



Solving the vehicle routing problem with time windows using modified football game algorithm

Zakir Hussain Ahmed^{a,*}, Fateme Maleki^b, Majid Yousefikhoshbakht^c, Habibollah Haron^d

^a Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Kingdom of Saudi Arabia

^b Department of Computer Science, Faculty of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

^c Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran

^d Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia

ARTICLE INFO

Keywords:

Heuristic search
Meta-heuristics
Football game algorithm
Vehicle routing problem with time windows
Solomon benchmark
NP-hard problem

ABSTRACT

The vehicle routing problem with time windows (VRPTW) is a famous problem in logistics and transportation that has many applications in real life. The objective of the problem is to find the minimum distance covered by a set of vehicles that set to start together from the warehouse and visit some prefixed customers within certain time frames. In this paper, an efficient optimization algorithm, called modified football game algorithm (MFGA), is proposed to solve the VRPTW as an NP-hard problem. MFGA imitates the behavior of football players during a game to find the best positions to score a goal under the supervision of a team coach. In the presented algorithm, a new method is presented to create an initial solution for the VRPTW problem. In addition, a more efficient method has been designed to create a new location for the player, which makes the algorithm more efficient. The performance of the proposed MFGA is validated against Solomon's VRPTW benchmark instances. Experimental results confirm that MFGA produces competitive results compared to several state-of-the-art algorithms in terms of various solution quality indicators. The proposed algorithm obtains 17 best-known solutions (BKs) and improves 12 BKs in the literature.

1. Introduction

Today, transportation is considered one of the important components of the national economy, and due to its infrastructural role, it has a great influence on every country has a process of economic growth. This section includes activities that are widespread in all fields. The production, distribution and consumption of goods and services are ongoing and plays an undeniable role in the set of economic activities. Without the transportation network, auxiliary facilities and equipment, and the desired fleet, the general growth and development of the country does not exist and seems impossible. Basically, in the growth and development of the world economy and trade at the current time and its expansion process The role of transportation systems in optimizing costs, travel time, movement speed, safety and service level cannot be underestimated.

Transportation in its comprehensive sense, including management, infrastructure and superstructure (fleet), is one of the basic needs of today's societies, so that in every country, in order to increase its economic, cultural, security and political power, having an extensive and

reliable transportation network is considered one of the basic needs. Some even believe that transportation is one of the basic tools of development. In the past, because transportation was considered a secondary need to fulfill primary needs such as business, travel, employment, etc., it was not properly addressed. But the growth and development of the global economy, the efforts of countries to make optimal use of the capabilities and opportunities at their disposal, and the tightening of competition in global arenas led to the fact that transportation, due to its direct role in reducing the costs of production and access to the market, and finally increasing The ability to compete in the field of international trade, especially for countries with a high volume of foreign trade (exports and imports), has a high position in management, planning and investment, and even research. The effect of such a feature has caused the past attitudes towards transportation to be changed and transportation to be viewed as an important service-economic sector.

The vehicle routing problem (VRP) is an important type of transportation problem, which has many applications in real life such as logistics and transportation. This problem is an NP-hard [1] that needs to

* Corresponding author.

E-mail address: zaahmed@imamu.edu.sa (Z.H. Ahmed).

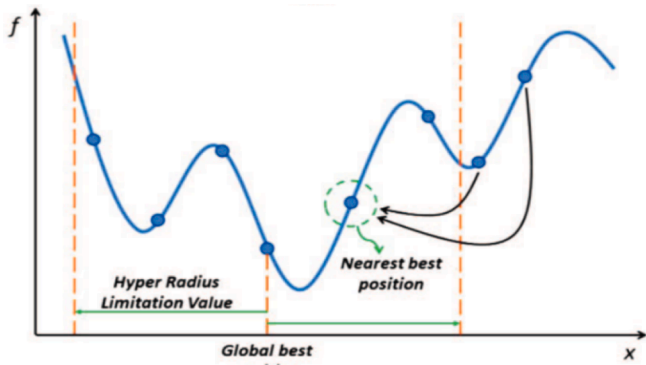


Fig. 1. The schematic representation of hyper radius penalty at the beginning of the optimization process.

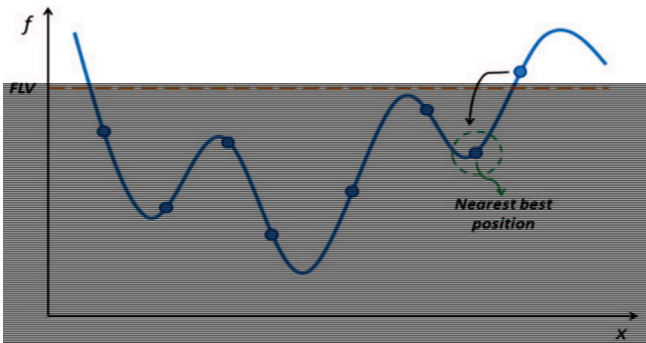


Fig. 2. The schematic representation of FP at the beginning of the optimization process.

find an optimal set of routes for serving a number of customers by some vehicles. In this problem, there are a number of customers in a geographical area with a warehouse so that each customer needs a certain amount of goods that must be delivered to them by a fixed fleet of vehicles. The goal is to determine a set of tours that start and end at the warehouse, provided that:

Each customer is visited exactly once by a vehicle.

The total customer demand of each tour should not exceed the capacity of the vehicle, which is considered Q .

Get the lowest total cost for all vehicle tours.

The emergence of new problems in the real world caused other restrictions to be imposed on the problem and other versions of the open vehicle routing problem (vehicles do not return to the warehouse after completing their mission and retrace their route at the same final nodes to end), the problem of VRP with pickup and delivery of goods (each

customer can have a request for the delivery and receipt of the goods and can solve this request only once or more by means of several vehicles), the problem of heterogeneous VRP (vehicles can be different from each other and different in terms of capacity) and vehicle routing problem with backhauls (some nodes must be visited before some other nodes) to come into being [2–5]. The VRP with time windows (VRPTW) is another difficult extension of the VRP in industry and practice. Here, each customer should be visited by only one vehicle during a defined time interval based on capacity constraints. The objective of the problem is to find the minimum distance covered by a set of vehicles that set to start together from the warehouse and visit some prefixed customers within certain time frames.

Many researchers have proposed exact and approximate methods (including heuristic/metaheuristic methods) for the optimization problem [6–8], and its specialized versions. Exact methods are able to attain optimal solutions, but they are only suggested for the problem instances of small sizes. These algorithms have poor performances for large-sized problem instances because of high computational time. Therefore, researchers prefer to use approximate methods for finding near optimal solutions quickly. Examples of approximate approaches for VRPTW including harmony search algorithm (HAS) [9] memetic algorithm (MA) [10], tabu search (TS) [11], variable neighborhood search (VNS) [12], etc.

Taha et al. [13] presented a combined bat algorithm with a large neighborhood search (LNS) for the VRPTW. In the LNS algorithm, for proper performance in the possible space of the problem and its global search, it used an efficient mechanism called generating a large neighborhood, which resulted in obtaining quality solutions to the problem. In [9], a hybrid algorithm based on the adaptive harmony method was proposed to solve the VRPTW. Since the proposed method has a well performance for global search of the solution space, several neighborhood search algorithms are used to increase the effectiveness of the solution in local search. Also, a hybrid simulated annealing (SA) method for the multi-objective type VRPTW using multiple temperatures was presented in [14]. In this case, two objectives are considered simultaneously, which include minimizing the distance traveled and balancing the routes. The second goal is to minimize the imbalance in the distance traveled by vehicles as well as the total items received by customers. It is observed that the first balance creates justice among vehicle drivers and, as far as possible, causes their working hours to be similar. The second balance is used, for example, for cases where there is a feeling of relative satisfaction among all customers.

Nagata [15] proposed a memetic algorithm to solve the VRP, which was used to propose a hybrid memetic algorithm for the VRPTW. This algorithm uses a penalty function to escape infeasible solutions. Besides, Nagata and Bräysy [16] used some structural algorithms to generate the initial feasible solutions, which were able to generate relatively high-quality solutions at an acceptable time. In addition, the crossover presented in [17] was considered and used for the VRPTW. In other words,

```

Consider initial value for input data including the players position  $X_i$  for  $i = 1, 2, \dots, N$ ,
fitness function  $f(X)$ , stop condition (max-iteration), etc.
Define parameters of algorithm including  $\theta, \lambda, \gamma$  and CMS
Repeat
    Obtain the fitness values for each  $i$ .
    Consider all the players in the field and put the best players in CM
    Identify the player who possess the ball  $X_{ball}^i$ 
    If  $(f(x_i) > FLV^t$  OR  $HD_i > HRLV^t)$ 
        Use a local solution on the nearest best solution and fine a new solution
        Update all parameters including  $\alpha_i, FLV$  and  $HRLV$ .
    Else
        Apply general movements to produce new positions
    End
Until  $(t \leq \text{max-iteration})$ 
    
```

Fig. 3. The pseudo code of football game algorithm.

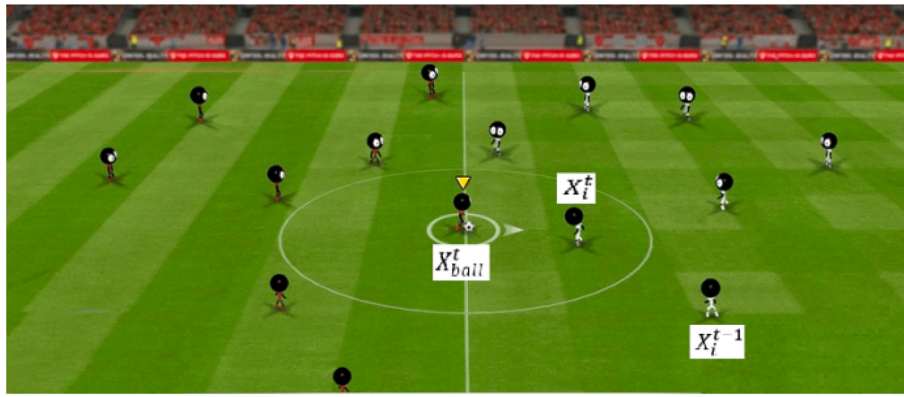


Fig. 4. The positions X_i^t , X_{ball}^t and X_i^{t-1} .

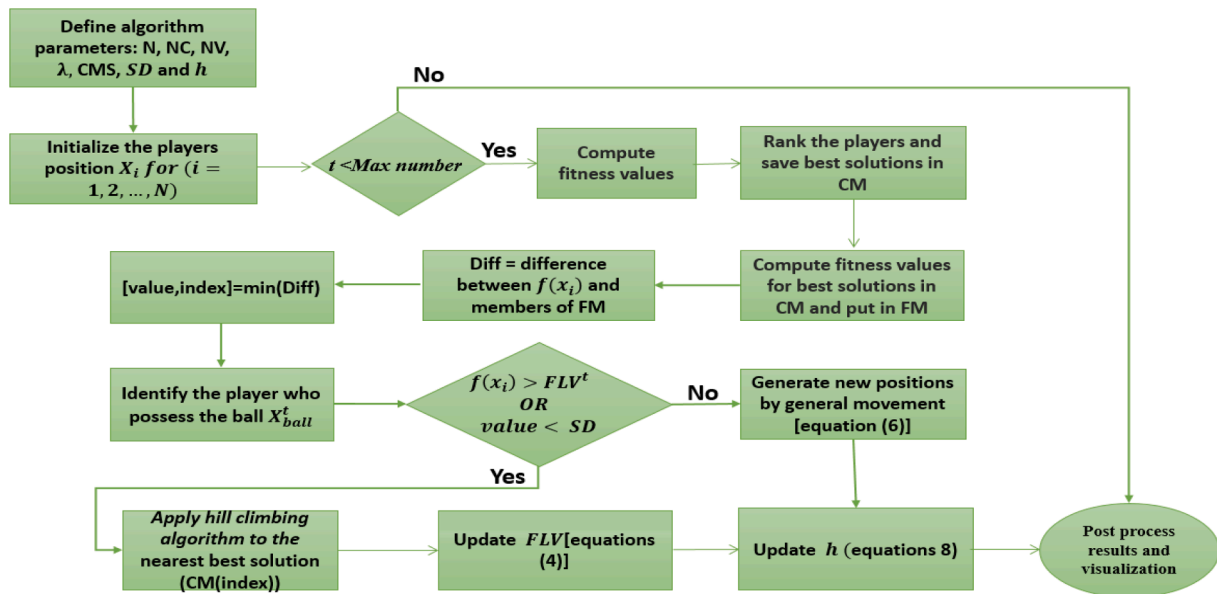


Fig. 5. Hill climbing algorithm.

using a new penalty function tried to prevent the time window from being violated. In addition, a multi-graph model for the VRPTW was proposed in [18] and then they used a hybrid algorithm based on large adaptive neighborhood search to solve this problem. Also, a new data structure and dynamic programming were used to set up the multi-graph, which was able to provide very good quality for answering problems. In [5], the VRPTW was solved by an ACO, and to increase its efficiency, they used a saving algorithm [19], local exchange mechanism and local search. Modifications were done to the algorithm that changed the pheromone setting, the presentation of any new method to enhance the solution further, and the use of a candidate list to select the next ant. These modifications enabled the algorithm to have good efficiency in intensification and diversification, thus preventing premature convergence and moving from local optimization to global optimization.

In [8], to solve the VRPTW, a hybrid search algorithm based on harmony algorithms is proposed which consists of two sub-algorithms HAS-optimizer and HAS-solver. In this algorithm, the HSA-optimizer is first used to be able to optimally configure the components of the second algorithm and create the solution to the problem, and then the HSA-solver algorithm considers the solution obtained by the first algorithm as input and improves it by using several neighborhood search algorithms. The algorithm achieves excellent results in comparison to other heuristic/metaheuristic algorithms and can be used to parameterize other values of heuristics algorithms, configure and upgrade the local

search algorithm as well as upgrade other meta-heuristic algorithms.

The football game algorithm (FGA) is a new metaheuristic for solving continuous global optimization problems proposed in [20]. This algorithm imitates the behavior of football players during a game to find the best positions to score a goal under the supervision of a team coach. In this paper, since the VRPTW is NP – hard, we propose a modified FGA (MFGA) as a new approximate method [21–25] and adapted its operators to solve it as a discrete problem for the first time. Further, we have used hill climbing (HC) method as local search method in which a new solution is generated by three neighborhood methods. The results in the Solomon’s VRPTW benchmarks show that our proposed MFGA is capable to find high quality solutions in comparison to the other heuristic/metaheuristic algorithms.

The paper is arranged as follows: The FGA is introduced in Section 2 and the proposed MFGA approach is explained in Section 3. In Section 4, the design of experiments and the results of the experiments will be discussed and will be compared via the state-of-the-art methods. Lastly, Section 5 presents the discussions and conclusions.

2. Football game algorithm

The FGA is a new metaheuristics that was used firstly for solving continuous global optimization problems. Due to the very good performance of this algorithm, in recent years, it has been tried to use this

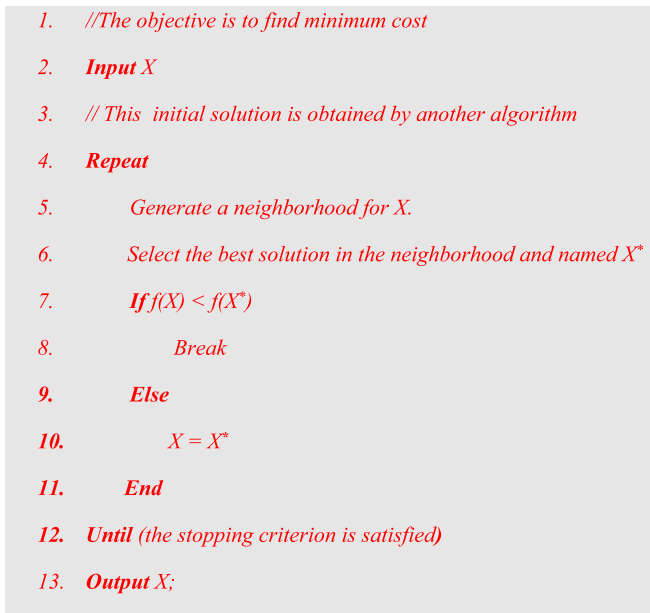


Fig. 6. Flowchart of the proposed algorithm.

Table 1
Variables and parameters of MFGA.

Name	Value
β	Calculated by Equation (7)
λ (inequation 4)	0.9
specified distance (SD)	100
Population size (N)	20
coach memory size (CMS)	N/2
h (inequation 8)	0.7
stopping criterion for MFGA	NI = 10
stopping criterion for HC	NI = 300

algorithm for traveling salesman problem and vehicle routing problem, although it has not been able to achieve the best solutions [26,27]. It was suggested in [20] and inspired the behavior of football players which try to find the best positions in the game. Unlike other existing algorithms, the FGA uses different strategies to create a balance between combination of diversification and intensification. The FGA algorithm uses the collective intelligence of humans, unlike many algorithms that use animal swarm intelligence. In other words, in this algorithm, like a football team whose objective is to score a goal, the goal is to achieve the best possible solution to the problem. In order to be able to use this algorithm in an optimization problem, we need to create a complete simulation between all the components of an optimization problem with a football match as follows:

- (i) The football pitch, football time and football players in the proposed algorithm are the feasible space of the problem, the amount of time to run the algorithm and the initial population of the problem.
- (ii) All the initial population which the algorithm starts working is the same team players who are attacking all the time and want to score as many goals as possible.
- (iii) In this algorithm, it is assumed that all the opposing players in the game are the same local optimum points within the search space. Therefore, these players of this team do not pay attention to the players of the first team and do not try to score goals.

If a player's worthy position is considered the same as the quality response of one of the members of the initial population, the goal is for each player to be around the soccer ball and the ball passes between the

players and reaches the player with the best position. It should be noted that it is the coach's duty to identify the best openers in the right position and to use tactics that will ultimately get the ball to these players. A very important point in the football game, which is also used in the algorithm, is that the coach can replace players who are not of well quality with other players during the game. In other words, in the proposed algorithm, a member of the population who does not have a quality objective function value can be removed from the population, and since the coach knows well solutions during the implementation of the algorithm, a solution with a good objective function value can be replaced. This strategy continues to reach the optimal point or goal until the end of the game, which is the time of the whole algorithm.

2.1. General movement of players

According to the game of football, if a player is not under the orders of his coach, then he/she is either simply walking on the field without a goal or he is moving towards the ball. Of course, in both cases, each player has a goal to reach his best place on the field to have a better position to score. It is noted that during the game, one very important place that other players always wish is the position of the player who has the ball. So, the other goal of the players is to get closer to the player with the ball (X_{ball}^t), receive it and create a better position for their team. As a result, Formula (1) shows the position of the player in repetition t. In this formula, $\beta \in [0,1]$ and $\varepsilon \in [-1,1]$ are two random variables that follow a uniform distribution. $\alpha > 0$ is also a parameter set by the user that takes values based on the problem that the algorithm solves. The value of the alpha parameter can be very important for local or global search of the algorithm because by decreasing this value, the algorithm searches locally and the power of the algorithm increases in intensification, while increasing the algorithm tends to global search or diversification. Gradually decreasing this parameter, the algorithm can be changed from global search at the starting of the algorithm to local search finally.

$$X_i^t = X_i^{t-1} + \alpha_i \varepsilon + \beta (X_{ball}^t - X_i^{t-1}) \tag{1}$$

Also in step t, it shows the player that has the ball. However, although the transfer of the ball between players is generally random, the player or member of the population who has a better position or goal function is more likely to own the ball. It should be noted here that a formula for the variable for considering alpha gradually can be provided by the Formula (2). In this formula, $\theta \in (0,1]$ and α_0 are the amount of constant and initial parameter which are randomly selected.

$$\alpha_i = \alpha_0 \theta^t \tag{2}$$

2.2. Coaching

The coach is very important in the game of football because he/she can achieve a good result by considering the game and using the appropriate tactics and changing players. Therefore, although what has been said about player movement, a team coach plays a key role in moving players and guiding them to more effective locations. Alternatively, the algorithm's memory completes its findings when it is executed, like the coach's memory (CM), and at all times keeps the best solutions and their values as the best positions on the field. Thus, it is very good to allocate some initial population size in the algorithm as the coach memory size (CMS) so that it can store these important solutions. The general purpose of using this memory is that the algorithm uses the following two strategies, like the natural version, to achieve the main objective of scoring goals on the football field and achieving the best possible solution in the algorithm.

Attack strategy: Because the main goal of the football game is to score goals and the general way to achieve this goal is that the defenders and midfielders have to move towards the opponent's field and crush that team in order to get better opportunities for themselves. Now,

Table 2
The results obtained by MFGA.

Data set	BKS		MFGA		Std	NV	Gap	CoVar	
	TD	NV	Ref	BS					AS
C1-01	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-02	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-03	828.06	10	[32]	828.06	832.03	4.19	10	0.00	0.50
C1-04	824.78	10	[32]	824.78	850.10	10.01	10	0.00	1.17
C1-05	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-06	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-07	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-08	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
C1-09	828.94	10	[32]	828.94	828.94	0.00	10	0.00	0.00
Avg (C1)	828.38	10.0		828.38	831.63	1.57	10.00	0.00	0.19
R1-01	1642.87	20	[30]	*1584	1609.50	14.00	19	-3.58	0.86
R1-02	1472.62	18	[30]	*1374.2	1410.90	15.77	16	-6.68	1.11
R1-03	1213.62	14	[32]	*1158.90	1191.30	15.85	13	-4.50	1.33
R1-04	982.01	10	[32]	996.95	1007.10	7.29	11	1.52	0.72
R1-05	1360.83	15	[30]	*1355.30	1365.70	7.23	15	-0.40	0.52
R1-06	1241.518	13	[30]	*1212.10	1231.30	6.32	13	-2.36	0.51
R1-07	1076.125	11	[30]	*1075.50	1084.10	2.69	11	-0.05	0.24
R1-08	948.573	10	[30]	959.88	965.35	4.14	10	1.19	0.42
R1-09	1151.839	13	[30]	1155.80	1169.10	9.70	12	0.34	0.82
R1-10	1080.36	11	[32]	1092.40	1098.80	3.72	12	1.11	0.33
R1-11	1053.49	12	[30]	1059.20	1066.80	8.11	12	0.54	0.76
R1-12	953.63	10	[32]	979.05	982.42	4.01	10	2.66	0.40
Avg (R1)	1181.45	13.08		1166.90	1181.90	8.23	12.83	-1.23	0.69
RC1-01	1623.58	15	[32]	*1595.90	1626.90	21.11	15	-1.70	1.29
RC1-02	1466.84	14	[30]	*1460.90	1475.60	8.74	14	-0.40	0.59
RC1-03	1261.67	11	[28]	1292.60	1298.60	0.01	11	2.45	0.00
RC1-04	1135.48	10	[29]	*1135.00	1150.80	12.19	10	-0.04	1.05
RC1-05	1518.60	16	[30]	*1510.10	1522.70	12.25	15	-0.55	0.80
RC1-06	1377.35	13	[30]	*1367.20	1390.00	13.43	13	-0.73	0.96
RC1-07	1212.83	12	[30]	1215.90	1217.90	3.75	12	0.25	0.30
RC1-08	1117.52	11	[30]	1120.10	1132.10	10.59	11	0.23	0.93
Avg (RC1)	1339.24	12.75		1337.20	1351.80	10.25	12.62	-0.15	0.75
C2-01	591.56	3	[32]	591.56	591.56	0.00	3	0.00	0.00
C2-02	591.56	3	[32]	591.56	591.56	0.00	3	0.00	0.00
C2-03	591.17	3	[32]	591.17	591.17	0.00	3	0.00	0.00
C2-04	590.60	3	[32]	590.60	612.11	15.04	3	0.00	2.45
C2-05	588.88	3	[32]	588.88	588.88	0.00	3	0.00	0.00
C2-06	588.49	3	[32]	588.49	588.49	0.00	3	0.00	0.00
C2-07	588.29	3	[32]	588.29	588.29	0.00	3	0.00	0.00
C2-08	588.32	3	[32]	588.32	588.32	0.00	3	0.00	0.00
Avg (C2)	589.86	3.0		589.86	529.54	1.88	3.0	0.00	0.35
R2-01	1147.80	9	[31]	1168.70	1180.30	8.5	6	1.82	0.72
R2-02	1039.32	5	[31]	1042.40	1044.50	2.11	6	0.29	0.20
R2-03	874.87	5	[31]	893.97	902.37	6.06	5	2.18	0.67
R2-04	735.80	3	[31]	744.02	748.50	6.36	4	1.11	0.84
R2-05	954.160	5	[31]	969.42	979.07	6.31	5	1.59	0.64
R2-06	884.25	4	[31]	*880.60	898.21	14.35	5	-0.41	1.59
R2-07	797.99	4	[31]	822.84	823.30	0.66	4	3.11	0.08
R2-08	705.62	3	[31]	736.55	737.00	0.64	4	4.38	0.08
R2-09	860.11	5	[30]	905.11	911.17	8.56	5	5.23	0.93
R2-10	910.98	5	[31]	937.06	946.31	13.17	4	2.86	1.39
R2-11	755.82	4	[31]	815.09	817.39	1.98	4	7.84	0.24
Avg (R2)	878.79	4.73		901.43	908.01	6.24	4.72	2.57	0.68
RC2-01	1266.11	9	[31]	1274.80	1303.30	5.86	6	0.68	0.44
RC2-02	1096.75	8	[31]	1115.70	1123.80	5.99	5	1.72	0.53
RC2-03	926.89	5	[31]	945.90	955.04	12.19	5	2.05	1.27
RC2-04	786.38	4	[31]	803.91	804.90	0.61	4	2.22	0.07
RC2-05	1157.55	7	[31]	1209.50	1210.10	4.02	6	4.48	0.33
RC2-06	1056.21	7	[31]	1098.00	1099.10	11.2	5	3.95	1.01
RC2-07	966.08	7	[31]	1010.40	1011.90	9.05	5	4.58	0.89
RC2-08	779.84	4	[30]	810.04	840.10	20.00	4	3.87	2.38
Avg (RC2)	1004.48	6.38	-	1033.50	1043.50	8.61	5.00	2.88	0.82
Avg (all)	978.00	8.50	-	983.20	-	6.21	8.23	-	0.57

because the opposing players in the proposed algorithm are the same quality solutions, so the goal is to bring low-quality solutions in the algorithm closer to the solutions that play the role of local optimizers. These optimizations are attributed so that the difference of their objective function is less than a certain value of hyper radius limitation value (HRLV) (Fig. 1), which according to Formula (3) this value gradually decreases. It should be noted that if for each member of the population, hyper distance (HD) is the distance to the best position,

some members of the population with a value higher than HRLV, move to the nearest quality solutions, which is called the hyper radius penalty (HRP) strategy [31].

$$HRLV^t = HRLV_{min} + \gamma(HRLV^{t-1} - HRLV_{min}). \tag{3}$$

where $\gamma \in (0.1)$ is the reduced constant of HRLV.

Substitution Strategy: This strategy is another way to further

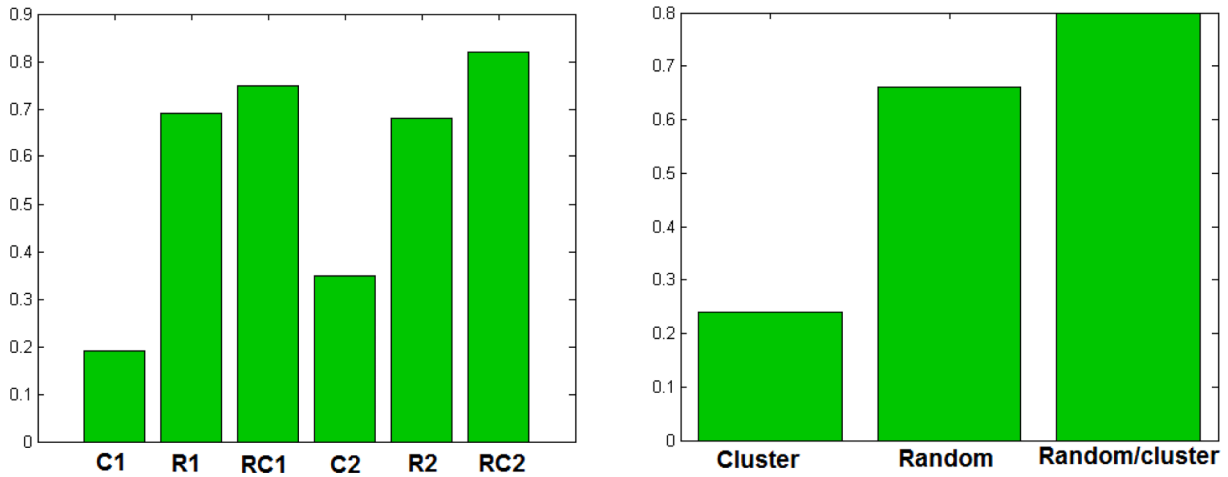


Fig. 7. Performance of MFGA on different types of datasets with respect to coefficient of variation.

improve the team to score, in which players who do not help the team much are replaced by other players to give the team more power to score. In the proposed algorithm, this solution is also considered, and in each iteration, poor quality solutions are removed from the popularization of the problem and the algorithm replaces them with better solutions to the problem. The specific amount on which the switch can be made is called the fitness limitation value (FLV), which replaces the solutions that have values worse than this value in each iteration. Since the solution quality is expected to improve during the algorithm execution, this value is supposed as a variable that is number of algorithm's iterations, its value improves (this value is more in maximization problems and less in minimum problems). This placement is shown in Fig. 2 using the FLV value, in which this strategy is called the efficient fitness penalty (FP) method.

$$FLV^t = FLV_{min} + \lambda(FLV^{t-1} - FLV_{min}). \quad (4)$$

Here, λ has the same role as γ has for Formula (4). In fact, the coaching section is the same as the local search section of the algorithm. After applying the strategies, the new position for players out of bounds is achieved by using random walk from the nearest best solution to their previous position (Formula (5)).

$$X_{new} = X_{nearestbest} + \alpha_i \epsilon \quad (5)$$

The pseudo code of the FGA is shown in Fig. 3.

3. The proposed method

The FGA is a population-based algorithm and can also be categorized in the memory usage algorithm recently introduced in 2016 to solve optimization problems. In this study, we suggest a modified FGA (MFGA) for solving the VRPTW as a discrete optimization problem. In the proposed MFGA, each player represents a VRPTW solution, each solution has a number of routes, and each route contains a number of customers serviced by a vehicle according to the constraints defined in Section 2. The initial population of players on the ground is randomly generated. To create a solution, we generate an empty route in the first step, then an un-routed customer is randomly selected and added to the current route that does not violate the VRPTW. Now, among the customers who have not been met by any vehicle so far, the closest customer is considered to be the last selected customer. Provided that no restrictions are violated, this customer will be added to the relevant route. In some iterations, it may not be possible to add a customer to the current route with the constraints of the problem, where the construction of the desired route is completed. This operation continues for all unmet customers until they all get in the way. As already said, each player in FGA randomly moves from his previous position (X_i^{t-1}) to find

a better position and goes toward the player that has the ball (X_{ball}^t) to receive it (general movement) or be guided by the coach to attack or substitute. In MFGA, the general movement of players is modeled with Formula (6). In this formula, player X_i^t moves to get a better position towards the previous position X_i^{t-1} and the position of the player who holds the ball X_{ball}^t .

$$X_i^t = \text{Replacing the route from } X_{ball}^t \text{ in } X_i^{t-1}. \quad (6)$$

where β specifies the number of routes to be replaced. In other words, we have modeled the player's move, toward his previous position and the ball, with Equation 15, in which the new position of each player (X_i^t) is generated by randomly replacing the number of routes of the player who has the ball (X_{ball}^t) in the routes of his previous position (X_i^{t-1}). β should be a number between the lowest (one route) and the maximum number of routes of two players X_{ball}^t and X_i^{t-1} (MR). So, we compute β using Formula (7), which gives us a number between 1 and MR. Also, the number of these routes has a direct impact on the exploration and exploitation of the proposed algorithm. Because VRPTW is a constrained problem and the routes in the previous position of players (X_i^{t-1}) is replaced to create a new position of player (X_i^t), the likelihood of duplication and loss of customers is high. That's why the solution is often infeasible, and routes must be repaired. Therefore, if β is a large number, more routes from the solution would be repaired, and the solution diversity and exploration would increase. If β is a small number, it means that less routes are replaced. It means that the number of routes to be repaired is less. Hence, exploitation of the search space in the algorithm increases. Furthermore, to create a balance between the exploration and exploitation capability in the MFGA, we control the value of parameter β with h , which decreases with increasing number of iterations using geometric scheduled by Formula (8):

$$\beta = 1 - h(1 - MR) \quad (7)$$

$$h = h \times \alpha. \quad (8)$$

Here, α is a reduction factor that we fixed to 0, and MR shows the maximum number routes of two players X_{ball}^t and X_i^{t-1} calculated by using Formula (9).

$$MR = \min(n_{routes}(X_{ball}^t), n_{routes}(X_i^{t-1})). \quad (9)$$

where n_{routes} specifies the number of routes for each position.

The generated new position X_i^t by Formula (6) is usually infeasible, because the VRPWT is a constrained problem and by replacing the routes in X_i^t to create a new position, the likelihood of duplication and loss of customers is high. For this reason, the below mechanism for

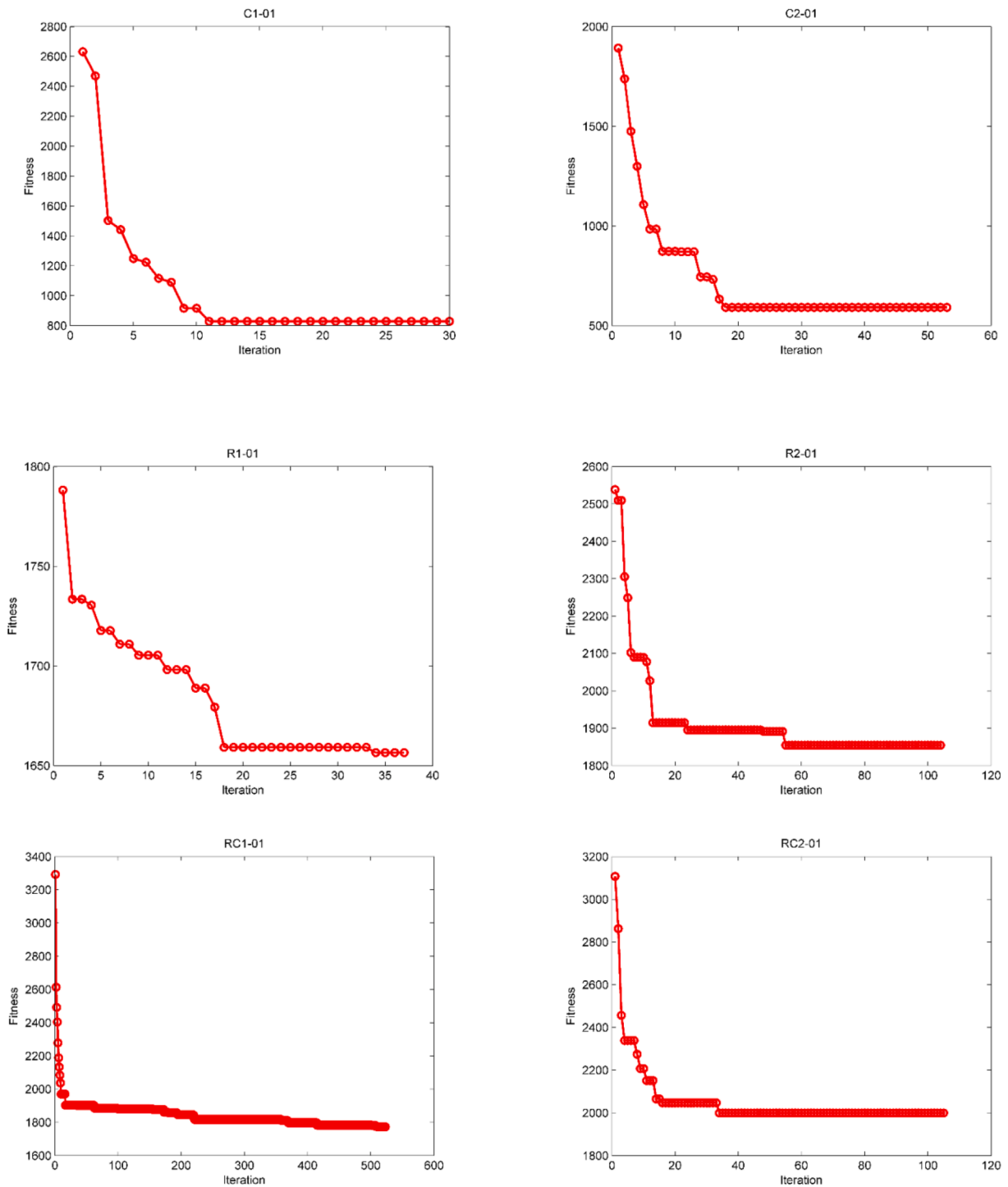


Fig. 8. The Changes of fitness relative to the increase in the number of iterations in FGA.

checking the feasibility of generated solutions is employed:

1. Identify duplicate and missing customers.
2. Remove duplicate customers from the latest routes.
3. Insert the missing customers to the first routes that can be inserted in it without violating the constraints of VRPTW.

In repairing infeasible solution, some duplicated customers are often found in the solution. If such customers are available, we need to remove duplicate customers from the routes to satisfy the constraint of the problem (each customer is served only once). Therefore, one of the

duplicate customers is kept and its repetitions are removed from the rest of the routes. For simplicity reasons, we assume that the duplicate customer, appearing in the first route, is kept and the rest are removed from the next routes. Then, lost customers, if available, are added to existing routes, so that missing customers will be added to the first route that can be inserted without violating the constraints. Otherwise, a new route will be created to route it. For example, suppose that we have two positions X_{ball}^t and X_i^{t-1} in Fig. 4 and the position of player X_i^t is created using Formula (6).

We suppose the positions shown in Fig. 4 that are considered as

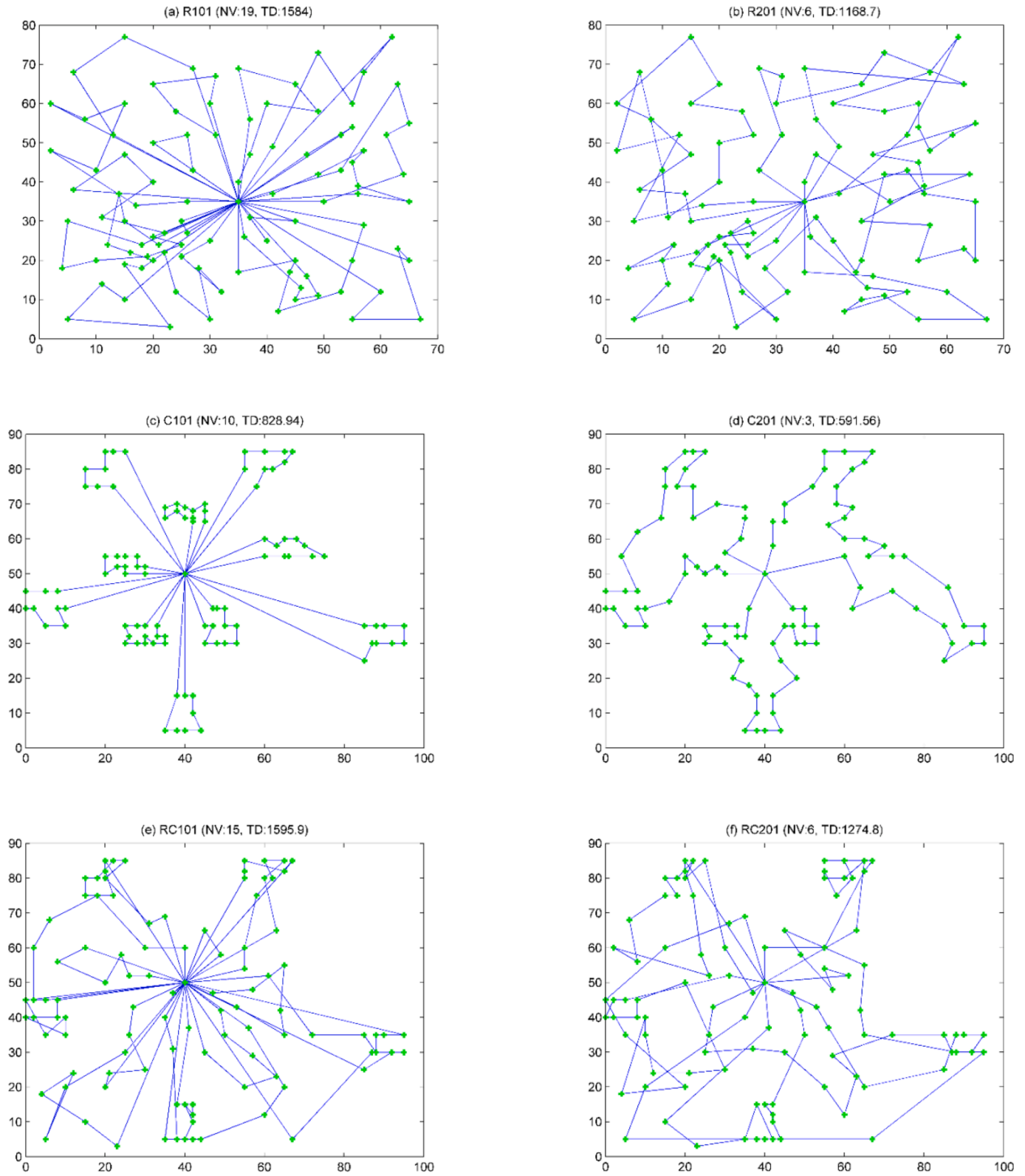


Fig. 9. Schematic representation of the best solutions found by FGA on 6 instances.

Table 3
Comparisons of heuristics applied to Solomon instances.

Instance	CPLA		PITSH		HSFLA		S-PSO		ACO-TS		MFGA	
	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD	NV	TD
R1-01	19	1656.20	19	1650.80	19	1650.80	19	1652.001	19	1655.03	19	1584.00
R1-02	17	1502.00	17	1486.12	17	1486.12	17	1500.809	18	1491.18	16	1374.20
R1-03	13	1295.60	13	1294.23	13	1292.67	14	1242.649	14	1243.22	13	1158.90
R1-04	9	1017.40	10	981.20	9	1007.31	10	1042.216	10	982.01	11	996.95
R1-05	14	1381.90	14	1377.11	14	1377.11	14	1385.082	16	1380.44	15	1355.30
R1-06	12	1258.80	12	1252.62	12	1252.03	12	1294.869	13	1265.36	13	1212.10
R1-07	10	1117.90	10	1104.66	10	1104.66	11	1123.981	11	1100.25	11	1075.50
R1-08	9	976.06	9	963.99	9	960.88	10	1011.682	9	958.66	10	959.88
R1-09	11	1229.70	11	1194.73	11	1194.73	12	1211.630	12	1101.99	12	1155.80
R1-10	10	1196.50	10	1118.84	10	1118.84	11	1190.362	12	1119.53	12	1092.40
R1-11	10	1123.60	10	1096.73	10	1096.73	11	1102.987	12	1091.11	12	1059.20
R1-12	9	1030.00	9	989.27	9	982.14	10	1029.124	10	974.73	10	979.05
Ave	11.92	1232.10	12	1209.19	11.92	1210.34	12.58	1232.300	13.0	1197.00	12.83	1166.90
C1-01	10	828.94	10	828.94	10	828.94	10	828.94	10	828.93	10	828.94
C1-02	10	828.94	10	828.94	10	828.94	10	829.71	10	828.94	10	828.94
C1-03	10	828.06	10	828.06	10	828.06	10	851.37	10	828.06	10	828.06
C1-04	10	824.78	10	824.78	10	824.78	10	868.52	10	828.2	10	824.78
C1-05	10	828.94	10	828.94	10	828.94	10	828.94	10	828.9	10	828.94
C1-06	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94
C1-07	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94	10	828.94
C1-08	10	828.94	10	828.94	10	828.94	10	828.94	10	830.94	10	828.94
C1-09	10	828.94	10	828.94	10	828.94	10	828.94	10	829.22	10	828.94
Ave	10.00	828.38	10.00	828.38	10.00	828.38	10.00	835.92	10.00	829.01	10.00	828.38
RC1-01	15	1626.10	14	1696.95	14	1696.95	15	1641.204	14	1650.14	15	1595.90
RC1-02	13	1486.20	12	1554.75	12	1554.75	13	1514.952	13	1514.85	14	1460.90
RC1-03	11	1268.80	11	1261.67	11	1261.67	11	1294.739	11	1277.11	11	1292.60
RC1-04	10	1136.30	10	1135.48	10	1135.48	10	1190.545	10	1159.37	10	1135.00
RC1-05	14	1542.30	13	1633.72	13	1629.44	14	1603.707	15	1617.88	15	1510.10
RC1-06	12	1394.10	11	1424.73	11	1424.73	12	1410.931	13	1387.63	13	1367.20
RC1-07	11	1234.10	11	1232.20	11	1230.48	11	1249.795	11	1280.01	12	1215.90
RC1-08	10	1155.10	10	1147.69	10	1139.82	11	1181.870	11	1157.44	11	1120.10
Ave	12	1355.40	12	1385.90	11.5	1384.16	12.13	1385.500	12.25	1380.60	12.62	1337.20
R2-01	4	1253.02	4	1252.37	4	1252.88	4	1274.97	7	1214.22	6	1168.70
R2-02	4	1086.08	3	1191.70	3	1192.27	3	1247.03	5	1105.20	6	1042.40
R2-03	3	945.80	3	941.08	3	939.95	3	1052.71	4	960.14	5	893.97
R2-04	3	752.13	2	825.52	2	826.31	3	844.16	4	771.47	4	744.02
R2-05	3	1017.93	3	994.43	3	994.80	3	1061.46	4	1050.26	5	969.42
R2-06	3	920.37	3	906.14	3	906.59	3	1016.35	4	954.85	5	880.60
R2-07	3	815.26	2	890.61	2	891.14	3	946.78	3	870.33	4	822.84
R2-08	2	729.42	2	726.82	2	727.60	2	834.72	3	777.72	4	736.55
R2-09	3	916.33	3	909.16	3	909.32	3	1003.19	3	934.21	5	905.11
R2-10	3	943.10	3	939.37	3	939.53	3	1040.54	5	949.02	4	937.06
R2-11	3	767.82	2	885.71	2	886.17	3	861.323	4	877.55	4	815.09
Ave	3.09	922.48	2.73	951.17	2.70	951.51	3.00	1016.700	4.18	951.36	4.72	901.43
C2-01	3	591.56	3	591.56	3	591.56	3	591.56	3	591.56	3	591.56
C2-02	3	591.56	3	591.56	3	591.56	3	591.56	3	591.56	3	591.56
C2-03	3	591.17	3	591.17	3	591.17	3	591.17	3	593.25	3	591.17
C2-04	3	590.60	3	590.60	3	590.60	3	615.43	3	595.55	3	590.60
C2-05	3	588.88	3	588.88	3	588.88	3	588.88	3	588.88	3	588.88
C2-06	3	588.49	3	588.49	3	588.49	3	588.88	3	588.49	3	588.49
C2-07	3	588.29	3	588.29	3	588.29	3	591.35	3	588.88	3	588.29
C2-08	3	588.32	3	588.32	3	588.32	3	588.49	3	588.03	3	588.32
Ave	3.00	589.86	3.00	589.86	3.00	589.86	3.00	593.41	3.00	590.78	3.00	589.86
RC2-01	4	1435.27	4	1406.94	4	1406.94	4	1423.52	5	1279.65	6	1274.80
RC2-02	4	1162.80	3	1367.09	3	1365.64	4	1193.59	5	1157.02	5	1115.70
RC2-03	3	1062.32	3	1050.64	3	1049.62	3	1123.42	6	1046.33	5	945.90
RC2-04	3	799.08	3	798.46	3	798.46	3	894.12	4	847.33	4	803.91
RC2-05	4	1303.68	4	1297.65	4	1297.65	4	1321.43	5	1334.55	6	1209.50
RC2-06	3	1155.33	3	1153.61	3	1146.32	3	1307.90	5	1112.20	5	1098.00
RC2-07	3	1095.37	3	1061.14	3	1061.14	3	1130.37	5	1078.52	5	1010.40
RC2-08	3	834.16	3	828.71	3	828.14	3	958.24	3	911.15	4	810.04
Ave	3.38	1106.00	3.25	1120.53	3.25	1119.24	3.38	1169.10	4.75	1095.80	5.00	1033.50

follows.

X_i^{t-1} :	Route 1	5	0	1	3	4	0	6	8	0	7	2	0
	Route 2												
	Route 3												
	Route 40												
X_{ball}^t :	Route 1	4	3	5	2	0	8	0	7	6	1	0	
	Route 2												
	Route 30												

These solutions consist of eight customers, 0 is the depot and each route is served by a vehicle. The new position X_i^t in MFGA is created using Formula (6) as follows:

- (i) Calculate parameter β that specifies the number of routes to be replaced.

$$MR = \min(n_{routes}(X_{ball}^t), n_{routes}(X_i^{t-1})) = \min(3,4) = 3$$

Table 4
Comparison of the performance of MFGA and different heuristics.

Algorithms		Group of instances		C1	C2	RC1	RC2
		R1	R2				
BKS	TD	1181.45	878.79	828.38	589.86	1339.24	1004.48
	NV	13.08	4.73	10.00	3.00	12.75	6.38
	%TD	+2.23	+11.19	+1.22	+2.64	+3.19	+9.42
HSFLA	TD	1210.34	951.51	828.38	589.86	1384.16	1119.24
	NV	11.92	2.70	10.00	3.00	11.50	3.25
	%TD	+2.45	+8.28	0.00	0.00	+3.35	+11.42
CPLA	TD	1232.10	922.48	828.38	589.86	1355.40	1106.00
	NV	11.92	3.09	10.00	3.00	12.00	3.38
	%TD	+4.29	+4.97	0.00	0.00	+1.21	+10.11
PITHS	TD	1209.17	951.17	828.38	589.86	1385.9	1120.53
	NV	12.00	2.73	10.00	3.00	12.00	3.25
	%TD	+2.35	+8.24	0.00	0.00	+3.48	+11.55
S-PSO	TD	1232.30	1016.70	835.92	593.42	1385.5	1169.1
	NV	12.58	3.00	10.00	3.00	12.13	3.38
	%TD	+4.30	+15.6	+0.91	+0.60	+3.45	+16.39
ACO-TS	TD	1197.00	951.36	829.01	590.78	1380.60	1095.80
	NV	13.00	4.18	10.00	3.00	12.25	4.75
	%TD	+1.32	+8.26	+0.08	+0.16	+3.09	+9.09
MFGA	TD	1166.90	901.43	828.38	589.86	1337.20	1033.50
	NV	12.83	4.72	10.00	3.00	12.26	5.00
	%TD	-1.23	+2.57	0.00	0.00	-0.15	+2.88

$$\beta = \lfloor 1 - h(1 - MR) \rfloor = \lfloor 1 - 0.9 \times (1 - 3) \rfloor = \lfloor 2.8 \rfloor = 2$$

So, the two routes should be replaced.

- (ii) First, X_i^{t-1} is copied to X_i^t , then two routes are randomly selected from X_{ball}^t and copied in two routes of X_i^t . In this example, routes 1 and 2 are two randomly selected routes of X_{ball}^t and are replaced by two randomly selected routes 1 and 3 of the X_i^t .

X_i^t :	Route 1	Route 2	Route 3	Route 4	8	0	1	3	4	0	4	3	5	2	0	7	2	0	40
-----------	---------	---------	---------	---------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----

- (iii) The new position X_i^t is indeed infeasible because customers 2, 3 and 4 are duplicate and customer 6 is missed. Therefore, a repair mechanism is necessary to repair the new generated positions that are infeasible. Besides, a mechanism must eliminate repeated customers and reassign the missed customers from the new position while maintaining VRPTW constraints. In general, to repair an infeasible position, first, one of each duplicate customer will be kept, and the rest of their repetitions will be removed from other routes. Second, missing customers are added to the first route if they can be inserted without violating any constraints of VRPTW problem (like customers 5 and 2 in Fig. 5). If we cannot put them in any of the existing routes, a new route will be created and added to it.

X_i^t :	Remove duplicate customers from the last routes																		
X_i^t :	Insert missing customers on routes																		
	Route 1	Route 2	Route 3	Route 4															
0					8	0	1	3	4	0	5	2	0	7	0				
Route 1	Route 2	Route 3	Route 4																
0					8	6	0	1	3	4	0	5	2	0	7	0			

So far, the general movement of players has been modeled by Formula (6). Now we model the guidance of the players by the coach, which includes the two below attack and substitute strategies.

Strategy attacking: Because each player moves towards player who

has a better position in the strategy of attack, this strategy cannot be used directly for VRPTW. Since the position of each member in a search space for a continuous problem is a point, but in VRPTW as a discrete problem, the position of each player consists of different points of the customers that together form a solution. For this reason, in the attack strategy for VRPTW, instead of comparing the position of the solutions, we use the comparison the objective function value of the positions. In such a way that the player will be pushed towards a better player whose fitness function difference is less than a specified distance (SD).

Strategy substitution: In MFGA, this strategy is the same as FGA by Formula (4), and the coach uses substitution option to change the poor players with a better player. Each player in the team (population) who has more fitness value than FLV will be replaced with another one around the nearest best player (solution). In this formula, FLV_{min} has the lowest fitness function value in each repetition and FLV^{t-1} is the highest fitness function value in the first iteration. Also λ is a reduction factor and fixed to 0.9.

In this study, as the coaching is the local search of the algorithm, the famous *Hill Climbing* (HC) method is used to enhance the MFGA exploitation by applying it to the feasible location constructed in the preceding step. In this step, after constructing any feasible solution, HC is used to improve the solution. HC is aimed to improve the exploitation in the neighborhood of the solution.

HC is a well-known local search method that begins through an initial solution, X, which then produces successively a neighbor solution X' by applying a neighborhood construct. In FGA, three neighborhood methods – *exchange*, *relocate* and *end-customer-exchange*, are applied to produce a neighboring solution. Then, if the solution X' is found better than X, then X is replaced by X'. Otherwise, delete X' and begin another iteration (see Fig. 5) [31]. The search process would be replicated till the terminating condition is met.

HC creates a solution applying the following three neighborhood constructs [32]:

1. Exchange operator that exchanges two customers placed on different routes.
2. Relocate operator that moves one customer from a route to another one.
3. End-customer exchange operator that exchanges two end customers on different routes.

Fig. 6 shows the flowchart of the proposed method.

4. Computational experiments

In this section, Solomon's VRPTW benchmark [32] is utilized to evaluate the effectiveness of MFGA. This benchmark consists of 56 samples with 100 customers, divided into six groups based on the customer's geographical location (R1, R2, C1, C2, RC1, RC2). In fact, the location of customers in this benchmark is both random (R1 and R2) or clustered (C1 and C2) or a combination of random and clustered locations (RC1 and RC2).

The proposed MFGA is coded in MATLAB 7.8 and runs on a PC with Windows 7 operating system, 2.40 GHz three-core Intel (R) processor, and 2 GB of RAM. The performance of metaheuristic methods significantly depends on proper choice of the most significant parameters, for example, size of the population (N), number of customers (NC), number of vehicles (NV), λ , coach memory size (CMS), specified distance (SD) and h , in our proposed MFGA. The values of NC and NV are determined with respect to the experimental data set and the parameter of h is used to control the intensification and diversification of MFGA directly. That is why choosing the appropriate values for these two parameters is a key value of the parameter setting. Besides, the MFGA performance is evaluated with values of 10, 15, 20 and 25 for parameter N, values of 50, 100, 150 and 200 for parameter SD, and values of 0.9, 0.8 and 0.7 for parameter h . The best values for these parameters are shown in Table 1 according to various experiments. The stopping criterion is adopted according to the number of non-improved iterations (NI), which has been fixed to 10 and 300 iterations without improvement for MFGA and HC respectively. Also, MFGA runs 30 times on each tested instance to obtain the best results.

Table 2 shows the characteristics of the six categories of test problems listed above, which include R1, R2, C1, C2, RC1, and RC2. The results of our proposed MFGA are then compared against the best results found for these instances. These instances are divided into three categories based on their first letter. In the first category R1 and R2, where 100 customers are randomly distributed around the warehouse, while in the second category C1 and C2, where same number of customers are scattered in clusters around the warehouse. Finally in the last category, there is a combination of the first and second categories. In other words, in the third category RC1 and RC2, where some customers are randomly, and others are in clusters around the depot. On these six categories instances, our algorithm is tested to show its performance.

In the first column of this table, the names of the instances of these six categories are given. The second, third and fourth columns summarize information for obtaining the best-known solution (BKS) values reported in the literature, particularly, distance traveled (TD), NV and the reference in which these results are reported, respectively. It is noted that some literatures [29–31] have reported several BKS without providing details, showing the solutions in the appendix. Because none of the solutions have been approved yet and some others may be wrong, their results are not presented here. The fifth to tenth columns report the obtained results using by our proposed MFGA. In this table, each of the instances is considered first and the algorithm is executed 31 times on each of them. Now the best solution (BS), the average solution (AS) and the standard deviation (Std) within 31 runs are presented in columns five to seven, respectively. The eighth column reports the number of vehicles (NV). The ninth column presents the percentage deviation (Gap) of the obtained solution from the BKS (using Formula (10)) and finally the tenth column presents the coefficient of variance (CoVar). This value is the percentage of standard deviation from the mean of the obtained solutions. Here, $d1$ and $d2$ are BS by MFGA and BKS respectively for the instances. It is noted that a negative value suggests that MFGA was able to upgrade BKS.

$$Gap = \left(\frac{d1 - d2}{d2} \right) \times 100 \quad (10)$$

The results indicate that the proposed MFGA could obtain very good

solutions to these standard instances so that the best solutions are obtained for 17 instances, which are shown in boldface. Further, the algorithm was able to improve best solutions to another 12 instances, which are shown in boldface and marked by a star. In addition, the average standard deviation for these instances is only 6.21 and the average variance coefficient is 0.57% (less than 1%), which is very high quality solution. On an overall average, the total travel distances for R1 instances are reduced by 1%. Although in this work we only aim to minimize the total distance traveled, MFGA has been able to achieve results with the maximum difference of two vehicles compared to BKSs. Even R1-01, R1-02, R1-03 and RC1-05 samples have improved with fewer vehicles. Besides, MFGA has the same performance as BKSs on the data sets C1 and C2 in which number of vehicles in C1 and C2 are 10 and 3, respectively. The percentage deviation demonstrates that MFGA results are relatively stable across all tested datasets and even acts 1.23% and 0.15% better than the best known on R1 and RC1 datasets respectively and only 2.57% and 2.88% away from BKSs on R2 and RC2 respectively.

The performance of MFGA based on the average coefficient of variation on different types of datasets is shown in Fig. 7. The best performance of MFGA on the C1 dataset with coefficient of variation equal to 0.19, followed by C2 dataset (0.35), R2 dataset (0.68), R1 dataset (0.69), RC1 dataset (0.75), and finally RC2 dataset (0.82) are presented in this figure. MFGA showed best performance on clustered datasets with an overall average coefficient of variation equal to only 0.24, followed by the random datasets (0.66), and finally the mixed datasets (0.80). Fig. 8 shows fitness variation with increasing number of iterations for six samples. As shown in this figure, FGA has been successful in all samples and converges to optimal response after several iterations. The Fig. 9 shows a schematic representation of the best solutions found for the six samples (R101, R201, C1-01, C201, RC101 and RC2-01).

To evaluate the effectiveness of MFGA, the obtained results are compared against the results produced by following state-of-the-art methods proposed for the VRPTW.

1. **CPLA**: Cooperative population learning algorithm of Barbuca [33].
2. **PITSH**: Parallel iterated tabu search heuristic of Cordeau and Maischberger [34].
3. **HSFLA**: Novel hybrid shuffled frog leaping algorithm of Luo et al. [35].
4. **S-PSO**: Discrete particle swarm optimization approach of Gong et al. [36].
5. **ACO-TS**: A hybrid approach, which consists of ant colony optimization (ACO) and tabu search of Yu et al. [37].

The detailed experimental results are reported in Table 3. All instances are tested and repeatedly solved 31 times. The first column of these tables is the instance name, the next columns include distance traveled (TD) and number of vehicles (NV) values for CPLA, PITSH, HSFLA, S-PSO, ACO-TS and MFGA. For each of the problem groups (R1, R2, C1, C2, RC1, and RC2), TD and NV are reported, and the best results are represented in bold face. Looking at the results shown in Table 4, the proposed MFGA presents very competitive results with respect to the compared methods, so that MFGA has shown better performance than other comparable methods on 46 instances of 56 instances.

Table 4 presents the average number of vehicles (NV) and the average best quality solutions obtained (TD: total distances) via the application of MFGA compared to the five algorithms presented above, including HSFLA, CPLA, PITSH, S-PSO and ACO-TS on Solomon's benchmark datasets (R1, R2, C1, C2, RC1 and RC2). Each row in the table consists of three parts called NV, TD, and %TD (%TD is the percent deviation between the algorithms and best-known solutions). The percentage deviation demonstrates that MFGA results are stable across all tested datasets and even acts 1.23% and 0.15% better than BKSs on R1 and RC1 datasets, respectively. In the problems of C1 and C2, MFGA achieved the best published results in all datasets and MFGA is 2.57 %,

and 2.88% away from BKSs respectively only in the problems of R2 and RC2. Also, compared with other methods in this table, the MFGA has the best performance on the R1, R2, RC1 and RC2 datasets. It is finally concluded that although MFGA did not manage to beat the best-known results for all datasets, the obtained results by the proposed algorithm for these datasets are very competitive with the mentioned algorithms.

5. Conclusions and future research

In this study, we have suggested MFGA to effectively solve the VRPTW, whose objective is to minimize the total distance traveled by the fixed number of vehicles. Since the standard FGA is a continuous optimization algorithm, we first adapted the operators of FGA to tackling the VRPTW and then proposed an FGA (MFGA) to solve it. MFGA uses HC as local search algorithms and three neighborhood structures to further improve the solutions. For evaluating the proposed algorithm, its results are reported for 56 instances of Solomon for VRPTW. The obtained results show the superior performance of MFGA, with lower overall distances compared to other heuristic algorithms found in the literature. The MFGA has the percent deviation 1.23% and %15 better than BKSs on R1 and RC1 datasets. Furthermore, MFGA has also attained 17 best-known solutions and 12 new best-known solutions on different dataset types. For future research, more restrictions on the problem can be considered. For example, two objective functions for the problem, including minimizing the distance traveled by the fleet and the number of vehicles, can be considered simultaneously, and also introduce further ad-hoc techniques to reduce the number of routes and to substantially increase the effectiveness of the algorithm for other VRP versions. Additionally, we plan to study this problem by applying other metaheuristic methods, as genetic algorithms [38], tabu search [2], etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-RP23030).

References

- [1] Yousefikhoshbakht M, Didehvar F, Rahmati F. An effective rank based ant system algorithm for solving the balanced vehicle routing problem. *Int J Indust Eng* 2016; 23(1):330–41.
- [2] Ahmed ZH, Yousefikhoshbakht M. An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows. *Alex Eng J* 2023;64:349–63. <https://doi.org/10.1016/j.aej.2022.09.008>.
- [3] Ahmed ZH, Hameed AS, Mutar ML, Wang S. Hybrid Genetic Algorithms for the Asymmetric Distance-Constrained Vehicle Routing Problem. *Math Probl Eng* 2022; 2022:1–20.
- [4] Ashouri M, Yousefikhoshbakht M. A Combination of Meta-heuristic and Heuristic Algorithms for the VRP, OVRP and VRP with Simultaneous Pickup and Delivery. *BRAIN Broad Res Artificial Intell Neurosci* 2017;8(2):81–95.
- [5] Yousefikhoshbakht M, Sedighpour M. An optimization algorithm for the capacitated vehicle routing problem based on ant colony system. *Aust J Basic Appl Sci* 2011;5 (12):2729–37.
- [6] Subhan F, Aziz MA, Khan IU, Fayaz M, Wozniak M, Shafi J, et al. Cancerous tumor controlled treatment using search heuristic (GA)-based sliding mode and synergetic controller. *Cancers* 2022;14(17):4191.
- [7] Rani S, Koundal D, Kavita F, Ijaz MF, Elhoseny M, Alghamdi MI. An optimized framework for WSN routing in the context of industry 4.0. *Sensors* 2021;21(19): 6474.
- [8] Dogra R, Rani S, Kavita, Shafi J, Kim SeongKi, Ijaz MF. ESEERP: Enhanced smart energy efficient routing protocol for internet of things in wireless sensor nodes. *Sensors* 2022;22(16):6109.
- [9] Yassen ET, Ayob M, Nazri MZA, Sabar NR. An adaptive hybrid algorithm for vehicle routing problems with time windows. *Comput Ind Eng* 2017;113:382–91. <https://doi.org/10.1016/j.cie.2017.09.034>.
- [10] Hedar A-R, Bakr MA. Three strategies tabu search for vehicle routing problem with time windows. *Comput Sci Inform Technol* 2014;2(2):108–19. <https://doi.org/10.13189/csit.2014.020208>.
- [11] de Armas J, Melián-Batista B. Variable neighborhood search for a dynamic rich vehicle routing problem with time windows. *Comput Ind Eng* 2015;85:120–31. <https://doi.org/10.1016/j.cie.2015.03.006>.
- [12] Pureza V, Morabito R, Reimann M. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. *Eur J Oper Res* 2012;218(3): 636–47. <https://doi.org/10.1016/j.ejor.2011.12.005>.
- [13] Baños R, Ortega J, Gil C, Fernández A, de Toro F. A Simulated Annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *Expert Syst Appl* 2013;40(5):1696–707. <https://doi.org/10.1016/j.eswa.2012.09.012>.
- [14] Y. Nagata Edge assembly crossover for the capacitated vehicle routing problem C. Cotta J. van Hemert Evolutionary Computation in Combinatorial Optimization. *EvoCOP 2007 Lecture Notes in Computer Science vol 4446 2007 Springer Berlin, Heidelberg* 10.1007/978-3-540-71615-0_13.
- [15] Ursani Z, Essam D, Cornforth D, Stocker R. Localized genetic algorithm for vehicle routing problem with time windows. *Appl Soft Comput* 2011;11(8):5375–90. <https://doi.org/10.1016/j.asoc.2011.05.021>.
- [16] Nagata Y, Bräysy O. A powerful route minimization heuristic for the vehicle routing problem with time windows. *Oper Res Lett* 2009;37(5):333–8. <https://doi.org/10.1016/j.orl.2009.04.006>.
- [17] Y. Nagata New EAX crossover for large TSP instances T.P. Runarsson H.G. Beyer E. Burke J.J. Merelo-Guervós L.D. Whitley X. Yao Parallel Problem Solving from Nature - PPSN IX. *PPSN 2006 Lecture Notes in Computer Science vol 4193 2006 Springer Berlin, Heidelberg* 10.1007/11844297_38.
- [18] Ticha HB, Absi N, Feillet D, Quilliot A. Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows. *Comput Oper Res* 2019;104:113–26. <https://doi.org/10.1016/j.cor.2018.11.001>.
- [19] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 1964;12(4):568–81.
- [20] Fadakar, E., and Ebrahimi, M. (2016). A new metaheuristic football game inspired algorithm, *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 6–11. DOI: 10.1109/CSIEC.2016.7482120.
- [21] Maleki F, Yousefikhoshbakht M. A hybrid algorithm for the open vehicle routing problem. *Int J Optimiz Civil Eng* 2019;9(2):355–71.
- [22] Yousefikhoshbakht M, Malekzadeh N, Sedighpour M. Solving the traveling salesman problem based on the genetic reactive bone route algorithm whit ant colony system. *Int J Prod Manage Eng* 2016;4(2):65–73.
- [23] Nakhaei F, Irannajad M, Yousefikhoshbakht M. Simultaneous optimization of flotation column performance using genetic evolutionary algorithm. *Physicochem Probl Miner Process* 2016;52(2):874–93.
- [24] Nakhaei F, Irannajad M, Yousefikhoshbakht M. Flotation column performance optimisation based on imperialist competitive algorithm. *Int J Min Miner Eng* 2016;7(1):1–17.
- [25] Rahmani A, Yousefikhoshbakht M. Capacitated facility location problem in random fuzzy environment: using (α, β) -cost minimization model under the Hurwicz criterion. *J Intell Fuzzy Syst* 2013;25(4):953–64.
- [26] Purnomo HD, Fibriani C, Somya R, Wee HM. Soccer game optimization for travelling salesman problem. In: *In 2017 International Conference on Innovative and Creative Information Technology (ICITech)*. IEEE; 2017. p. 1–7.
- [27] Alif A, Garside AK, Amallynda I, Ramadhani BNIF. Solving Capacitated Vehicle Routing Problem Using Football Game Algorithm. *Jurnal Ilmiah Teknik Industri* 2022;21(1):27–36.
- [28] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper Res* 1987;35(2):254–65. <https://doi.org/10.1287/opre.35.2.254>.
- [29] Tan KC, Chew YH, Lee LH. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Comput Optim Appl* 2006;34: 115–51. <https://doi.org/10.1007/s10589-005-3070-3>.
- [30] Alvarenga GB, Mateus GR, de Tomi G. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Comput Oper Res* 2007;34(6):1561–84. <https://doi.org/10.1016/j.cor.2005.07.025>.
- [31] de Oliveira, H.C.B., Vasconcelos, G.C., Alvarenga, G.B., Mesquita, R.V., and de Souza, M.M. (2007). A robust method for the VRPTW with multi-start simulated annealing and statistical analysis, *2007 IEEE Symposium on Computational Intelligence in Scheduling*, pp. 198–205. DOI: 10.1109/SCIS.2007.367690.
- [32] Rochat Y, Taillard ÉD. Probabilistic diversification and intensification in local search for vehicle routing. *J Heuristics* 1995;1:147–67. <https://doi.org/10.1007/BF02430370>.
- [33] Barbucha D. A cooperative population learning algorithm for vehicle routing problem with time windows. *Neurocomputing* 2014;146:210–29. <https://doi.org/10.1016/j.neucom.2014.06.033>.

- [34] Cordeau J-F, Maischberger M. A parallel iterated tabu search heuristic for vehicle routing problems. *Comput Oper Res* 2012;39(9):2033–50. <https://doi.org/10.1016/j.cor.2011.09.021>.
- [35] Luo J, Li X, Chen M-R, Liu H. A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows. *Inf Sci* 2015;316:266–92. <https://doi.org/10.1016/j.ins.2015.04.001>.
- [36] Gong, Y.-J., Zhang, J., Liu, O., Huang, R.-Z., Chung, H.S.-H., and Shi, Y.-H. (2012). Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach, in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2), pp. 254-267. DOI: 10.1109/TSMCC.2011.2148712.
- [37] Yu B, Yang ZZ, Yao BZ. A hybrid algorithm for vehicle routing problem with time windows. *Expert Syst Appl* 2011;38(1):435–41. <https://doi.org/10.1016/j.eswa.2010.06.082>.
- [38] Ahmed ZH, Al-Otaibi N, Al-Tameem A, Saudagar AKJ. Genetic crossover operators for the capacitated vehicle routing problem. *Comput Mater Continua* 2023;74(1): 1575–605. <https://doi.org/10.32604/cmc.2023.031325>.