

AN ATTRACTORS-BASED PARTICLE SWARM OPTIMIZATION FOR MULTIOBJECTIVE CAPACITATED VEHICLE ROUTING PROBLEM

ABDOUL-HAFAR HALASSI BACAR^{1,*}  AND SAID CHARRIFFAINI RAWHOUDINE² 

Abstract. In this paper, a new multiobjective discrete particle swarm algorithm is presented for the Capacitated vehicle routing problem. The binary algorithm integrates particle displacement based on local attractors, a crowding distance as elitism policy and genetic operators. The proposed approach is first implemented on a set of well-known benchmarks for single-objective capacitated vehicle routing problems and compared to results underlined in the literature. The obtained results demonstrate its ability to achieve the main optimization solution and sometimes prove its efficiency from other given techniques. Then, the approach is applied to an academical example for the multiobjective Urban Bus Routing Problem with Route Balancing.

Mathematics Subject Classification. 90C29, 90B20.

Received February 8, 2021. Accepted August 4, 2021.

1. INTRODUCTION

The vehicle routing problem (VRP) is a combinatorial optimization problem seeking to service a number of customers with a fleet of vehicles. Introduced first by Dantzig and Ramser [8], the VRP is an important problem in the fields of transportation, distribution, and logistics. The problem is concerned with delivering goods to a set of customers with known demands through vehicle routes by minimizing the total route cost. Capacitated vehicle routing problem (CVRP) is the most elementary version of the VRP and seeks a number of routes for m number of vehicles with Q units homogeneous/inhomogeneous capacity to minimize total transportation cost of routes while satisfying the delivery demands of n number of customer nodes. Each route must start and finish at the depot and each customer has to be visited exactly once by one vehicle.

Various real cases versions of VRP can be found in literature and details with formulation, and solution methods can be found in [35]. The VRP is formulated as a combination of the bin packing problem (BPP) and the traveling salesman problem (TSP). Since both BPP and TSP are proven to be NP-hard problems [38]; the VRP and its CVRP variant are too. Among those CVRP variants are Vehicle Routing Problem with Simultaneous Pick-up and Delivery (VRP-SPD) [33] and Vehicle Routing Problem with time windows [9, 37].

Keywords. Particle swarm optimization, multiobjective optimization, vehicle routing problems, urban bus routing problems, simultaneous pick-up and delivery.

¹ Université des Comores, Faculté des Sciences et Techniques, Département des Mathématiques, Physique, Chimie et Informatique, Laboratoire des Mathématiques, Statistiques et Informatique (LMSI), BP 2585 Moroni, Comoros.

² Agence pour la Sécurité de la Navigation Aérienne en Afrique et à Madagascar, Aéroport International Prince Said Ibrahim, Moroni, Comoros.

*Corresponding author: halassi.abdoul@fst-udc.org; halassi.abdoul@gmail.com

In the VRBP-SPD version, customers require simultaneous collection of goods from their location in addition to distribution of goods to their location and a fleet of vehicles originated in a depot serves customers with pick-up and deliveries from/to their locations. Increasing importance of reverse logistics activities make it necessary to determine efficient and effective vehicle routes for simultaneous pick-up and delivery activities. Moreover, when taking into account the costumers waiting time, the vehicle running time and seeking to minimize that time for given reasons (using delay for the delivering goods, preserving some physical and/or medical qualities, reducing the vehicle motor running and the availability time windows of customers), we formulate a multiobjective optimization problem. That is, the new goal is not only to minimize the total transportation cost of route but also minimizing each driver's working time. This multiobjective version of the VRP is named the Vehicle Routing Problem with Route Balancing (VRP-RB) [17]. Numerous real-life applications of transportation problems can be derived from this version. We can cite for instance the Urban School Bus Routing [13,29] which is a challenging problem in urban densely populated areas.

Many natural phenomenon based techniques, for example ant colony optimization (ACOs) and genetic algorithms (GAs), have been developed in solving multi-objective optimization problems such as NSGA-II [10], PAES [20] and SPEA2 [40] in term of genetic algorithms and for Multiobjective Ant colony Optimization, one can refer to [2,26] among others. These population-based algorithms have the ability to explore the different parts of the Pareto front simultaneously [16].

During these last decades, many PSO [18] approaches have been proposed and most of them use the concept of Pareto dominance to classify the solutions [22,27]. Comparisons of different multiobjective algorithms that use such techniques have been outlined in [21,27] among others. These outlined works show that, in general, MOPSO based techniques are able to generate a best set of nondominated solutions close to the true Pareto front compared to GAs based algorithms. Note that in terms of diversity of the nondominated solutions, GA based algorithms can produce best results but are not able to cover the entire Pareto front in all test functions.

However, numerical approximation challenges reside on proposed optimization problems algorithms, such as how intensifying the final computed Pareto front close to the true Pareto known front and how improve diversification of these optimal solution over the whole Pareto front. First, it should be noted that the use of a new displacement of particles [15], which basically is based on local attractors, takes into account more the process of intensification and ensure optimal solution converge to a slongtable point of the true Pareto front. In other hand, the use of crowding distance for the nondominated sorting in [27], the local choice of global best guide of a particle and the personal best guide mutation coupled to a crossover operator may improve considerably the process of diversification.

In other hand, several combinatorial Particle Swarm Optimization (PSO) techniques have been proposed to solve different versions of the Vehicle Routing Problem [1,24,25] and results have shown that PSO based algorithms are able to solve efficiently this kind of combinatorial optimization problem. So, the proposed algorithm extends PSO in solving combinatorial multiobjective capacitated vehicle routing problem with route balancing by changing the classical displacement to the one based on local attractors and incorporating the mechanism of crowding distance computation in the insertion method and the deletion method of the external archive of nondominated solutions. The proposed approach is named the Discrete Multiobjective Particle Swarm Optimization based on Attractors (DMOPSO-A). The attractors based displacement, the crowding distance mechanism together with a mutation based on personal best guides and crossover operators maintain the diversity and the intensity of nondominated solutions in the external archive.

In order to evaluate the performance of the proposed DMOPSO-A approach, several test problems from the literature benchmarks are considered. Results of some known solvers are compared with the solution provided by the proposed approach. And then, an academic version of the urban bus routing problem is solved by the actual approach.

This paper is organized as follows; in Section 2, the mathematical formulation and details of CVRP with route balancing is presented. In Section 3, details of the proposed DMOPSO-A approach are introduced and explained. Numerical experiments and performance validations of the proposed approach are evaluated in Section 4. Finally, Section 5 concludes the paper with future perspectives.

2. THE MULTIOBJECTIVE CAPACITATED VEHICLE ROUTING PROBLEM

In this section, the CVRP is formulated as a Mixed-integer linear programming (MILP) model based on model. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a graph with $\mathcal{V} = (v_0, v_1, \dots, v_n)$ a set of vectors where v_0 is the depot and v_1, \dots, v_n are the n customers to be served; and $\mathcal{A} = A_{ij} = \{(v_i, v_j) : (v_i, v_j)_{i \neq j} \in \mathcal{V}\}$ a set of arcs, representing a route from customer i to customer j . The graph \mathcal{G} is considered to be complete, *i.e.*, for each couple $(i, j)_{i \neq j}$ of customers, there exists an arc a_{ij} (a direct route) from customer i to customer j . Matrices D and T with entries d_{ij} and t_{ij} represent the length of the arc a_{ij} and the average time for the road cross a_{ij} respectively. Each customer i has a given positive delivery q_i , a pick-up quantity p_i and a service duration s_i . At the depot, m vehicles are located and each vehicle has a maximal capacity Q_k and a maximal service duration \mathcal{D}_k . Each vehicle has also a fixed cost f and a variable cost g per distance unit.

The multiobjective CVRP consists on choosing at most m circuits so that

- (1) Each circuit starts and ends at the depot;
- (2) Each customer is visited once with exactly one vehicle;
- (3) The total discharge in a vehicle j along a given arc must not exceed the maximal capacity Q_k of the vehicle;
- (4) The total service duration of each vehicle (including the service duration) must not exceed the limit duration \mathcal{D}_k of the vehicle;
- (5) The total cost and the route balancing of the whole routing must be as minimal as possible.

Let define the CVRP decision variables as a binary variable x_{ijk} which indicates whether the arc (i, j) is traversed by the vehicle k or not. $x_{ijk} = 1$ is the vehicle k follows the route (i, j) and $x_{ijk} = 0$ if not; and a variable y_{ijk} which represents the weight of vehicle k when traversing the route (i, j) . The objective functions of this problem are expressed as:

$$F_1(x, y) = f \sum_{k=1}^m \sum_{j=1}^n x_{0jk} + g \sum_{i=0}^n \sum_{j=1}^{n+1} \sum_{k=1}^m d_{ij} \times x_{ijk}, \quad (2.1)$$

$$F_2(x, y) = \max_{k \in \{1, \dots, m\}} \left(\sum_{i=0}^n \sum_{j=1}^{n+1} x_{ijk} \times d_{ij} \right) - \min_{k \in \{1, \dots, m\}} \left(\sum_{i=0}^n \sum_{j=1}^{n+1} x_{ijk} \times d_{ij} \right) \quad (2.2)$$

and the optimization problem is stated by

$$\text{Minimize } F(x, y) = (F_1(x, y), F_2(x, y)) \quad (2.3)$$

$$\text{subject to } \sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1, \quad 1 \leq i \leq n \quad (2.4)$$

$$\sum_{j=0}^n x_{jik} = \sum_{j=1}^{n+1} x_{ijk} \quad 1 \leq i \leq n, 1 \leq k \leq m \quad (2.5)$$

$$\sum_{j=1}^n x_{0jk} \leq 1, \quad 1 \leq k \leq m \quad (2.6)$$

$$\sum_{j=1}^n t_{ijk} + s_j \leq \mathcal{D}_k, \quad 0 \leq i \leq n, 1 \leq k \leq m \quad (2.7)$$

$$y_{ijk} \leq x_{ijk} \times Q_k, \quad 0 \leq i, j \leq n, 1 \leq k \leq m \quad (2.8)$$

$$x_{ijk} \in \{0, 1\}, \quad 0 \leq i \leq n, 0 \leq j \leq n+1, 1 \leq k \leq m \quad (2.9)$$

$$y_{ijk} \geq 0, \quad 0 \leq i \leq n, 0 \leq j \leq n+1, 1 \leq k \leq m. \quad (2.10)$$

Constraints (2.4) and (2.5) form the feasible vehicle routes, such that each customer is served by exactly one vehicle (2.4) and each vehicle that arrives to a customer leaves that customer (2.5) and each vehicle is used to

serve at most one round (2.6). The constraint (2.7) imposes to each vehicle to respect its maximum time service allowed during its serving time and constraint (2.8) controls the maximum capacity of each vehicle during the service. Finally, constraints (2.9) and (2.10) are used for the definition of the decision variables. Note that these constraints are those imposed for the capacitated vehicle routing problems. But when solving another version of CVRP, VRP-SPD or VRPTW for example, additional constraints are needed.

3. THE BINARY PARTICLE SWARM OPTIMIZATION BASED ON ATTRACTORS

3.1. The discrete particle swarm optimization

The initial binary particle swarm optimization algorithm was proposed by Kennedy and Eberhart [19] in 1997 for discrete optimization problems and later other versions are studied in [5, 6, 28] among others. In these methods, each particle must take “true” or “false” values, “yes” or “no” values, or solution alike which indicate a potential solution. A fitness solution is always introduced in order to evaluate the appropriateness of solutions. Each particle is located in some dimensional space and each element of a particle position can take the binary value of 0 or 1 which can change from 1 to 0 and *vice versa*. Each particle has also a velocity vector with elements defined in range $[-V_{\max}, V_{\max}]$. The DPSO velocities are defined in terms of probabilities that a bit will be in one state or the zero state. At the beginning of the algorithm, a number of particles and their velocities are initialized randomly. Then in each iteration step, the algorithm searches on how reaching the optimal solutions based on its predefined fitness function. The global best guide and the personal best guides used to update velocity vectors are chosen in the same manner as for the continuous optimization. Eventually, the position of the particles is updated using velocity vectors.

With this definition, the velocity vector must be restricted in the interval $[0, 1]$. So, a normalization and control function is introduced to force the real values of velocity vectors to be in $[0, 1]$. In this study, the sigmoid function [36] is defined by

$$V'_{ij} = \text{sigm}(V_{ij}(t)) = \frac{1}{1 + \exp(-V_{ij}(t))}. \quad (3.1)$$

The velocity is updated in the same way as in continuous optimization and is expressed by

$$V_{k,t+1} = \omega V_{k,t} + c_1 r_1 (P_{k,t} - x_{k,t}) + c_2 r_2 (G_{b,t} - x_{k,t}), \quad (3.2)$$

where ω is the inertia factor that influences the local and global abilities of the algorithm, $V_{k,t}$ is the velocity of the particle k at iteration step t , c_1 and c_2 are weights affecting the cognitive and social factors, respectively. r_1 and $r_2 \sim \mathcal{U}(0, 1)$; $P_{k,t}$ stands for the best value found by particle k (pBest) and $G_{b,t}$ denotes the global best found by the entire swarm (gBest).

And the new position of the particle is obtained using the equation

$$X_{k,t+1} = \begin{cases} 1 & \text{if } r_{xk} \leq V'_{k,t+1} = \text{sigm}(V_{k,t+1}) \\ 0 & \text{otherwise,} \end{cases} \quad (3.3)$$

where r_{xk} is a uniform random number in the range $[0, 1]$.

3.2. Displacement based on attractors

Attractors, in PSO, describe the path marked by particle positions which any particle passes by before landing on its final target. These intermediate positions are so called local intermediate positions. Let us define $S_{k,t}$ and $T_{k,t}$, these local intermediate positions, the particle positions is represented by the values position and velocity (3.2) and (3.3), the best personal performance $P_{k,t}$ and the best global performance $G_{b,t}$ of the swarm. So far the particle $(X_{k,t+1}; V_{k,t})$ travels by the local positions $(S_{k,t}$ and $T_{k,t})$ before it reaches the next particle position $(X_{k,t+1}; V_{k,t+1})$. The operators which recently inserted in [15] perform the continuous and/or combinatorial

optimization. The increment process grants the application a benefit of the convergency and the diversity. The equations of attractors are stated as follows:

$$\begin{aligned} S_{k,t} &= X_{k,t} + c_1 V_{k,t}, \\ T_{k,t} &= S_{k,t} + c_2 r_2 (P_{k,t} - S_{k,t}), \\ X_{k,t+1} &= T_{k,t} + c_3 r_3 (G_{k,t} - X_{k,t}), \\ V_{k,t+1} &= \alpha V_{k,t} + \beta (P_{k,t} - X_{k,t}) + \gamma (G_{k,t} - X_{k,t}), \end{aligned} \quad (3.4)$$

where

$$\alpha = c_1 (1 - c_2 r_2) \times (1 - c_3 r_3), \quad \beta = c_2 r_2 (1 - c_3 r_3), \quad \gamma = c_3 r_3, \quad (3.5)$$

The constants c_1 , c_2 , and c_3 are arbitrary numbers and r_1 , r_2 and $r_3 \sim \mathcal{U}(0, 1)$. However, it is suitable to secure the convergency of the approach. Therefore, the arbitrary constants will be constraint as follows (one set at a time) [15]:

$$0 < c_1 < 0.9; \quad 0 < c_2 < 2; \quad 0 < c_3 < 2, \quad (3.6)$$

$$0 < c_1 < 0.9; \quad 2 \leq c_2 < 4; \quad 2 \leq c_3 < 4. \quad (3.7)$$

Introducing the sigmoid function, the attractors aforementioned are reshaped to correspond as follows. The velocity remains as it is typed above (3.4), however, the intermediate position are formulated

$$S_{k,t} = \begin{cases} 1 & \text{if } r_{sk} \leq V'_{k,t} = \text{sigm}(X_{k,t} + c_1 V_{k,t}) \\ 0 & \text{otherwise,} \end{cases} \quad (3.8)$$

$$T_{k,t} = \begin{cases} 1 & \text{if } r_{tk} \leq \text{sigm}(S_{k,t} + c_2 r_2 |P_{k,t} - S_{k,t}|) \\ 0 & \text{otherwise,} \end{cases} \quad (3.9)$$

so that the following particle position is written

$$X_{k,t+1} = \begin{cases} 1 & \text{if } r_{xk} \leq \text{sigm}(T_{k,t} + c_3 r_3 |G_{k,t} - X_{k,t}|) \\ 0 & \text{otherwise,} \end{cases} \quad (3.10)$$

where $(r_{sk}, r_{tk}, r_{xk}) \sim \mathcal{U}(0, 1)$ are random numbers and $|x|$ is the absolute value of x .

3.3. Elitist policy

The diversity is a quality performance which quantify an approach for its abilities to spread particles and have a comparable Pareto front. Hypercubes techniques [7] and Crowding Distance (CD) [23, 27] are well known in operations. They aim to select the best particles in order to respect an adequate distribution of the Pareto. The process consists of mining particles nondominated solutions. The following work, The CD gathers to perform this tedious work and it delivers value of density of solutions concerning its surroundings. The obtained values are used to for comparison amongst nondominated solutions, therefore to maintain a well-distributed Pareto front. To illustrate, if a given archive set is full, one of the techniques (hypercubes techniques or CD) is applied to select the best solution between a nondominated solution in the archive set and a new nondominated solution and stored in the archive.

For instance, the nondominanted solutions rest in the archive set; they are arranged in order, from lower to the upper of objective function values. These values are computed for all nondominated solutions; besides the value represents the distance between two neighboring nondominated solutions of the considered solution. Two particular values remain permanent in the achieve set: the lower and the higher objective function values; their

CD possess a significant value and they are also the boundaries. The algebraic form of the CD of the $kieth$ particle at iteration step t is stated as follows:

$$CD(k) = \sum_{i=1}^2 (f_i(x_{k+1,t}) - f_i(x_{k-1,t})), \quad (3.11)$$

The two particles sent in computation for the kth are the $k-1$ and $k+1$ particles respectively before and after the considered particle. Therefore, the CD values saved will be compared so that the particles with small CD values are exposed to be replaced for another nondominated solutions. The nondominated solutions altogether which are kept in the external archive set are eligible to be a global best guide (each one). Moreover, a random selection is practicable for all the swarm at each iteration. However, a single steady global best guide occur an efficient convergency. On the other hand, the diversification is jeopardized in all the process.

As regards of the approximated Pareto front, all the nondominated solutions recorded in the archive set might be selected to serve as a guide. From the start of the program, a set of particles is run in the PSO variant, then the nondominated solutions are stored in the archive set. Then, at each iteration a nondominated solution is selected as the global best guide randomly. Obviously any nondominated solution can play the role of global best guide. In process of finding the approximated Pareto Front, solutions will converge towards the nondominated solution. According to the parameters' set, the application can guarantee a compelling convergence. Nevertheless the distribution over the archive set remains debatable as in each step all the offspring relay on a random global best guide. In term of perusing the process till the end, many nondominated solutions may gather partially. Therefore, the display of Pareto front could appeal with scattered dense part of nondominated solutions.

Based on the late information, it is suitable to add another tool to hold the diversification. For the current circumstances, a particle with the highest CD value represents the global best guide for the swarm at each iteration. Although the satisfactory was not accurate enough; an additional tool is subsequently added, the DMOPSO-A. This application reinforces the intensification by allowing each particle to possess its own best guide $G_{k,t}$ at each iteration along the entire process. The DMOPSO-A withstands both the intensification and the diversification; by the end the nondominated solutions dwell in a set with flawless diversity characteristic.

3.4. Adaptation of genetic operators

During this last decade, operation researchers focus on adapting native evolutionary operators to particle swarm approaches such as mutation [7, 27] and elitism [20].

3.4.1. Mutation operator

The mutation operator consists on applying a mutation (a genetic modification) of a particle to improve its position. Applied to a given particle chosen randomly, the mutation operator changes its coordinates with a certain probability. An uniform mutation operator selects with a certain probability a particle to be muted. Then, with another given probability, it chooses a dimension (one of the particle coordinates in the decision space) and apply modification of that position with respect of the search domain space.

In this work, we select the mutation operator [34] which applies the mutation of the personal best guide of the particle instead of the particle itself. The mutation operator is defined by

$$\mathcal{P}_{k,t}^i = \begin{cases} P_{k,t}^i & \text{if } r_{kt} \leq \text{sigm}(\sigma'_{k,t} \times \text{Rand}^i()), \\ \bar{P}_{k,t}^i & \text{otherwise,} \end{cases} \quad (3.12)$$

where $P_{k,t}^i$ is the $ieth$ coordinate of the personal best guide $P_{k,t}$ of the $kieth$ particle at iteration step t , $\text{Rand}^i() \sim \text{Beta}(\alpha, \beta)$ is a beta distributed random number with parameters $\alpha, \beta \leq 1$. $\bar{P}_{k,t}^i$ is the opposite of $P_{k,t}^i$ value, i.e., if $P_{k,t}^i = 1$, then $\bar{P}_{k,t}^i = 0$ and *vice versa*. The parameter $\sigma'_{k,t}$ is defined by

$$\sigma'_{k,t} = \sigma_{k,t} \times \exp(\tau \times \mathcal{N}(0, 1) + \tau' \times \mathcal{N}^i(0, 1)),$$

where $\sigma_{k,t}$ is a uniform generated random number named the *strategy parameter*, \mathcal{N}^i means that a normal distributed random number is generated for each dimension coordinate; $\tau = \frac{1}{\sqrt{2K}}$ and $\tau' = \frac{1}{\sqrt{2\sqrt{K}}}$ with K is the population size.

Note that after the mutation operator applied, comparisons between the personal best guide $P_{k,t}$ and its corresponding muted particle $\mathcal{P}_{k,t}$ are made to determine if the new muted particle $\mathcal{P}_{k,t}$ dominates $P_{k,t}$ or not. $P_{k,t}$ is replaced only if its is dominated by its muted particle $\mathcal{P}_{k,t}$.

3.4.2. Crossover operator

In addition to the mutation operator and the elitist policy, the DMOPSO-A approach integrates a crossover operator. This approach, introduced in [34] and named DE-PSO, is formulated as a Differential Evolution algorithm. A target particle is fixed, then a new candidate particle is created by crossing other particles randomly selected in the swarm.

In this work, the crossover algorithm uses three particles selected such that they are pairwise different and all different to the target particle. If k denotes the target particle and k_1, k_2, k_3 the selected particles for crossover, the crossover operator is defined by

$$\tilde{x}_{k,t}^i = \begin{cases} x_{k_1,t}^i & \text{if } r_{kt} \leq \text{sigm} \left(F \times |x_{k_2,t}^i - x_{k_3,t}^i| \right), \\ \bar{x}_{k_1,t}^i & \text{otherwise,} \end{cases} \quad (3.13)$$

where $F \sim \mathcal{U}(0,1)$, $x_{k_1,t}^i$, $x_{k_2,t}^i$ and $x_{k_3,t}^i$ are the i th coordinates of the particles k_1, k_2, k_3 at iteration step t respectively; and $\tilde{x}_{k,t}^i$ is the i th coordinate of the candidate (crossed) particle \tilde{k} and $\bar{x}_{k_1,t}^i$ is the opposite bit of $x_{k_1,t}^i$ value. If the new created particle \tilde{k} dominates the target particle k , the candidate particle replaces the target one.

3.5. Adaptation of the DMOPSO-A for the multiobjective CVRP

Let us describe now the formulation of DMOPSO-A algorithm for the multiobjective CVRP. Scheduling vehicle routing is one of the key issues for a successful application of the DMOPSO-A algorithm to the multi-objective CVRP. The goal is how finding a suitable mapping from problem solution to DMOPSO-A particle. The following modelization has been presented in [4] for the Capacitated Vehicle Routing Problem.

Let us consider a problem of which N customers must be visited by M vehicles. We can generate a search space with particles positions vectors so that each particle length has $N \times M$ components. Each particle will contain M sections, and each section will be formed by N discrete points with values 0 or 1. If the value is 1 that means that the corresponding customer is served by the considered vehicle and if the value is 0 this will mean that the vehicle selected does not serve the corresponding customer. An example of this modelization can be set as in Figure 1.

In this example, eight customers must be served by two vehicles. And a possible vehicle routing problem instance (customer, vehicle) is generated. For the decoding process, each particle is represented as a matrix with two lines. For a VRP problem when N customers must be served by M vehicles, the first line of the matrix is composed by a vector $S = (s_1, s_2, \dots, s_{N \times M})$ with entries integer values arranged from 1 to $N \times M$, i.e., $s_l = l (l = 1, \dots, N \times M)$. The route construction is formed as follows:

- R_{ijk} represents a tensor where $k = 1, \dots, K$ is the candidate particle, i the vehicle and j the customer.
- $R_{ijk} = 1$ if the particle K the i th vehicle serves the j th customer and 0 otherwise.
- The vehicle number is computed from the position X by using the following equation

$$i = \left\lfloor \frac{s_l - 1}{N} \right\rfloor + 1, \quad (3.14)$$

where $\lfloor x \rfloor$ is the integer portion of x .

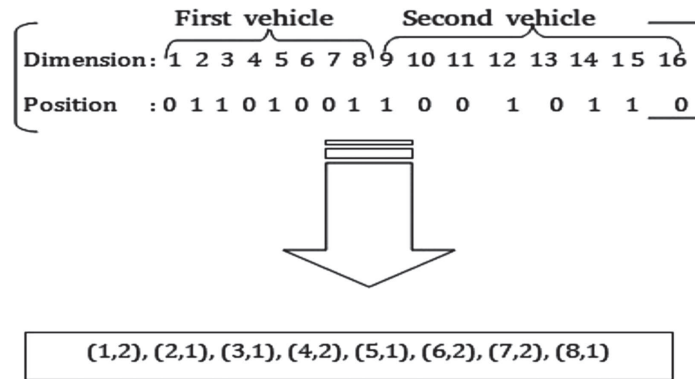


FIGURE 1. An example of mapping for the DMOPSO-A approach in the CVRP.

- The customer number is extracted using equation

$$j = s_l - N \times \left\lfloor \frac{s_l - 1}{N} \right\rfloor. \quad (3.15)$$

- Finally, the scheduled route is obtained by the value contained in the second line of the column l . If that value is 1, then the customer j is served by the vehicle i .

For handling the CVRP constraints, each customer has to be served exactly once by one vehicle, the total vehicle length must not exceed the maximal distance constraint and the total discharge of any route must not exceed the capacity of the vehicle. To check such constraints, the matrix R is used. For a given particle k_0 , each column (customer service) j is examined.

- If there exists only one line i such that $R_{ijk_0} = 1$, then the column (customer) j respects the serving constraints.
- If $R_{ijk_0} = 0$, $\forall i = 1, \dots, M$ (the customer j is not visited by any vehicle), a line i_0 is randomly pulled and the value is changed to $R_{i_0jk_0} = 1$.
- If there exists at least two indices (lines) i_1 and i_2 such that $R_{i_1jk_0} = R_{i_2jk_0} = 1$ (the customer j is served by at least two vehicles), a line i_0 is randomly chosen from those with values 1 and its value is maintained while all the others indices will be set to zero.
- If the total length in the route exceeds the limited value or the total successive discharge of each direct route exceeds the capacity of the vehicle, the solution is infeasible. For infeasible solutions, carry out the DMOPSO-A operation until the solutions become feasible.

4. NUMERICAL EXPERIMENTS

In this section, computational results are conducted in order to discuss the performance of the proposed algorithm. Some benchmarks from literature are selected and obtained results are compared to those proposed in literature. All the tests are carried on Matlab R2011b programs under Linux in a PC with AMD E1-1200 APU – 2 GB RAM.

4.1. Dell'Amico benchmark

The first computational experiment is conducted on the Vehicle Routing problem with Simultaneous Pick-up and Delivery (VRP-SPD) which is a particular case of the Capacitated Vehicle Routing Problem where the vehicle capacity must be revalued after each customer service. Each customer is affected to a demand to be

TABLE 1. Comparison of the DMOPSO-A best solution found by DMOPSO-A, PSO and the handled Dell’Amico *et al.* optimal solution.

Instance class	Average total cost		
	Dell’Amico [11]	The Jin Ai [1]	DMOPSO-A
Class 1	522.5	524.7	523.9
Class 2S	236 822.7	236 826.4	236 824.1
Class 2C	341 481.2	341 994.3	341 484.0
Class 3S	12 082.3	11 912.3	12 082.9
Class 3C	15 979.6	15 984.6	15 980.6

delivered and a given good to be picked. This is a single objective optimization problem where the only need is to minimize the routes cost of the fleet. For instance, the benchmark data set taken into account is recorded in the work of Dell’Amico *et al.* [11]. This problem comprises five classes of VRP-SPD instances. Each class composed of 20 or 40 customers consists of instances. Class 1 consists of 12 instances, while Class 2S, 2C, 3S, and 3C consist of 18 instances. The following problem parameters are affected in the proposed DMOPSO-A method: fixed cost per vehicle, $f = 0$; variable cost per distance unit, $g = 1$; maximal service duration $\mathcal{D}_j = \infty$; and the number of vehicles available is the number of vehicle in the optimal known solution. The computation test is run during 100 iterations, based on a mutation rate equal to 0.6, a crossover rate equal to 0.39, $c_1 = 0.75$, $c_2 = 1.2$, $c_3 = 2.6$. This problem has been considered in [1] where authors resolve it using a simple PSO algorithm. Their results are using here for comparison.

The comparison of the best solution among 100 iterations of the DMOPSO-A is presented in Table 1. The average total cost of the instances in each class is compared to the one found by Dell’Amico *et al.* [11]. It has been mentioned that Dell’Amico *et al.* [11] provided the optimal solution of 75 out of 84 instances and the upper bound represents the optimal solution found. Hence, results in Table 1 demonstrate that DMOPSO-A approach can be able to produce high quality solutions that are very close to the best solution found. We can also remark that only for the 3S class, The Jin approach [1] produces a best solution compared to the proposed approach. In addition, the solutions can be reached in a few computational iterations and a short computation time.

4.2. Dethloff test problem

Next, we consider another computational experiment that is based of the benchmark problem presented by Dethloff [12]. This is composed by four data sets named SCA3, SCA8, CON3 and CON8, which consist of 10 instances of a problem of 50 customers. Each one is different of another one by specific characteristics: SCA data are composed by customers distributed uniformly in the service area while the CON problems are generated with half of costumers located uniformly in the service region and the other half are situated in given part of the service affectation. The number after SCA or CON indicated the parameter for determining vehicle capacity.

This VRPSPD is an multiobjective optimization problem where the goal is to minimize the total traveled distance by maximizing the capacity constraint of the vehicle. Therefore, the problem parameters are set as follows: the cost per vehicle is fixed, and $f = 0$; the variable cost per distance unit is $g = 1$; there is no constraint on service duration, so $D = \infty$; and the number of available vehicles is equal to the number of available vehicle in the best known solution.

To compare our approach to those found in the literature, we collect the best solution among 10 iterations based on a mutation rate equal to 0.5, a crossover rate equal to 0.25, $c_1 = 0.58$, $c_2 = 1.5$, $c_3 = 2.4$. The found solutions are compared to those obtained by The Jin [1] using PSO, Tang and Galvao [32] using a Tabu search approach, Bianchessi and Righini [3] by using an heuristic algorithms and the reference solutions in [12]. The overage results of the numerical results over the 10 iterations of each data set are presented in Table 2. The presented results show that the proposed DMOPSO-A performs best results compared the

TABLE 2. Comparison of the 10 iterations overage solution optimal solutions found by DMOPSO-A and others published methods on the Dethloff benchmark.

Set	Average total cost				
	Dethloff [12]	Tang [32]	Bianchessi [3]	The Jin [1]	DMOPSO-A
SCA3	746.6	674.2	684.6	675.8	679.5
SCA8	1166.4	1044.4	1035.7	1041.8	1043.4
CON3	597.3	564.2	568.5	569.6	575.2
CON8	860.6	774.3	776.4	798.3	801.6

TABLE 3. Computational comparison between the benchmark of the Salomon's RC instances and the current results.

Instance	Best known solution		Ghoseiri [14]		Proposed DMOPSO-A Approach			
			Average of solutions in 10 runs		Best solution during the 10 runs		Average of solutions in 10 runs	
	# Vehicles	Distance	# Vehicles	Distance	# Vehicles	Distance	# Vehicles	Distance
RC101	15	1636.92	15.3	1693.2	15.4	1689.5	15	1696.99
RC102	13	1470.26	14.5	1521.0	13.5	1509.1	14	1554.77
RC103	11	1261.67	12.2	1357.4	13	1250.8	12	1261.66
RC104	10	1135.48	11	1213.5	11	1120.4	10	1135.47
RC105	16	1590.25	15.9	1610.5	16	1615.7	16	1629.44
RC106	11	1427.13	13.5	1437.1	12	1422.1	12	1424.75
RC107	11	1230.48	12.2	1287.9	11	1223.1	11	1230.50
RC108	10	1142.66	11.3	1197.9	11	1136.5	11	1139.83
RC201	6	1134.91	4	1457.0	5	1401.2	6	1406.98
RC202	4	1181.99	4	1381.9	5	1364.3	5	1365.71
RC203	4	1026.61	4.9	1196.7	4	1056.9	4	1057.99
RC204	3	798.46	3	926.74	4	795.54	3	798.76
RC205	4	1300.25	4	1411.3	4	1300.5	4	1298.01
RC206	3	1153.93	4	1195.5	3	1144.1	3	1146.52
RC207	4	1040.67	4	1070.3	4	1043.4	4	1061.15
RC208	4	785.93	3.7	905.07	4	828.40	4	828.70

others approaches considered for comparison. Moreover, the proposed MOPSO based approach achieves best results in less computational time. Once more, the considered test problem prove that the proposed approach is competitive and improve solution for this kind of benchmarks.

4.3. Solomon's problem

Next, we consider the Solomon's problem which consists of a vehicle routing problem with time windows. The problem is defined as a bi-objective vehicle routing problem with route balancing with 25, 50 and 100 customers. These Solomon's instances are generally used as references in operation research. These are devised into three categories [30, 31, 39]: category C plus a number of instances where customers are distributed in a cluster, category R plus a number of instances where customers are randomly distributed, and category RC plus a number of instances where the repartition of customers is handled so that distribution is between random and clusters. This problem is solved in [14] using goal programming and genetic algorithm. Their obtained results are used in the current work for comparison. In this example, only the last category with 100 customers is taken

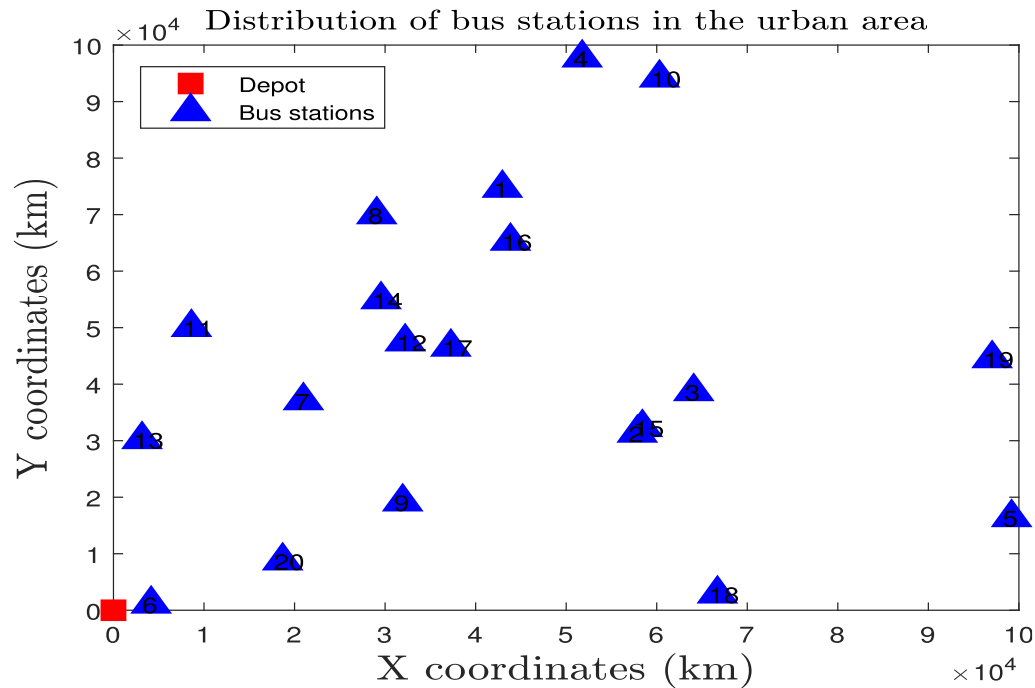


FIGURE 2. Geographical location of the 20 bus stations from the depot.

TABLE 4. Longitudinal and latitudinal positions of the 20 bus stations from the depot.

Station number	1	2	3	4	5	6	7	8	9	10
<i>x</i> coordinate (km)	42 954	57 855	64 085	51 767	99 193	4160	20 978	29 074	31 946	60 305
<i>y</i> coordinate (km)	74 720	31 390	38 737	97 777	16 444	1146	37 130	70 034	19 222	94 225
Station number	11	12	13	14	15	16	17	18	19	20
<i>x</i> coordinate (km)	8611	32 214	3157	29 549	58 417	43 846	37 265	66 703	97 045	18 686
<i>y</i> coordinate (km)	50 092	47 538	30 251	54 998	32 380	65 350	46 608	2932	44 565	8785

TABLE 5. The delivery, pick-up and service duration parameters for the stations.

Station	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Deliveries	4	5	3	5	2	3	3	3	5	4	4	4	3	3	3	2	4	1	2	4
Pick-up	1	0	1	2	1	2	1	2	2	1	1	2	2	0	2	2	2	1	3	1
Service (min)	3	2	2	3	3	2	3	2	2	2	3	2	2	3	2	3	2	2	2	3

TABLE 6. The fleet capacities and maximal service duration.

Vehicle	1	2	3	4	5	6	7	8	9	10
Maximal capacity	27	34	30	38	48	33	30	42	42	50
Maximal service duration (min)	187	190	119	192	186	132	177	150	182	135

into account. The computation test is run during 10 iterations, based on a population size settled to be equal to 150, a mutation rate equal to 0.5, a crossover rate equal to 0.35, $c_1 = 0.75$, $c_2 = 1.45$, $c_3 = 1.85$.

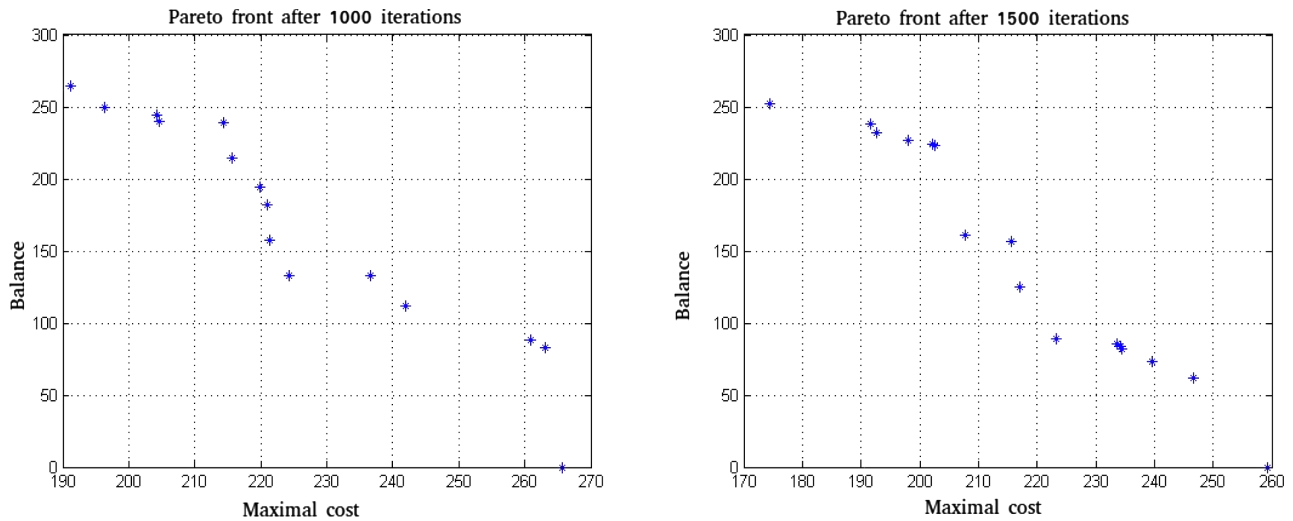


FIGURE 3. Compromise surface of the studied problem after 1000 iterations (*left*) and 1500 iterations (*right*).

TABLE 7. Compromise surface of the studied problem after 500, 1000 and 1500 iterations.

500		1000		1500	
Cost	Balance	Cost	Balance	Cost	Balance
267.7704	0	263.0799	82.8680	246.6556	62.1773
258.0097	151.3161	241.9217	111.6078	207.7286	160.9848
227.8329	186.9130	221.3454	157.5654	202.2095	224.1326
220.3364	258.3727	215.6187	214.8524	217.0931	125.2366
243.2008	164.8209	260.8721	88.5379	223.2417	88.6984
254.7722	151.8112	224.3150	133.3050	259.2252	0
208.3095	271.0049	236.5776	132.5957	215.7201	156.1479
262.6538	111.2472	204.2400	143.8645	191.6524	238.3541
		221.0789	182.4075	207.7286	160.9848
		219.8852	194.5006	239.5300	73.6239
		214.3199	239.0917	192.5655	232.2075
		191.2347	264.8391	198.0644	226.8551
		204.6563	239.5708	233.5968	85.9236
		265.7499	0	234.3565	82.3965
		196.3839	249.3279	234.3351	83.8069
				174.3803	251.8756
				202.5823	222.9275

Experimental results are reported in Table 3 where the first column presents the instance name of the considered Solomon's benchmark [30], the second and third columns point to the best solution (number of vehicles and distance cost) known from literature, the fourth and fifth columns reproduces the average solutions from Ghoseiri's results published in [14]. Finally, the computed best optimal solutions by the current DMOPSO-A approach are listed in the sixth and seventh columns while the eighth and ninth columns represent the average solutions for the 10 runs. The best solution known from literature is taken according to those considered in [14].

TABLE 8. Two examples of bus orientation found in the nondominated solutions.

Bus	Particle 2					Particle 14				
1	0	11	0	—	—	0	11	0	—	—
2	0	12	2	10	0	0	7	10	0	—
3	0	9	19	0	—	0	4	9	0	—
4	0	17	0	—	—	0	14	17	0	—
5	0	7	3	13	0	0	1	5	0	—
6	0	8	14	0	—	0	12	13	19	0
7	0	4	5	0	—	0	6	20	0	—
8	0	15	16	0	—	0	8	2	15	18
9	0	18	0	—	—	0	3	0	—	—
10	0	1	6	20	0	0	16	0	—	—

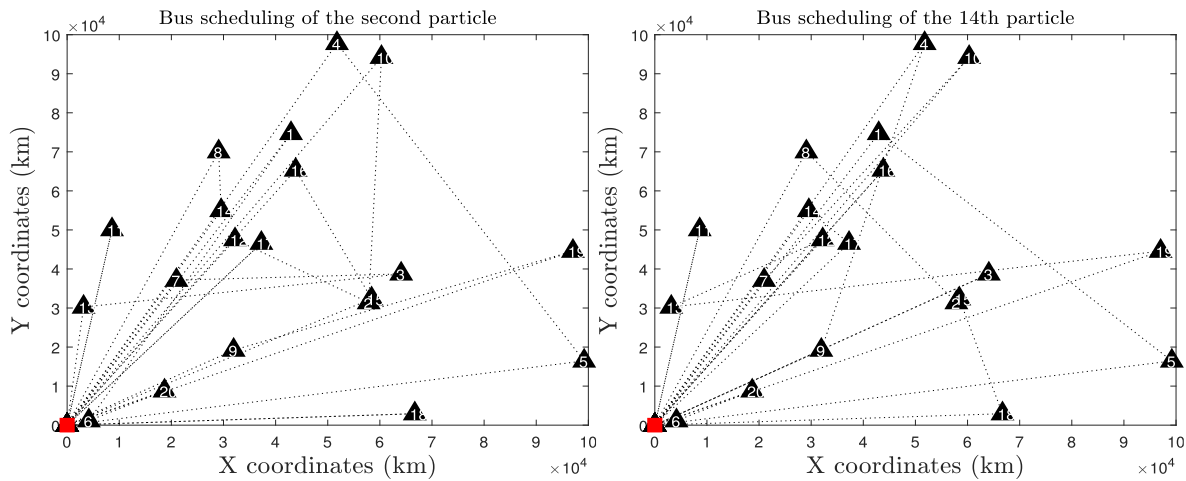


FIGURE 4. Route balance of the selected nodominated solutions in Table 8.

These results show that the use of the proposed approach to solve this biobjective optimization problem for the vehicle routing problem based on the Solomon's benchmark is more interesting and competitive compared to results in the literature. This approach lead us to get interesting route balance optimal solutions without considerably affecting the optimal distance cost. We can therefore conclude that the proposed DMOPSO-A approach is adapted to solve VRP with simultaneous pickup and deliveries or with time windows.

4.4. Urban school bus routing with route balancing

Finally, we consider an urban school bus routing problem with route balancing. The problem consists of affecting a set of buses to serve a given number of stations to deposit and/or retrieve schoolboys and schoolgirls. In the real-life process, the number of students to be picked or deposed at a given station is stochastic and cannot be fixed *a priori*. But, in this example, for reasons of simplicity and as preliminary example, this number is considered to be *a priori* known. This problem is one of the famous applications of the Vehicle Routing Problem with Simultaneous Pick-up and Delivery (VRP-SPD). Because pupils must not arrive late to their school/home, this problem is also one of the applications of the vehicle routing problem with time windows. To take into account the time windows constraint, we use here the route balancing objective. This will lead vehicles to minimize routes circuits and minimize students waiting time in the station and inside the bus. Each

bus must return to the depot after the end of service. In this current example, we do not consider the problem of stochastic route traffic which may change the route travel time.

In this example, 20 stations have to be served by at most 10 school buses. To facilitate the modeling, the geographic distribution of the stations must be known. The stations distribution is depicted in Figure 2.

For each station j , there are assigned the longitude and latitude localisation as a two-dimensional orthonormal positions, a number of students to be deposited, a number of students to be picked and a service time duration s_j in minutes. The initial delivery number of students are considered to pick-up into the bus from the depot (bus terminal). These are summarized in Tables 4 and 5.

The fleet of buses is considered to be inhomogeneous. So, each vehicle has its own maximal transport capacity and its own maximal service duration. These characteristics are summarized in Table 6.

For comparison, simulations are made for the proposed DMOPSO-A approach. The swarm population contains 20 particles, the mutation rate is equal to 0.75, the crossover rate is set to 0.6; $c_1 = 0.48$ $c_2 = 0.9$ and $c_3 = 2.0$. The code is run up to 500, 1000 and 1500 iterations. Next, Pareto front results corresponding to 1000 and 1500 iterations are plotted in Figure 3.

The compromise solutions are summarized in the Table 7 where the first column of each class corresponds to the total cost of a route while the second column is the route equity (the route balance).

Two examples of nondominated particle solutions reported in Table 8 and prove that each route starts from the depot and finishes on the depot; and each station is visited once by exactly one bus. The majority of the buses have almost the same number of stations to serve. This last remark is in concordance of the route balancing desire. These selected two routes balances are depicted in Figure 4.

5. CONCLUSION

The Discrete Multiobjective Particle Swarm approach, based on a probability of particle bit update, local attractors, crowding distance, and genetic operators; has been used in this work to solve vehicle routing problems. The performance of this new approach over all the benchmark problems studied turned out to be satisfactory in the sense that the aim was to determine if the new algorithm was able to obtain results close to those underlined in the literature.

Numerical results are shown for three well-known test examples on vehicle routing problems and compared to results from the literature. The obtained results demonstrate that, in spite of its simplicity, handling and flexibility, the DMOPSO-A is a promising approach to multiobjective optimization because its performance was generally better and encouraging. Further work will consider stochastic vehicle routing problems such as Urban School Bus routing without any *a priori* knowledge of number of students to be picked and/or deposited at a given bus station and by taking into account stochastic traffic flow.

Acknowledgements. The authors wish to thank the anonymous referees for their valuable comments that lead us to improve the quality of the present work.

REFERENCES

- [1] T.J. Ai and V. Kachitvichyanukul, A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.* **36** (2009) 1693–1702.
- [2] J.S Angelo and H.J.C. Barbosa, On Ant Colony Algorithms for Multiobjective Optimization, chap.5 in: *Ant Colony Optimization – Methods and Applications*, edited by A. Ostfeld. InTech (2011) 53–74.
- [3] N. Bianchessi and G. Righini, Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Comput. Oper. Res.* **34** (2007) 578–94.
- [4] A.-L. Chen, G.-K. Yang and Z.-M. Wu, Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *J. Zhejiang Univ. Sci. A* **7** (2006) 607–614.
- [5] W.-N. Chen, J. Zhang, H.S.H. Chung, W.-L. Zhong, W.-G. Wu and Y.-H. Shi, A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Trans. Evol. Comput.* **14** (2010) 278–300.
- [6] M. Clerc, Discrete particle swarm optimization, illustrated by the traveling salesman problem. In: *New Optimization Techniques in Engineering*. Vol. 141 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin-Heidelberg (2004) 219–239.

- [7] C.A. Coello Coello and M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization. In: Vol. 2 of Congress on Evolutionary Computation. CEC'02 (2002) 1051–1056 DOI: [10.1109/CEC.2002.1004388](https://doi.org/10.1109/CEC.2002.1004388).
- [8] G.B. Dantzig and J.H. Ramser, The truck dispatching problem. *Manage. Sci.* **6** (1959) 80–91.
- [9] J. de Armas and B. Melián-Batista, Variable neighborhood search for a dynamic rich vehicle routing problem with time windows. *Comput. Ind. Eng.* **85** (2015) 120–131.
- [10] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Parallel Problem Solving from Nature VI Conference (2000) 849–858.
- [11] M. Dell'Amico, G. Righini and M. Salani, A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transp. Sci.* **40** (2006) 235–247.
- [12] J. Dethloff, Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR Spectr.* **23** (2001) 79–96.
- [13] J. Euchi and R. Mrahihi, The urban bus routing problem in the Tunisian case by the hybrid artificial ant colony algorithm. *Swarm Evol. Comput.* **2** (2012) 15–24.
- [14] K. Ghoseiri and S.F. Ghannadpour, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Appl. Soft Comput.* **10** (2010) 1096–1107.
- [15] A. Halassi, An attractor-based multiobjective particle swarm optimization. *Int. J. Appl. Comput. Math.* **3** (2017) 1019–1036.
- [16] K.N. Ishibuchi, N. Tsukamoto and N. Nojima, An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. *Eur. J. Oper. Res.* **188** (2008) 57–75.
- [17] N. Jozefowiez, F. Semet and E.G. Talbi, Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **189** (2008) 296–309.
- [18] J. Kennedy and R. Eberhart, Particle swarm optimization. In: Proceedings of the Fourth IEEE International Conference on Neural Networks, Perth, Australia (1995).
- [19] J. Kennedy and R.C. Eberhart, A discrete binary version of the particle swarm algorithm. In: Vol. 5 of 1997 IEEE International Conference on Systems, Man, and Cybernetics. *Computational Cybernetics and Simulation* (1997) 4104–4108.
- [20] J. Knowles and D. Corne, Approximating the nondominated front using the Pareto archived evolution strategy. *Evol. Comput.* **8** (2000) 149–172.
- [21] M.F. Leung, S.C. Ng, C.C. Cheung and A.K. Lui, A new strategy for finding good local guides in MOPSO. In 2014 IEEE Congress on Evolutionary Computation (CEC) (2014) 1990–1997.
- [22] X. Li, A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: Genetic and Evolutionary Computation-GECCO 2003. Vol. 2723 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg (2003) 37–48.
- [23] W.-X. Li, Q. Zhou, Y. Zhu and F. Pan, An improved MOPSO with a crowding distance based external archive maintenance strategy, edited by Y. Tan, Y. Shi and Z. Ji, In: Vol. 7331 of *ICSI*. Springer (2012) 74–82.
- [24] Y. Marinakis, M. Marinaki and G. Dounias, A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Eng. App. Artif. Intell.* **23** (2010) 463–472.
- [25] S.A. MirHassani and N. Abolghasemi, A particle swarm optimization algorithm for open vehicle routing problem. *Expert Syst. App.* **38** (2011) 11547–11551.
- [26] C. Özkale and A. Fiğlali, Evaluation of the multiobjective ant colony algorithm performances on biobjective quadratic assignment problems. *Appl. Math. Modell.* **37** (2013) 7822–7838.
- [27] C.R. Raquel and P.C. Naval Jr, An effective use of crowding distance in multiobjective particle swarm optimization. In: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, GECCO'05. ACM (2005) 257–264.
- [28] R. Roy, S. Dehuri and S.B. Cho, A novel particle swarm optimization algorithm for multi-objective combinatorial optimization problem. *Int. J. Appl. Metaheuristic Comput. (IJAMC)* **2** (2012) 41–57.
- [29] P. Schittekat, J. Kinable, K. Sorensen, M. Sevaux, F. Spieksma and J. Springael, A metaheuristic for the school bus routing problem with bus stop selection. *Eur. J. Oper. Res.* **229** (2013) 518–528.
- [30] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35** (1987) 254–265.
- [31] 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. App.* **34** (2006) 115–151.
- [32] F.A. Tang and R.D. Galvao, A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput. Oper. Res.* **33** (2006) 595–619.
- [33] A.S. Tasan and M. Gen, A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Comput. Ind. Eng.* **62** (2012) 755–761.
- [34] R. Thangaraj, M. Pant, A. Abraham and P. Bouvry, Particle swarm optimization: hybridization perspectives and experimental illustrations. *Appl. Math. Comput.* **217** (2011) 5208–5226.
- [35] P. Toth and D. Vigo, editors. The Vehicle Routing Problem (Monographs on Discrete Mathematics and Applications). SIAM (2001).
- [36] L. Wang, X. Wang, J. Fu and L. Zhen, A novel probability binary particle swarm optimization algorithm and its application. *J. Softw.* **3** (2008) 28–35.
- [37] C. Wang, D. Mu, F. Zhao and J.W. Sutherland, A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows. *Comput. Ind. Eng.* **8** (2015) 111–122.
- [38] L.A. Wolsey, Integer Programming. John Wiley & Sons (1998).
- [39] W. Zhou, T. Song, F. He and X. Liu, Multiobjective vehicle routing problem with route balance based on genetic algorithm. *Discrete Dyn. Nat. Soc.* **2013** (2013) 1–9.

- [40] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm, edited by K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou and T. Fogarty. In: EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems (2000).

Subscribe to Open (S2O)

A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/maths-s2o-programme>