International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 1277–1296

APPLICATION OF PARALLEL GREY WOLF OPTIMIZER FOR OPTIMAL SOLVING MULTI-DEPOT VEHICLE ROUTING PROBLEM

Supaporn Suwannarongsri

Department of Materials Handling and Logistics Engineering
Faculty of Engineering
King Mongkut's University of Technology North Bangkok
1518 Pracharaj-1 Road, Bangsue, Bangkok 10800, Thailand
supaporn.s@eng.kmutnb.ac.th

Received March 2025; revised June 2025

Abstract. *This paper proposes the application of the parallel grey wolf optimizer (PGWO), one of the newest modified versions of the grey wolf optimizer (GWO), to solve the multi-depot vehicle routing problem (MDVRP). The PGWO was developed by using the GWOs as the search units for running on a single-CPU platform associated with the partitioning mechanism (PM) for dividing an entire search area into many sub-search areas for each GWO to search for solutions in non-overlapping areas, the sequencing mechanism (SM) for organizing GWOs to run one-by-one on each iteration, and the discarding mechanism (DM) for eliminating some inferior GWOs to speed up the search process. The objective of the MDVRP consisting of many depots and vehicles is to determine the optimal vehicle route to minimize the total distance satisfying the particular constraints and criteria. In this paper, the PGWO is applied to optimally solving ten selected real-world MDVRP consisting of approximately 100-300 locations. Experimental results obtained by the PGWO will be compared with those obtained by the original GWO, cuckoo search (CS), particle swarm optimization (PSO), and genetic algorithm (GA). From experimental results, it was found that the PGWO can provide optimal solutions of all ten selected MDVRP with shorter total distance than the original GWO, CS, PSO, and GA, significantly.*
Keywords: Parallel grey wolf optimizer, Grey wolf optimizer, Cuckoo search, Particle swarm optimization, Genetic algorithm, Multi-depot vehicle routing problem

1. **Introduction.** One of the real-world logistic engineering problems is the vehicle routing problem (VRP). It is considered as a class of the NP-hard combinatorial optimization problem with multiple practical applications in transportation, distribution, and collection [1-3]. For the classical VRP, a fleet of vehicles from a depot serves a geographically spread group of customers with known demands and then returns to that depot. The objective is to find the optimal distribution solution with minimum distances while satisfying the customer demands, routing constraints, vehicle restrictions, and depot limitations.

In 1959, Dantzig and Ramser [4] firstly proposed the multi-depot vehicle routing problem (MDVRP) as a generalization of the VRP. The MDVRP focuses on the pickup and/or delivery of products from several depots to many customers. In general, there is a set of customer locations to be served by a set of vehicles from a set of depots established in different places. The objective of the MDVRP is to minimize total distance in order to minimize the overall costs and to maximize the customers' demand by optimizing the

DOI: 10.24507/ijicic.21.05.1277

sequence of customer locations visited by each vehicle, satisfying such conditions and criteria as distance, time, and cost involved in the operation. Consisting of a fleet of vehicles, the MDVRP includes different service requirements (pickup and/or delivery of products) at each location, different capacities, and time constraints of each vehicle in the fleet. In the MDVRP, vehicles leave from one of the depots, serve customers along the routes, and return to the depot where they leave after completion of their routes. In addition, each location will be visited exactly once by any vehicle in a fleet [4,5]. For example, the simple MDVRP consisting of 2 depots and 9 customer locations is illustrated in Figure 1, where customer locations are expressed with circles and depots are mentioned by using squares. Traveling distances between 2 consecutive locations are characterized by the numbers (expressed in kilometer) on the dash-dotted lines as visualized in Figure 1.

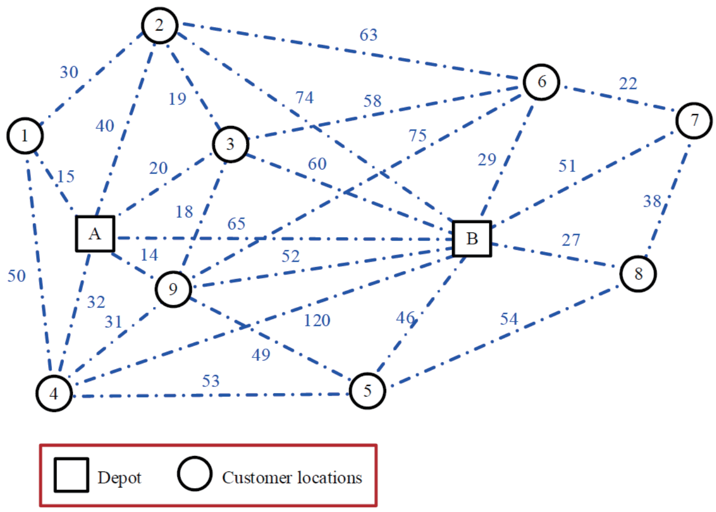


FIGURE 1. Example of MDVRP consisting of 2 depots and 9 customer locations

As one of the NP-hard combinatorial problems, many researchers developed exact methods for solving the MDVRP [6,7]. In the literature, the MDVRP can be efficiently solved by the efficient metaheuristic optimization search techniques based on the modern optimization approach, for instance, tabu search (TS) [8-10], genetic algorithm (GA) [11,12], variable neighborhood search (VNS) [13], particle swarm optimization (PSO) [14-18], ant colony optimization (ACO) [19-23], cuckoo search (CS) [24-26], artificial bee colony (ABC) [27], iterated local search (ILS) [28], sector combination optimization (SCO) [29], and modern metaheuristic algorithm (MoMA) [30]. However, these metaheuristic techniques do not guarantee optimal solutions, but they generally promise a near optimal solution within a reasonable solution search time.

Nowadays, one of the newest metaheuristic optimization search techniques named the parallel grey wolf optimizer (PGWO) has been proposed for global optimization in 2025 [31]. The PGWO algorithm was developed to enhance the search performance of the original grey wolf optimizer (GWO), which mimics the leadership hierarchy and hunting mechanisms of grey wolves in nature. The hunting mechanisms of grey wolves consist of four main strategies, i.e., encircling prey, hunting for prey, searching for prey, and attacking prey [32-35]. The PGWO was proposed for running on a single-CPU. By using the original GWO as the search unit, the PGWO possesses the partitioning mechanism (PM) utilized to divide an entire search area into many sub-search areas for each GWO to search for solutions in non-overlapping areas, the sequencing mechanism (SM) used to organize the search units (GWOs) to run one-by-one on each iteration as the time sharing with multiple points single strategy (MPSS), and the discarding mechanism (DM) conducted to eliminate some inferior GWOs to speed up the search process. The PGWO was tested against several benchmark optimization problems to perform its effectiveness and search performance. When compared to the original GWO, which has demonstrated superior search performance over PSO, gravitational search algorithm (GSA), differential evolution (DE), evolutionary programming (EP), and evolution strategy (ES) [32], the PGWO achieved even more efficient global optimization, exhibiting higher success rates, fewer search iterations, and less computation times. In addition, the PGWO requires only a few search parameters, making the algorithm simple and easy to use. Bearing in mind that the application of the PGWO to real-world MDVRP problems has not yet been reported in the literature, this paper applies the PGWO to optimally solving ten real-world MDVRP instances, each consisting of approximately 100-300 locations selected from literature. The results obtained by the PGWO are compared with those from the original GWO, CS, PSO, and GA to evaluate its effectiveness. Experimental results reveal that the PGWO consistently achieves shorter total distances across all ten problem instances, significantly outperforming the original GWO, CS, PSO, and GA, as reported in this paper. These findings demonstrate both the novelty and the practical relevance of the proposed approach.

This paper consists of five sections. After an introduction proposed in Section 1, the remaining of the paper is organized as follows. The problem formulation including the MDVRP model, the objective and constrained functions are illustrated in Section 2. The PGWO algorithms for the MDVRP optimization are described in Section 3. Experimental results and discussions are provided in Section 4. Finally, conclusions and future research are given in Section 5.

2. **Problem Formulation.**

In this section, the VRP model is firstly described. Then, the MDVRP model is illustrated as follows.

2.1. **VRP model.**

The simple VRP is modelled upon a connected and directed graph $G = (N, A)$, where N is the set of nodes or vertices which represent the customer locations and A is the set of arcs which represents the distances. The set of customers C and a depot 0 ($N = C \cup 0$) is also given. The distance (or cost depending on the distance) using the arc (i, j) is represented by $C_{i,j}$ (because the fleet is homogeneous, the cost is the same for all the vehicles). The index set of available vehicles to serve the demand is K and the demand of every customer is d_i. Lastly, as this is a homogeneous fleet, the capacity, q, of all the vehicles is equal. In order to model the VRP, two variables are usually considered, i.e., the binary variable $X_{ij}^k \in \{0, 1\}$ and an auxiliary variable $U_i \geq 1$. If the arc (i, j) is used by the kth vehicle, then the binary variable $X_{ij}^k = 1$. Otherwise, $X_{ij}^k = 0$. The auxiliary variable U_i represents the location in which node i is visited on its route. The objective of this variable is to ensure that the routes of each vehicle are well-defined and avoid cycling (sub-tour).

The VRP model is expressed in (1) to minimize the total distance of the system having served every customer in the network satisfying to the constrained functions as stated in (2)-(10) [4,36].

$$\text{Minimize } f_{\text{VRP}} = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij} X_{ij}^k \tag{1}$$

$$\text{subject to } \sum_{k \in K} \sum_{j \in N} X_{ij}^k \geq 1, \quad \forall i \in C \tag{2}$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^k \leq q, \quad \forall k \in K \tag{3}$$

$$\sum_{j \in N} X_{oj}^k = 1, \quad \forall k \in K \tag{4}$$

$$\sum_{i \in N} X_{io}^k = 1, \quad \forall k \in K \tag{5}$$

$$\sum_{i \in N} X_{ih}^k - \sum_{j \in N} X_{hj}^k = 0, \quad \forall h \in C, \ \forall k \in K \tag{6}$$

$$U_0 = 1 \tag{7}$$

$$U_j \geq (U_i + 1) - M\left(1 - \sum_{k \in K} X_{ij}^k\right), \quad \forall i \in N, \ \forall j \in C \tag{8}$$

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

$$2 \leq U_i \leq |N|, \quad \forall i \in C \tag{10}$$

The constrained function in (2) ensures that every customer is visited at least once. The constrained function in (3) limits the amount of customer locations served by a vehicle according to their capacity. The constrained functions in (4) and (5) guarantee that the vehicles depart and finish the route at the depot 0. The constrained function in (6) assures that the flow continues through the network. The constrained function in (7) is used to secure that the first node to be visited according to auxiliary variables U is the depot. The constrained function in (8) determines that the value of U_j must be higher than U_i, when i is the previously visited node. The constrained functions in (9) and (10) are used to avoid the possibility of having sub-tours in the routes.

2.2. **MDVRP model.** The MDVRP is the particular case of the VRP. In other words, the MDVRP is a generalization of the VRP [4]. It is modelled upon a connected and directed graph $G = (N, A)$, where N is the set of nodes or vertices which represent the customer locations and A is the set of arcs which represents the distances as the VRP. The set of customers C and depots D is given ($N = C \cup D$). The distance (or cost depending on the distance) using the arc (i, j) is represented by $C_{i,j}$. The set of available vehicles to attend the demand is K. As a homogeneous fleet, the capacity, q, of all the vehicles is equal. The demand to satisfy every customer is d_i. Lastly, as the depots have a finite capacity to attend the vehicles, the maximum capacity of depot i is added by $MaxDepo_i$. In order to model the MDVRP, two variables used in the VRP are kept, and a new variable is added, i.e., the binary variable $X_{ij}^k \in \{0, 1\}$, an auxiliary variable $U_i \geq 1$, and binary variable $W_d^k \in \{0, 1\}$. If the arc (i, j) is used by the kth vehicle, then the binary variable $X_{ij}^k = 1$. Otherwise, $X_{ij}^k = 0$. The auxiliary variable U_i represents the location in which node i is visited on its route. The objective of this variable is to ensure that the routes of each vehicle are well-defined and avoid cycling. If the vehicle k is served from depot d, then the binary variable $W_d^k = 1$. Otherwise, $W_d^k = 0$.

The MDVRP model is stated in (11) to minimize the total distance of the system having served every customer in the network satisfying to the constrained functions as stated in (12)-(21) [4,36,37].

$$\text{Minimize } f_{\text{MDVRP}} = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij} X_{ij}^k \tag{11}$$

$$\text{subject to } \sum_{k \in K} \sum_{j \in N} X_{ij}^k \geq 1, \quad \forall i \in C \tag{12}$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^k \leq q, \quad \forall k \in K \tag{13}$$

$$\sum_{j\in N} X_{hj}^k = W_h^k, \quad \forall h \in D, \, \forall k \in K \tag{14}$$

$$\sum_{i\in N} X_{ih}^k = W_h^k, \quad \forall h \in D, \, \forall k \in K \tag{15}$$

$$\sum_{i\in N} X_{ih}^k - \sum_{j\in N} X_{hj}^k = 0, \quad \forall h \in C, \, \forall k \in K \tag{16}$$

$$\sum_{k\in K} W_i^k \leq MaxDepo_i, \quad \forall i \in D \tag{17}$$

$$U_i = 1, \quad \forall i \in D \tag{18}$$

$$U_j \geq (U_i + 1) - M\left(1 - \sum_{k\in K} X_{ij}^k\right), \quad \forall i \in N, \, \forall j \in C \tag{19}$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \, \forall k \in K \tag{20}$$

$$2 \leq U_i \leq |N|, \quad \forall i \in C \tag{21}$$

Regarding to the model in (11) and constrained functions in (12)-(21), the formulation of the MDVRP follows the same ideas as the one explained before for the VRP, and in general for the routing problems. It is necessary to ensure well-defined routes. The difference is that now constraints in (4) and (5) which ensure that all vehicles depart and finish the route at the depot 0, are subject to variable W of constraints in (14) and (15). Thus, if a vehicle is not associated to a depot, it cannot depart from that depot or return to that depot. Also, a new constraint in (17) is needed to guarantee that the capacity of the depot is not violated.

3. PGWO Algorithm for MDVRP Optimization.

The PGWO algorithm is briefly described in this section. Then, the MDVRP optimization by the PGWO is illustrated in detail as follows.

3.1. PGWO algorithm.

Referring to the original GWO proposed in 2014 [32], the GWO was developed by mimicking the leadership hierarchy and hunting mechanism of grey wolves in nature consisting of four main mechanisms, i.e., encircling prey, hunting for prey, searching for prey, and attacking prey, by using four gray wolf types, i.e., alpha (α), beta (β), delta (δ), and omega (ω). The alpha (α) is the most powerful wolf standing for the best solution. The second and third best solutions are beta (β) and delta (δ), respectively. The rest of the candidate solutions are assumed to be omega (ω). In the original GWO algorithm, the hunting (optimization) is guided by alpha (α), beta (β), and delta (δ). The omega (ω) wolves follow these three wolves [32].

For the encircling prey mechanism, the mathematical relationships expressed in (22) and (23) are conducted, where t is the current iteration, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\vec{X}_p$ is the position vector of the prey, and $\vec{X}$ is the position vector of a grey wolf. The vectors $\vec{A}$ and $\vec{C}$ can be calculated by (24) and (25), respectively, where the component $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations, and r_1, r_2 are random vectors in $[0, 1]$.

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \tag{22}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{23}$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \tag{24}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{25}$$

For the hunting for prey mechanism, the mathematical relationships stated in (26)-(28) are employed.

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)\right|, \ \vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)\right|, \ \vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)\right| \quad (26)$$

$$\vec{X}_1(t) = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \left(\vec{D}_\alpha\right), \ \vec{X}_2(t) = \vec{X}_\beta(t) - \vec{A}_2 \cdot \left(\vec{D}_\beta\right), \ \vec{X}_3(t) = \vec{X}_\delta(t) - \vec{A}_3 \cdot \left(\vec{D}_\delta\right) \quad (27)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3} \quad (28)$$

For the searching for prey and the attacking prey mechanisms, the fluctuation range of $\vec{A}$ in (24) is decreased by $\vec{a}$. In other words, $\vec{A}$ is a random value in the interval $[-a, a]$, where a is decreased from 2 to 0 over the course of iterations. When random values of $\vec{A}$ are in $[-1, 1]$, the next position of a search agent can be in any position between its current position and the position of the prey. If $|A| \geq 1$, the searching for prey mechanism (exploration or diversification of the algorithm) will be invoked, and if $|A| < 1$, the attacking prey mechanism (exploitation or intensification of the algorithm) will be activated [32-35]. The original GWO algorithm can be represented by the pseudo code as shown in Figure 2.

```
Initialize:
- Initialize the objective function f(x) and search spaces
- Initialize the grey wolf population Xi (i = 1,2,…,n)
- Initialize a, A, and C
- Evaluate all agents Xi via f(x)
- Ranking Xα = the best agent, Xβ = the second best agent, and
  Xδ = the third best agent
while (t ≤ Max number of iterations)
    for i = 1 : n (all n grey wolf population)
        - Update the position of the current search by using (28)
    end for
    - Update a, A, and C
    - Evaluate all agents Xi via f(x)
    - Update Xα, Xβ and Xδ
    - t = t + 1
end while
- Report the best solution Xα
```

FIGURE 2. Pseudo code of the original GWO algorithm

The PGWO algorithm was developed and firstly proposed in 2025 for running on a single-CPU [31]. By using the original GWO as the search unit, the PGWO algorithm utilizes the PM to divide an entire search area into many sub-search areas for each GWO. The SM is conducted to organize the search units (GWOs) to run one-by-one on each iteration. Also, the DM is used to eliminate some inferior GWOs to speed up the search process.

The algorithm of the PGWO will initialize the objective function $f(\boldsymbol{x})$, entire search spaces, number of GWOs, N, (as the search units of the PGWO), maximum iteration Max_Iter (as the termination criteria: TC), and initial iteration Iter = 1 (as a counter). Then, the algorithm will activate the PM to generate N sub-search-spaces. After that, the algorithm will invoke the SM and execute GWO_1, GWO_2, …, GWO_N, for $N = N - K$, $K \leq N - 1$, $N_{\min} = 1$. Then, the algorithm will activate the DM to eliminate some inferior GWOs. If the TC is met (Iter $\geq$ Max_Iter), terminate the search process and report the best solution found. Otherwise, the algorithm will update Iter = Iter + 1 for proceeding

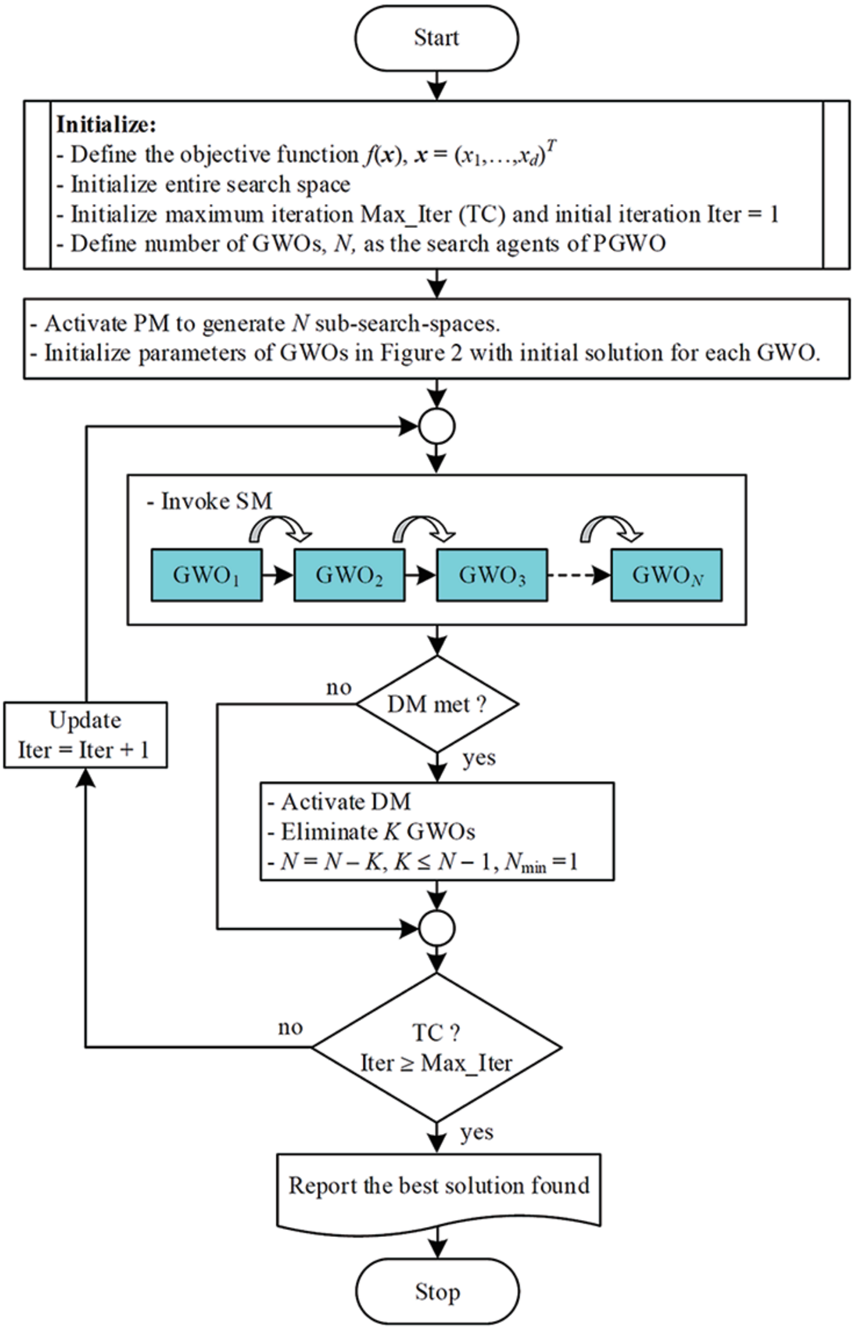


FIGURE 3. Flow diagram of the PGWO algorithm [31]

a next iteration. The PGWO algorithm can be represented by the flow diagram as shown in Figure 3, where GWO_1, GWO_2, …, GWO_N are the original GWO algorithms as represented by the pseudo code shown in Figure 2.

3.2. **PGWO-based MDVRP optimization.** In this work, the PGWO algorithm is applied to optimally solving ten selected real-world MDVRP. Regarding to the PGWO algorithm, the MDVRP optimization by the PGWO can be described step-by-step as follows.

Step-1 *For MDVRP*: initialize the objective function f_{MDVRP} as stated in (11) and the constrained functions as expressed in (12)-(21), entire search space, number of customer locations N and their corresponding distances $C_{i,j}$, number of vehicles K in the fleet, and number of depots D.

For GWO: initialize the grey wolf population n, and the values of a, A, and C. Evaluate all agents X_i via the objective function f_{MDVRP} in (11) and the constrained functions in (12)-(21). Ranking X_α as the best agent, X_β as the second-best agent, and X_δ as the third-best agent.

For PGWO: initialize the number of GWOs = N, maximum iteration Max_Iter, and initial iteration Iter = 1.

Step-2 Activate the PM to generate N sub-search-spaces with initial solutions $\boldsymbol{x}_i$ for each GWO.

Step-3 Invoke the SM and execute GWO$_1$, GWO$_2$, ..., GWO$_N$, for $N = N - K$, $K \leq N-1$, $N_{\min} = 1$. All N GWOs are executed one-by-one on this current iteration according to Figure 2. From GWO$_1$ to GWO$_N$, for all grey wolf population n, update the position of the current search by using (28). If $f_{\text{MDVRP}}(\boldsymbol{x}^*) < f_{\text{MDVRP}}(\boldsymbol{x})$, update $\boldsymbol{x} \leftarrow \boldsymbol{x}^*$. Then, update the values of a, A, and C. Evaluate all agents X_i via the objective function f_{MDVRP} in (11) and the constrained functions in (12)-(21). After that, update X_α, X_β, and X_δ.

Step-4 Activate the DM to eliminate some inferior GWOs. K GWOs, $K \leq N-1$, $N_{\min} = 1$, having lower convergence than $N - K$ GWOs will be discarded. N GWOs, $N = N - K$, $K \leq N - 1$, $N_{\min} = 1$, will be maintained to continue the search for the optimal solutions.

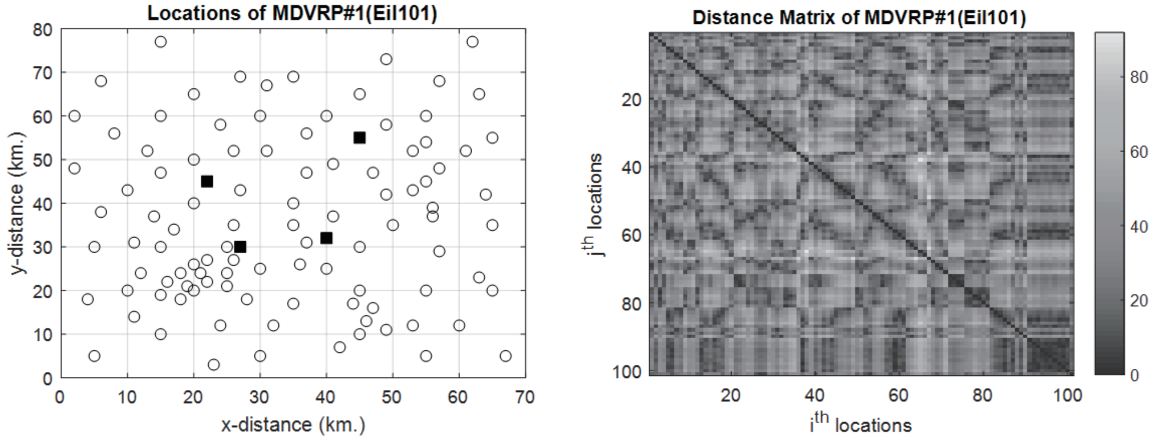
Step-5 If the TC is met (Iter $\geq$ Max_Iter), terminate the search process. All best solutions of N GWOs will be ranked. Among them, the best solution will be reported. Otherwise, go to Step-6.

Step-6 Update Iter = Iter + 1 and go back to Step-3 for executing a next iteration.

4. **Experimental Results and Discussions.** In this work, the objective of the MDVRP optimization based on the PGWO algorithm is to minimize total distance. Therefore, the pickup/delivery time requirements, and the traffic situation are neglected. The load capacity q of all available vehicles K is equal as a homogeneous fleet. Also, the symmetric case, $C_{ij} = C_{ji}$, is assumed. Ten real-world MDVRP consisting of approximately 100-300 locations are selected from [38-40] summarized in Table 1. For example, the 101 customer locations and its 4 depots of MDVRP#1 (Eil101) are plotted in Figure 4(a), where the white circles stand for the customer locations and the black squares stand

Table 1. Ten selected real-world MDVRP problems

Problems	Names	Number of customer locations	Optimal tour for one vehicle (km.)	Comments
MDVRP#1	Eil101	101	629	Eilon
MDVRP#2	Ch130	130	6,110	Churritz
MDVRP#3	Ch150	150	6,528	Churritz
MDVRP#4	Pu195	195	2,323	Pulleyblank
MDVRP#5	Re198	198	15,780	Reinelt
MDVRP#6	Gr202	202	40,160	Groetschel
MDVRP#7	Gr229	229	134,602	Groetschel
MDVRP#8	Pr264	264	49,135	Padberg/Rinaldi
MDVRP#9	Pr299	299	48,191	Padberg/Rinaldi
MDVRP#10	Lin318	318	42,029	Lin/Kernighan



(a) Locations (b) Distance matrix

Figure 4. Customer and depot locations and distance matrix of MDVRP#1 (Eil101)

for the depots. When the symmetric case is assumed, $C_{ij} = C_{ji}$, the distance matrix of MDVRP#1 (Eil101) is depicted in Figure 4(b), when the shade of gray represents the distance between the i-th and the j-th locations. This implies that the lighter the shade of gray, the greater the distance.

For comparative purposes, the GA, PSO, CS, and GWO algorithms are conducted due to their widespread adoption as population-based metaheuristic optimization techniques. The GA was developed from the principles of natural selection and evolution [41,42]. It lies in three main operators, i.e., selection, crossover, and mutation. These operators are used to evolve a population of candidate solutions towards an optimal solution. The GA has been widely applied to a variety of fields, including optimization, machine learning, and engineering. For further details on the GA algorithm and its application, readers may refer to the recent review papers in [43,44]. For PSO [45,46], it was originally developed for simulating social behavior and, in particular, motion of organisms in a bird flock or fish school. The main algorithm of the PSO involves iteratively updating the velocity and position of particles in a swarm, guided by their personal best and the global best positions discovered so far. Each particle represents a potential solution, and the swarm collectively explores the search space, converging towards optimal solutions. The PSO has been widely applied to a diverse range of optimization problems, particularly in the fields of engineering, science, and data science. For more details on the PSO algorithm and its application, readers may refer to the recent review papers in [47,48]. For CS [49,50], it was developed as one of the nature-inspired metaheuristic algorithms. The CS algorithm mimics the obligate brood parasitism behavior of some cuckoo species in combination with the Lévy flights, a type of random walk representing the behavior of some birds and fruit flies, to explore the search space and find optimal solutions. The CS has been widely applied to various real-world optimization problems, including engineering design, resource allocation, scheduling, and image processing. For comprehensive discussions on the CS algorithm and its applications, readers may refer to the recent review papers in [51,52].

For solving ten selected real-world MDVRP in Table 1 and comparison, the GA, PSO, CS, GWO, and PGWO algorithms were coded by MATLAB version 2018b run on Intel(R) Core(TM) i5-3470 CPU@3.60GHz, 4.0GB-RAM. For each MDVRP, 50 trial-runs are executed to search for its best solution. The depot's locations are assumed to be arbitrary defined. Each depot should serve at least 10 customer locations. The daily working time of any vehicle is defined as no longer than 8 hr. Also, 80 km/hr. is approximated for the average speed of all vehicles. Thus, the overall distance of each vehicle must not exceed 640 km/day [30]. These data are employed to define the number of depots D and the number of vehicles K in a fleet of each MDVRP problem.

From the preliminary studies of the original GWO, the numbers of grey wolf population $n = 10, 20, 30, \ldots, 200$ are tested against ten selected MDVRP problems. It was found that the best value of n is 25-40 allowing the GWO algorithm can reach the optimal solutions of all problems. Thus, $n = 30$ is set for the GWO. For a fair comparison of the MDVRP optimization, the searching parameters of other algorithms are set from the recommendations and the preliminary studies of the GWO as follows.

For GA [41,42], number of populations = 30 (fixed as n of GWO), crossover probability = 0.95 and mutation probability = 0.05.

For PSO [45,46], number of swarms = 30 (fixed as n of GWO), cognitive learning rate = 2.0 and social learning rate = 2.0.

For CS [49,50], number of nests (or cuckoos) = 30 (fixed as n of GWO) and discovery probability = 0.25.

For PGWO [31], the PGWO having 2, 4, 8, and 16 GWOs are denoted as PGWO#2, PGWO#4, PGWO#8, and PGWO#16, respectively. The number of grey wolf population $n = 30$ is also set for all GWOs in PGWO algorithm. The PM is set as the symmetrical boundaries of the sub-search-spaces for 2, 4, 8, and 16 GWOs for each MDVRP. The SM is set as a time-sharing technique for all GWOs running one-by-one as sequential manner in each iteration based on a single-CPU. The search process is repeated until one of the GWOs hits the optimal solution or some inferior GWOs are discarded by the DM. For all MDVRP problems, the DM is set by (29)-(31) [31], where q_i is the iteration at which the DM is invoked, and D_i is the number of GWOs remaining after DM is activated each time. q_i and D_i can be calculated by (30) and (31), respectively. In each time of the DM activation, the number of GWOs will be discarded by half. Finally, there is only one GWO left to continue the search for the global solutions.

$$N = 2^J, \quad N = 2, 4, 8, 16 \tag{29}$$

$$q_i = \text{Round}\left[\frac{(\text{Max\_Iter})i}{(J+1)}\right], \quad i = 1, \ldots, J \tag{30}$$

$$D_i = \frac{N}{2^i}, \quad i = 1, \ldots, J \tag{31}$$

For all algorithms, the maximum iteration (or maximum generation) Max_Iter = 2,000 is set as the TC. From results over 50 trial-runs, the convergent curves of the GA, PSO, CS, and GWO of the MDVRP#1 (Eil101) are depicted in Figure 5 as the example, while those of the GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16 are plotted in Figure 6 as the example. The convergent curves of other MDVRP problems are omitted because they have a similar form to those in Figure 5 and Figure 6. From Figure 5, it can be observed that the GWO performs higher performance with better solution and faster convergent rate than the CS, PSO, and GA, respectively. From Figure 6, it can be seen that the PGWO#16 performs higher performance with better solution and faster convergent rate than the PGWO#8, PGWO#4, PGWO#2, and GWO, respectively. All experimental results of ten selected MDVRP optimization obtained by the GA, PSO, CS, GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16 are summarized in Table 2. For instance, the optimal tours of the MDVRP#1 (Eil101) obtained by the GA, PSO, CS, GWO,

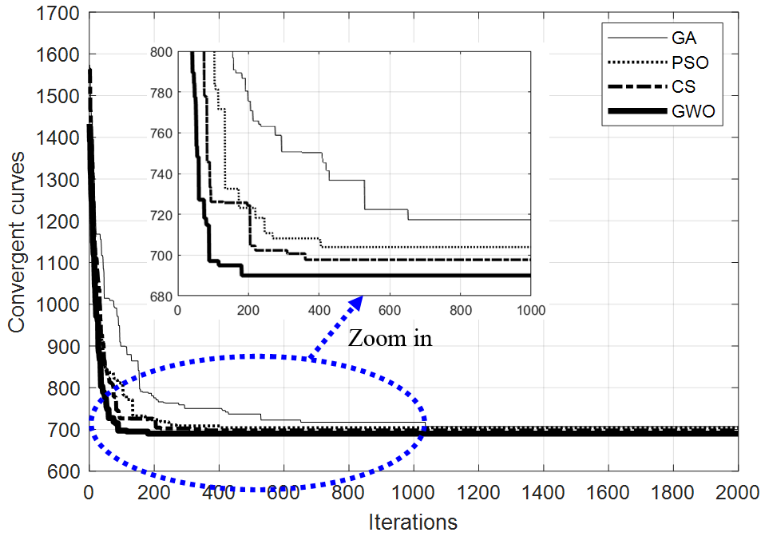


FIGURE 5. Convergent curves of GA, PSO, CS, and GWO for MDVRP#1 (Eil101)

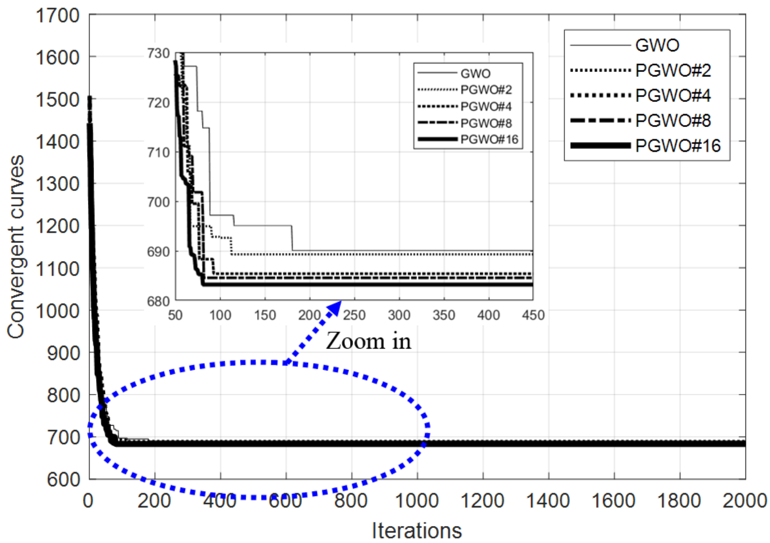


FIGURE 6. Convergent curves of GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16 for MDVRP#1 (Eil101)

PGWO#2, PGWO#4, PGWO#8, and PGWO#16 are plotted in Figures 7-14, respectively. Results in Figures 7-14 and Table 2 are further analyzed as follows.

For the MDVRP#1 (Eil101), it possesses the number of customer locations $N = 101$, the number of depots $D = 4$, and the number of vehicles $K = 4$. From Figures 7-14 and Table 2, the GA provides the total distance of 706.93 km., the PSO performs the total distance of 703.95 km., the CS yields the total distance of 697.74 km., the GWO gives the total distance of 690.03 km., the PGWO#2 provides the total distance of 689.29 km., the PGWO#4 performs the total distance of 685.40 km., the PGWO#8 yields the total distance of 684.57 km., while the PGWO#16 gives the total distance of 683.21 km.

For the MDVRP#2 (Ch130), it has $N = 130$, $D = 8$, and $K = 12$. From Table 2, the GA provides the total distance of 9,670.67 km., the PSO performs the total distance of 9,508.44 km., the CS yields the total distance of 9,312.26 km., the GWO gives the total distance of 8,947.84 km., the PGWO#2 provides the total distance of 8,791.62 km., the

Table 2. Optimal tours of MDVRP obtained by GA, PSO, CS, GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16

Problems	No. of D	No. of K	Optimal tour (km.)							
			GA	PSO	CS	GWO	PGWO#2	PGWO#4	PGWO#8	PGWO#16
MDVRP#1	4	4	706.93	703.95	697.74	690.03	689.29	685.40	684.57	683.21
MDVRP#2	8	12	9,670.67	9,508.44	9,312.26	8,947.84	8,791.62	8,403.06	7,895.18	7,726.43
MDVRP#3	10	14	9,794.51	9,772.05	9,592.40	9,046.17	8,810.78	8,584.52	8,107.75	7,905.56
MDVRP#4	10	6	3,545.23	3,308.78	3,236.87	3,114.36	2,901.24	2,729.35	2,602.94	2,548.22
MDVRP#5	12	26	18,933.45	18,605.29	18,464.31	18,214.23	18,096.55	17,892.14	17,463.73	17,105.92
MDVRP#6	15	70	46,985.32	46,711.54	46,509.78	46,107.65	45,960.43	45,882.27	45,696.08	45,235.47
MDVRP#7	20	220	139,049.70	138,568.62	138,098.51	137,730.72	137,522.08	136,965.41	136,848.07	136,522.53
MDVRP#8	16	80	54,918.62	54,787.38	54,501.06	54,132.35	53,925.13	53,798.22	53,638.31	53,607.50
MDVRP#9	16	80	53,655.08	53,312.47	53,188.92	52,994.84	52,816.24	52,703.19	52,645.89	52,512.31
MDVRP#10	15	70	49,671.98	49,382.65	49,173.54	48,801.05	48,526.82	48,499.50	48,378.02	48,121.76

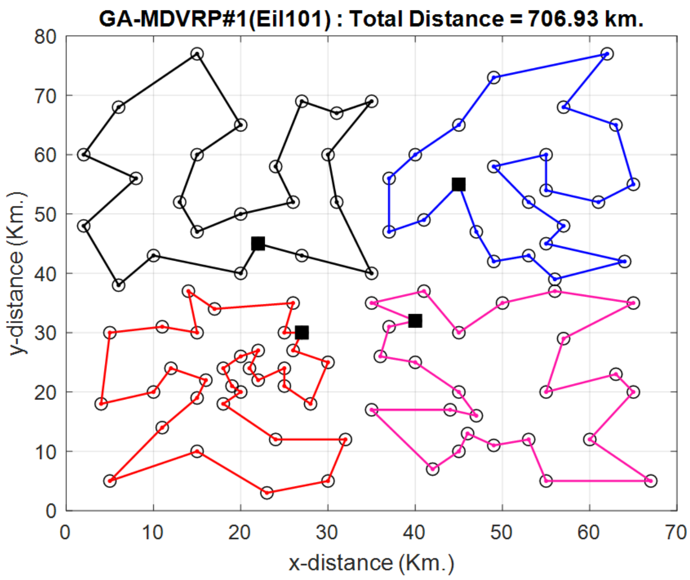


Figure 7. Optimal tour of MDVRP#1 (Eil101) obtained by GA

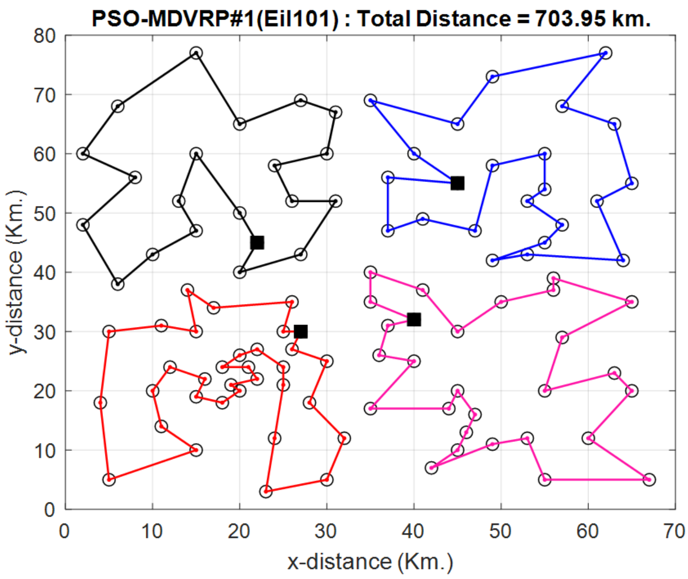


Figure 8. Optimal tour of MDVRP#1 (Eil101) obtained by PSO

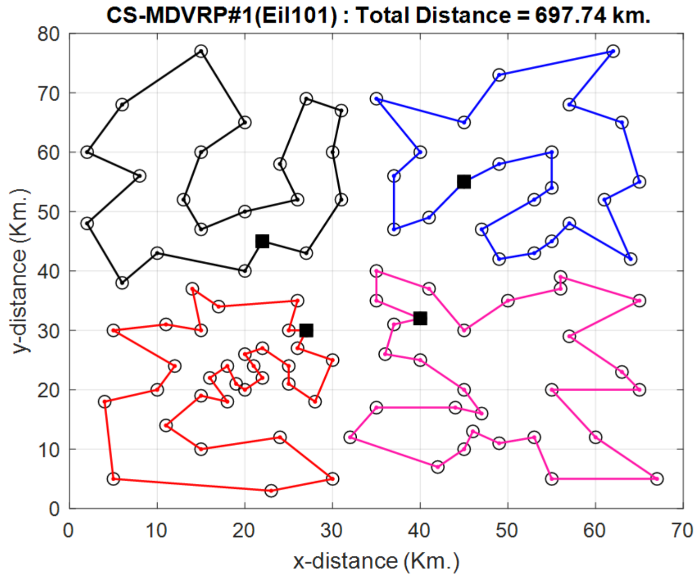


Figure 9. Optimal tour of MDVRP#1 (Eil101) obtained by CS

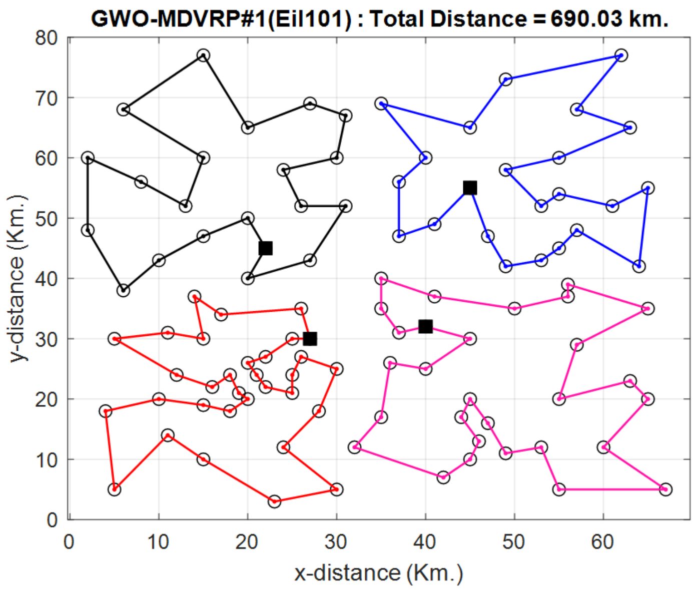


Figure 10. Optimal tour of MDVRP#1 (Eil101) obtained by GWO

PGWO#4 performs the total distance of 8,403.06 km., the PGWO#8 yields the total distance of 7,895.18 km., while the PGWO#16 gives the total distance of 7,726.43 km.

For the MDVRP#3 (Ch150), it possesses $N = 150$, $D = 10$, and $K = 14$. From Table 2, the GA provides the total distance of 9,794.51 km., the PSO performs the total distance of 9,772.05 km., the CS yields the total distance of 9,592.40 km., the GWO gives the total distance of 9,046.17 km., the PGWO#2 provides the total distance of 8,810.78 km., the PGWO#4 performs the total distance of 8,584.52 km., the PGWO#8 yields the total distance of 8,107.75 km., while the PGWO#16 gives the total distance of 7,905.56 km.

For the MDVRP#4 (Pu195), it has $N = 195$, $D = 10$, and $K = 6$. From Table 2, the GA provides the total distance of 3,545.23 km., the PSO performs the total distance of 3,308.78 km., the CS yields the total distance of 3,236.87 km., the GWO gives the total distance of 3,114.36 km., the PGWO#2 provides the total distance of 2,901.24 km., the

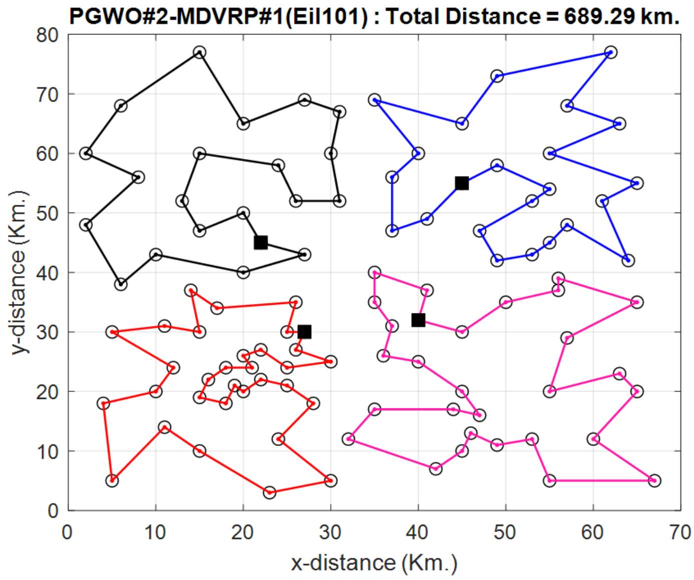


FIGURE 11. Optimal tour of MDVRP#1 (Eil101) obtained by PGWO#2

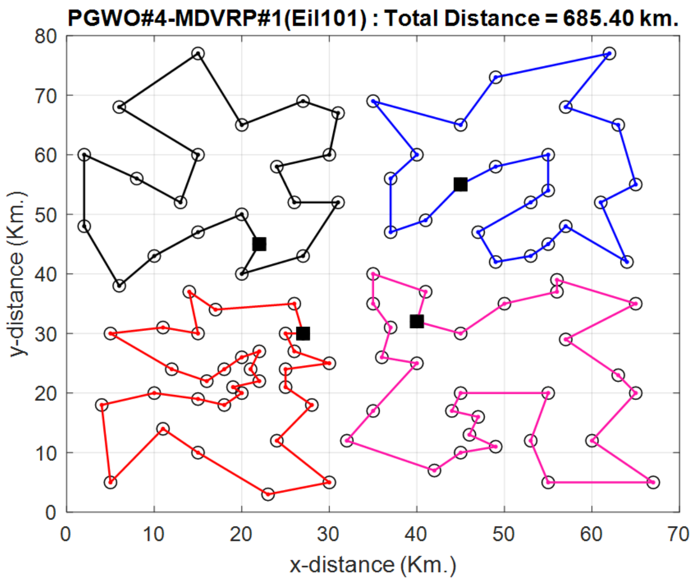


FIGURE 12. Optimal tour of MDVRP#1 (Eil101) obtained by PGWO#4

PGWO#4 performs the total distance of 2,729.35 km., the PGWO#8 yields the total distance of 2,602.94 km., while the PGWO#16 gives the total distance of 2,548.22 km.

For the MDVRP#5 (Re198), it possesses $N = 198$, $D = 12$, and $K = 26$. From Table 2, the GA provides the total distance of 18,933.45 km., the PSO performs the total distance of 18,605.29 km., the CS yields the total distance of 18,464.31 km., the GWO gives the total distance of 18,214.23 km., the PGWO#2 provides the total distance of 18,096.55 km., the PGWO#4 performs the total distance of 17,892.14 km., the PGWO#8 yields the total distance of 17,463.73 km., while the PGWO#16 gives the total distance of 17,105.92 km.

For the MDVRP#6 (Gr202), it has $N = 202$, $D = 15$, and $K = 70$. From Table 2, the GA provides the total distance of 46,985.32 km., the PSO performs the total distance of 46,711.54 km., the CS yields the total distance of 46,509.78 km., the GWO gives the total distance of 46,107.65 km., the PGWO#2 provides the total distance of 45,960.43 km., the

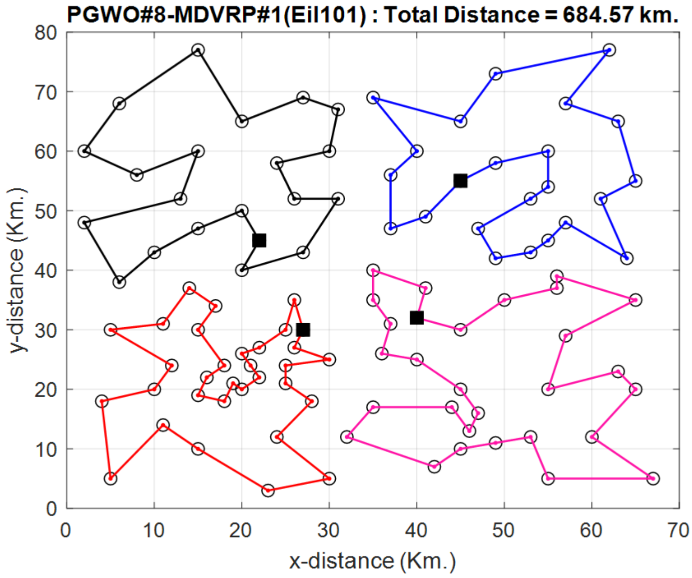


FIGURE 13. Optimal tour of MDVRP#1 (Eil101) obtained by PGWO#8

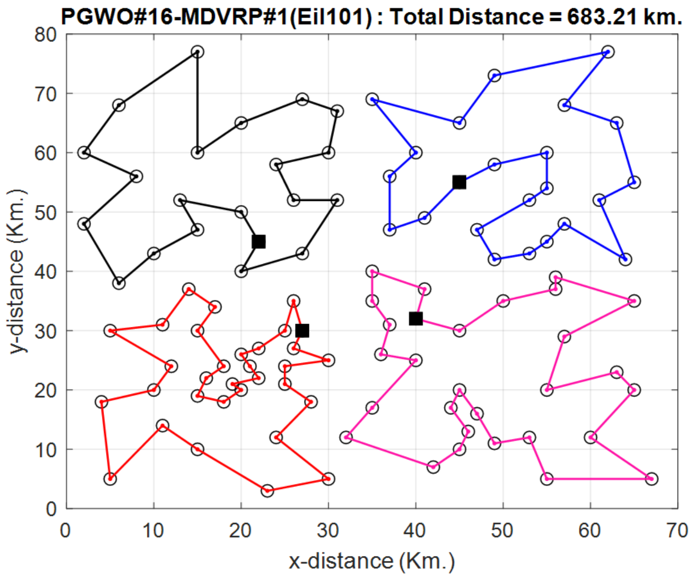


FIGURE 14. Optimal tour of MDVRP#1 (Eil101) obtained by PGWO#16

PGWO#4 performs the total distance of 45,882.27 km., the PGWO#8 yields the total distance of 45,696.08 km., while the PGWO#16 gives the total distance of 45,235.47 km.

For the MDVRP#7 (Gr229), it possesses $N = 229$, $D = 20$, and $K = 220$. From Table 2, the GA provides the total distance of 139,049.70 km., the PSO performs the total distance of 138,568.62 km., the CS yields the total distance of 138,098.51 km., the GWO gives the total distance of 137,730.72 km., the PGWO#2 provides the total distance of 137,522.08 km., the PGWO#4 performs the total distance of 136,965.41 km., the PGWO#8 yields the total distance of 136,848.07 km., while the PGWO#16 gives the total distance of 136,522.53 km.

For the MDVRP#8 (Pr264), it has $N = 264$, $D = 16$, and $K = 80$. From Table 2, the GA provides the total distance of 54,918.62 km., the PSO performs the total distance of 54,787.38 km., the CS yields the total distance of 54,501.06 km., the GWO gives the total distance of 54,132.35 km., the PGWO#2 provides the total distance of 53,925.13 km., the

PGWO#4 performs the total distance of 53,798.22 km., the PGWO#8 yields the total distance of 53,638.31 km., while the PGWO#16 gives the total distance of 53,607.50 km.

For the MDVRP#9 (Pr299), it possesses $N = 299$, $D = 16$, and $K = 80$. From Table 2, the GA provides the total distance of 53,655.08 km., the PSO performs the total distance of 53,312.47 km., the CS yields the total distance of 53,188.92 km., the GWO gives the total distance of 52,994.84 km., the PGWO#2 provides the total distance of 52,816.24 km., the PGWO#4 performs the total distance of 52,703.19 km., the PGWO#8 yields the total distance of 52,645.89 km., while the PGWO#16 gives the total distance of 52,512.31 km.

For the MDVRP#10 (Lin318), it has $N = 318$, $D = 15$, and $K = 70$. From Table 2, the GA provides the total distance of 49,671.98 km., the PSO performs the total distance of 49,382.65 km., the CS yields the total distance of 49,173.54 km., the GWO gives the total distance of 48,801.05 km., the PGWO#2 provides the total distance of 48,526.82 km., the PGWO#4 performs the total distance of 48,499.50 km., the PGWO#8 yields the total distance of 48,378.02 km., while the PGWO#16 gives the total distance of 48,121.76 km.

From overall results of all ten selected real-world MDVRP summarized in Table 2, it was found that the PSO can give shorter total distance than the GA. The CS can yield shorter total distance than the PSO, while the GWO can provide shorter total distance than the CS, PSO, and GA, respectively. Between GWO and PGWO, it was found that the PGWO#2 can give shorter total distance than the GWO. The PGWO#4 can provide shorter total distance than the PGWO#2. The PGWO#8 can yield shorter total distance than the PGWO#4, while the PGWO#16 can give shorter total distance than the PGWO#8, PGWO#4, PGWO#2, and GWO, respectively.

Table 3 provides the average search time (AST) consumed by the GA, PSO, CS, and GWO for all ten selected real-world MDVRP over 50 trial-runs. It was found that the GWO consumes the AST less than the CS, PSO, and GA, respectively.

Table 3. AST of GA, PSO, CS, and GWO for MDVRP optimization over 50 trial-runs

Problems	Average search time: AST (sec.)			
	GA	PSO	CS	GWO
MDVRP#1	35.29	22.64	18.46	15.37
MDVRP#2	52.78	36.82	31.14	26.04
MDVRP#3	59.25	38.47	35.82	28.56
MDVRP#4	48.94	31.53	23.30	19.83
MDVRP#5	122.67	84.90	62.73	38.22
MDVRP#6	145.71	86.48	78.52	42.44
MDVRP#7	187.38	133.56	126.17	74.12
MDVRP#8	136.04	94.21	85.45	56.09
MDVRP#9	132.52	91.08	83.06	51.68
MDVRP#10	129.87	87.59	81.87	45.13
Averages	105.05	70.73	62.65	39.75

Table 4 shows the AST consumed by the GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16 for all ten selected real-world MDVRP over 50 trial-runs. It can be observed that the PGWO#16 consumes the AST more than the PGWO#8, PGWO#4, PGWO#2, and GWO, respectively. This is because the PGWO was proposed for running on a single-CPU platform. The more the search units (GWOs) used in the PGWO, the

Table 4. AST of GWO, PGWO#2, PGWO#4, PGWO#8, and PGWO#16 for MDVRP optimization over 50 trial-runs

Problems	Average search time: AST (sec.)				
	GWO	PGWO#2	PGWO#4	PGWO#8	PGWO#16
MDVRP#1	15.37	26.42	32.45	42.72	61.83
MDVRP#2	26.04	42.61	56.83	71.34	89.26
MDVRP#3	28.56	48.95	67.32	86.40	102.38
MDVRP#4	19.83	34.04	44.06	57.95	70.67
MDVRP#5	38.22	52.57	63.51	74.70	83.55
MDVRP#6	42.44	71.60	94.38	124.56	152.14
MDVRP#7	74.12	121.52	158.87	183.43	208.29
MDVRP#8	56.09	85.76	125.92	150.81	174.33
MDVRP#9	51.68	78.41	112.70	142.28	172.05
MDVRP#10	45.13	73.28	98.84	131.05	164.72
Averages	39.75	63.52	85.49	106.52	127.92

Table 5. EAST and PDAST of PGWO#2, PGWO#4, PGWO#8, and PGWO#16 with respect to GWO for MDVRP optimization

Problems	Equivalent average search time: EAST (sec.)				
	GWO	PGWO#2	PGWO#4	PGWO#8	PGWO#16
MDVRP#1	15.37	13.21	8.11	5.34	3.86
MDVRP#2	26.04	21.31	14.21	8.92	5.58
MDVRP#3	28.56	24.48	16.83	10.80	6.40
MDVRP#4	19.83	17.02	11.02	7.24	4.42
MDVRP#5	38.22	26.29	15.88	9.34	5.22
MDVRP#6	42.44	35.80	23.60	15.57	9.51
MDVRP#7	74.12	60.76	39.72	22.93	13.01
MDVRP#8	56.09	42.88	31.48	18.85	10.90
MDVRP#9	51.68	39.21	28.18	17.79	10.75
MDVRP#10	45.13	36.64	24.71	16.38	10.30
Averages	39.75	31.76	21.37	13.32	8.00
PDAST (%)	0%	20.10%	46.24%	66.49%	79.87%

more the search time consumed [31]. However, the AST of the PGWO can be reduced by running it on multicore or parallel processors.

The numeric data in Table 4 can be converted to the equivalent AST (EAST) with respect to the GWO by using the formulation as expressed in (32), where N is number of GWOs used in the PGWO#N. The percent decrease of AST (PDAST) of the equivalent PGWO with respect to the GWO can be calculated by using the formulation as stated in (33) for comparison as summarized in Table 5. Regarding to results in Table 5, it can be noticed that the equivalent PGWO#2, PGWO#4, PGWO#8, and PGWO#16 averagely consume the AST less than the GWO by 20.10%, 46.24%, 66.49%, and 79.87%, respectively.

$$\mathrm{EAST}_{\mathrm{PGWO}\#N} = \frac{\mathrm{AST}_{\mathrm{PGWO}\#N}}{N} \tag{32}$$

$$\mathrm{PDAST}_{\mathrm{PGWO}} = 100 \times \left(\frac{\mathrm{AST}_{\mathrm{GWO}} - \mathrm{EAST}_{\mathrm{PGWO}}}{\mathrm{AST}_{\mathrm{GWO}}} \right) \tag{33}$$

5. **Conclusions.** The application of the PGWO to solving the MDVRP problems has been proposed in this paper. As one of the newest versions of the original GWO, the PGWO algorithm was developed for running on a single-CPU platform by using GWOs as the search units. In this work, the PGWO has been applied to solving ten selected real-world MDVRP consisting of approximately 100-300 customer locations. Results obtained by the PGWO containing 2, 4, 8, and 16 GWOs (PGWO#2, PGWO#4, PGWO#8, and PGWO#16) have been compared with those obtained by the GWO, CS, PSO, and GA. From experimental results, the GWO can provide the optimal tours with shorter total distance than the CS, PSO, and GA, respectively. Moreover, the GWO has consumed the AST less than the CS, PSO, and GA, respectively. As results between the GWO and PGWO, the PGWO#16 can provide the optimal tours with shorter total distance than the PGWO#8, PGWO#4, PGWO#2, and GWO, respectively. This can be concluded that the PGWO can give the optimal solutions of all ten selected MDVRP with shorter total distance than the GWO, CS, PSO, and GA, respectively. However, by running on a single-CPU platform, the AST consumed by the PGWO is greater than that consumed by the GWO. This AST of the PGWO can be reduced by running it on multicore or parallel processors. Once comparing the EAST of the PGWO with respect to the GWO, the equivalent PGWO#2, PGWO#4, PGWO#8, and PGWO#16 have averagely consumed the AST less than the GWO by 20.10%, 46.24%, 66.49%, and 79.87%, respectively. For the future research, the PGWO will be further applied to solving the more specific real-world MDVRP including the capacitated MDVRP (CMDVRP), the MDVRP with time windows (MDVRPTW), and the non-heterogeneous fleet MDVRP (NHMDVRP) for minimizing total distance and balancing the number of vehicles, number of customer locations, traveling time of each vehicle, and the traveling distance of each vehicle in the fleet on each depot.

Acknowledgment. This paper was funded by King Mongkut's University of Technology North Bangkok with contract no. KMUTNB-68-BASIC-60.

REFERENCES

[1] R. W. Bent and P. Van Hentenryck, Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research*, vol.52, no.2, pp.977-987, 2004.
[2] J. Mańdziuk and A. Żychowski, A memetic approach to vehicle routing problem with dynamic requests, *Applied Soft Computing*, vol.48, pp.522-534, 2016.
[3] A. Ahkamiraad and Y. Wang, Capacitated and multiple cross-docked vehicle routing problem with pickup, delivery, and time windows, *Computers & Industrial Engineering*, vol.119, pp.76-84, 2018.
[4] G. B. Dantzig and J. H. Ramser, The truck dispatching problem, *Management Science*, vol.6, no.1, pp.80-91, 1959.
[5] T. Tlili, S. Krichen, G. Drira and S. Faiz, On solving the multi-depot vehicle routing problem, *Proc. of the 3rd International Conference on Advanced Computing, Networking and Informatics*, pp.103-108, 2016.
[6] G. Laporte, Y. Nobert and D. Arpin, Optimal solutions to capacitated multi-depot vehicle routing problems, *Congressus Numerantium*, vol.44, pp.283-292, 1984.
[7] C. Contardo and R. Martinelli, A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints, *Discrete Optimization*, vol.12, pp.129-146, 2014.
[8] J. F. Cordeau, M. Gendreau and G. Laporte, A tabu search heuristic for periodic and multi-depot vehicle routing problems, *Networks*, vol.30, pp.105-119, 1997.
[9] J. W. Escobar, R. Linfati, P. Toth and M. Baldoquin, A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem, *Journal of Heuristics*, vol.20, no.5, pp.1-27, 2014.
[10] L. Shen, F. Tao and S. Wang, Multi-depot open vehicle routing problem with time windows based on carbon trading, *International Journal of Environmental Research and Public Health*, vol.15, no.9, 2018.
[11] K. Ghoseiri and S. Ghannadpour, A hybrid genetic algorithm for multi-depot homogenous locomotive assignment with time windows, *Applied Soft Computing*, vol.10, pp.53-65, 2010.

[12] M. Mirabi, N. Shokri and A. Sadeghieh, Modeling and solving the multi-depot vehicle routing problem with time window by considering the flexible end depot in each route, *International Journal of Supply and Operations Management*, vol.3, no.3, pp.1373-1390, 2016.

[13] M. Polacek, R. Hartl, K. Doerner and M. Reimann, A variable neighborhood search for the multi-depot vehicle routing problem with time windows, *Journal of Heuristics*, vol.10, pp.613-627, 2004.

[14] P. Sombuntham and V. Kachitvichyanukul, Multi-depot vehicle routing problem with pickup and delivery requests, *IAENG Transactions on Engineering Technologies*, vol.5, pp.71-85, 2010.

[15] P. Sombuntham and V. Kachitvichayanukul, A particle swarm optimization algorithm for multi-depot vehicle routing problem with pickup and delivery requests, *Proc. of the International Multi-Conference of Engineers and Computer Scientists (IMECS 2010)*, Hong Kong, pp.1998-2003, 2010.

[16] Y.-J. Gong, J. Zhang, O. Liu, R.-Z. Huang, H. S.-H. Chung and Y.-H. Shi, Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.42, no.2, pp.254-267, 2012.

[17] S. Zou, J. Li and X. Li, A hybrid particle swarm optimization algorithm for multi-objective pickup and delivery problem with time windows, *Journal of Computers*, vol.8, no.10, pp.2583-2589, 2013.

[18] M. M. Tavakoli and A. Sami, Particle swarm optimization in solving capacitated vehicle routing problem, *Bulletin of Electrical Engineering and Informatics*, vol.2, no.4, pp.252-257, 2013.

[19] B. Yu, Z. Z. Yang and Y. X. Xie, A parallel improved ant colony optimization for multi-depot vehicle routing problem, *Journal of the Operational Research Society*, vol.62, pp.183-188, 2011.

[20] T. Demirel and S. Yilmaz, A new solution approach to multi-depot vehicle routing problem with ant colony optimization, *Journal of Multiple-Valued Logic and Soft Computing*, vol.18, pp.421-439, 2012.

[21] Y. Kao, M.-H. Chen and Y.-T. Huang, A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems, *Mathematical Problems in Engineering*, pp.1-17, 2012.

[22] K. V. Narasimha, E. Kivelevitch, B. Sharma and M. Kumar, An ant colony optimization technique for solving min-max multi-depot vehicle routing problem, *Swarm and Evolutionary Computation*, vol.13, pp.63-73, 2013.

[23] N. E. Toklu, L. M. Gambardella and R. Montemanni, A multiple ant colony system for a vehicle routing problem with time windows and uncertain travel times, *Journal of Traffic and Logistics Engineering*, vol.2, no.1, pp.52-58, 2014.

[24] H. Zheng, Y.-Q. Zhou and Q. Luo, A hybrid cuckoo search algorithm-GRASP for vehicle routing problem, *Journal of Convergence Information Technology*, vol.8, no.3, pp.821-828, 2013.

[25] M. Alssager and Z. A. Othman, Cuckoo search algorithm for capacitated vehicle routing problem, *Journal of Theoretical and Applied Information Technology*, vol.88, no.1, pp.11-19, 2016.

[26] A. Goli, A. Aazami and A. Jabbarzadeh, Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions, *International Journal of Artificial Intelligence*, vol.16, no.1, pp.88-112, 2018.

[27] S. George and S. Binu, Vehicle route optimisation using artificial bees colony algorithm and cuckoo search algorithm – A comparative study, *International Journal of Applied Engineering Research*, vol.13, no.2, pp.953-959, 2018.

[28] A. A. Juan, I. Pascual, D. Guimarans and B. Barrios, Combining biased randomization with iterated local search for solving the multi-depot vehicle routing problem, *International Transactions in Operational Research*, vol.22, pp.647-667, 2014.

[29] Y. Shi, L. Lv, F. Hu and Q. Han, A heuristic solution method for multi-depot vehicle routing-based waste collection problems, *Applied Science*, vol 10, no.7, 2403, 2020.

[30] S. Suwannarongsri, Optimal solving multi-depot vehicle routing problem by modern metaheuristic algorithm, *International Journal of Innovative Computing, Information and Control*, vol.19, no.6, pp.1753-1767, 2023.

[31] S. Suwannarongsri, Development of parallel grey wolf optimizer for global optimization, *International Journal of Innovative Computing, Information and Control*, vol.21, no.5, pp.1255-1276, 2025.

[32] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, vol.69, pp.46-61, 2014.

[33] Y. Liu, A. As'arry, M. K. Hassan, A. A. Hairuddin and H. Mohamad, Review of the grey wolf optimization algorithm: Variants and applications, *Neural Computing and Applications*, vol.36, pp.2713-2735, 2024.

[34] W. Xiao, H. Deng, Y. Sheng and L. Hu, Factored grey wolf optimizer with application to resource-constrained project scheduling, *International Journal of Innovative Computing, Information and Control*, vol.14, no.3, pp.881-897, 2018.
[35] K. Jiang, H. Ni, R. Han and X. Wang, An improved multi-objective grey wolf optimizer for dependent task scheduling in edge computing, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2289-2304, 2019.
[36] G. Nagy and S. Salhi, Heuristics algorithm for single and multiple depot vehicle routing problem with pickups and deliveries, *European Journal of Operational Research*, vol.162, pp.126-141, 2005.
[37] S. Salhi and M. Sari, Models for the multi-depot vehicle fleet mix problem, *European Journal of Operational Research*, vol.103, pp.95-112, 1997.
[38] TSPLIB95, *Symmetric Traveling Salesman Problem*, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/, 2022.
[39] R. E. Bland and D. F. Shallcross, Large traveling salesman problems arising from experiments in X-ray crystallography: A preliminary report on computation, *Operations Research Letters*, vol.8, pp.125-128, 1989.
[40] M. Grötschel and O. Holland, Solution of large-scale symmetric travelling salesman problems, *Mathematical Programming*, vol.51, pp.141-202, 1991.
[41] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.
[42] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison Wesley, 1989.
[43] S. Katoch, S. S. Chauhan and V. Kumar, A review on genetic algorithm: Past, present, and future, *Multimedia Tools and Applications*, vol.80, pp.8091-8126, 2021.
[44] F. M. Isa, W. N. M. Ariffin, M. S. Jusoh and E. P. Putri, A review of genetic algorithm: Operations and applications, *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol.40, no.1, pp.1-34, 2024.
[45] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proceedings of the IEEE Conference on Neural Networks*, vol.4, pp.1942-1948, 1995.
[46] R. C. Eberhart and Y. Shi, Particle swarm optimization: Developments, applications and resources, *Proceedings of the 2001 Congress on Evolutionary Computation*, vol.1, pp.81-86, 2001.
[47] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh and S. Mirjalili, Particle swarm optimization: A comprehensive survey, *IEEE Access*, vol.10, pp.10031-10061, 2022.
[48] A. G. Gad, Particle swarm optimization algorithm and its applications: A systematic review, *Archives of Computational Methods in Engineering*, vol.29, pp.2531-2561, 2022.
[49] X.-S. Yang and S. Deb, Cuckoo search via Lévy flights, *Proc. of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp.210-214, 2009.
[50] X.-S. Yang and S. Deb, Engineering optimisation by cuckoo search, *Journal of Mathematical Modelling and Numerical Optimisation*, vol.1, no.4, pp.330-343, 2010.
[51] M. Guerrero-Luis, F. Valdez and O. Castillo, A review on the cuckoo search algorithm, *Studies in Computational Intelligence*, vol.940, pp.113-124, 2021.
[52] A. S. Joshi, O. Kulkarni, G. M. Kakandikar and V. M. Nandedkar, Cuckoo search optimization – A review, *Materials Today: Proceedings*, vol.4, no.8, pp.7262-7269, 2017.

Author Biography

Supaporn Suwannarongsri received the B.Eng. degree in Industrial Engineering from Thonburi University (TRU), Thailand, 2004; the M.Eng. degree in Industrial Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, 2008; the Ph.D. degree in Sustainable Energy and Environment Technology and Management from Rajamangala University of Technology Rattanakosin (RMUTR), Thailand, 2014.

Dr. Suwannarongsri is currently an associate professor at the Department of Materials Handling and Logistics Engineering, Faculty of Engineering, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. Her research interests include logistics and materials handling, operation research, production planning and design, and applications of metaheuristic algorithms to various real-world industrial engineering problems. She has published over 100 papers in journals and conferences, nationally and internationally.