E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia Science & Technology Asia
Vol.31 No.1 January - March 2026 Page: [38-56]
Original research article

Parameter Tuning for Modified Ebola Optimization Search Algorithm in Vehicle Routing Problem with Time Constraints

Sirichai Yodwangjai[1,*], Kanokporn Boonjubut[2]

[1]*Department of Industrial Engineering Technology, College of Industrial Technology, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand*
[2]*Department of Engineering Management, Faculty of Industrial Technology, Nakhon Ratchasima Rajabhat University, Nakhon Ratchasima 30000, Thailand*

Received 14 March 2025; Received in revised form 1 March 2026
Accepted 5 March 2026; Available online 27 March 2026

ABSTRACT

Vehicle Routing Problem with Time Windows (VRPTW) is a challenging problem because it involves finding the best routes for a fleet of vehicles to serve customers within time windows. VRPTW is an NP-hard Combinatorial optimization problem. In this paper, we propose an Ebola Optimization Search Algorithm (EOSA) framework to solve the VRPTW. An improved Ebola Optimization Search Algorithm modified with a control parameter is designed to solve the problem. Also, modified in EOSA is the impact of both the exploitation and exploration stages. However, the contact rate is a parameter that influences solution quality. The Taguchi method is used for tuning the parameters of the number of solutions and the contact rate. The experiment results showed the effectiveness of the proposed method, which was compared to the best-known solution.

Keywords: Design of experiment; Ebola optimization search algorithm; Parameter tuning; Time constraints; Vehicle routing problem

1. Introduction

The Vehicle Routing Problem (VRP) is a crucial issue in transportation and supply chain management, representing one of the most significant challenges for logistics companies today. First introduced as the Truck Dispatching Problem by Dantzig and Ramser [1], it serves as the paradigmatic case for the distribution of goods from a central depot to dispersed customers, aiming to minimize total cost by

***Corresponding author:** sirichai.y@cit.kmutnb.ac.th **doi:** xx.xxxxx/scitechasia.xxxx.xx

determining the optimal number of vehicles. The VRP has been various extensions and variations over time, including capacitated VRP (CVRP), multi-depot VRP (MDVRP), periodic VRP (PVRP), time-dependent VRP (TDVRP), heterogeneous fleet VRP (HFVRP), and dynamic VRP [2].

VRPTW is a complex variation of the VRP often found in industrial and practical applications. In this problem, each customer must be visited by exactly one vehicle within a specified time window, considering the vehicle's capacity constraints. The goal is to minimize the total distance traveled by a fleet of vehicles, which all start from a central warehouse and must visit designated customers within their respective time frames. There are two types of time windows restrictions for demand points. The first is the hard time windows, which mandates that a vehicle must begin serving customers within the specified time frame. If the vehicle arrives early, it must wait; if it arrives late, service is rejected [3]. The second type is the soft time windows, which permits the vehicle to start service outside the designated time frame but incur a penalty for doing so [4].

The VRP is an NP-hard problem. The exact methods can find optimal solution to the problem in small size. It cannot be solved in an optimal time range. Furthermore, its real-life VRP applications are complex and considerable large in scale. Therefore, metaheuristics are often more suitable for practical applications. Metaheuristics algorithms have been developed inspired by various swarm based, evolutionary-based, physical-based, human-based method [5]. Examples of such modern metaheuristics methods for solving the VRPTW include metaheuristics like the Tabu Search (TS) [6], Simulated Annealing (SA) [7], Genetic algorithm (GA) [8], Ant Colony Optimization (ACO) [9], Particle Swarm Optimization (PSO) [10], Variable Neighborhood Search (VNS) [11], etc. Apart from modern metaheuristics, many metaheuristics have been proposed for solving problems. These proposed metaheuristics are mainly based on natural inspiration systems. Tan et al. [12] proposed adaptive learning Bacterial Forging Optimization. Zhang et al. [13] presented a Hybrid Scatter Search with PSO. Wu et al. [14] proposed a novel ACO based on Improved Brainstorming Optimization. Shen et al. [15] proposed hybridizing Ant Colony System (ACS) and Brain Storm Optimization (BSO). Wei et al. [16] introduced a novel hybrid a Large Neighborhood Search with modified Rat Swarm Optimization. He et al. [17] proposed adaptive VNS with Ant Colony Algorithm. Aggarwal and Kumar [18] proposed modification in the randomness factor in Firefly Algorithm. Ahmed et al. [19] proposed a modified Football Game Algorithm. Wu et al. [20] proposed a neighborhood comprehensive learning PSO and, Chai et al. [21] applied Harris Hawks Optimizer in a real case.

The Ebola Optimization Search Algorithm (EOSA) is a population-based solution that wasnewly introduced by Oyelade et al. [22]. EOSA is inspired by the propagation strategy of Ebola virus disease based on SIR model. EOSA is able to generate a competitive solution for an NP-hard problem. Furthermore, the approach of EOSA was conducted in different problems such as agriculture [23], telecommunication [24], electrical distribution network [25], medical diagnosis [26-28]. The parameter of the metaheuristic has a strong impact on the performance of an optimization algorithm [29, 30]. The EOSA consists of multiple parameters that are impact to solution quality [26]. The contact rate is one of the vari-

able number that is tuning in each problem but some published research determine fixed value at 0.1 [31, 32]. They can tuning for improve the solution quality. Taguchi experiment design is one of methods that use for tuning the parameters and reducing time budget [33].

Moreover, the following are the technical contribution of this study: 1. To propose a Ebola Optimization Search Algorithm for Vehicle Routing Problem with Time Windows. 2. To tuning parameters of Ebola Optimization Search Algorithm for various problem types. 3. To evaluate and compare the modified algorithm with other algorithms.

The remaining sections of the paper are organized as follows: Section 2 presents a mathematical programming model in VRPTW. Section 3 present framework of the Ebola Optimization Search Algorithm. Section 4 presents the step of modified Ebola Optimization Search Algorithm for VRPTW. Section 5 presents the results obtained and discussion on the findings, while the study's concluding remarks and future research directions are presented in Section 6.

2. Mathematical model

In this section, we present a mathematical programming model for our problem. We assume that vehicles are homogeneous fleets. Each customer has time windows for receiving service. Let us define a graph, $G = (V, A)$ as a transportation network. V is defined as a vertex set, $V = \{0, 1, 2, ..., n\}$ where 0 denoted the center and the set of vertexes, $V = \{1, 2, ..., n\}$ corresponds to the customers where n is the number of customers. In this graph, a distance d_{ij} and travel time t_{ij} are associated with every arc $(i, j) \in A$, where A is arcs set. Each customer i has a demand q_i, a service time s_i and a time windows (e_i, l_i), where e_i and l_i are the earliest time and the latest time to start the service of customer i. In the case where the vehicle to serve the customer i arrives prior to e_i, it must wait at customer i location until time e_i. For depot, we set $q_0 = 0$, $e_0 = 0$ and $l_0 = T$, where T is the latest time in which vehicles can return to the depot.

Sets and indices

i, j the number of customers.
k the number of vehicles.
N set of customers.
K set of vehicles.

Parameters

d_{ij} the distance from customer i to j.
t_{ij} the travel time from customer i to j.
q_i the demand of customer i.
Q_k the capacity of vehicle k.
s_i the service time for customer i.
e_i the earliest arrival time for customer i.
l_i the latest arrive time for customer i.
w_i the waiting time of vehicle at customer i.
M the large positive number.

Decision variable

$$x_{ijk} = \begin{cases} 1, & \text{if the vehicle } k \text{ travels from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

$A_{ik} =$ starting time of customer i by vehicle k

Objective function

$$MinZ = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} \qquad (2.1)$$

subject to

$$\sum_{k \in K} \sum_{i \in N} x_{ijk} = 1; \forall j \in N, i \neq j \qquad (2.2)$$

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1; \forall i \in N, j \neq i \qquad (2.3)$$

$$\sum_{i \in N} \sum_{j \in N} q_i x_{ijk} \leq Q_k; \forall k \in K \quad (2.4)$$

$$\sum_{j \in N, j \neq 0} x_{0jk} \leq 1; \forall k \in K \quad (2.5)$$

$$\sum_{j \in N} x_{0jk} = \sum_{i \in N} x_{i0k}; \forall k \in K \quad (2.6)$$

$$A_{ik} + s_i + t_{ij} + w_i - M(1 - x_{ijk}) \leq A_{jk} ; \forall i, j \in N, k \in K \quad (2.7)$$

$$e_i \leq A_{ik} + w_i \leq l_i; \forall i \in N, k \in K \quad (2.8)$$

$$w_i = max\{e_i - A_{ik}, 0\}; \forall i \in N, k \in K \quad (2.9)$$

$$A_{ik} \geq 0; \forall i \in N, k \in K \quad (2.10)$$

$$x_{ijk} = \{0, 1\}; \forall i, j \in N, k \in K \quad (2.11)$$

The objective function (2.1) minimizes the total distance. Constraints (2.2) and (2.3) guarantee that each customer is served by exactly one vehicle. Constraint (2.4) is the capacity constraint, which states that for each route, the accumulated demand from the customers who are served by a vehicle must not exceed the vehicle capacity. Constraint (2.5) - (2.6) guarantee that the vehicles start from and return to the depot. Constraint (2.7) specifies the time relationship between two customers who are served by the same vehicle. Constraints (2.8) - (2.9) are related to time windows. Constraint (2.10) - (2.11) are decision variables.

3. Ebola Optimization Search Algorithm

In this section, we provide a comprehensive explanation of the EOSA. The mathematical of EOSA is used for explain in update position in each iteration.

3.1 Framework of Ebola Optimization Search Algorithms

A thorough understanding of the SEIR model in the context of disease transmission is essential for the development of robust optimization techniques. This study presents an algorithm that incorporates distinct epidemiological compartments, specifically tailored to the transmission dynamics of the Ebola virus. Human-to-human transmission is modeled as occurring exclusively through infection from a designated reservoir, with quarantined and vaccinated individuals factored into the model to reflect their impact on transmission rates. These elements are consistent with observed epidemiological patterns of the virus.

The proposed optimization technique is based on the SEIR-HDFVQ framework, which expands the classical SEIR model

Table 1. Parameter of MEOSA.

Symbols	Description	Range
π	Recruitment rate of susceptible human individuals	Value from experiment
β_1	Contact rate of infection human individuals	
β_2	Contact rate pathogen individuals/ environment	
β_3	Contact rate of deceased human individuals	
β_4	Contact rate of recovered human individuals	
Γ	Disease-induced death rate of human individuals	Random value in range 0-1
γ	Recovery rate of human individuals	
η	Decay rate of Ebola virus in the environment	
α	Rate of hospitalization of infected individuals	
τ	Natural death rate of human individuals	
δ	Rate of burial of deceased human individuals	
ϑ	Rate of vaccination of individuals	
ϖ	Rate of response to hospital treatment	
μ	Rate response to vaccination	
ξ	Rate of quarantine of infected individuals	

by including compartments for Quarantine (Q), Vaccination (V), Funeral (F), Death (D), Hospitalization (H), Recovery (R), Infection (I), Exposure (E), and Susceptibility (S). All parameters are shown in Table 1. Additionally, the model considers the potential risk posed by recovered individuals who may retain the virus in bodily fluids, thereby maintaining the capacity to infect healthy individuals. This comprehensive modeling approach enhances our understanding of disease propagation and informs the development of effective intervention strategies to control outbreaks [22].

3.2 The mathematical model of EOSA

An individual within the S is positioned in designated area and allowed to move, effectively illustrating the concept of infectiousness as they transition into the infected (I) compartment. The position of exposure individual in S is computed using Eq. (3.1).

$$mI_i^{t+1} = mI_i^t + \rho M(I), \tag{3.1}$$

where ρ represent the scale factor of displacement of an individual mI_i^{t+1} and mI_i^t are the updated and original positions respectively at time t and $t+1$. $M(I)$ is the movement rate made by individuals and is defined thus:

$$M(I) = srate * rand(0,1) + M(Ind_{best}), \tag{3.2}$$

$$M(S) = lrate * rand(0,1) + M(Ind_{best}). \tag{3.3}$$

The exploitation and exploration stage are design based on infected individual within distance of movement. The $srate$ is short-distance movement in exploitation stage. Meanwhile, the exploration stage, the $lrate$ is average distance of infected individual between susceptible.

3.2.1 Initialization of susceptible population

The initial susceptible S is created by a random number in the search space. The dimension of the initial susceptible equals the number of customers where $PSize$ is the population size of susceptible. An individual is positioned within a space and is allowed to move around to demonstrate the concept of infectiousness so that the individual can transit to the infected (I) compartment. The individual are initialized by Eq. (3.4)

$$individual_i = L_i + rand(0,1) * (U_i + L_i), \tag{3.4}$$

where U_i can be defined as upper bounds, L_i can be defined as lower bounds. and i is defined as an integer from 1 to N in the size of population.

3.2.2 Update position

For updating the Susceptible (S), Infected (I), Hospitalized (H), Exposed (E), Vaccinated (V), Recovered (R), Funeral (F), Quarantine (Q), Dead (D). In our case, the application of differential calculus intends to obtain the rates of change of quantities S, I, H, V, R, Q, and D with respect to time t. Hence, the Eq. (3.5) - Eq. (3.11) are as follows:

$$\frac{\partial S(t)}{\partial t} = \pi - (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2(PE))S - (\tau S + \Gamma I)), \tag{3.5}$$

$$\frac{\partial I(t)}{\partial t} = (\beta_1 I + \beta_3 D + \beta_4 R + \beta_2(PE)\lambda)S - (\Gamma + \gamma)I - (\tau)S, \tag{3.6}$$

$$\frac{\partial H(t)}{\partial t} = \alpha I - (\gamma + \varpi)H, \tag{3.7}$$

$$\frac{\partial R(t)}{\partial t} = \gamma I - \Gamma R, \tag{3.8}$$

$$\frac{\partial V(t)}{\partial t} = \gamma I - (\mu + \vartheta)V, \tag{3.9}$$

$$\frac{\partial D(t)}{\partial t} = (\tau S + \Gamma I) - \delta D, \tag{3.10}$$

$$\frac{\partial Q(t)}{\partial t} = (\pi I - (\gamma R + \Gamma D)) - \xi Q. \tag{3.11}$$

We assume that Eq. (3.5) – Eq. (3.11) are scalar functions meaning that each has one number as a value, which can be represented as a float.

3.2.3 Selection

The best solution, the current best solution, and the global best solution are denoted as $sbest$, $gbest$ and $cbest$ at time t. We distinguish as $gbest$ and $cbest$ infected individuals who are super-spreaders and spreaders of the Ebola virus by Eq. (3.12)

$$sbest = \begin{cases} gbest, & f(cbest) < f(gbest) \\ cbest, & f(cbest) \geq f(gbest) \end{cases}. \tag{3.12}$$

4. Modified Ebola Optimization Search Algorithm for VRPTW

In this section, we present an improved EOSA designed for the VRPTW, referred to as the Modified Ebola Optimization Search Algorithm (MEOSA). We enhance the algorithm by introducing a control parameter for exploration-exploitation switching and incorporating local search to improve solutions. The pseudocode in Algorithm 1

4.1 Encoding

In the encoding process, each individual is structured as an array of length customer. Each value in a dimension was initially generated with a uniform random number between 0 and 1. For example, the number of dimensions was set to 9, which was equal to the number of customers. Fig. 1 illustrates an individual with a random number from 0.1 to 0.9.

Algorithm 1 : MEOSA for VRPTW

Input: $Objfunc, PSize, epoch, evdincub$
Output: $gbest, Sols$

$S, E, I, H, R, V, Q, Sols \leftarrow \emptyset$
generate the index case, $icase$ by random number
determine the $icase$ to S in $PSize$
set the current best, $cbest$ and global best, $gbest$
while $t \leq epoch \wedge len(I) > 0$ **do**
    $Q \leftarrow$ apply the quarantine case by $rand(0, Eq.\ (3.11) \times I)$
    $fracI \leftarrow I - Q$ is difference of infected case from quarantine case
    for $i \leftarrow 1$ to $len(fracI)$ **do**
        update position, pos_i via movement rate by $Eq.\ (3.1)$
        $d_i \leftarrow$ generate random number $rand(0, 1)$
        generate newly infected case, $newI$
        if $d_i > evdincub$ **then**
            $newI \leftarrow$ random selection in pos_i
            if $newI < 0.5$ **then**
                $newI \leftarrow$rand(0,[(Eq. (3.6) $\times I \times srate$) + $(Rand \times I \times a)$])
            else
                $newI \leftarrow$rand(0,[(Eq. (3.6) $\times I \times lrate$) - $(Rand \times I \times a)$])
            end if
            $newI+ \leftarrow newI$;
        end if
        $I+ \leftarrow newI$;
    end for
    $h \leftarrow rand(0, Eq.\ (3.7) \times I), H+ \leftarrow h$;
    $r \leftarrow rand(0, Eq.\ (3.8) \times I), R+ \leftarrow r$;
    $v \leftarrow rand(0, Eq.\ (3.9) \times I), V+ \leftarrow v$;
    $d \leftarrow rand(0, Eq.\ (3.10) \times I), D+ \leftarrow d$;

Algorithm 1 : MEOSA for VRPTW

27: $I+ \leftarrow I - add(r, d)$
28: $S+ \leftarrow r$;
29: $S- \leftarrow d$;
30: $cbest = fitness(objfunc, I)$;
31: **if** $cbest > gbest$ **then**
32: $gbest \leftarrow cbest$;
33: $Sols \leftarrow gbest$
34: **end if**
35: **end while**
36: return $gbest, Sols$

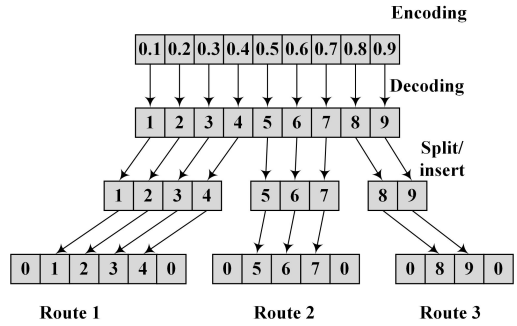


Fig. 1. Encoding example of the representation of VRPTW.

4.2 Decoding

The floating value is transformed into a permutation number by the least rank value. When an individual is transformed into the solution representation of VRP, the procedure of decoding is described as follows. Considering the problem constraints, the vehicle starts with the first customer point of the individual. If the customer point can be inserted into the route without violating the constraints of the capacity and time windows, it is inserted into the route. Otherwise, a new route is constructed to insert a customer point. Repeat this process until all customer points are located in the routes. If the total demand in each vehicle is greater than the vehicle capacity, then remove one customer from the exceed list and add it to the end of the next vehicle. As shown in Algorithm 2.

The operational procedure of the MEOSA, as depicted in Algorithm 1, is structured into the following sequential steps:

Step 1 Initialization: The process begins on line 1 with the initialization of the biological status sets (S, E, I, H, R, V, Q) and relevant algorithmic parameters.

Step 2 Index case generation: On line 2, an initial candidate solution, or index case ($icase$), is generated using a random distribution.

Step 3 Population assignment: The $icase$ is assigned to the susceptible population (S) within the defined $PSize$ on line 3.

Step 4 Feasibility check and initial evaluation: On line 4, the initial individuals are evaluated. If the initial solution violates the vehicle capacity constraints, a repair mechanism is applied as per Algorithm 2. Subsequently, both the current best solution ($cbest$) and the global best solution ($gbest$) are established.

Step 5 Iterative optimization loop: The algorithm enters the main execution loop on line 5, which continues until the termination criteria are satisfied or the infected population is depleted. The loop consists of the following sub-procedures:

Step 5.1 Quarantine mechanism (Lines 6–7): The quarantine status is applied to the infected set based on Eq. (3.11). The active infected population ($fracI$) is then determined by excluding quarantined individuals, effectively focusing the search.

Step 5.2 Infection and mutation (Lines 8–22): For each remaining infected individual, the position (pos_i) is updated via the movement rate defined in Eq. (3.1). If the random value d_i exceeds the incubation threshold ($evdincub$), a new infection

Algorithm 2 : Repair Infeasible

Input: An infeasible solution.
Output: A feasible solution.

1: $i = 0$
2: **for** each vehicle, v in V **do**
3: **while** d_i exceeds the Q_k **do**
4: $i = i + 1$
5: find the $CusEx_i$
6: remove $CusEx_i$ from vehicle v
7: put $CusEx_i$ in $ExceedList$
8: **end while**
9: next v
10: **end for**
11: $CapLeft(v)$ the remaining capacity of v
12: Demand $CusEx_i$ $\leftarrow$ the demand of $CusEx_i$
13: **while** $ExceedList \neq 0$ **do**
14: **for** each member; $CusEx_i \in ExceedList$ **do**
15: $VFound \leftarrow FALSE$
16: **while** not $VFound$ **do**
17: **if** $CapLeft(v) > Dem(CusEx_i)$ **then**
18: put $CusEx_i$ in the last location of route of v
19: remove $CusEx_i$ from $ExceedList$
20: update $CapLeft(v)$
21: $CapLeft(v) \leftarrow CapLeft(v) - Dem(CusEx_i)$
22: $VFound \leftarrow TRUE$
23: **end if**
24: **end while**
25: next $CusEx_i$
26: **end for**
27: **end while**

($newI$) is triggered.

Step 5.3 Exploration vs. Exploitation (Lines 14–21): The search strategy is governed by a selection threshold. If the selection value is less than 0.5, the algorithm performs a local search (exploitation) using the short-range rate ($srate$). Otherwise, a long-range rate ($lrate$) is employed to facilitate exploration across the search space.

Step 5.4 State Transitions and population update (Lines 23–29): The statuses of all subgroups (Hospitalized, Recovered, Vaccinated, and Deceased) are updated using the control rates defined in Eq. (3.7) - Eq. (3.10). Recovered individuals return to the susceptible pool to allow for potential reinfection, while deceased individuals are permanently removed.

Step 5.5 Fitness evaluation (Lines 30–34): The fitness of the current infected population is computed. The $cbest$ is compared against the $gbest$, and updates are performed if a superior solution is discovered.

Step 6 Termination: Finally, on line 36, the algorithm terminates and returns the global best solution ($gbest$) alongside the set of all identified optimal solutions ($Sols$).

4.3 Control parameter

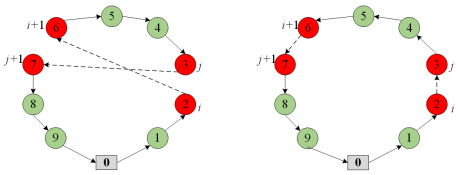
Furthermore, the convergence trajectory of the algorithm is regulated by a controllable parameter a, which undergoes a linear decay from 2 to 0 over the duration of the search. This attenuation process is formalized in Eq. (4.1), where t and $epoch$ denote the current iteration and the maximum iteration threshold, respectively. To maintain a balance between exploration and exploitation, the stochastic component ($Rand \times I \times a$), incorporating a random coefficient $Rand \in [0, 1]$, is employed to modulate the transition between the algorithm's operational phases. In the initial iterations, the elevated magnitude of a (approaching 2) induces significant fluctuations in the search vector, thereby facilitating extensive exploration and the dispersion of agents across the solution space. The re-

sult of $(Rand \times I \times a)$ is used for the exploitation stage in line 15 and the exploration stage in line 17 of Algorithm 1.

$$a = 2 - t \times \frac{2}{epoch}. \qquad (4.1)$$

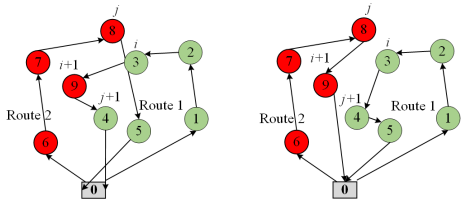
4.4 Local search

The improvement method is used for improving the initial solution. We applied the exchange operator to determine when the solution is in a local optimum. They are used for both intra-route and inter-route. The 2-opt is used to swap 2 edges in the intra-route. The two edges $(i, i+1)$ and $(j, j+1)$ are randomly selected for exchange. The $(i, i+1)$ and $(j, j+1)$ have to be replaced by the edges (i, j) and $(i+1, j+1)$. as shown in Fig.2. The 2-Opt* is used to swap in inter-route as shown in Fig.3



(a) Before **(b)** After

Fig. 2. 2-Opt Operator.



(a) Before **(b)** After

Fig. 3. 2-Opt* Operator.

5. Computational results

5.1 Implementation and instances

The algorithm was coded in Python 3.7. All experiments were run on personal computers. The machine is configured with Intel®core™ i7-1260P CPU 2.10 GHz and Ram 16 GB. The evaluation is based on well-known Solomon's VRPTW instances [24], which are divided into six data sets C1, C2, R1, R2, RC1 and RC2. In sets R1 and R2 customers are uniformly distributed, in sets C1 and C2 customers are clustered in groups and in sets RC1 and RC2 customers are semi-clustered. Each data instance contains the number and capacity of vehicles and the position, demand, service time, time windows of all the customers. In addition, there are small time windows and small vehicle capacity in sets C1, R1 and RC1, while there are large time windows and large vehicle capacity in sets C2, R2 and RC2. Table 2 shows the relevant information for each instance.

5.2 Parameters setting

Based on our preliminary evaluation, we define characteristics of the approach method for the experiments. They are comprised of parameters including η, α, Γ, γ, τ, δ, ϑ, ω, μ and ξ that are set at random value. The π, β_1, β_2, β_3, β_4 are recruiting rate and contact rate which tuned out to be the most influential on the solution quality [26]. This approach is effective in a fair number of solutions. The combination of the number of $PSize$ and the number of $epoch$ determines the amount of search. Three levels of $PSize/epoch$ were considered 5/20,000, 10/10,000 and 20/5,000. The number of solutions was fixed at 100,000 for each independent run.

The selection of appropriate parameter values significantly impacts the quality of solutions generated by an algorithm. To determine the optimal parameter settings with minimal computational effort, this study applies the Taguchi's Design of Experiments (TDOE) methodology. Specif-

Table 2. Instance information.

Instance class	Vehicle capacity	Service time	Time windows	Distribution type
C1	200	90	1236	Cluster
C2	700	90	3390	Cluster
R1	200	10	230	Random
R2	1000	10	1000	Random
RC1	200	10	240	Random and cluster
RC2	1000	10	960	Random and cluster

ically, a three-level orthogonal array was employed to evaluate the control factors, encompassing 27 experimental runs. The specific parameters, including their respective ranges and levels, are detailed in Table 3, while the Taguchi's L_{27} orthogonal matrix is presented in Table 4.

The Taguchi experimental design, widely utilized for optimization problems, employs two primary tools: the Orthogonal Array (OA) and the Signal-to-Noise (S/N) ratio. The OA serves as a numerical matrix that structures experimental plans across various factor levels, ensuring a robust evaluation. To assess experimental outcomes, the S/N ratio is applied to quantify the relationship between the mean (the desired 'signal') and the standard deviation (the undesirable 'noise') [34]. Given that the objective of this study is to minimize the total travel distance, the S/N ratio is computed based on the 'smaller-the-better' criterion, which is calculated by the Eq. (5.1)

$$(S/N)_{ij} = -10\log_{10}\left(\frac{\sum z_{ij}^2}{n}\right) \quad ;\forall i, j, \tag{5.1}$$

where z_{ij} is the objective function value using parameter i on level j and n is the number of times level j of parameter i is repeated over the runs of all tails.

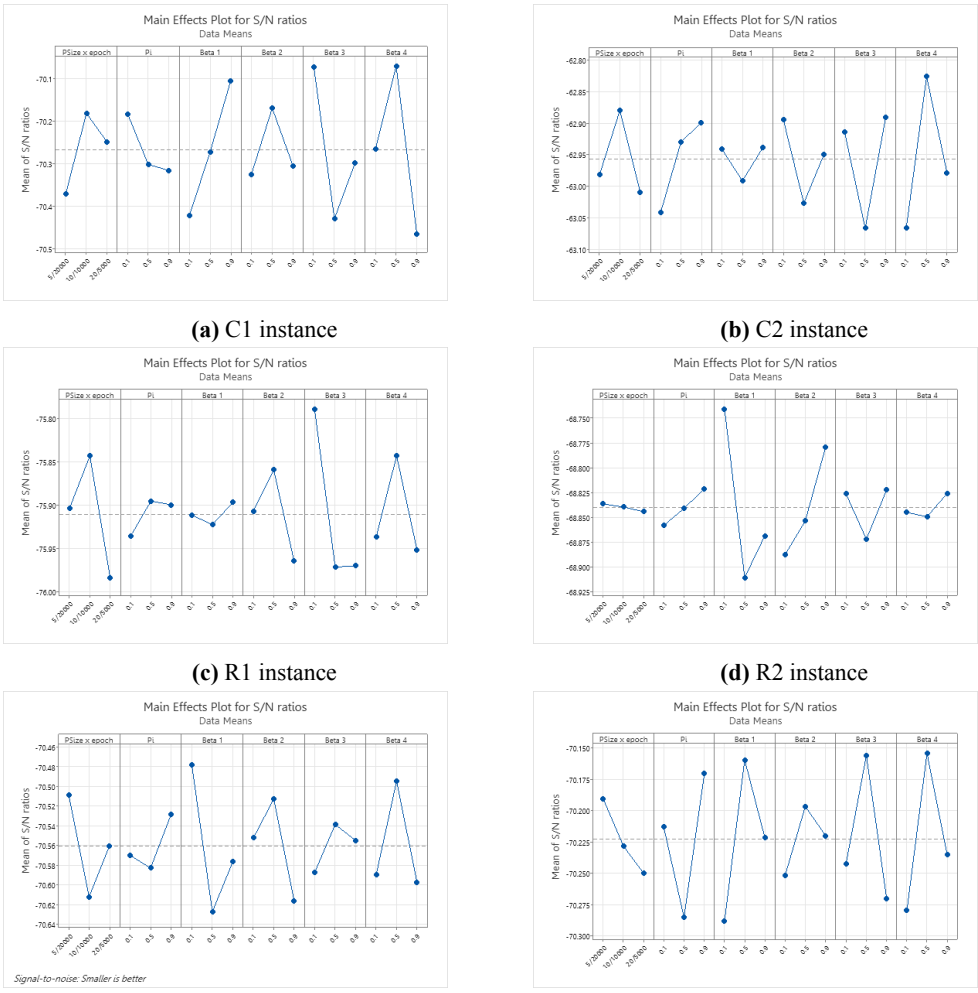
From Fig.4, which presents the mean of S/N ratio of instance, it can be concluded that the optimal parameter value of MEOSA is that corresponding to the highest S/N ratio. This confirms the effectiveness of the

Table 3. Parameter tuning of MEOSA for Taguchi method.

Parameter	Level 1	Level 2	Level 3
$PSize \times epoch$	5 / 20,000	10 / 10,000	20 / 5,000
π	0.1	0.5	0.9
β_1	0.1	0.5	0.9
β_2	0.1	0.5	0.9
β_3	0.1	0.5	0.9
β_4	0.1	0.5	0.9

Table 4. Taguchi orthogonal matrix.

Design	parameter					
	$PSize \times epoch$	π	β_1	β_2	β_3	β_4
1	5/20,000	0.1	0.1	0.1	0.1	0.1
2	5/20,000	0.1	0.1	0.1	0.5	0.5
3	5/20,000	0.1	0.1	0.1	0.9	0.9
4	5/20,000	0.5	0.5	0.5	0.1	0.1
5	5/20,000	0.5	0.5	0.5	0.5	0.5
6	5/20,000	0.5	0.5	0.5	0.9	0.9
7	5/20,000	0.9	0.9	0.9	0.1	0.1
8	5/20,000	0.9	0.9	0.9	0.5	0.5
9	5/20,000	0.9	0.9	0.9	0.9	0.9
10	10/10,000	0.1	0.5	0.9	0.1	0.5
11	10/10,000	0.1	0.5	0.9	0.5	0.9
12	10/10,000	0.1	0.5	0.9	0.9	0.1
13	10/10,000	0.5	0.9	0.1	0.1	0.5
14	10/10,000	0.5	0.9	0.1	0.5	0.9
15	10/10,000	0.5	0.9	0.1	0.9	0.1
16	10/10,000	0.9	0.1	0.5	0.1	0.5
17	10/10,000	0.9	0.1	0.5	0.5	0.9
18	10/10,000	0.9	0.1	0.5	0.9	0.1
19	20/5,000	0.1	0.9	0.5	0.1	0.9
20	20/5,000	0.1	0.9	0.5	0.5	0.1
21	20/5,000	0.1	0.9	0.5	0.9	0.5
22	20/5,000	0.5	0.1	0.9	0.1	0.9
23	20/5,000	0.5	0.1	0.9	0.5	0.1
24	20/5,000	0.5	0.1	0.9	0.9	0.5
25	20/5,000	0.9	0.5	0.1	0.1	0.9
26	20/5,000	0.9	0.5	0.1	0.5	0.1
27	20/5,000	0.9	0.5	0.1	0.9	0.5



(a) C1 instance

(b) C2 instance

(c) R1 instance

(d) R2 instance

(e) RC1 instance

(f) RC2 instance

Fig. 4. Main effect plot for the S/N ratio.

tuned parameters in enhancing MEOSA's performance. Table 5 provides the tuned parameter values for MEOSA in each instance. Based on the Taguchi experimental design, the selected parameters for the C1 instance are as follows: $PSize \times epoch$ = 10/10000, π=0.1, β_1=0.9, β_2=0.5, β_3=0.1, and β_4 =0.5.

5.3 Convergence analysis

This subsection presents the convergence analysis of MEOSA, comparing its performance under different parameter variations. To ensure a fair comparison, we define the variant parameter and set a fixed constraint value for all runs. The recruiting rate and contact rate set at 0.1, 0.5 and 0.9, while other parameters remain constant. The figure illustrates the convergence differences across five parameters (π, β_1, β_2, β_3, and β_4). The vertical axis represents the objective function value at each iteration, while the horizontal axis denotes the number of iterations. Most solutions

Table 5. Tuned parameter of MEOSA.

Parameter	C1	C2	R1	R2	RC1	RC2
$PSize \times epoch$	10 / 10,000	10 / 10,000	10 / 10,000	5/ 20,000	5/ 20,000	5/ 20,000
π	0.1	0.9	0.5	0.9	0.9	0.9
β_1	0.9	0.9	0.9	0.1	0.1	0.5
β_2	0.5	0.1	0.5	0.9	0.5	0.5
β_3	0.1	0.9	0.1	0.9	0.5	0.5
β_4	0.5	0.5	0.5	0.9	0.5	0.5

converge early within the set limit of 1,000 iterations for the C1 and C2 instances, as shown in Figs. 5 - 6. In Fig. 5, higher parameter values result in slower convergence for the C1 instances. Conversely, Fig. 6 shows that setting the C2 instance parameters to 0.5 leads to faster convergence and improved objective function values.

5.4 Experimental result

In addition, to display the overall performance of the algorithm on all instances, the best results are measured by the number of vehicles (NV) and the total distance (TD). In addition, the performance of the MEOSA is summarized using the minimum, mean, and maximum values, along with the standard deviation. Tables 6 - 8 shows the results obtained by the MEOSA, as well as those of the EOSA and the best known solution (BKS) in terms of the best solution. The MEOSA shows the best solution from all 30 runs. Concurrently, the $\%Gap$ is defined Eq. (5.2) where BS denotes the optimal solution achieved by the current method, and BKS represents the best known solution for the benchmark instance, and $\%Gap$ signifies the error of BS.

$$\%Gap = (BS - BKS)/BKS * 100\%. \tag{5.2}$$

Table 6 highlights the performance of MEOSA in the C instances. The MEOSA achieved the optimal solution in 13 out of 17 problems within these instance. Additionally, Table 7 found the optimal solution in 9 out of 23 problems, with an average gap of 0.28%. While MEOSA uses more vehicles than BKS, its total travel distance is shorter than that of EOSA. However, in instance R105, MEOSA recorded a distance of 1,475.50, which is 1.75% higher than EOSA. Conversely, as shown in the Table 8, MEOSA achieved a total distance of 19,886.91, which is 0.06% lower than BKS. Furthermore, the number of vehicles (NV) used by MEOSA is lower than in EOSA, though BKS remains more efficient in achieving the target NV.

We selected two instances from the C instance group as a representative case to assess the effectiveness of the proposed method in solving the problem. From the Table 9, MEOSA achived the best found distance 842.16, which is 2.11% higher than the optimal solution. The number of vehicles used remains consistent at 10; however, the number of customers served (NCS) varies. Additionally, route 1, 2, 5 and 8 in MEOSA follow a sequential arrangement but differ from the optimal solution in an indirect opposite order. Notably, customer 10, which is assigned to route 7 in the optimal solution, is instead placed in route 3 (cyan route) in MEOSA. The detail shown in Fig. 7 and Table 9.

In Fig. 8 shown the C202 instance

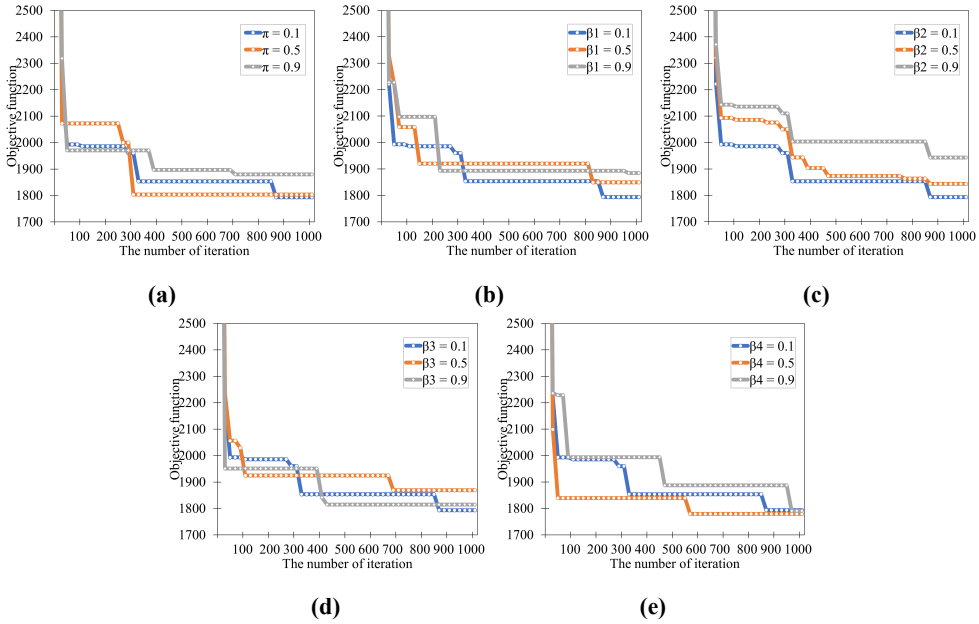


Fig. 5. Convergence of contact rate in C1 instance.

Table 6. Computational result C.

Problem	BKS		EOSA		MEOSA					$\%Gap_{BKS}$	$\%Gap_{EOSA}$
	NV	*TD*	*NV*	*TD*	*NV*	*Min.*	*Mean*	*Max.*	*Stdev.*		
C101	10	828.94	10	828.94	10	828.94	858.41	901.69	30.83	0.00 %	0.00 %
C102	10	828.94	10	828.94	10	828.94	848.77	888.43	34.35	0.00 %	0.00 %
C103	10	828.06	10	887.78	10	828.06	843.72	852.76	10.92	0.00 %	-6.73 %
C104	10	824.78	10	950.06	10	842.16	914.26	950.54	62.44	2.11 %	-11.36 %
C105	10	828.94	10	828.94	10	828.94	849.47	899.42	31.20	0.00 %	0.00 %
C106	10	828.94	10	828.94	10	828.94	865.05	890.47	21.38	0.00 %	0.00 %
C107	10	828.94	10	828.94	10	828.94	849.87	894.15	26.30	0.00 %	0.00 %
C108	10	828.94	10	828.94	10	828.94	869.55	916.18	30.78	0.00 %	0.00 %
C109	10	828.94	10	828.94	10	828.94	841.07	863.39	19.36	0.00 %	0.00 %
C201	3	591.56	3	591.56	3	591.56	591.56	591.56	0.00	0.00 %	0.00 %
C202	3	591.56	4	626.67	4	618.58	662.06	704.61	45.72	4.57 %	-1.29 %
C203	3	591.17	4	648.42	4	628.08	642.83	651.98	12.90	6.24 %	-3.14 %
C204	3	590.6	4	654.5	4	603.40	652.60	689.64	36.06	2.17 %	-7.81 %
C205	3	588.88	3	588.88	3	588.88	593.20	604.11	5.45	0.00 %	0.00 %
C206	3	588.49	3	588.49	3	588.49	624.75	638.48	21.08	0.00 %	0.00 %
C207	3	588.29	3	588.29	3	588.29	684.83	714.14	47.94	0.00 %	0.00 %
C208	3	588.32	3	588.32	3	588.32	607.95	620.29	11.82	0.00 %	0.00 %

is characterized large time windows. The MEOSA found *NV* in all route at 4. The customer no. 66, 67 and 69 were assigned to a newly established route. The best distance achieved by MEOSA is 618.58, which exceeds the optimal solution by 4.57%. The detail of routing is shown in Fig. 8 and Table 10.

6. Conclusion

This paper investigates the VRPTW, which aims to minimize travel distance

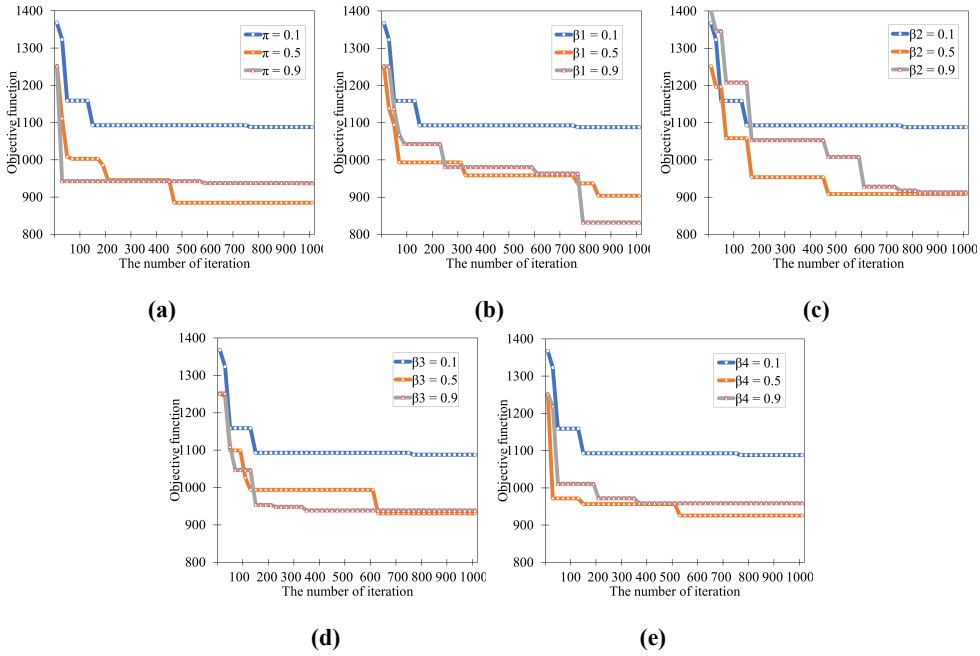


Fig. 6. Convergence of contact rate in C2 instance.

Table 7. Computational result R.

Problem	BKS		EOSA		MEOSA					%Gap$_{BKS}$	%Gap$_{EOSA}$
	NV	*TD*	*NV*	*TD*	*NV*	*Min.*	*Mean*	*Max.*	*Stdev.*		
R101	19	1650.8	21	1685.97	22	1678.92	1685.97	1691.99	6.60	1.70 %	-0.42 %
R102	17	1486.12	20	1552.26	17	1473.66	1520.56	1559.22	42.86	-0.84 %	-5.06 %
R103	13	1292.68	16	1391.22	17	1315.28	1403.55	1452.98	47.97	1.75 %	-5.46 %
R104	9	1007.31	13	1109.36	11	1041.79	1051.23	1059.67	7.32	3.42 %	-6.09 %
R105	14	1377.11	18	1450.09	18	1475.50	1498.27	1523.71	24.22	7.14 %	1.75 %
R106	12	1252.03	16	1364.11	16	1342.53	1360.47	1369.16	12.14	7.23 %	-1.58 %
R107	10	1104.66	13	1168.01	12	1083.14	1120.09	1168.01	36.73	-1.95 %	-7.27 %
R108	9	960.88	12	1041.27	11	1029.92	1042.78	1050.00	9.50	7.19 %	-1.09 %
R109	11	1194.73	15	1245.09	12	1166.15	1210.25	1245.16	40.97	-2.39 %	-6.34 %
R110	10	1118.84	13	1153.24	12	1102.09	1131.22	1154.62	26.75	-1.50 %	-4.44 %
R111	10	1096.72	14	1159.32	14	1116.59	1154.21	1182.85	33.02	1.81 %	-3.69 %
R112	9	982.14	11	1071.35	11	1034.34	1050.78	1071.35	15.30	5.31 %	-3.45 %
R201	4	1252.37	14	1344.22	11	1230.86	1341.47	1411.88	96.97	-1.72 %	-8.43 %
R202	3	1191.7	13	1301.76	9	1134.82	1146.53	1157.44	11.65	-4.77 %	-12.82 %
R203	3	939.5	5	936.12	5	927.59	959.87	1013.08	35.08	6.03 %	-1.27 %
R204	2	825.52	5	818.62	5	807.60	842.65	946.22	69.05	-2.17 %	-1.35 %
R205	3	994.43	9	1111.82	9	1036.18	1041.08	1044.67	3.71	4.20 %	-6.80 %
R206	3	906.14	7	991.11	6	944.13	947.18	951.89	3.47	4.19 %	-4.74 %
R207	2	890.61	6	871.8	6	869.62	870.80	871.80	0.93	-2.36 %	-0.25 %
R208	2	726.82	5	801.73	5	763.69	768.75	774.01	5.85	5.07 %	-4.74 %
R209	3	909.16	8	995.17	8	930.16	933.04	937.03	3.44	2.31 %	-2.69 %
R210	3	939.37	9	995.17	7	968.89	988.67	995.39	13.19	3.14 %	-2.64 %
R211	2	885.71	6	841.99	6	815.74	816.32	817.00	0.66	-7.90 %	-3.12 %

while ensuring that deliveries to multiple customers occur within predefined time

Table 8. Computational result RC.

Problem	BKS		EOSA		MEOSA					%Gap$_{BKS}$	%Gap$_{EOSA}$
	NV	*TD*	*NV*	*TD*	*NV*	*Min.*	*Mean*	*Max.*	*Stdev.*		
RC101	14	1696.95	18	1751.45	18	1732.30	1760.81	1809.97	33.99	2.08 %	-1.09 %
RC102	12	1554.75	20	1552.16	14	1442.45	1450.96	1463.53	9.20	-7.22 %	-7.07 %
RC103	11	1261.67	13	1348.02	13	1306.73	1330.50	1348.03	17.89	3.57 %	-3.06 %
RC104	10	1135.48	12	1239.99	11	1227.51	1264.41	1285.70	25.80	8.10 %	-1.01 %
RC105	13	1629.44	17	1672.63	17	1651.24	1669.31	1696.23	20.38	1.34 %	-1.28 %
RC106	11	1424.73	13	1388.86	13	1381.37	1432.68	1480.73	55.01	0.70 %	-0.54 %
RC107	11	1230.48	12	1297.24	12	1291.74	1305.44	1332.47	19.07	4.98 %	-0.42 %
RC108	10	1139.82	12	1239.92	11	1234.60	1251.31	1259.43	9.94	8.32 %	-0.43 %
RC201	4	1406.94	10	1393.61	9	1326.28	1489.92	1565.24	110.46	-5.73 %	-4.83 %
RC202	3	1365.65	8	1219.02	10	1184.17	1191.12	1199.23	6.80	-13.29 %	-2.86 %
RC203	3	1049.62	8	1033.05	7	993.68	997.41	1003.57	5.37	-5.33 %	-3.81 %
RC204	3	798.46	6	868.59	4	830.08	848.32	883.90	17.89	3.96 %	-4.43 %
RC205	4	1297.65	10	1261.19	10	1258.78	1260.34	1261.20	1.15	-3.00 %	-0.19 %
RC206	3	1146.32	8	1167.34	8	1146.50	1156.03	1163.50	7.06	0.02 %	-1.79 %
RC207	3	1061.14	11	1864.59	7	1060.55	1668.09	1875.05	405.05	-0.06 %	-43.12 %
RC208	3	828.14	7	857.2	5	818.93	821.58	823.82	2.47	-1.11 %	-4.46 %

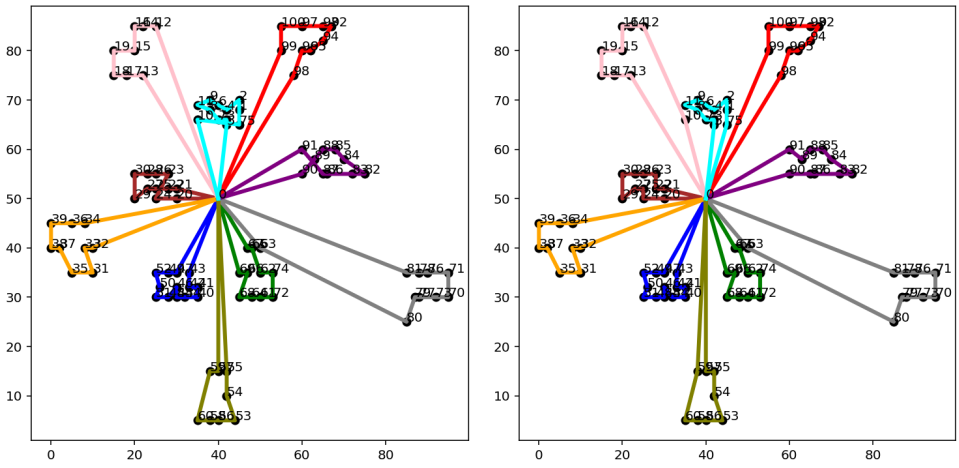
Table 9. Optimal solution of C104.

Route	MEOSA (TD = 842.16, NV=10, Q_k=200)	Demand	NCS.	Optimal solution (TD = 824.78, NV = 10, Q_k=200)	Demand	NCS.
1	0-91-87-86-83-82-84-85-88-89-90-0	170	10	0-90-87-86-83-82-84-85-88-89-91-0	170	10
2	0-21-22-25-27-29-30-28-26-23-24-20-0	170	11	0-20-24-25-27-29-30-28-26-23-22-21-0	170	11
3	0-5-3-7-8-11-9-6-4-2-1-75-10-0	180	12	0-5-3-7-8-11-9-6-4-2-1-75-0	170	11
4	0-67-65-62-74-72-61-64-68-66-69-0	150	10	0-67-65-62-74-72-61-64-68-66-69-0	150	10
5	0-34-36-39-38-37-35-31-33-32-0	200	9	0-32-33-31-35-37-38-39-36-34-0	200	9
6	0-43-42-41-40-44-45-46-48-51-50-52-49-47-0	190	13	0-43-42-41-40-44-46-45-48-51-50-52-49-47-0	190	13
7	0-13-17-18-19-15-16-14-12-0	190	8	0-13-17-18-19-15-16-14-12-10-0	200	9
8	0-55-54-53-56-58-60-59-57-0	200	8	0-57-55-54-53-56-58-60-59-0	200	8
9	0-98-96-95-94-92-93-97-100-99-0	190	9	0-98-96-95-94-92-93-97-100-99-0	190	9
10	0-81-78-76-71-70-73-77-79-80-63-0	200	10	0-81-78-76-71-70-73-77-79-80-63-0	200	10

Table 10. Optimal solution of C202.

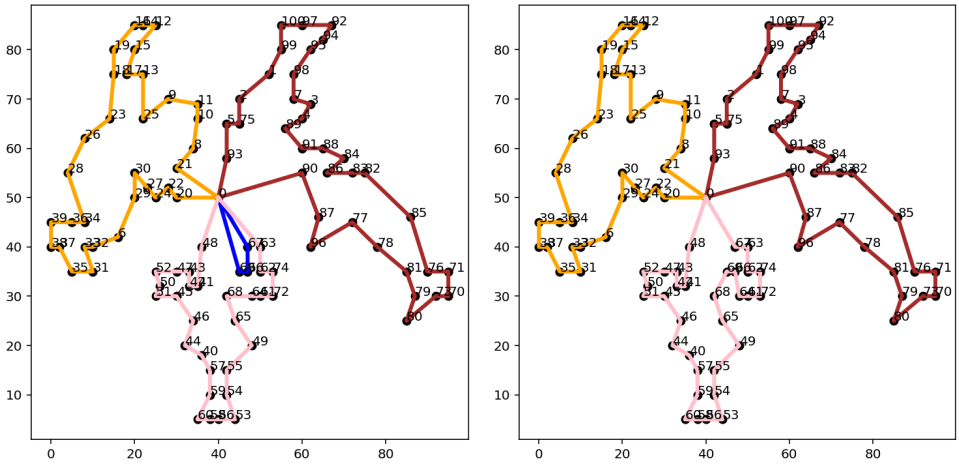
Route	MEOSA (TD = 618.58, NV= 4, Q_k=700)	Demand	NCS.	Optimal solution (TD = 591.56, NV = 3, Q_k=700)	Demand	NCS.
1	0-20-22-24-27-30-29-6-32-33-31-35-37-38-39-36-34-28-26-23-18-19-16-14-12-15-17-13-25-9-11-10-8-21-0	630	33	0-20-22-24-27-30-29-6-32-33-31-35-37-38-39-36-34-28-26-23-18-19-16-14-12-15-17-13-25-9-11-10-8-21-0	630	33
2	0-63-62-74-72-61-64-68-65-49-55-54-53-56-58-60-59-57-40-44-46-45-51-50-52-47-43-42-41-48-0	530	29	0-67-63-62-74-72-61-64-66-69-68-65-49-55-54-53-56-58-60-59-57-40-44-46-45-51-50-52-47-43-42-41-48-0	560	32
3	0-93-5-75-2-1-99-100-97-92-94-95-98-7-3-4-89-91-88-84-86-83-82-85-76-71-70-73-80-79-81-78-77-96-87-90-0	620	35	0-93-5-75-2-1-99-100-97-92-94-95-98-7-3-4-89-91-88-84-86-83-82-85-76-71-70-73-80-79-81-78-77-96-87-90-0	620	35
4	0-67-66-69-0	30	3	-	-	-

constraints. The VRPTW is an NP-hard problem that requires significant computational effort to solve efficiently. Ebola Search Optimization Algorithm is a meta-



(a) MEOSA of C104 **(b)** Optimal solution of C104

Fig. 7. solution in C104 instance.



(a) MEOSA of C202 **(b)** Optimal solution of C202

Fig. 8. solution in C202 instance.

heuristic algorithm that has been successfully applied to various optimization problems; however, it has not yet been utilized for solving the VRPTW. In this study, we introduce the Ebola Optimization Search Algorithm (EOSA) to address the VRPTW. Additionally, the Taguchi method is employed to evaluate the impact of the contact rate on solution quality. To further enhance the solution process, we propose a modified version of EOSA, known as the Modified EOSA, which incorporates control parameters. Experimental results indicate that MEOSA effectively minimizes travel distance but is less efficient in reducing the number of vehicles required. Future research could explore the complexity of the problem and its applicability to real-world scenarios to enhance solution effectiveness.

Acknowledgments

This research was funded by College of Industrial Technology, King Mongkut's University of Technology North Bangkok (Grant No. Res-CIT314/2023).

References

[1] Dantzig GB, Ramser JH. The truck dispatching problem. Management Science. 1959;6(1):80-91.

[2] Elshaer R, Awad H. A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants. Computers & Industrial Engineering. 2020;140:106242.

[3] Iqbal S, Kaykobad M, Rahman MS. Solving the multi-objective vehicle routing problem with soft time windows with the help of bees. Swarm and Evolutionary Computation. 2015;24:50-64.

[4] Miranda DM, Conceição SV. The vehicle routing problem with hard time windows and stochastic travel and service time. Expert Systems with Applications. 2016;64:104-16.

[5] SS VC, HS A. Nature inspired meta heuristic algorithms for optimization problems. Computing. 2022;104(2):251-69.

[6] Cordeau JF, Laporte G, Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society. 2001;52(8):928-36.

[7] Vincent FY, Susanto H, Jodiawan P, Ho TW, Lin SW, Huang YT. A simulated annealing algorithm for the vehicle routing problem with parcel lockers. IEEE Access. 2022;10:20764-82.

[8] Berger J, Barkaoui M. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Computers & Operations Research. 2004;31(12):2037-53.

[9] Ding Q, Hu X, Sun L, Wang Y. An improved ant colony optimization and its application to vehicle routing problem with time windows. 2012;98:101-7.

[10] Marinakis Y, Marinaki M, Migdalas A. A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows. Information Sciences. 2019;481:311-29.

[11] Bräysy O. A reactive variable neighborhood search for the vehicle-routing problem with time windows. INFORMS Journal on Computing. 2003;15(4):347-68.

[12] Tan L, Lin F, Wang H. Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows. Neurocomputing. 2015;151:1208-15.

[13] Zhang J, Yang F, Weng X. An evolutionary scatter search particle swarm optimization algorithm for the vehicle routing problem with time windows. IEEE Access. 2018;6:63468-85.

[14] Wu L, He Z, Chen Y, Wu D, Cui J. Brainstorming-based ant colony optimization for vehicle routing with soft time windows. IEEE Access. 2019;7:19643-52.

[15] Shen Y, Liu M, Yang J, Shi Y, Middendorf M. A hybrid swarm intelligence algorithm for vehicle routing problem with time windows. IEEE Access. 2020;8:93882-93.

[16] Wei X, Xiao Z, Wang Y. Solving the vehicle routing problem with time windows using modified rat swarm optimization algorithm based on large neighborhood search. Mathematics. 2024;12(11):1702.

[17] He M, Wei Z, Wu X, Peng Y. An adaptive variable neighborhood search ant colony algorithm for vehicle routing problem with soft time windows. IEEE Access. 2021;9:21258-66.

[18] Aggarwal D, Kumar V. Performance evaluation of distance metrics on firefly algorithm for VRP with time windows. International Journal of Information Technology. 2021;13(6):2355-62.

[19] Ahmed ZH, Maleki F, Yousefikhoshbakht M, Haron H. Solving the vehicle routing problem with time windows using modified football game algorithm. Egyptian Informatics Journal. 2023;24(4):100403.

[20] Wu Q, Xia X, Song H, Zeng H, Xu X, Zhang Y, et al. A neighborhood comprehensive learning particle swarm optimization for the vehicle routing problem with time windows. Swarm and Evolutionary Computation. 2024;84:101425.

[21] Chai S, Kamaluddin M, Rashid MFFA. Optimisation of vehicle routing problem with time windows using Harris hawks optimiser. Journal of Mechanical Engineering and Sciences. 2022;16(3):9056-65.

[22] Oyelade ON, Ezugwu AES, Mohamed TI, Abualigah L. Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. IEEE Access. 2022;10:16150-77.

[23] Ashwini C, Sellam V. EOS-3D-DCNN: Ebola optimization search-based 3D-dense convolutional neural network for corn leaf disease prediction. Neural Computing and Applications. 2023;35(15):11125-39.

[24] Matheen M, Sundar S. A novel technique to mitigate data redundancy and improve network lifetime using fuzzy criminal search ebola optimization for WMSN. Sensors. 2023;23(4):2218.

[25] Zare P, Davoudkhani IF, Mohajery R, Zare R, Ghadimi H, Ebtehaj M. Multi-objective coordinated optimal allocation of distributed generation and D-STATCOM in electrical distribution networks using ebola optimization search algorithm. In: 2023 8th International Conference on Technology and Energy Management (ICTEM). IEEE; 2023. p. 1-7.

[26] Oyelade ON, Ezugwu AE. EOSA-GAN: Feature enriched latent space optimized adversarial networks for synthesization of histopathology images using ebola optimization search algorithm. Biomedical Signal Processing and Control. 2023;84:104734.

[27] Oyelade ON, Ezugwu AE. Immunity-based ebola optimization search algorithm for minimization of feature extraction with reduction in digital mammography using CNN models. Scientific Reports. 2022;12(1):17916.

[28] Mohamed TI, Oyelade ON, Ezugwu AE. Automatic detection and classification of lung cancer CT scans based on deep learning and ebola optimization search algorithm. PLoS One. 2023;18(8):e0285796.

[29] Hoos HH. Automated algorithm configuration and parameter tuning. In: Autonomous Search. Springer; 2012. p. 37-71.

[30] Kazikova A, Pluhacek M, Senkerik R. Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison? In: Mendel. vol. 26; 2020. p. 9-16.

[31] Akinola O, Oyelade ON, Ezugwu AE. Binary ebola optimization search algorithm for feature selection and classification problems. Applied Sciences. 2022;12(22):11787.

[32] Oyelade ON, Agushaka JO, Ezugwu AE. Evolutionary binary feature selection using adaptive ebola optimization search algorithm for high-dimensional datasets. PLoS One. 2023;18(3):e0282812.

[33] Gümüş DB, Özcan E, Atkin J. An analysis of the Taguchi method for tuning

a memetic algorithm with reduced computational time budget. In: Computer and Information Sciences: 31st International Symposium, ISCIS 2016, Kraków, Poland; 2016. p. 12-20.

[34] Taguchi G, Chowdhury S, Wu Y. Taguchi's quality engineering handbook. John Wiley & Sons; 2005.