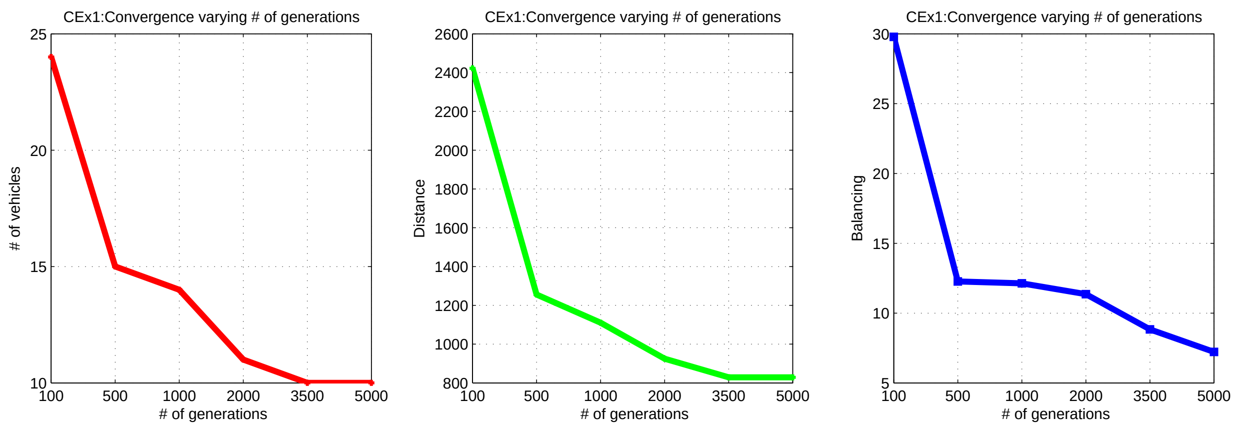


Fig. 6. Control Experiment 1: Convergence of objective functions while varying the # of generations parameter γ_m.

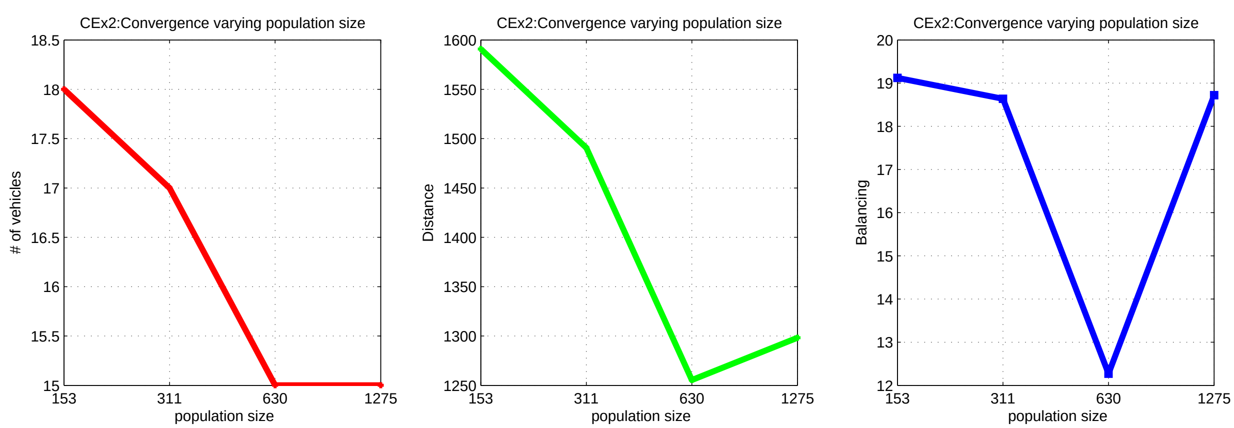


Fig. 7. Control Experiment 2: Convergence of objective functions while varying the population size M.

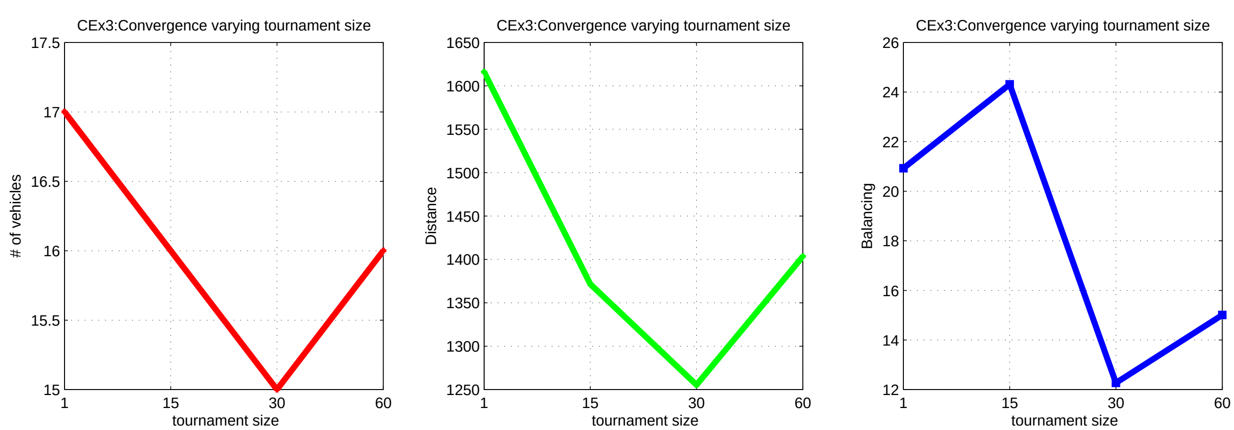


Fig. 8. Control Experiment 3: Convergence of objective functions while varying the tournament size τ.

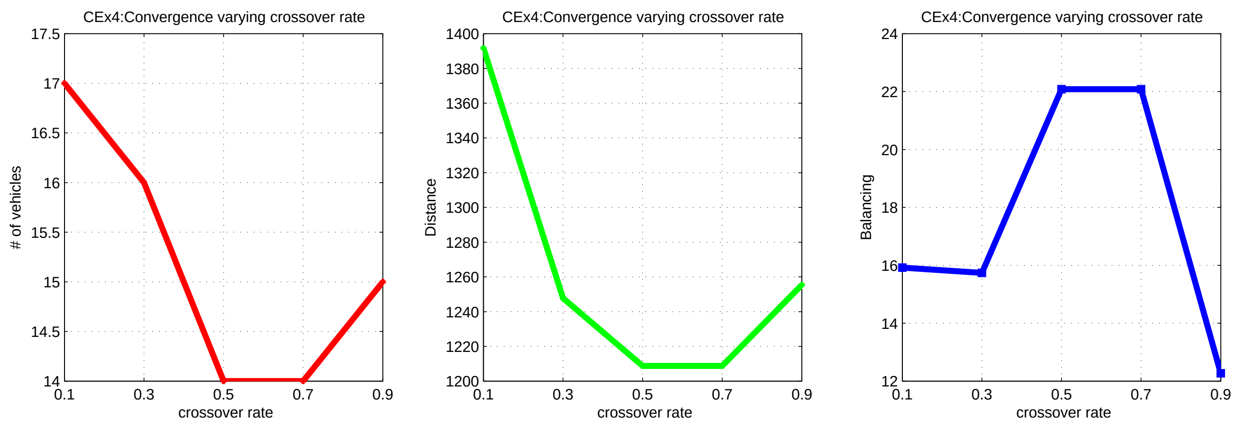


Fig. 9. Control Experiment 4: Convergence of objective functions while varying the crossover rate c_r.

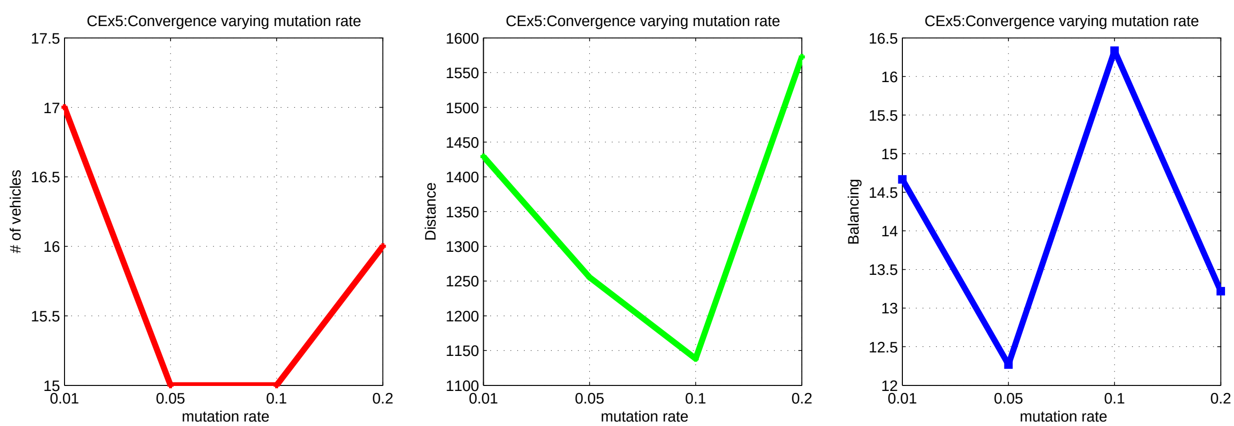


Fig. 10. Control Experiment 5: Convergence of objective functions while varying the mutation rate m_r.

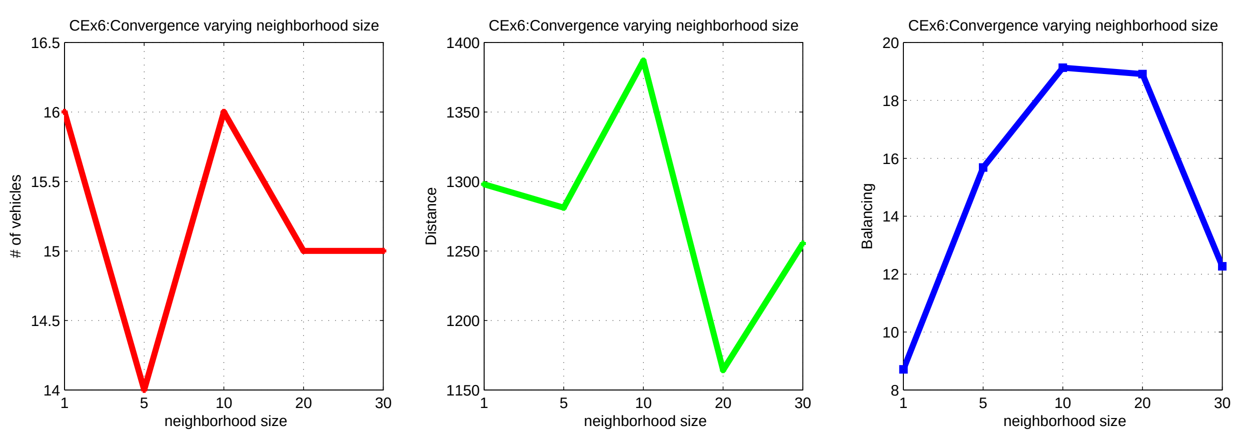


Fig. 11. Control Experiment 6: Convergence of objective functions while varying the neighborhood size T.

escaping from local optima, but on the other hand, they do not modify the existing solutions much for deteriorating their fitness.

Finally, in Figure 11, control experiment CEx6 shows the sensitivity of the proposed approach with respect to the neighborhood size. The results show that this parameter influences differently the three objective functions. That is, the # of vehicles and balancing objective prefer a small neighborhood size, where the distance objective prefers a relatively large neighborhood size. Having said that, note that the largest neighborhood value examined in CEx6 provide adequate results for all three objective functions. We believe that this behavior is due to the fact that the neighborhood size directly affects both the convergence and the diversity of the MOEA/D, since the largest the T is, the more subproblems will have the same high quality solution in the internal population (recall the neighborhood size calculation in Section IV for more details). In complex and multi-dimensional optimization problems this may provide the opportunity for the genetic operators and the local search heuristics to search high quality solutions more frequently and therefore increase the possibility of finding better and close to optimum solutions.

Experimental Series 2 - MOEA/D-aLS vs. MOEA/D:

Figure 12 shows that the MOEA/D-aLS improves the performance of the conventional MOEA/D in terms of both convergence and diversity on a subset of the Solomon test instances (C101, C201, R101, R201, RC101 and RC201). 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 II that summarizes the statistical performance of MOEA/D-aLS and MOEA/D in terms of the Coverage (C) and the Distance from the reference set (I_D) on the entire Solomon dataset test instances. The results show that the non-dominated solutions obtained by the MOEA/D-aLS dominate the non-dominated solutions obtained by MOEA/D (C-metric) and performs better in terms of I_D in 55 and 54 out of total 56 test instances, respectively. That is, the proposed MOEA/D approach with the adaptive local search heuristic outperforms the conventional MOEA/D in more than 98% of the test instances in both quality and diversity. In particular, the solutions of the PF obtained by MOEA/D-aLS for classes C and RC dominate more than 87% (average of C(M-aLS,M) within each class of instances) of those obtained by the conventional MOEA/D. When the same figure is calculated for the class R, the average drops to around 80%, which is still considered to be substantial.

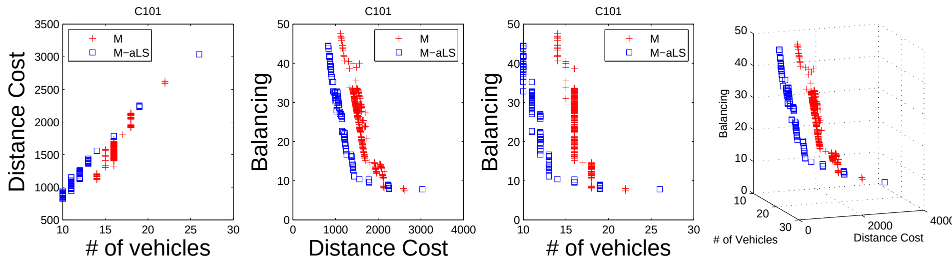
TABLE II
PROPOSED MOEA/D WITH ADAPTIVE LS (M-ALS) COMPARED TO CONVENTIONAL MOEA/D (M) IN TERMS OF C AND I_D METRICS.

Test Inst.	C(M-aLS,M)	C(M,M-aLS)	I_D(M-aLS)	I_D(M)
C101:	0.9254	0.0352	45.00	50.98
C102:	1.0000	0.0000	16.1869	70.5637
C103:	1.0000	0.0000	16.2652	46.0676
C104:	1.0000	0.0000	22.1395	33.2264
C105:	0.6747	0.0000	4.5422	153.5722
C106:	1.0000	0.0000	8.5140	87.5886
C107:	0.9806	0.0000	15.8914	76.7894
C108:	1.0000	0.0000	9.8942	57.9962
C109:	0.9677	0.0000	6.0180	48.6797
C201:	0.200	0.010	13.70	33.60
C202:	0.9472	0.0474	10.5454	45.0907
C203:	1.0000	0.0000	42.8598	45.6517
C204:	1.0000	0.0000	18.5364	28.3809
C205:	0.4533	0.4603	3.0346	32.0000
C206:	0.8806	0.0000	7.1499	28.7793
C207:	0.7917	0.0169	12.3011	19.6839
C208:	0.9992	0.0000	3.7775	39.9049
R101:	0.600	0.01	2.60	14.80
R102:	1.0000	0.0000	9.7864	29.8021
R103:	1.0000	0.0000	8.6037	26.4186
R104:	0.8936	0.0000	8.7431	112.5842
R105:	0.9172	0.0000	8.9812	28.5395
R106:	0.3846	0.2782	10.9755	18.9079
R107:	0.9964	0.0000	5.6054	40.5698
R108:	0.8312	0.0000	4.0837	36.3509
R109:	0.9970	0.0000	5.8439	27.6830
R110:	0.9038	0.0000	8.3456	29.8997
R111:	0.9983	0.0000	2.8367	26.5533
R112:	0.8709	0.0000	3.3246	274.5008
R201:	0.300	0.050	2.000	17.60
R202:	0.7015	0.1696	25.2719	67.9228
R203:	0.4488	0.3071	37.3635	28.1987
R204:	0.9803	0.0000	5.7384	23.6597
R205:	0.9749	0.0000	4.2222	11.0538
R206:	0.8407	0.0000	21.4840	36.3410
R207:	0.9570	0.0000	13.6464	44.2890
R208:	0.0000	0.3101	34.2739	32.0445
R209:	0.9019	0.0000	3.7543	49.1394
R210:	0.9782	0.0000	12.3150	35.6985
R211:	0.7738	0.0000	4.2728	53.6483
RC101:	0.1000	0.0000	42.80	55.60
RC102:	0.6154	0.0000	10.3781	41.7276
RC103:	0.9829	0.0000	5.4576	26.2972
RC104:	1.0000	0.0000	4.4443	573.2388
RC105:	0.9976	0.0000	10.8924	64.3919
RC106:	1.0000	0.0000	6.2760	57.5871
RC107:	0.8347	0.0000	6.7537	118.6446
RC108:	0.9960	0.0000	7.5085	93.0992
RC201:	0.820	0.0000	18.60	33.500
RC202:	1.0000	0.0000	23.6881	67.7323
RC203:	0.9107	0.0000	17.6818	65.8201
RC204:	0.8367	0.0000	11.5361	62.8168
RC205:	1.0000	0.0000	21.6881	82.6331
RC206:	1.0000	0.0000	15.7154	86.9275
RC207:	0.9217	0.0000	17.7751	88.0324
RC208:	1.0000	0.0000	6.8093	100.2798

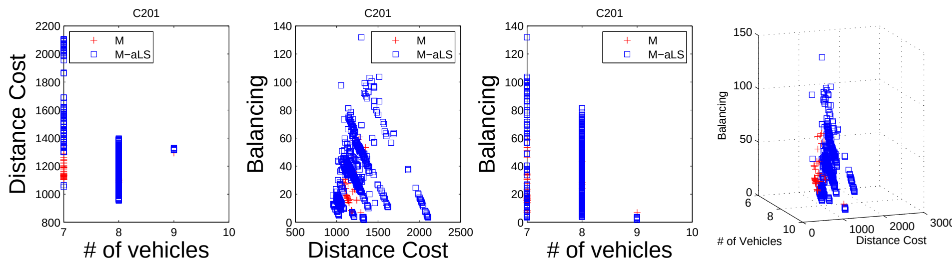
Experimental Series 3 - MOEA/D-aLS vs. MOEA/D-rLS:

Figure 13 shows that the performance of MOEA/D-aLS is better than MOEA/D-rLS in terms of both convergence and diversity on a subset of the Solomon dataset (C101, C201, R101, R201, RC101 and RC201). In particular, the MOEA/D-aLS 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.

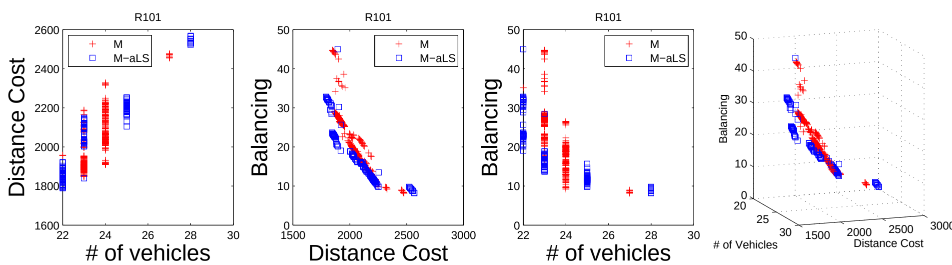
Again, this is more evident in Table III that summarizes the performance of MOEA/D-aLS and MOEA/D in terms of the Coverage (C) and the Distance from the reference set (I_D) in all 56 Solomon test instances. The results show that the non-dominated solutions obtained by the MOEA/D-aLS dominate most of the non-dominated solutions obtained by MOEA/D-rLS and that our proposed approach performs better in terms of I_D. In particular, the MOEA/D-aLS outperforms MOEA/D-rLS in 48 out of 56 test instances, i.e., it provides



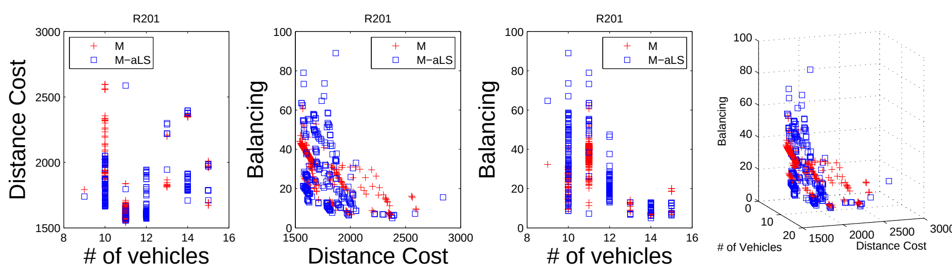
(a) C101



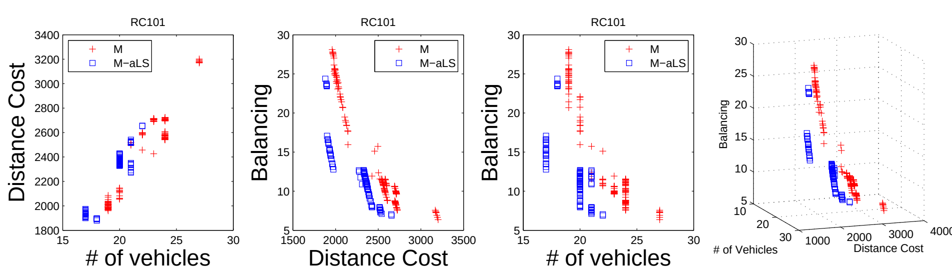
(b) C201



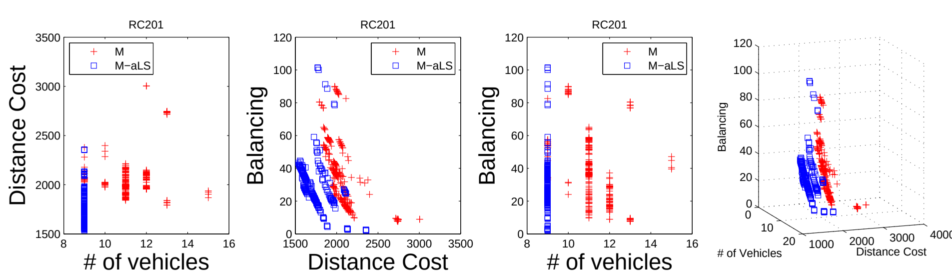
(c) R101



(d) R201

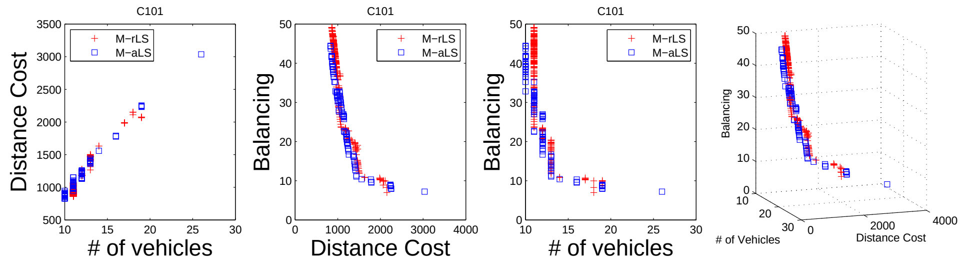


(e) RC101

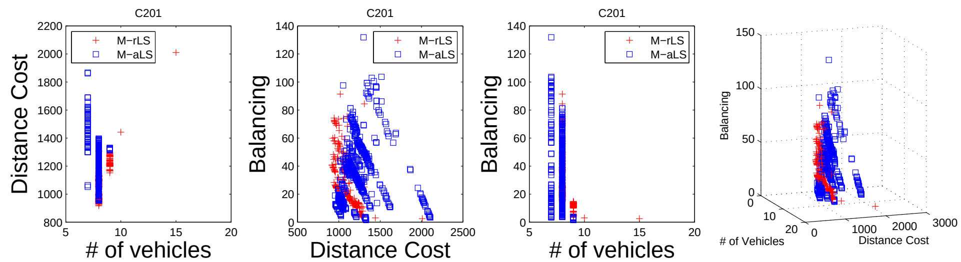


(f) RC201

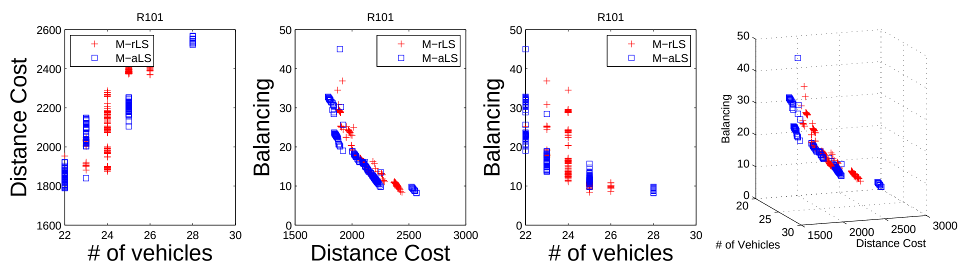
Fig. 12. Comparison between the proposed MOEA/D with adaptive Local Search (MOEA/D-aLS) and the conventional MOEA/D.



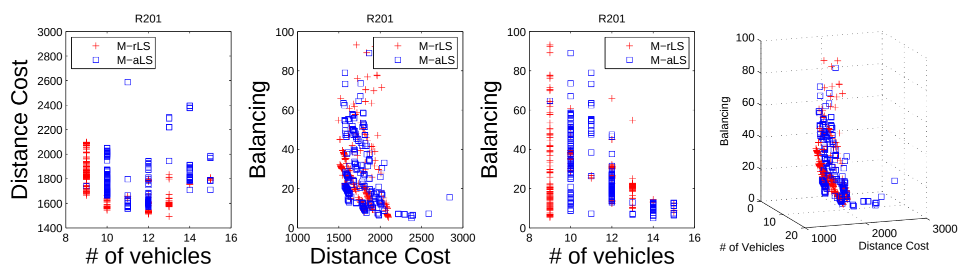
(a) C101



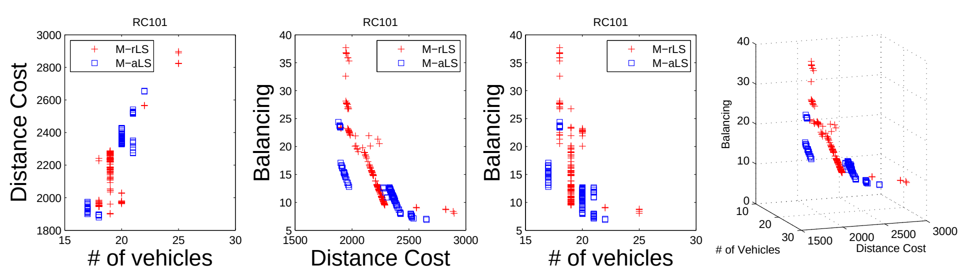
(b) C201



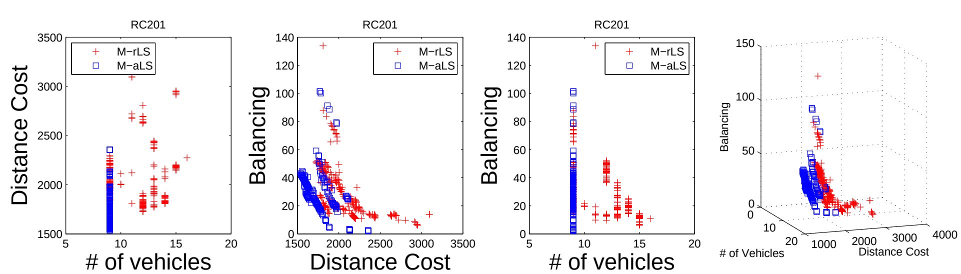
(c) R101



(d) R201



(e) RC101



(f) RC201

Fig. 13. 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 III
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-aLS, M-rLS)	C(M-rLS, M-aLS)	I_D(M-aLS)	I_D(M-rLS)
C101:	0.5700	0.5100	45.00	45.00
C102:	0.8108	0.1338	16.1870	25.1422
C103:	1.0000	0.0000	16.2653	18.3169
C104:	0.9123	0.0000	22.1395	15.9686
C105:	1.0000	0.0000	4.5502	53.5151
C106:	0.3711	0.0494	8.5128	15.0741
C107:	0.8581	0.0549	15.9056	8.2695
C108:	0.7467	0.0000	9.8949	14.3187
C109:	0.7984	0.1367	6.0180	7.7708
C201:	0.6600	0.4500	13.700	14.870
C202:	0.9667	0.0128	10.5454	13.0121
C203:	0.9793	0.0000	42.8598	37.8997
C204:	0.8981	0.0000	18.5364	14.9857
C205:	0.9906	0.0323	4.8392	12.4235
C206:	0.9083	0.0000	7.1499	28.7335
C207:	0.8565	0.1501	12.8931	12.6655
C208:	0.9984	0.0028	3.7777	10.7057
R101:	0.600	0.200	2.60	31.70
R102:	0.7600	0.1151	9.7868	15.9861
R103:	0.6747	0.0000	8.6037	17.4263
R104:	1.0000	0.0000	8.7496	10.3383
R105:	0.7219	0.0571	8.9839	17.8700
R106:	0.5210	0.3028	10.9833	5.9599
R107:	0.9730	0.0000	5.6055	10.4212
R108:	0.9675	0.0758	4.0837	7.9049
R109:	0.9970	0.0000	5.9754	11.0515
R110:	0.3782	0.4484	12.3672	4.9977
R111:	0.9983	0.0000	2.8384	6.8581
R112:	1.0000	0.0000	4.3903	5.3078
R201:	0.600	0.201	2.020	2.200
R202:	0.0149	0.9852	26.7705	8.1937
R203:	0.3701	0.3711	31.3426	20.5700
R204:	1.0000	0.0000	5.7384	31.3253
R205:	0.9749	0.0000	4.2222	11.0538
R206:	0.0000	0.8431	24.6603	7.9243
R207:	0.9606	0.0000	13.6476	36.2269
R208:	0.0000	0.7341	47.1448	3.2491
R209:	0.0011	0.5000	5.1973	19.0129
R210:	0.0000	0.9814	16.7154	20.1256
R211:	0.4178	0.6110	5.0982	4.1596
RC101:	0.500	0.430	42.80	26.60
RC102:	0.6709	0.0000	10.3796	19.5948
RC103:	0.2171	0.6597	5.9040	9.6797
RC104:	0.9788	0.0000	4.4449	8.9320
RC105:	0.9976	0.0000	10.8924	16.8726
RC106:	1.0000	0.0000	6.2760	6.2853
RC107:	0.9597	0.0000	6.7537	9.3652
RC108:	0.7557	0.0141	7.5085	6.6495
RC201:	0.1003	0.0000	18.60	37.30
RC202:	0.5517	0.1033	24.6459	17.9281
RC203:	0.0000	0.8213	18.0369	19.3742
RC204:	0.0000	1.0000	15.9901	10.7663
RC205:	0.2429	0.2947	21.6885	34.5763
RC206:	0.0411	0.0000	18.8320	7.9089
RC207:	0.9181	0.0000	17.7751	11.8165
RC208:	1.0000	0.0000	6.9002	8.9560

a higher quality PF in around 85% of the VRP instances. The proposed MOEA/D-aLS approach performs worst mainly when there is more randomness in the distribution of the customers. For example, the MOEA/D-rLS approach provides non-dominated solutions with higher quality than the proposed MOEA/D-aLS in the following test instances: R110, R202, R206, R211, RC103, RC203, RC204 and RC206. On average, MOEA/D-aLS dominate around 85% of the non-dominated solutions obtained by MOEA/D in class C. This improvement is reduced to 60% and 56% on average for classes R and RC, respectively (and even further to 40% and 36%, when we only consider subclasses R2 and RC2, respectively). This is due to the fact that the adaptive selection of particular local search heuristics may worsen the global search of the MOEA/D approach and therefore search the multi-dimensional objective space less adequately. This fact can be further supported from the performance of the two approaches with respect to the I_D metric, where the MOEA/D-aLS approach performs better in just around 70% of the entire Solomon dataset.

Another important observation that can be drawn from both Experimental Series 2 and 3 is that there is some variability on the percentage of the improvement provided by the proposed MOEA/D-aLS compared to the MOEA/D-rLS approach. This provides some insights that the proposed MOEA/D with adaptive local search heuristics approach should not only take into consideration the preference of each subproblem (i.e., objective functions) when applying certain problem-specific heuristics for locally optimizing existing solutions, but also other parameters related to the test instances such as the distribution of the customers and the size of the time windows.

Finally, Table IV shows the best solutions obtained for each objective function by each MOEA/D variant for all 56 test instances of the Solomon dataset. For every test instance, the best solution found for each objective is shown in bold. These results show that, in general, the MOEA/D-aLS outperforms the other two MOEA/D variants. In this experiment, out of the 56 test instances, the results obtained by MOEA/D-aLS are:

- [at least as good / the best] with respect to the min. number of vehicles V in $[87.5\%/46.4\%]$ of the cases, respectively,
- the best with respect to the min. distance cost D in 66.1% of the cases, and
- [at least as good / the best] with respect to the min. balancing offset B in $[55.4\%/39.3\%]$ of the cases, respectively.

The hybrid MOEA/Ds seem to clearly outperform the conventional MOEA/D, and as already observed, in previous experiments (Experimental Series 3, Table III), MOEA/D-rLS only seems to match the performance of our proposed method in some cases where there is more randomness in the distribution of the customers, and in particular in the subclasses R2 and RC2 of the Solomon dataset.

An example of the best solution in terms of number of vehicles obtained by MOEA/D-aLS is illustrated in Figure 14.

TABLE IV
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-aLS			MOEA/D-rLS			MOEA/D		
	V	D	B	V	D	B	V	D	B
C101:	**10.0**	**828.9**	7.2	11.0	860.2	**6.9**	14.0	1116.0	7.4
C102:	**11.0**	**994.0**	**5.7**	12.0	1159.9	**5.7**	14.0	1351.6	9.3
C103:	**10.0**	**1148.3**	**2.5**	12.0	1343.2	**2.5**	13.0	1295.3	4.6
C104:	**10.0**	**1237.0**	**0.8**	**10.0**	1324.9	4.3	11.0	1260.0	1.7
C105:	**11.0**	**956.4**	8.3	12.0	1028.9	**5.6**	16.0	1469.8	7.5
C106:	**12.0**	**1029.4**	6.4	**12.0**	1037.4	4.9	15.0	1333.2	**4.6**
C107:	**10.0**	**889.3**	9.4	11.0	991.4	**3.6**	13.0	1360.6	5.4
C108:	**10.0**	**947.8**	3.0	11.0	1110.5	**2.8**	12.0	1127.8	5.7
C109:	**10.0**	**992.9**	**2.2**	**10.0**	1074.9	2.5	12.0	1267.5	2.5
C201:	**7.0**	954.0	**1.5**	8.0	**917.0**	2.5	**7.0**	1085.0	6.8
C202:	**5.0**	**838.3**	**0.5**	6.0	949.9	0.9	7.0	1149.3	0.7
C203:	**5.0**	**945.1**	**0.3**	6.0	1188.1	0.5	6.0	1184.6	0.5
C204:	**4.0**	**1095.1**	**0.1**	5.0	1159.9	0.2	5.0	1129.3	0.2
C205:	6.0	854.8	**0.5**	**5.0**	**787.1**	0.8	**5.0**	886.8	1.3
C206:	**4.0**	**763.6**	**0.1**	6.0	977.7	0.5	5.0	958.1	0.4
C207:	**4.0**	858.5	**0.3**	**4.0**	**846.6**	**0.3**	5.0	928.0	**0.3**
C208:	**4.0**	**711.9**	0.3	4.0	785.3	**0.2**	5.0	942.6	0.3
R101:	**22.0**	**1789.0**	**8.0**	**22.0**	1874.0	8.4	**22.0**	1848.0	8.2
R102:	**19.0**	**1603.3**	3.9	**19.0**	1623.1	**3.6**	20.0	1661.8	6.6
R103:	**15.0**	**1391.6**	**2.0**	16.0	1436.8	3.9	17.0	1521.3	4.1
R104:	**12.0**	**1216.3**	1.9	13.0	1223.9	**1.4**	15.0	1702.6	11.2
R105:	**17.0**	**1556.8**	3.5	**17.0**	1606.8	**3.0**	19.0	1591.8	4.3
R106:	16.0	**1437.4**	3.6	16.0	1484.9	**1.4**	**15.0**	1477.2	5.1
R107:	**13.0**	**1268.4**	**1.6**	14.0	1295.5	**1.6**	14.0	1461.1	4.3
R108:	**11.0**	**1085.4**	**1.3**	**11.0**	1144.7	1.7	12.0	1336.4	2.6
R109:	**14.0**	1362.9	2.7	15.0	**1359.1**	**2.4**	15.0	1451.6	3.8
R110:	**13.0**	1273.9	**1.5**	**13.0**	**1214.2**	1.9	14.0	1370.9	4.2
R111:	**13.0**	**1267.3**	3.1	14.0	1311.4	**2.0**	**13.0**	1414.0	2.5
R112:	**11.0**	1199.8	1.7	**11.0**	**1173.3**	**1.2**	14.0	1579.8	10.8
R201:	**9.0**	**1538.0**	**5.0**	**9.0**	1555.0	5.0	**9.0**	1540.0	6.0
R202:	**7.0**	1368.1	1.0	**7.0**	**1341.1**	0.8	**7.0**	1433.2	**0.7**
R203:	**5.0**	1258.6	**0.2**	**5.0**	1294.4	0.6	**5.0**	**1182.7**	**0.2**
R204:	**4.0**	**1024.2**	**0.2**	5.0	1131.0	**0.2**	**4.0**	1099.2	0.3
R205:	**5.0**	**1231.5**	0.6	**5.0**	1276.6	**0.4**	**5.0**	1276.6	**0.4**
R206:	5.0	1250.3	**0.1**	**4.0**	**1209.2**	0.2	5.0	1302.7	0.7
R207:	**3.0**	**1120.3**	0.3	4.0	1143.2	**0.1**	4.0	1242.1	0.3
R208:	**3.0**	1095.9	0.2	**3.0**	**889.3**	**0.1**	3.0	1103.0	**0.1**
R209:	**4.0**	1167.3	**0.2**	**4.0**	**1119.7**	**0.2**	5.0	1351.6	0.5
R210:	5.0	1274.3	0.3	**4.0**	**1189.1**	**0.2**	6.0	1276.3	0.4
R211:	**4.0**	934.8	0.2	**4.0**	**902.1**	0.2	**4.0**	1181.4	**0.1**
RC101:	**17.0**	**1879.0**	**6.4**	18.0	1899.0	8.0	19.0	1960.0	**6.4**
RC102:	**15.0**	**1651.7**	3.5	17.0	1747.3	**3.3**	16.0	1869.8	6.2
RC103:	**13.0**	1515.1	**1.7**	**13.0**	**1503.2**	2.4	14.0	1563.4	5.2
RC104:	**12.0**	**1331.4**	0.9	**12.0**	1370.4	**0.7**	16.0	2173.9	17.9
RC105:	**17.0**	**1727.3**	**4.6**	19.0	1871.9	5.2	20.0	2042.2	9.6
RC106:	**14.0**	**1528.9**	**2.5**	**14.0**	1554.3	2.8	18.0	1873.3	12.1
RC107:	**13.0**	**1438.2**	**2.6**	14.0	1507.1	2.8	17.0	1997.7	11.7
RC108:	**13.0**	**1363.1**	**1.5**	**13.0**	1390.8	1.9	15.0	1820.7	10.4
RC201:	**9.0**	**1535.0**	**2.0**	**9.0**	1726.0	6.1	**9.0**	1792.0	7.6
RC202:	**7.0**	1564.7	1.3	9.0	**1553.6**	**1.0**	8.0	1792.9	4.7
RC203:	6.0	1342.0	**0.3**	**5.0**	**1334.7**	0.5	7.0	1687.4	3.6
RC204:	4.0	1263.1	0.3	**3.0**	**1181.0**	**0.1**	5.0	1374.7	0.3
RC205:	8.0	**1576.7**	**0.9**	**7.0**	1670.8	1.0	9.0	1969.5	2.8
RC206:	**6.0**	1530.9	0.5	7.0	**1445.9**	**0.3**	7.0	1909.9	3.7
RC207:	**4.0**	**1281.7**	**0.2**	5.0	1365.6	**0.2**	7.0	1757.4	5.1
RC208:	**3.0**	1117.4	0.3	4.0	**1113.9**	**0.1**	6.0	1557.9	0.9

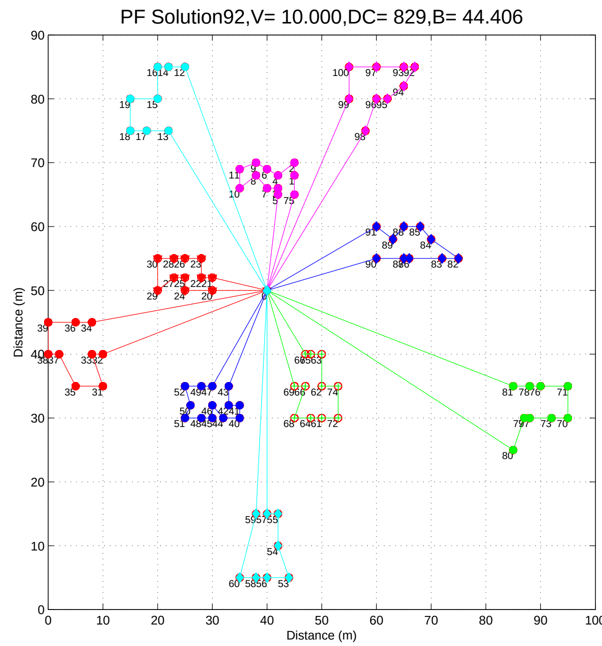


Fig. 14. Routes corresponding to best solution for C101 using MOEA/D-aLS

VII. 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) heuristics. The MOEAD-aLS decomposes the proposed MOP into a set of scalar subproblems which are solved simultaneously using at each generation multiple LS heuristics adaptively selected based on objective preferences and instant requirements. We evaluate our proposition on the entire Solomon dataset that is composed of 56 benchmark test instances. The results show that the MOEA/D-aLS clearly improves the performance of the traditional MOEA/D in almost all cases and in most cases of the MOEA/D hybridized with randomly selected LS heuristics.

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. Furthermore, relaxing our tri-objective optimization problem and applying the proposed approach on an existing two-objective optimization, so as to evaluate its performance compared to the best-known solutions, is also a future direction. Finally, we plan to apply our hybrid MOEA/D with adaptive LS heuristic approach on other benchmark problems.

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 metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.