



Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

<http://scientiairanica.sharif.edu>



Improved multi-ant-colony algorithm for solving multi-objective vehicle routing problems

R.K. Goel^{a,b,*} and R. Maini^a

a. Department of Computer Engineering, Punjabi University, Patiala, India (147002).

b. Department of Computer Science, Government College, Naraingarh, India (134101).

Received 27 September 2018; received in revised form 17 June 2019; accepted 29 July 2019

KEYWORDS

Ant colony system;
Firefly optimization;
Load balancing;
Multi-objective
optimization;
Vehicle routing
problem.

Abstract. Classical Vehicle Routing Problems (VRP) involve the supply of goods/services from a central depot to geographically scattered customers. Besides the classical objective of minimizing the total traveled distance, the present work also considers simultaneous optimization of two additional objectives namely minimizing makespan and minimizing distance imbalance. A mathematical model has been developed to deal with this multi-objective version of VRP (MO-VRPTW). A meta-heuristic based on multiple ant colony systems for solving this MO-VRPTW has also been proposed. Firefly optimization Algorithm (FA) has also been applied to avoid local optima. Two new migration operators named Migration-I and Migration-II and multi-pheromone matrices have been developed to further improve the solution quality. The proposed algorithm has been tested on a number of benchmark problems and its superiority over other state-of-the-art approaches and Non Dominated Sorting Algorithm-II (NSGA-II) is demonstrated.

© 2021 Sharif University of Technology. All rights reserved.

1. Introduction

Vehicle Routing Problem (VRP) is a well-known classical combinatorial optimization problem in transportation logistics and supply chain management. It is concerned with the supply of goods/services from a central distribution/collection center to a finite number of geographically dispersed customers efficiently and economically. As the cost and time of transportation of goods have a direct influence on business and industry, VRP has gained importance amongst the researchers to produce business-enterprise economical solutions.

Furthermore, global warming, traffic jams, road congestion, and other issues such as depleting natural resources have motivated researchers to search for efficient routing strategies for supply chain activities.

Graphically, VRP can be formulated as an undirected graph $G(V, E)$, where the vertex set $N = \{0, 1, 2, 3, \dots, n\}$ is the set of nodes and $E = \{(i, j) | i, j \in V, i \neq j\}$ is the set representing the link between nodes (i, j) . Node 0 is considered as the central depot where a fleet of homogeneous vehicles (each with identical capacity Q) are available to serve customers $(N/\{0\})$, each having a fixed demand q_i . Each arc (i, j) is associated with a fixed symmetric travel cost d_{ij} and travel time t_{ij} satisfying triangle inequality, i.e., $d_{ij} \leq d_{ik} + d_{kj} \forall i, j, k \in N$.

The objective of VRP is to find an optimal set of routes with the optimal sequence of customers satisfying the following constraints:

*. Corresponding author.

E-mail addresses: rcse123@gmail.com (R.K. Goel);
research_raman@yahoo.com (R. Maini)

- Each route must start from the central depot and end there;
- Each of the customers must be fully serviced by a single vehicle in a single visit;
- Demand of any customer does not exceed the vehicle capacity.

Most of the real-life models of VRP primarily aim at optimizing a single objective of VRP such as minimizing the total traveled distance or minimizing the required fleet size. However, supply chain management/logistic management and other transportation industries mostly encounter multi-objective scenarios of minimizing simultaneously the number of vehicles as well as total traveled distance. Multi-objective Optimization Problems (MOP) are mainly concerned with the optimization of more than one objective simultaneously under the same set of conditions. The introduction of additional objectives to be optimized simultaneously further enhances the utility of classically defined a single-objective optimization problem [1].

Much of the available literature on the subject deals with optimization of single-objective functions such as minimizing the total traveled distance or minimizing the overall fleet size required. In fact, most of the companies/enterprises try to not only minimize the primary objective of total traveled distance but also focus on balancing the individual route lengths (i.e., makespan), number of customers covered by a single vehicle and balancing of working hours of the drivers, etc. Such types of problems result in MOP with conflicting objectives.

In this paper, firstly, a new mathematical model has been developed to tackle the MO-VRPTW considering the minimization of total traveled distance, length of the longest route (makespan), and driver's load imbalance (in terms of distance traveled) simultaneously. Another goal of the proposed work is to develop an algorithm that can solve the developed model efficiently and effectively.

In order to achieve this goal, an improved Ant Colony System (ACS) based multi-objective optimization algorithm has been proposed to optimally solve the proposed Multi-Objective Vehicle Routing Problem (MO-VRP) with time windows. The proposed Multiple ACS (named MACS) can enhance the basic ACS in the following three aspects:

- Multiple pheromone matrices have been introduced to fit MACS for solving different objective functions of the proposed MO-VRPTW;
- Proposed ACS has been hybridized with Firefly optimization Algorithm (FA) to avoid being trapped into local optima;
- Furthermore, two new migration operators have been proposed to improve the obtained solutions.

The rest of the paper is organized as follows. Literature related to MO-VRPs is presented in Section 2. The mathematical formulation of the present multi-objective VRPTW is presented in Section 3. Section 4 describes the details of the proposed algorithm. Experimentation details and results are presented in Section 5. Conclusions based on the present study are finally drawn in Section 6.

2. Literature review

The literature currently available on multi-objective VRP is generally considered Capacitated Vehicle Routing Problem (CVRP) without time windows constraint. Borgulya [2] presented a heuristic algorithm to minimize the total traveled distance as well as to balance the route length of vehicles. Similarly, Jozefowicz et al. [3] proposed an evolutionary algorithm along with an elitist diversification-based metaheuristic to simultaneously reduce the total length of routes as well as optimize route balancing. Kritikos and Ioannou [4] worked towards balancing of load carried by each vehicle. Recently, Zhou et al. [5] proposed a genetic algorithm-based approach for bi-objective CVRP where they considered the total traveled distance as well as distance balance of active vehicles simultaneously. Melián-Batista et al. [6] designed two scatter search-based algorithms and tested them for the bi-objective VRPTW. They tackled a real-life problem presented by a Spain-based company to minimize the total tour length as well as try to balance time spent by each of the drivers on duty. Ombuki et al. [7] presented a hybrid meta-heuristic to solve MO-VRPTW. Genetic Algorithm (GA) and Tabu search were employed to optimize the number of vehicles and Travelled Distance (TD), respectively. Jozefowicz et al. [8] thoroughly discussed the potential benefits of using MO-VRPs. In 2011, Garcia-Najera and Bullinaria [9] proposed an improved multi-objective evolutionary algorithm for MO-VRPTW. They proposed similarity measure to improve the diversity and quality of the Pareto optimal front solution sets. Ten different types of mutation operators were defined and tested and the superiority of the proposed improved evolutionary algorithm was shown on a standard bench mark set while considering bi-objective and tri-objective formulations. Recently, Dong et al. [10] presented a tissue p-system based on discrete glowworm evolution mechanism and variable neighborhood evolution mechanism to solve bi-objective VRPTW involving minimization of the fleet size primarily and total traveled distance. They were able to find some new Pareto front optimal solution sets for small-scale instances and provide comparable results for large benchmark problems. Additional information on recent methods dealing with MO-VRPTW is available in [1,11–14].

Much of the literature considered above mainly emphasizes the minimization of traveled distance and/or fleet size required to complete the task. However, besides the minimization of transportation cost, real-life VRP also deals with some other additional objectives like customer satisfaction, workload balancing (in terms of time and distance), etc. For instance, Norwegian distribution company “Innovation AS” provides routing services for newspaper distribution in large parts of the country. Its main goal is to find well-balanced routes with a maximum of 20% imbalance. Hasle [15] and Najafi et al. [16] solved MO-VRP for designing a robust evacuation plan in case of an earthquake. They proposed a hierarchical multi-objective optimization model including the minimization of waiting time for the medical treatment of injured persons as well as minimization of the response time for the supply of relief materials with the least number of emergency vehicles. Zhou and Wang [17] defined the workload imbalance as the difference between the load carried out by maximum utilized and minimum utilized vehicles. Distance imbalance directly influences the time traveled by the particular vehicle. Lacomme et al. [18] claimed that in the waste collection problem, the collection should be finished as soon as possible so that the remaining time can be utilized to categorize the collected waste material. Furthermore, the delivery of perishable food items, emergency services, and disaster evacuation strategies require the shortest travel time. Most of the works in the field of MO-VRP focused on managerial benefits, ignoring customer satisfaction and driver’s work distribution. However, such factors greatly influence the market share of the companies. For example, late delivery of items due to longer makespan makes customers unsatisfied and may cause loss of future sales. Additionally, the workload imbalance may create differences among the lorry drivers which results in lower throughput. Hence, such real-life situations need consideration of multi-objective routing plans involving the total traveled distance, minimizing the length of the longest route (i.e., makespan), as well as balancing traveled route lengths of each vehicle.

Like VRP, MO-VRP is also an NP-hard problem. Exact solution algorithms are possible for problems with small instances only. Moreover, in an MOP, more than one objective is to be combined, which cannot be separated or solved sequentially. As a result, exact methods are seldom available for such problems (Dong et al., 2017 [10]). Therefore, in recent years, meta-heuristic-based approaches have been adopted to solve MO-VRP. Evolutionary algorithms can approximate the whole Pareto front. As a result, a large number of studies have used evolutionary algorithms to deal with MO-VRPs [9,13]. However, evolutionary algorithms suffer from the problem of being trapped

in local optima and a significant computational effort is generally required to escape from being trapped in local optima. Besides, EAs [5] proposed a local search-based optimization technique namely LSMO-VRPTW. They handled the multi-objective problem sequentially by optimizing different objectives, one at a time. However, in such a case, there may be chances of distortion of objectives that are already optimized. Some other recent papers in the literature on MOVRPs are tabulated, as given in Table 1.

Interested readers may also refer to [27] for a detailed survey on MO-VRPs.

No doubt that MO-VRPTW is still a less explored topic in the field of MO-VRPs. Our MO-VRPTW aims to minimize total tour length (from an economical perspective), minimizing makespan (for early delivery) and route imbalance (from the social perspective of fair distribution of work among drivers).

3. Mathematical formulation

Before formulating the mathematical model of multi-objective VRPTW, we first briefly describe the classical VRPTW model.

VRPTW is an extension of VRP with additional constraints of time windows associated with each of the customers. This study assumes the case of hard time windows where customers must be serviced within a specified time window. If a vehicle arrives earlier than the starting/opening time of the customer’s window, then it has to wait till the opening of the window. However, if a vehicle arrives after the closing time of the window, then this customer cannot be served at all by this vehicle.

The set of notations used in the mathematical formulation is presented in Table 2. With these notations, the mathematical model of VRPTW is formulated as follows:

$$\min f_1 = \sum_{k \in K} \sum_{i,j \in V} d_{ij} X_{ij}^k, \quad (1)$$

subjected to:

$$\sum_{k \in K} \sum_{j \in V} X_{ij}^k = 1, \quad \forall i = N/\{0\}, \quad (2)$$

$$\sum_{j=1}^N X_{0j}^k = 1, \quad \forall k \in K, \quad (3)$$

$$\sum_{i=1}^N X_{i0}^k = 1, \quad \forall k \in K, \quad (4)$$

$$\sum_{i=1}^N X_{ih}^k - \sum_{j=1}^N X_{hj}^k = 0, \quad \forall h = \frac{N}{\{0\}}, \quad \forall k \in K, \quad (5)$$

Table 1. Recent literature on MOVRP.

Author/s	Problem	Objectives	Approach	Metric	Data set
[19]	DVRPTW with fuzziness	TD, total travel time, max customer, preference, NV	GA	–	Solomon
[20]	MOVRPTW	TD, NV	EA	Coverage	Solomon
[17]	MOVRPTW	TD, makespan, total waiting time, NV	LS	IGD, HV, coverage metric	Real life
[21]	MDVRP	Reachability time, profit, satisfaction level, NV	PSO	–	Real life
[22]	MOVRP	TD, NV	EA	HV	Solomon
[23]	MOVRPTW	Customer satisfaction, energy consumption NV	EA	–	Random
[24]	MOVRP	Total travel time, total reliability	FA	Spacing metric	Real world
[25]	MOMDVRPTW	TD, NV, makespan, total waiting time and total delay time	EA	HV	Real world
[26]	MOVRPTW	Total travel time, servicing level	Memetic algorithm	–	Modified solomon

Table 2. Mathematical notations used in MO-VRPTW.

Symbol used	Description
X_{ij}^k	Binary variable (0/1) equals 1 if edge $(i - j)$ is followed by vehicle k , otherwise 0
d_{ij}	Distance between nodes i and j
K	Maximum number of vehicles
Q	Maximum capacity of each vehicle
q_i	Demand of customer i
s_i	Service time at customer i
w_i	Waiting time at customer i
T_i^k	Service start time for customer i by vehicle k
$[e_i, l_i]$	Earliest start time and latest start time specified for customer i

$$\sum_{i=1}^N q_i \sum_{j=1}^N X_{ij}^k \leq Q, \quad \forall k \in K, \quad (6)$$

$$T_i^k + s_i + t_{ij} + w_j \leq T_j^k, \quad \forall i, j = N/\{0\}, \quad \forall k \in K, \quad (7)$$

$$e_i \leq T_i^k \leq l_i, \quad \forall i, j = N/\{0\}, \quad \forall k \in K, \quad (8)$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall k \in K, \quad \forall i, j \in N. \quad (9)$$

Objective Function (1) seeks to minimize the total traveled distance. Constraint (2) specifies that every customer is visited by exactly one vehicle and splitting of deliveries is forbidden. Constraints (3) and (4) ensure that all routes start and end at the central depot, respectively. Constraint (5) ensures that the vehicle has to leave the customer's location being serviced after visiting it. Inequality (6) restricts the violation of capacity constraints such that total serving on the route must be less than or at most equal to the available capacity of the vehicle. Eqs. (7) and (8)

preserve the time window constraints by checking that the sum of traveling times to the customer, service time, and waiting time, if any, are less than the closing time of the customer's time window. Finally, partial or split deliveries are prohibited by Eq. (9).

The literature indicates that MO-VRPTW is mostly formulated as a bi-objective VRPTW, working towards the minimization of total fleet size as well as total traveled distance simultaneously. In this paper, along with these objectives, an attempt is made to solve the VRP with hard time windows, minimizing makespan as well as the distance imbalance at the same time. The latter two objectives gain insight from the social perspective and guarantee a fair-share scheduling of the work. These objectives can be mathematically formulated as follows:

$$f_2 = \min \left(\max_{k \in K} (\text{makespan}_k) \right)$$

$$\text{where } \text{makespan}_k = \sum_{i,j \in V} d_{ij} X_{ij}^k, \quad (10)$$

$$f_3 = \left(\max_{k \in K} (\text{makespan}_k) - \min_{k \in K} (\text{makespan}_k) \right) / \max_{k \in K} (\text{makespan}_k). \quad (11)$$

However, the objective function f_2 as given in Eq. (10) corresponds to the minimization of makespan, i.e., length of the longest route, f_3 , given in Eq. (11) defines the distance imbalance among the distances covered by vehicles to complete the delivery. Makespan can directly affect the cost of transportation. For example, as mentioned in [28], in the real-life scenario of school bus routing where certain route lengths far exceed the average route due to large distances between the pickup points. As a result, there are fewer chances of using the full capacity of the bus, in case when most of customers belong to the last locations. It may also provide late servicing for end customers. f_2 is mainly associated with timely delivery and satisfaction of customers [27].

An MOP can be mathematically formulated as follows:

$$\min f(x) = \{f_1(x), f_2(x), \dots, f_n(x)\}. \quad (11a)$$

Here, $f_i(x)$ ($i = 1, 2, \dots, n$) are the objective functions to be optimized simultaneously in a specified set of constraints. MOP attempts to find a set of non-dominated solutions called Pareto Sets (PS). The dominance among the solution sets is defined as follows:

Definition 1: A solution set $A = (a_1, a_2, a_3, \dots, a_n)$ dominates another solution set $B = (b_1, b_2, b_3, \dots, b_n)$ iff $\forall i \ b_i \leq a_i$ and $\exists i \in (1, 2, \dots, n)$ s.t. $b_i < a_i$, solution

set A must be better than set B in at least one objective and not worse in others. If $b_i = a_i \forall i$, then solutions are indifferent, i.e., incomparable. Furthermore, a solution set Z is said to be Pareto optimal if there does not exist any other solution Z' such that Z' dominates Z .

Solution methods for dealing with MOP are broadly categorized into two groups: (i) a priori approach and (ii) a posteriori approach [8]. In a priori approach, a preference index is associated with each of the objectives under consideration. Then, all of the objectives are aggregated into a single-objective function. However, the problem with this approach is that weights associated with different objective functions are very difficult to decide, particularly when there are tradeoffs between the objectives (as are generally there in real-life applications). In a posteriori approach, all the possible Pareto fronts or a subset of them are produced. However, in multi-objective optimization, the objectives are generally conflicting in nature.

4. Proposed MACS for MOP

It has been observed that some of the commonly used approaches such as GA, Particle Swarm Optimization (PSO), and Tabu search have certain disadvantages. These do not use heuristic information for solution construction; rather, they rely on their self-optimization mechanism for the same. However, ACS takes advantage of heuristic function during the solution construction process [27]. Furthermore, as discussed in Section 1, EAs also suffer from a huge computation burden to avoid trapping out of local optima. However, ants in ACS due to their cooperative behavior are better suited for combinatorial optimization problems such as VRP and bin packing problems.

4.1. Ant Colony System (ACS)

ACS gets inspired from the real ants. Ants in nature live in colonies. Their food search begins with searching for paths for the food source in a random manner, i.e., exploring various possible alternatives. Real ants release a smelling agent called pheromone on the paths they follow. At the return trip from the source, ant deposits the pheromone on the path, which guides other ants for food sources. The greater the amount of pheromone on the path, the more likely it is that this path will be followed by other ants. However, the pheromone left by an ant gradually decays over time. In addition to pheromone information, ants also use their insight to explore new paths. By keeping these facts in check, some efficient ant colony optimization algorithms have been designed. For applying ACO to real-life problems, the problem has to be modeled as connected graphs and the solution to a problem is an ordered sequence of all the nodes. In the mathematical model developed on this base, an ant probabilistically

visits from the current node i to the next node j . The state transition from current node i to the next node is influenced by exploration and exploitation, and mathematically defined by the following formula:

$$j = \begin{cases} \arg \max_{j \in J} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in J} \tau_{ij}^\alpha \eta_{ij}^\beta} & \text{if } \varphi \leq \varphi_o \text{ (exploitation)} \\ J & \text{otherwise (exploration)} \end{cases} \quad (12)$$

Exploitation refers to the selection of edges already visited by most previous ants, whereas to explore a new solution set, the ant chooses the next node to be visited by following the pseudo transition probability rule:

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in J} \tau_{ij}^\alpha \eta_{ij}^\beta} & \text{if } j \in J \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Here, p_{ij} is the probability of moving from current node i to the next node j , τ_{ij} and η_{ij} denote the pheromone concentration and heuristic value of the edge $i - j$, respectively, α and β parameters show the relative importance of pheromone and heuristic, respectively. J is the set of nodes currently available for visit from node i . The solution construction process in ACO involves pheromone concentration and heuristic information. The two factors are biased by different weights. For MOP, multiple pheromones and/or heuristics contribute to the construction of the solution.

Heuristic information (η_{ij}): In general, the heuristic function can be calculated as the reciprocal of the objective function (for minimization problems). For MOP, heuristic is the aggregation of different objectives to be optimized, i.e., multiple objectives are aggregated into the heuristic calculation. In our formulated MO-VRPTW model, the 2nd and 3rd objectives (i.e., f_2 and f_3) correspond to the minimization of make span and route imbalance, respectively. Since the aforesaid two objectives are known only after the completion of the route, these two cannot be included in heuristic η_{ij} while constructing the route using ACO. Therefore, we have assumed that the heuristic function η_{ij} depends simply upon the first objective f_1 and is given by:

$$\eta_{ij} = \frac{1}{d_{ij} + w_{ij}}, \quad (14)$$

where w_{ij} is the waiting time at location j (if reached earlier than opening time a_j of location j ; otherwise, it is 0).

Pheromone matrix (τ_{ij}): As discussed earlier, the proposed approach uses Pareto-front optimal solutions and, therefore, we avoid combining multiple objectives in the heuristic calculation, but we use multiple

pheromone matrices corresponding to each objective. Since our formulated model includes three objectives, here we consider three pheromone matrices namely $\tau(1)$, $\tau(2)$, and $\tau(3)$ corresponding to the proposed three objectives namely f_1 , f_2 , and f_3 , respectively. Initially, all the three matrices are initialized to equal values given by $\tau_{ij} = 1/d_{ij}$, i.e., pheromone trail on each of the arc $i - j$ in all matrices is reciprocal of the arc length. Furthermore, to deal with different objectives, we divide the ant system into three different colonies (col_u) $u = 1, 2, 3$. Each colony constructs an independent solution corresponding to its pheromone matrix. When all ants of the corresponding colony construct their solutions, the best solution of that colony updates pheromone locally according to the quality of the corresponding objective amongst those solutions. Pareto front solutions among three colonies are used to update pheromone globally. Local pheromone updating can be done through the following equation:

$$\tau_{ij}^{new}(u) = \begin{cases} (1 - \rho)\tau_{ij}^{old}(u) + \rho\Delta\tau_{ij} & \text{if } arc(i - j) \\ & \text{followed by} \\ & u^{th} \text{ ant} \\ (1 - \rho)\tau_{ij}^{old}(u) & \text{otherwise} \end{cases} \quad (15)$$

where $\Delta\tau_{ij}$ is the pheromone deposited on arc $(i - j)$ by the ants following that arc (if any). $\Delta\tau_{ij}$ is defined for three colonies differently as three different colonies have been used to optimize three different objectives accordingly. It can be given by the following equation as follows:

$$\Delta\tau_{ij} = \begin{cases} \frac{\bar{f}_1}{k_1 * f_1} & \text{if } u = 1 \\ \frac{\bar{f}_2}{k_2 * f_2} & \text{if } u = 2 \\ \frac{\bar{f}_3}{k_3 * f_3} & \text{if } u = 3 \end{cases} \quad (15a)$$

\bar{f}_u is the average value of the objective functions of all the ants of u th colony. After several simulations, it can be observed that $k_1 = 5$, $k_2 = 5$, and $k_3 = 5$ give the most appropriate outcome.

Similarly, in each iteration, after finding the best solution, the pheromone can be updated globally using the following equation:

$$\tau_{ij}^{new} = \begin{cases} (1 - \rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij} & \text{if } arc(i - j) \\ & \text{followed the current} \\ & \text{best solution} \\ (1 - \rho)\tau_{ij}^{old} & \text{otherwise} \end{cases} \quad (16)$$

where,

$$\Delta\tau_{ij} = 1 / \sum f_u. \quad (16a)$$

Here, the denominator is the sum of all the objective functions for the current best solution.

ACS is well suited for solving VRP and VRPTW in terms of computational complexity and convergence rate. However, the traditional ACS has the drawback of pre-mature convergence and getting easily trapped into local optima. Therefore, the researchers have improved basic ACS to solve VRP and its variants by hybridizing ACS with GA, PSO, Local search, neighborhood search, etc. [1].

On the other hand, the advantages of FA in various optimization fields have been analyzed in the survey presented by [29]. It has also been noticed that while ACS has a relatively high convergence rate, FA provides a more diversified solution space due to its multi-modality nature [30]. Despite such a high popularity of FA and multi-modality nature, it has been rarely applied in the field of routing and multi-objective routing as well.

Thus, it is felt that the appropriate hybridization of these two approaches might prove more effective in solving multi-objective VRPs. Hence, to take advantage of the good features of the above-said approaches, we combined FA with ACS to solve MO-VRPTW.

4.2. Firefly Algorithm (FA)

FA is a recent population-based stochastic algorithm that uses a probabilistic approach to problem solutions. It is based on flashing behavior and the phenomenon of bioluminescent communication of fireflies. Fireflies flash the light for mating primarily. Interested users can refer to [30,31] for more technical details of FA.

In the simplest form as suggested by Fister et al. [30], attractiveness is proportional to the light seen by the observing fireflies; the variation in attractiveness with distance is approximated as:

$$\mathfrak{B} = \mathfrak{B}_0 e^{-\gamma r_{ij}^2}, \quad (17)$$

where \mathfrak{B}_0 is the attractiveness at $r = 0$ and r_{ij} is the Euclidean distance between two fireflies in positions i and j . Finally, the movement (x_i) of firefly (i) towards the firefly (j) is determined as follows:

$$x_i = x_i + \mathfrak{B} e^{-\gamma r_{ij}^2} + \sigma. \quad (18)$$

Here, the second term is the attractiveness and the third term σ is the randomness control parameter such that $\sigma \in [0, 1]$.

The ants in ACS follow the path with a higher concentration of pheromone, causing the ACS to get trapped into local optima. So, FA is applied to search for other promising regions that are less explored.

4.2.1. Firefly encoding scheme

To adapt FA for discrete optimization problems and to encode fireflies, here we adopted the representation scheme proposed by Yang [32]. The fireflies are encoded as a set of customers whose index represents customer visiting sequence number. “0” denotes the start of a new route and last “0” end of the overall tour. For example, for the tours (T1 and T2), the fireflies are represented, as depicted in Figure 1.

Another key issue in defining the distance r_{ij} between two fireflies is defined as the sum of the links connected by different customers in between two fireflies (shown by red links in Figure 2). For instance, the distance between Fireflies 1 and 2 is 3, as depicted in Figure 2.

4.2.2. Pseudocode for FA search

Step 1: Initialization. The ‘ m ’ tours generated by ACS can be treated as the population of ‘ m ’ fireflies;

Step 2: Light intensity calculation. Since VRPs aim to minimize objective problems, the intensity of the flashing light of each firefly is set equal to the inverse of the objective function value;

Step 3: Attractiveness and movement. Sort ‘ m ’ fireflies according to their light intensity. The firefly with the least objective function value can be considered the best firefly. The attractiveness \mathfrak{B} of other fireflies towards the best firefly is determined as a light intensity function and random move (σ) and given by Eq. (18);

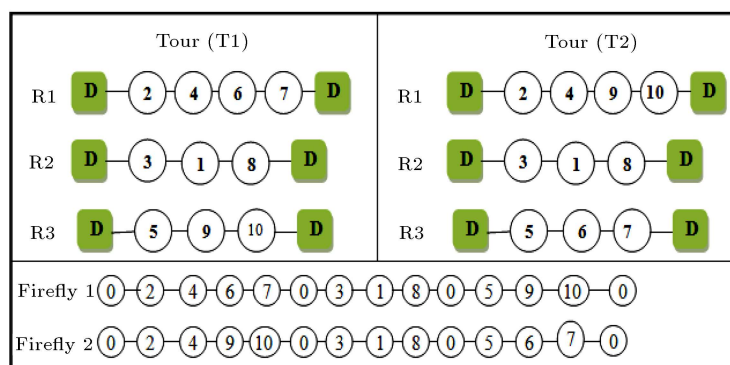


Figure 1. Firefly encoding.

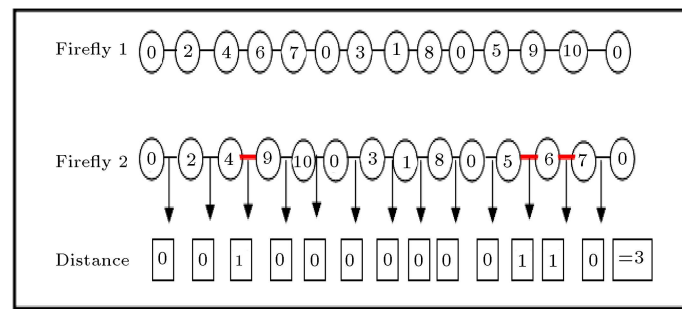


Figure 2. Distance between fireflies 1 and 2.

Step 4: Return ‘ m ’ explored tours to ACS algorithm.

The values of parameters can be taken as $m = 15$ and $\sigma = 0.4$, as suggested by Yang [32].

4.3. Migration operators

For combinatorial optimization problems, it is a general practice to augment local search with basic meta-heuristic algorithms to enhance the results obtained from the basic algorithms. However, for MOP, Pareto dominance-based Local Search (called PLS) is applied. In PLS, the new solutions are accepted on the basis of directions in which the improvement is expected. In this paper, PLS based on migration moves is applied. Migration refers to the removal of a customer from one route and reinsertion into another route, provided that the obtained solution is feasible and better.

Here, two migration operators namely Migration-I and Migration-II are proposed and brief details are given as follows:

Migration-I: This operator is used to reduce the makespan of the largest route. A route with the largest makespan is selected and a customer nearest to the neighboring route is selected as seed. The seed is removed from its parent route and reinserted into the neighboring route. The new solution is accepted on the basis of the quality of improved makespan and solution feasibility. Diagrammatical Migration-I is shown below in Figure 3. The lastly served customer in route R1 can be migrated from that route R1 to R2 as the first customer of route R2 (without any constraint violation) to shorten the makespan.

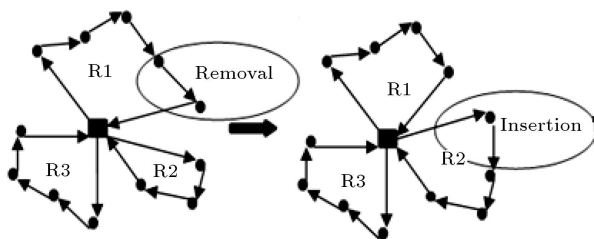


Figure 3. Migration-I operator.

Migration-II: This operator attempts to minimize f_3 . The route with a length less than half of the average route length (if any) is selected. The customers of this route are considered one by one and readjusted to other neighboring routes (if possible), resulting in minimization of distance imbalance. It is depicted in Figure 4. Additionally, the operator also tries to reduce the number of routes. For instance, route R4, given in Figure 4, has only two customers which can be migrated to routes R1 and R3 (without any constraint violation), reducing the overall distance imbalance among vehicles.

The general framework of our proposed ACS algorithm is given by the MACS as shown in Algorithm 1.

\mathbb{S} in line 7 of Algorithm 1 represents the tour constructed using the nearest neighbor heuristic. \mathbb{U} is the archive set of Pareto front solutions and, initially, it contains only a single solution \mathbb{S} , as described in line 8. Lines 11–37 represent the main loop iterations until a specified termination criterion is found. Loop involving lines 13–29 undergoes repetition for all of the ants. Inner while loop (lines 15–26) is repeated until all the customers are not served. Lines 19–23 decide the exploration of new possibilities or exploitation of the previous routes according to Eq. (12). In line 30, local search is applied to improve the obtained results. Line 33 selects all the non-dominated solutions among current k ants. In line 34, the archive set \mathbb{U} is updated by inserting all non-dominated solutions and discarding dominated solutions. Pheromone trails are updated in lines 27 and 35, as described in Section 3.2. Finally, all the Pareto front solutions are presented in line 38.

5. Experimentation and discussion

Evaluation of the performance of multi-objective optimization algorithms is more complicated than that of single-objective algorithms. The researchers have proposed a number of valuable criteria. However, till now no single criteria is available to evaluate the overall performance of MO- algorithms. Here, we use two most commonly used metrics namely Hyper Volume (HV) and Coverage Metric (C-Metric) to evaluate the

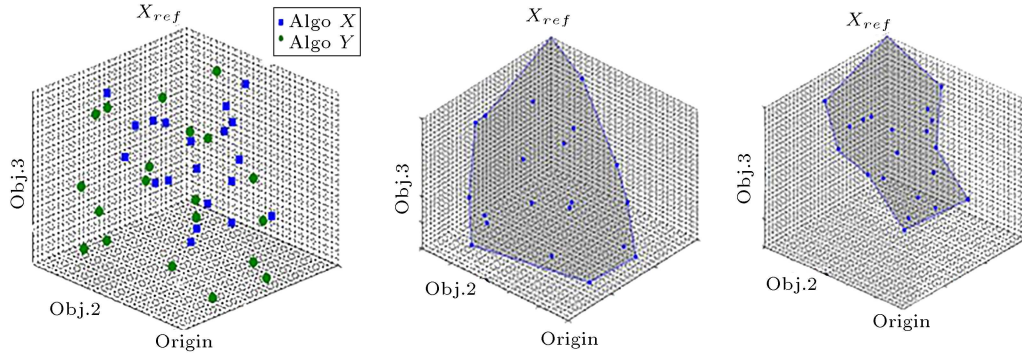


Figure 4. (a) C-Metric (algorithm X, algorithm Y), (b) algorithm Y hyper-volume, and (c) algorithm X hyper-volume.

```

1. /* Initialization*/
2. Parameter setting: Set values of no. of ants (m), Stopping criterion, i.e., max number of
3. Initialize Pareto front  $\mathbb{U}$  as empty.
4. Initialize Pheromone matrix  $\tau_0$ .
5. /* Main loop*/
6. /* Construct initial solution*/
7. Apply nearest neighbor heuristic and construct initial feasible solution  $\mathbb{S}$ .
8. Update  $\mathbb{U}$  as  $\mathbb{U} = \mathbb{U} \cup \mathbb{S}$ .
9. /* MACS*/
10. Set  $itr = 1$ 
11. while  $itr \leq N_{max}$  do
12.   for each ant colony ( $col_u$ ) ( $u = 1:3$ ) do
13.     for  $i = 1$  to  $m$  (no. of ants) do
14.       set pool of visited customers  $\mathcal{S}_i = \emptyset$  and veh_used i.e.  $k = 1$ 
15.       while  $\mathcal{S}_i \neq V$  do
16.         for each of unvisited customer  $v$  do
17.           calculate the probability of movement acc. to transition rule (13).
18.           Generate a random number  $\varphi$ 
19.           if  $\varphi \leq \varphi_0$ 
20.             exploration
21.           otherwise
22.             exploitation
23.           end if
24.           If vehicle gets full or any other constraint violation set  $k = k + 1$ 
25.         end for
26.       end while
27.       update pheromone locally as  $\tau_{ij}^{new}(u)$ 
28.       Apply FA search as described in Section 3.2.
29.     end for
30.     find the best solution  $best\_sol(u)$  among all ants for each colony
31.     Apply Migration-I and Migration-II operators on  $best\_sol(u)$ 
32.   end for
33.   Among the  $u$  best solutions, find the set  $\mathbb{S}$  of non-dominated Pareto solutions.
34.   Update  $\mathbb{U}$  as  $\mathbb{U} = \mathbb{U} \cup \mathbb{S}$  and remove all of the dominated solutions (if any).
35.   For each of the  $\mathbb{S}$ , apply update pheromone globally
36.   Set  $itr = itr + 1$ 
37. end while
38. Output Pareto front  $\mathbb{U}$ 

```

Algorithm 1. MACS algorithm

performance of the proposed MACS [17,34]. The mentioned two criteria are defined as follows.

5.1. Performance metrics

HV: As proposed in [35], HV indicates the area bounded by a reference point X_{ref} , which is dominated

by at least one of the non-dominated solutions. Let $X = x_1, x_2, \dots, x_n$ be a set of non-dominated solutions. Therefore, the larger the HV (x), the better the value of X will be. In general, HV determines the closeness of solution space to the Pareto fronts. Diagrammatically, HV for the three-objective function

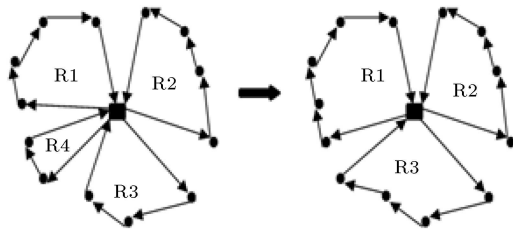


Figure 5. Migration-II operator.

is shown in Figure 5. Since the volume of solution set Y is greater than that of solution set X , the set Y will be better than the set X as the former is closer to Pareto front. In our case, X_{ref} is the reference point which is equal to twice the objective function obtained from the initial nearest neighbor heuristic, as suggested by Baños et al. [13].

C-metric: This metric proposed by [36] compares two non-dominated algorithms X and Y (say) based on the following metric:

$$C(X, Y) = \frac{|\{y \in Y | \exists x \in X : x \geq y\}|}{|Y|}, \quad (19)$$

where $C(X, Y)$ is an ordered pair representing the percentage of non-dominated solutions obtained by algorithm Y , which are dominated by at least one solution of algorithm X . Furthermore, $C(X, Y) + C(Y, X)$ does not require to be equal to 1 as there may be some solutions that are non-comparable. Moreover, the greater the value of $C(X, Y)$, the better the value of X will be than Y . According to Figure 5(a), it is noticed that $C(X, Y) = 55\%$ and $C(Y, X) = 72\%$. Hence, algorithm Y performs better than algorithm X . Furthermore, HV of algorithm Y (Figure 5(b)) is wider than algorithm X (Figure 5(c)) and it also verifies the better performance of Y over X .

5.2. Benchmarks and parameter setting

Since no standard benchmarks are available for comparing the currently proposed MO-VRPTW, the proposed MACS algorithm is applied to standard benchmark problems of Solomon's datasets available in [37]. The 56 problem sets are broadly categorized into 6 groups, namely C-1, R-1, RC-1, C-2, R-2, and RC-2. In C-type problems, the customers are clustered into groups, whereas in R datasets, the customers are randomly scattered. RC type is a collection of partially grouped as well as partially geographically scattered customers. Moreover, type-1 problems are associated with smaller time windows, while the time span for type-2 problems is large. In type-2 problems, the capacity of vehicles is also larger than that of type-1 sets. Hence, type-2 problems use smaller fleet size than type-1 problem sets. All the computations have been performed using MATLAB R-by a laptop with 2.0 GB

Table 3. Decision parameters.

Parameter	Description	Value
N_{\max}	Max no. of iterations	5000
m	Number of ants	10
φ_0	Exploration vs exploitation decision	0.9
α	Importance of pheromone	1
β	Importance of heuristic	5
ρ	Pheromone evaporation constant	0.1

Ram, Core-2 Duo processor having a speed of 2.3 GHZ. As proposed, MACS uses a number of parameters that significantly affect the performance of the algorithm. Therefore, optimal values of these parameters should be selected appropriately. For our experimentation, we choose these values from the recent literature on the use of ACS for VRPTW [32]. Used values are given as in Table 3.

5.3. Simulation results

To check the performance, the results of the proposed MACS algorithm are compared with those of other approaches available in the literature with a greater focus on a single objective, two objectives, and then all the three objectives simultaneously.

5.3.1. Comparison among different versions of the proposed MACS

Here, first of all, we have analyzed the effect of introducing FA search and migration operators in the proposed MACS. The performance comparison among different versions of the proposed MACS (i.e., MACS with FA search only, MACS with migration operators only, and MACS with FA search and migration operators both) on all the considered objectives is presented in Table 4. The first column in this table presents the dataset taken and the other three main columns namely TD, Makespan, and Load balancing represent the considered three objectives. From the TD column, it has been found that FA search tries to optimize the total traveled distance as it is providing better results (for all the datasets taken) than the results obtained from migration operators only. Alternately, migration operators perform well for load balancing among the drivers/routes, as shown in the last column. However, the performance of FA and Migration operators is incomparable in case of makespan, as shown by bold entries in the respective column. It might be due to the fact that Migration-I tries to reduce the makespan, whereas Migration-II operator will contribute to larger makespan by eliminating the routes with a very small number of visited customers. The underlined italic

Table 4. Performance comparison among different versions of Multiple ACS (MACS).

Data type	TD			Makespan			Load balancing		
	MACS with FA only	MACS with Migration only	Proposed MACS	MACS with FA only	MACS with Migration only	Proposed MACS	MACS with FA only	ACS with migration only	Proposed MACS
R101	1847.63	1882.16	1650.799	142.16	143.93	132.49	27%	35%	56%
R102	1532.04	1583.94	1486.859	143.04	141.26	134.25	47%	57%	70%
R103	1391.27	1444.46	1292.675	142.16	143.93	132.49	37%	44%	57%
R104	1158.03	1193.25	1007.31	145.62	141.78	134.25	27%	43%	61%
R105	1465.41	1498.04	1377.11	140.37	136.42	132.49	29%	46%	58%
R106	1338.15	1375.26	1252.03	143.04	146.94	134.25	28%	40%	56%
R107	1209.82	1255.1	1104.655	140.37	136.42	132.49	47%	60%	72%
R108	1024.32	1072.17	960.8753	145.62	136.42	134.25	37%	48%	59%
R109	1257.8	1302.18	1194.734	140.37	141.78	132.49	57%	68%	76%
R110	1204.06	1256.2	1118.838	145.62	146.94	134.25	37%	48%	59%
R111	1201.97	1249.14	1096.726	140.37	145.62	132.49	47%	63%	79%
R112	1083.14	1152.9	982.1392	145.62	146.94	134.25	37%	49%	62%
R201	1426.85	1500.07	1252.371	481.6	466.07	427.87	72%	78%	81%
R202	1256.22	1322.62	1191.703	492.71	458.26	438.72	90%	91%	94%
R203	979.65	997.44	939.5029	419.72	458.26	358.4	72%	83%	90%
R204	907.06	907.06	825.5188	<i>419.72</i>	501.4	476.65	90%	91%	93%
R205	1083.16	1107.14	994.4272	<i>365.27</i>	384.28	384.28	74%	80%	87%
R206	998.14	1052.84	906.1416	365.27	351.97	335.37	89%	91%	93%
R207	953.18	1008.26	890.6078	<i>365.27</i>	453.27	453.27	91%	92%	93%
R208	848.01	943.16	726.8223	<i>365.27</i>	379.82	367.44	91%	91%	93%
R209	1024.11	1097.52	909.1629	330.16	328.41	316.34	87%	91%	91%
R210	1032.9	1097.52	939.3722	<i>310.16</i>	328.41	324.46	89%	89%	94%
R211	949.62	997.44	885.7109	481.6	481.6	469.24	87%	88%	91%
RC101	1803.89	1881.83	1696.949	<i>151.04</i>	173.85	162.7	72%	80%	86%
RC102	1701.2	1762.07	1554.747	173.16	173.85	162.7	72%	81%	85%
RC103	1345.12	1389.04	1261.671	173.16	149.06	142.73	66%	71%	82%
RC104	1261.95	1321.17	1135.479	140.84	152.24	134.45	90%	90%	93%
RC105	1802.65	1881.83	1629.436	<i>151.04</i>	170.81	167.05	74%	83%	89%
RC106	1508.23	1544.04	1424.733	173.16	173.85	167.76	88%	91%	91%
RC107	1295.26	1321.17	1230.477	<i>140.84</i>	152.24	150.63	46%	59%	73%
RC108	1198.46	1237.85	1139.821	145.58	170.81	136.29	90%	92%	93%
RC201	1510.02	1568.06	1406.94	462.88	481.4	447.66	88%	90%	93%
RC202	1446.37	1506.24	1365.645	557.16	543.02	526.87	74%	81%	89%
RC203	1123.85	1144.16	1049.624	510.96	503.34	498.26	60%	64%	76%
RC204	849.01	919.43	798.4632	343.14	343.14	321.72	69%	72%	84%
RC205	1391.43	1402.18	1297.647	436.02	423.5	418.43	75%	81%	90%
RC206	1208.27	1256.92	1146.317	499.16	510.12	476.61	70%	76%	82%
RC207	1203.18	1256.92	1061.144	436.02	431.67	419.83	55%	66%	79%
RC208	861.48	880.06	828.1411	343.14	316.9	309.42	79%	80%	86%
C101	857.16	861.28	828.9366	141.4	129.75	127.3	70%	74%	81%
C102	839.15	857.16	828.9366	141.4	130.08	127.3	68%	73%	74%
C103	863.25	863.25	828.0646	138.07	132.92	128.04	70%	73%	82%
C104	849.46	857.16	824.7765	141.4	130.08	128.04	50%	59%	69%
C105	901.12	909.26	828.9366	138.07	135.6	127.3	49%	59%	68%
C106	863.38	884.62	828.9366	138.07	132.92	127.3	38%	49%	59%
C107	863.38	909.26	828.9366	138.07	135.6	127.3	68%	72%	80%
C108	849.46	863.25	828.9366	141.4	127.3	127.3	39%	51%	70%
C109	863.38	884.62	828.9366	138.07	129.75	127.3	48%	58%	68%
C201	603.14	647.62	591.5563	262.76	251.06	238.21	87%	90%	92%
C202	603.14	647.62	591.5563	262.76	251.06	238.21	86%	91%	92%
C203	617.26	650.14	591.1732	273.14	257.81	238.21	88%	91%	93%
C204	642.32	642.32	590.5985	273.14	249.01	238.21	76%	82%	91%
C205	617.26	647.62	588.8757	273.14	251.06	235.53	88%	91%	90%
C206	642.32	650.14	588.4926	262.76	257.81	235.53	66%	70%	78%
C207	617.26	647.62	588.286	273.14	251.06	235.53	85%	89%	90%
C208	607.24	624.46	588.3235	273.14	249.64	235.53	83%	87%	92%

Table 5. Comparison with other approaches considering single-objective f_1 .

Data set	C_1	C_2	R_1	R_2	RC_1	RC_2
ACO [10]	881.44	641.25	1383.2	1098.22	1211.12	1209.44
	10	3.5	13.6	4.1	12.7	5.6
	112	89	405	395	204	188
ACO–Tabu [10]	841.92	612.75	1213.16	952.3	1415.62	1120.37
	10	3.3	13.1	4.6	12.7	5.6
	210	142	698	655	317	407
MOEA [38]	828.38	591.49	1236.17	912.81	1392.09	1162.4
	10	3	13.89	4	12.63	5.13
	–	–	–	–	–	–
MOGA [7]	828.48	590.6	1253.86	907.38	1370.84	1070.38
	10	3	13.89	4.37	12.75	4.75
	–	–	–	–	–	–
PDVA [10]	828.38	591.49	1228.6	1033.53	1362.09	1068.26
	10	3	12.92	3.45	12.75	3.75
	3800	4000	3800	3900	3800	7700
HAFA [33]	843.32	599.04	1244.48	976.2	1347.72	1163.98
	10	3	13.18	3.18	12.96	3.5
	–	–	–	–	–	–
EDFA [39]	907.11	666.23	1442.71	1243.18	1568.94	1490.36
	10	3	13.18	3.18	12.96	3.5
	–	–	–	–	–	–
Proposed MACS	828.38	599.04	1250.61	1024.41	1374.82	1158.97
	10	3	13.9	3.5	13.1	4
	500	2000	3600	3750	3400	3400
BKS	828.38	589.86	1209.89	951.19	1245.79	1119.35
	10	3	12.92	3.45	12.37	4
	–	–	–	–	–	–

entries in the makespan column (Table 4) show the best makespan among all the three versions. However, these best results are the cost of unfair load balancing.

5.3.2. Comparison of the proposed MACS with some of other available techniques

In this subsection, we analyze the performance of our proposed MACS with some of the available techniques available in the literature.

Observations for f_1 alone: The proposed MACS is

firstly tested on and applied to the Solomon datasets considering only single-objective minimization of total traveled distance. To this end, the number of ant colonies, i.e., $u = 1$, single pheromone matrix, and only one local pheromone updating ($\tau_{ij}^{new}(1)$) are required. A total of 100 simulation runs have been performed for all the 56 datasets, and their average compared with other proposed approaches available in the literature is reported in Table 5. The proposed MACS has been compared with standard ACS, ACO hybridized with Tabu (ACO-Tabu), p -system based evolutionary

algorithm (PDVA) (all three results are reported by [10]), MOEA [38], MOGA [7], HAFA [18] and [39], and Best-Known Solutions (BKS) reported in the literature. For each approach, Table 5 has three rows representing total traveled distance, number of vehicles used, and computation time in seconds, respectively: (–) in any cell indicates that the data are not available.

It was observed that our approach outperformed ACO, ACO-Tabu, and EDFA in terms of TD as well as number of vehicles used to complete the tour. Moreover, for C1 and C2 datasets, it is comparable to other approaches with a maximum percentage error of 1% from PDVA. For R1 and R2 problems, it appears to be better than MOEA, MOGA, and EDFA in terms of TD (except for MOEA and MOGA in R2). Additionally, the proposed approach requires a shorter computation time (for all data sets) and produces better results (in case of R2) than PDVA. However, there are 2%, 1%, and 8% errors in TD of R1, RC1, and RC2 with our approach in comparison to the BKS. Hence, the overall performance of the proposed algorithm (for single-objective problems) is found to be comparable to other existing algorithms designed especially for single-objective VRPTWs.

5.3.3. Observations for MO-VRPTW

The next proposed MACS has been tested on the same Solomon benchmarks while considering all the three objectives simultaneously. Since NSGA-II (Non-dominated Sorting GA-II) is one of the successfully used evolutionary algorithms for MOPs [34,40,41], the NSGA-II is taken as a comparative algorithm for the proposed approach. The approach used by Sánchez-Oro et al. [34] has been adopted to fit NSGA-II for MO-VRPTW and accordingly, a population of 200 chromosomes and 100000 generations has been used. The crossover rate and probability of mutation are 0.6 and 0.3, respectively. Again, a total of 2000 simulation runs have been performed for both the algorithms and the average values of the metrics defined in Section 5.1, as presented in Table 6. Since true Pareto fronts are unknown for the current problem, Pareto fronts obtained using the current approach have been taken as true Pareto fronts. Additionally, the value of each objective function obtained by using the nearest neighbor approach has been taken as the reference point for calculating HV.

Bold entries in Table 6 depict the better values among the two compared approaches. The last row in Table 6 summarizes the overall results of 56 instances as (b/s/w), indicating that the proposed MACS performs better, nearly equal to and worse than NSGA-II, respectively.

Considering HV metric, it has been found that HV values range from 0.32 to 1.03 and 0.37 to 1.68 for NSGA-II and MACS, respectively, confirming better

Pareto fronts of the proposed approach than that of NSGA-II. Furthermore, it has been also concluded that HV obtained using MACS is better in the case of 49 instances with the maximum increase of 178% for RC-107 and the minimum increase of 6% for R-211 with an average increase of 57%. On the other hand, NSGA-II appears to be better in only 5 cases with a maximum rise of 37% (for R-205). The superiority of the proposed approach can also be supported by the results obtained while comparing C-Metric values. It has been found that in 42 cases (out of 56), the proposed MACS dominates NSGA-II, while the latter could dominate the former only in 6 cases. Moreover, both the approaches are non-comparable (i.e., no one dominated another) in 8 cases. Upon comparing the computation time needed for the two approaches, it was found that our algorithm performed better in all the 56 instances with nearly one-tenth of the time needed than NSGA-II. Hence, the proposed approach proved to be more stable. The two algorithms are further compared in terms of total traveled distance and makespan (the two most conflicting objectives). The resulting graphs of six problems (one from each category of Solomon Dataset) are depicted in Figure 6. From all the graphs, the superiority of the proposed MACS over better NSGA-II both in terms of solution quality and solution diversity has been confirmed.

6. Conclusion and scope for future work

This study investigated the multi-objective vehicle routing problem with time windows focusing on simultaneous minimization of total traveled distance, makespan, and route imbalance using ant colony system. The ant system was divided into multiple colonies to deal with each objective separately. Moreover, multiple pheromone matrices were used to update the pheromone according to the obtained Pareto front solutions. Firefly Algorithm (FA) was hybridized to avoid being trapped into local optima. Two new local search operators namely Migration-I and Migration-II were designed to further improve the solution quality. The algorithm was validated by testing on 56 Solomon benchmark problems considering the minimization of total traveled distance. Furthermore, Multiple ACS (MACS) proved to be much better than the widely used multi-objective NSGA-II while considering multiple objectives of VRPTW. The proposed MACS can also be easily adapted to many-objective problems by only changing the number of colonies and pheromone matrices corresponding to each objective. In the future, the proposed approach can be extended further to real-life applicability of VRPTW such as VRPTW with split delivery, stochastic VRPTW and time-dependent VRPTW, etc. Future works also include experimen-

Table 6. Comparison with Non Dominated Sorting Algorithm-II (NSGA-II) in terms of average values of Hyper Volume (HV), Coverage Metric (C-Metric), and computation time on Solomon benchmarks considering all the three objectives.

Data type	HV		C-Metric		Computation time (s)	
	NSGA-II	MACS	C (NSGA-II, MACS)	C (MACS, NSGA-II)	NSGA-II	MACS
R101	0.4357	0.6267	0.3233	0.3964	11575	2028
R102	0.4227	0.6093	0.3901	0.4216	13229	2219
R103	0.4906	0.8922	0.3531	0.3528	14707	2127
R104	0.4853	0.8698	0.4002	0.3542	10348	2563
R105	0.3156	0.5549	0.5216	0.2709	14264	2214
R106	0.3769	0.5483	0.4137	0.431	14314	2302
R107	0.4227	0.665	0.3143	0.5249	15938	1997
R108	0.4373	0.6034	0.5609	0.5385	12649	2553
R109	0.4706	1.0533	0.2145	0.6085	11872	2626
R110	0.4518	1.1976	0.4722	0.5102	16439	2321
R111	0.3905	0.363	0.4308	0.467	9507	1964
R112	0.532	0.4904	0.4461	0.4793	13108	1927
R201	0.7001	1.0962	0.4678	0.5106	14482	1900
R202	0.6806	1.045	0.2534	0.3891	10990	2119
R203	0.6477	0.904	0.3502	0.6215	12077	2310
R204	0.5908	1.025	0.4583	0.4583	10463	2171
R205	0.9021	0.5645	0.4426	0.5238	12467	2444
R206	0.6386	0.9455	0.438	0.4904	14182	2634
R207	0.6102	0.8349	0.4398	0.46	11108	2627
R208	0.5344	0.7046	0.4701	0.5113	11562	2537
R209	0.5906	1.1609	0.6218	0.5365	13608	2750
R210	0.6004	0.9324	0.3204	0.4128	12427	2600
R211	0.5284	0.5592	0.338	0.6581	11352	2154
RC101	0.5407	1.0042	0.2932	0.2947	13134	2156
RC102	0.4642	0.7967	0.2541	0.6548	11729	1947
RC103	0.5495	0.9821	0.4205	0.5488	10821	2090
RC104	0.4882	1.1523	0.4273	0.5679	12587	1888
RC105	0.379	0.7503	0.4094	0.5802	10403	1793
RC106	0.456	0.4557	0.3984	0.3992	13788	2068
RC107	0.44	1.2238	0.4386	0.5578	10555	2515
RC108	0.5186	0.4816	0.4901	0.4943	9140	2137
RC201	0.8221	0.9084	0.4374	0.6075	14356	2881
RC202	0.7802	1.6803	0.4607	0.4632	14496	2719
RC203	0.8379	0.8388	0.6255	0.3578	12875	2410
RC204	0.6846	1.4303	0.449	0.5672	11332	2291
RC205	0.7205	1.1266	0.4916	0.4908	14697	2752
RC206	0.6793	1.3835	0.2729	0.7041	13076	1876
RC207	0.4609	0.584	0.3886	0.6248	12167	2153
RC208	0.6628	0.8764	0.2908	0.6108	12534	1825
C101	0.9692	1.5189	0.3871	0.5389	10256	2282
C102	1.039	1.615	0.3442	0.6304	4515	929
C103	0.9255	0.697	0.316	0.2938	4833	922
C104	0.5612	0.7724	0.3549	0.4802	7371	982
C105	0.6378	0.9278	0.3004	0.6103	8604	1012
C106	0.6004	1.2699	0.3794	0.5349	5693	1127

Table 6. Comparison with Non Dominated Sorting Algorithm-II (NSGA-II) in terms of average values of Hyper Volume (HV), Coverage Metric (C-Metric), and computation time on Solomon benchmarks considering all the three objectives (continued).

Data type	HV		C-Metric		Computation time (s)	
C107	0.6128	0.933	0.3052	0.4586	8679	1050
C108	0.5587	1.2751	0.2556	0.4702	8854	1148
C109	0.728	1.009	0.2578	0.2594	3225	1138
C201	0.605	1.3785	0.3557	0.4027	5706	835
C202	0.5892	1.034	0.4314	0.4805	4928	1132
C203	0.5904	0.5822	0.4372	0.5621	3920	858
C204	0.5591	0.8942	0.3627	0.5427	7077	810
C205	0.5994	0.7247	0.4619	0.5093	5668	1098
C206	0.6237	0.8248	0.5243	0.5226	8516	883
C207	0.559	0.7935	0.5738	0.4166	5488	866
C208	0.6702	0.8567	0.4644	0.5003	8058	724
b/s/w	49/2/5		42/8/6		56/0/0	

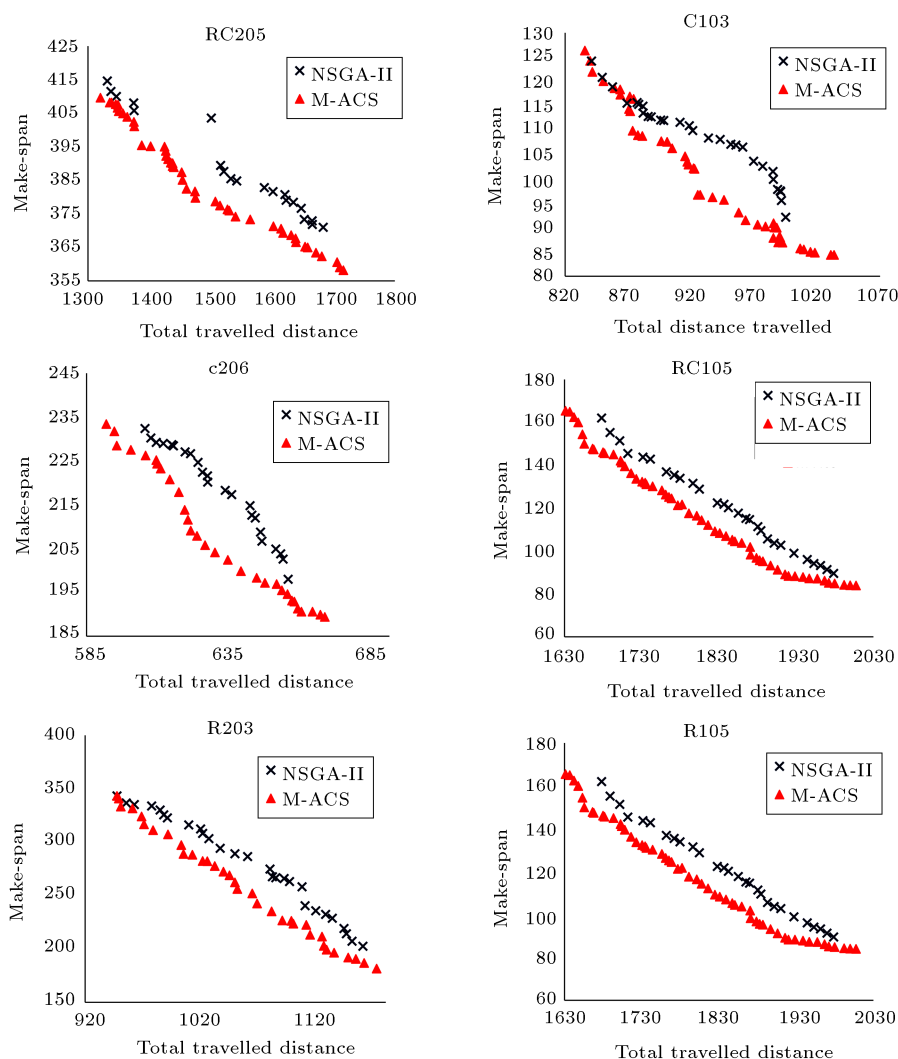


Figure 6. Makespan vs total travelled distance.

tion on hybridization of the proposed MACS with EAs to solve many-objective real-life VRPTWs.

References

- Goel, R. and Maini, R. "Vehicle routing problem and its solution methodologies: a survey", *International Journal of Logistics Systems and Management*, **28**(4), pp. 419–435 (2017).
- Borgulya, I. "An algorithm for the capacitated vehicle routing problem with route balancing", *Central European Journal of Operations Research*, **16**(4), pp. 331–343 (2008).
- Jozefowiez, N., Semet, F., and Talbi, E.G. "An evolutionary algorithm for the vehicle routing problem with route balancing", *European Journal of Operational Research*, **195**(3), pp. 761–769 (2009).
- Kritikos, M.N. and Ioannou, G. "The balanced cargo vehicle routing problem with time windows", *International Journal of Production Economics*, **123**(1), pp. 42–51 (2010).
- Zhou, W., Song, T., He, F., and Liu, X. "Multiobjective vehicle routing problem with route balance based on genetic algorithm", *Discrete Dynamics in Nature and Society*, **2013**, pp. 1–9 (2013).
- Melián-Batista, B., De Santiago, A., Angel, Bello, F., et al. "A bi-objective vehicle routing problem with time windows: A real case in Tenerife", *Applied Soft Computing*, **17**, pp. 140–152 (2014).
- Ombuki, B., Ross, B.J., and Hanshar, F. "Multi-objective genetic algorithms for vehicle routing problem with time windows", *Applied Intelligence*, **24**(1), pp. 17–30 (2006).
- Jozefowiez, N., Semet, F., and Talbi, E.G. "Multi-objective vehicle routing problems", *European Journal of Operational Research*, **189**(2), pp. 293–309 (2008).
- Garcia-Najera, A. and Bullinaria, J.A. "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows", *Computers & Operations Research*, **38**(1), pp. 287–300 (2011).
- Dong, W., Zhou, K., Qi, H., He, C., and Zhang, J. "A tissue P system based evolutionary algorithm for multi-objective VRPTW", *Swarm and Evolutionary Computation*, **39**, pp. 310–322 (2018).
- Jozefowiez, N., Semet, F., and Talbi, E.G. "Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem", In *International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Heidelberg, pp. 271–280 (2002).
- Lau, H.C., Chan, T.M., Tsui, W.T., et al. "A fuzzy guided multi-objective evolutionary algorithm model for solving transportation problem", *Expert Systems with Applications*, **36**(4), pp. 8255–8268 (2009).
- BañOs, R., Ortega, J., Gil, C., et al. "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows", *Computers & Industrial Engineering*, **65**(2), pp. 286–296 (2013).
- Widyadana, G.A. and Irohara, T. "Modelling multi-tour inventory routing problem for deteriorating items with time windows", *Scientia Iranica*, **26**(2), pp. 932–941 (2019).
- Hasle, G. "Routing applications in newspaper delivery", SINTEF Report A23753, Oslo, Norway (2012).
- Najafi, M., Eshghi, K., and Dullaert, W. "A multi-objective robust optimization model for logistics planning in the earthquake response phase", *Transportation Research Part E: Logistics and Transportation Review*, **49**(1), pp. 217–249 (2013).
- Zhou, Y. and Wang, J. "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows", *IEEE Systems Journal*, **9**(3), pp. 1100–1113 (2015).
- Lacomme, P., Prins, C., Sevaux, M., et al. "A genetic algorithm for a bi-objective capacitated arc routing problem", *Computers and Operations Research*, **33**(12), pp. 3473–3493 (2006).
- Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R., et al. "A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application", *Applied Soft Computing*, **14**, pp. 504–527 (2014).
- Chiang, T.C. and Hsu, W.H. "A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows", *Computers and Operations Research*, **45**, pp. 25–37 (2014).
- Kaiwartya, O., Kumar, S., Lobiyal, D.K., et al. "Multiobjective dynamic vehicle routing problem and time seed based solution using particle swarm optimization", *Journal of Sensors*, **2015**, pp. 1–14 (2015).
- Qi, Y., Hou, Z., Li, H., et al. "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows", *Computers & Operations Research*, **62**, pp. 61–77 (2015).
- Ghannadpour, S.F. and Hooshfar, M. "Multi-objective vehicle routing problem with time windows and fuel consumption minimizing", In *ICORES*, pp. 92–99 (2016).
- Gharib, Z., Bozorgi-Amiri, A., Tavakkoli-Moghaddam, R., et al. "A cluster-based emergency vehicle routing problem in disaster with reliability", *Scientia Iranica, Transactions E, Industrial Engineering*, **25**(4), pp. 2312–2330 (2018).
- Wang, J., Weng, T., Zhang, Q., et al. "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows", *IEEE Transactions on Cybernetics*, **(99)**, pp. 1–12 (2018).
- Corberán, A., Fernández, E., Laguna, M., et al. "Heuristic solutions to the problem of routing school buses with multiple objectives", *Journal of the Operational Research Society*, **53**(4), pp. 427–435 (2002).

27. Miranda, D.M., Branke, J., and Conceição, S.V. “Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time”, *Applied Soft Computing*, **70**, pp. 66–79 (2018).
28. Bansal, S. and Goel, R. “Multi objective vehicle routing problem: A survey”, *Asian Journal of Computer Science and Technology*, **7**(3), pp. 1–6 (2018).
29. Zhong, Y.G. and Ai, B. “A modified ant colony optimization algorithm for multi-objective assembly line balancing”, *Soft Computing*, **21**(22), pp. 6881–6894 (2017).
30. Fister, I., Fister Jr, I., Yang, X.S., et al. “A comprehensive review of firefly algorithms”, *Swarm and Evolutionary Computation*, **13**, pp. 34–46 (2013).
31. Yang, X. “Firefly algorithms for multimodal optimization”, *Stochastic Algorithms: Foundations and Applications*, Springer Berlin Heidelberg, pp. 169–178 (2009).
32. Yang, X. “Firefly algorithm, stochastic test functions and design optimisation”, *International Journal of Bio-Inspired Computation*, **2**(2), pp. 78–84 (2010).
33. Goel, R. and Maini, R. “A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems”, *Journal of Computational Science*, **25**, pp. 28–37 (2018).
34. Sánchez-Oro, J., López-Sánchez, A.D., and Colmenar, J.M. “A general variable neighborhood search for solving the multi-objective open vehicle routing problem”, *Journal of Heuristics*, **26**(3), pp. 423–452 (2020).
35. Zitzler, E., Laumanns, M., and Thiele, L. “SPEA2: Improving the strength Pareto evolutionary algorithm”, *TIK-report*, **103**, pp. 1–22 (2001).
36. Zitzler, E. and Thiele, L. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”, *IEEE Transactions on Evolutionary Computation*, **3**(4), pp. 257–271 (1999).
37. <http://w.cba.neu.edu/~msolomon/problems.htm>
38. Yu, B., Yang, Z.Z., and Yao, B.Z. “A hybrid algorithm for vehicle routing problem with time windows”, *Expert Systems with Applications*, **38**(1), pp. 435–441 (2011).
39. Osaba, E., Carballedo, R., Yang, X.S., et al. “An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows”. In *Nature-Inspired Computation in Engineering*, pp. 21–41, Springer, Cham (2016).
40. Deb, K., Pratap, A., Agarwal, S., et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182–197 (2002).
41. Mandal, S.K., Pacciarelli, D., Løkketangen, A., et al. “A memetic NSGA-II for the bi-objective mixed capacitated general routing problem”, *Journal of Heuristics*, **21**(3), pp. 359–390 (2015).

Biographies

Rajeev Goel is working as an Assistant Professor in Govt. College Naraingarh, Ambala at Computer Science Department. He received his BS and MS degrees from Kurukshetra University Kurukshetra, India. Currently, he has done his PhD from Punjabi University Patiala India. His research interest includes metaheuristics and operations research.

Raman Maini received BTech (Computer Science & Engineering) from Beant College of Engineering, Gurdaspur, Punjab, India in 1999 and MTech (Computer Science & Engineering) from PAU, Ludhiana, India in 2002. He received Merit certificate in his MTech thesis at PAU. He is currently working as a Professor in Computer Engineering at University College of Engineering, Punjabi University, Patiala, India. He is a life member of ISTE (Indian Society of Technical Education), India, and IETE (Institution of Electronics & Telecommunication Engineers), India. His current areas of research include computer vision (specialty noise reduction in medical images, edge detection, and image enhancement) and development of soft computing-based algorithms.