Journal of Network Intelligence ©2024 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **9**, Number **2**, May **2024**

Logistics Job Intelligent Scheduling Model Based on Discrete Grey Wolf Optimization Algorithm

Tian-Juan Gao, Yuan Zhou*
Department of Logistics Management
Hebei Jiaotong Vocational and Technical College, Shijiazhuang 050035, China
gaotianjuan@126.com, zhouyuanhbjt@163.com

Thomas Masurat
Technische Hochschule Wildau, Brandenburg 14467, Germany
daisyzhou.fly@gmail.com

*Corresponding author: Yuan Zhou
Received October 25, 2023, revised December 27, 2023, accepted February 4, 2024.

Abstract. *Logistics job scheduling is playing an increasingly important role in various aspects of social development and regional infrastructure. In recent years, scholars have conducted extensive research on logistics job scheduling. In real-world logistics job scheduling problems, there are numerous optimization objectives, such as transportation costs, time constraints, and traffic pressures. This paper begins by establishing a fundamental mathematical model and formulating a multi-objective optimization function for intelligent logistics job scheduling, taking into account the constraints involved. To address the combinatorial optimization challenges of logistics scheduling, the paper introduces the Grey Wolf Optimization (GWO) algorithm. However, because the solution space in logistics job scheduling optimization is considered a hypercube, and the GWO algorithm is designed for continuous optimization problems, the paper discretizes the GWO algorithm. It transforms the real-number positions of grey wolf individuals in continuous space into integer vectors and designs a repair and optimization algorithm based on a greedy strategy. The paper ultimately designed and implemented an improved Discrete Grey Wolf Optimization algorithm(DGWO). In order to verify the applicability of the improved DGWO algorithm in solving logistics job scheduling problems, this paper utilizes 10 data instances from reference 11 and compares the DGWO algorithm with GA_El, MPCEA, and TTLS algorithms proposed in references 4, 11, and 25, respectively. The results show that the improved DGWO algorithm in this paper exhibits good performance in terms of solving optimization problems, convergence, and robustness. It can be effectively used to address real-world logistics job scheduling problems.*
Keywords: logistics job scheduling; combinatorial optimization; Discrete Gray wolf Optimization algorithm; encoding transformation; greedy strategy

1. **Introduction.** Modern logistics has become one of the core components reflecting the economic vitality, overall competitiveness, and urban image of a region. Logistics job scheduling can significantly improve the efficiency of logistics processes such as transportation and warehousing. Through intelligent scheduling systems, resources can be allocated efficiently, reducing wait times and minimizing empty vehicle loads, thereby lowering costs [1,2]. It plays an increasingly crucial role in urban development and daily life.

In recent years, numerous scholars have conducted extensive research on logistics job scheduling. Delaram and Valilai [3] proposed a mathematical model for task scheduling in cloud manufacturing systems to minimize logistics costs for intercity transportation of semi-finished products. Elgendy et al. [4] addressed logistics scheduling problems with an improved genetic algorithm optimizing for maximum completion time and transportation costs. Taniguchi and Shimamoto [5] explored dynamic vehicle routing and scheduling problems with variable travel times. Noroozi et al. [6] aimed to enhance overall profit by considering order revenue and logistics scheduling costs, solving scheduling models using an adaptive genetic algorithm. However, in practical vehicle routing problems, there are numerous optimization objectives, such as transportation costs, time constraints, and traffic pressures, which involve various influencing factors and related information. Therefore, the study of vehicle routing and scheduling problems with multiple objectives and constraints is of significant importance.

This paper, starting from the perspective of modern logistics job scheduling, provides a foundational information model for logistics scheduling problems and a mathematical model for comprehensive evaluation. It also introduces an improved Discrete Grey Wolf Optimization algorithm. The effectiveness of this algorithm in logistics scheduling is ultimately validated through practical examples.

2. **Definition and mathematical model of Logistics Job scheduling problem.** The logistics job scheduling problem can be described as follows: Given the locations of customers and dispatch starting points, and under the constraints of the vehicle's maximum loading capacity and the latest transportation time, the goal is to minimize transportation costs. This problem involves transporting n tasks using m vehicles to their respective destinations [7,8].

The cargo matrix provides detailed information about the goods at the logistics center, including the item name, item number, weight, packaging volume, inventory identification, and so on. Its mathematical representation is shown in Equation (1):

$$[G] = \begin{bmatrix} G_{11} & \cdots & G_{1j} & \cdots & G_{1w} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ G_{i1} & \cdots & G_{ij} & \cdots & G_{iw} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ G_{n1} & \cdots & G_{ni} & \cdots & G_{nw} \end{bmatrix} \tag{1}$$

where G_{ji} represents the i-th description information of cargo j, where $j = 1, 2, \ldots, n$, and n is the cargo space number; $i = 1, 2, \ldots, w$. For example, $i = 1$ represents cargo name; $i = 2$ represents cargo code; $i = 3$ represents cargo weight, and so on.

The vehicle information matrix describes the basic information of the scheduling vehicles, including vehicle name, code number, carrying capacity, transportation cost, and so on. Equation (2) provides a mathematical description of vehicle information.

$$[V] = \begin{bmatrix} V_{11} & \cdots & V_{1h} & \cdots & V_{1v} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ V_{k1} & \cdots & V_{kh} & \cdots & V_{kv} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ V_{m1} & \cdots & V_{mh} & \cdots & V_{mv} \end{bmatrix} \tag{2}$$

where V_{kh} represents the h-th character information of vehicle k, where $k = 1, 2, \ldots, m$, and m is the total number of vehicles that can be scheduled from the center. $h = 1, 2, \ldots, v$, where $h = 1$ represents vehicle name, $h = 2$ is vehicle number, and $h = 3$ represents the vehicle's carrying capacity, and so on.

The transportation route and condition information matrix describe the distance between two traffic points and road condition factors [9, 10, 11]. Equation (3) provides its mathematical description.

$$[D] = \begin{bmatrix} D_{0-0} & D_{0-1}/R_{0-1} & \cdots & D_{0-(s-1)}/R_{0-(s-1)} & D_{0-s}/R_{0-s} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ D_{s-0}/R_{s-0} & D_{s-1}/R_{s-1} & \cdots & D_{s-(s-1)}/R_{s-(s-1)} & D_{s-s} \end{bmatrix} \quad (3)$$

In some cases, the traffic routes are one-way and not reversible, so $D_{g \to p} \neq D_{p \to g}$, which means that the mutual delivery distances between two logistics points are different.

Logistics job scheduling aims to meet customer demands while maximizing cost savings under the constraint of maintaining smooth traffic [12,13]. The main constraints in optimizing vehicle routes include customer wait times, vehicle quantity limits, and transportation costs, among others. The mathematical model is represented by schedulable vehicles (S), transportation routes (D), and transportation costs (C), as shown in Equations (4), (5), and (6).

$$S = \sum_{k=1}^{m} S(V_k) \quad (4) \qquad (4)$$

$$D = \sum_{k=1}^{m} D_{\max}(V_k) \quad (5) \qquad (5)$$

$$C = \sum_{k=1}^{m} \left[C_1(V_k) + \sum_{j=1}^{n_k} C_2(V_k) D_j R_j + C_3(V_k) T_k \right] \quad (6) \qquad (6)$$

where $C_1(V_k)$, $C_2(V_k)$ and $C_3(V_k)$ represent the startup cost, transportation cost per unit, and transportation service cost of vehicle V_k, respectively. D_j is the transportation distance for cargo j, corresponding to the transportation route information matrix D_{g-p}, R_j is the road condition coefficient, and T_k is the operating time for vehicle V_k.

$$T_a^f(j) \leq T_a^d(j) \quad (7) \qquad (7)$$

$$W(V_k) \leq V_k^3 \quad (8) \qquad (8)$$

Where $T_a^f(j)$ is the estimated arrival time for cargo j, $T_a^d(j)$ is the actual arrival time required for cargo j, V_k is the estimated carrying capacity of the vehicle, and $W(V_k)_f$ is the actual load capacity of the vehicle.

3. **Basic grey wolf optimization algorithm.** The Grey Wolf Optimization (GWO) algorithm is a nature-inspired swarm intelligence optimization algorithm, drawing inspiration from the social structure and hunting behavior of grey wolves in the wild. This algorithm was first introduced by Seyedali in 2014. GWO has gained recognition due to its simplicity, ease of parameter tuning, and straightforward implementation. It has been successfully applied in various domains, including 0-1 knapsack problems, numerical optimization, multi-layer perceptron training, and engineering design, among other practical problems [14,15,16,17].

GWO models this hierarchy with four main levels: α, β, δ, and ω. α is at the top and symbolizes the leader of the wolf pack. Beta is the second-in-command, serving under Alpha and assisting in decision-making. When Alpha loses its dominance, Beta becomes the best candidate for the new Alpha. Delta is at the bottom of the hierarchy, following the commands of both Alpha and Beta. Omega is at the lowest level, representing the majority of the wolf pack, responsible for executing the decisions made by the higher-ranked wolves and maintaining internal relationships within the pack.

During the optimization process of GWO, the best solution is considered as Alpha (α), the second-best and third-best solutions are Beta (β) and Delta (δ), and the remaining solutions are Omega (ω). The iterative process of GWO relies on Alpha, Beta, and Delta to guide Omega. The global optimization search is achieved through a task distribution process involving encircling, hunting, and attacking phases assigned to the wolves at each hierarchical level [18, 19, 20].

3.1. **Tracking and approaching.** During the encirclement of the prey, the calculation of the gray wolf's position is determined by the prey's location, coefficient vector **A**, and the distance between the gray wolf and the prey, as shown in Equations (9), (10), and (11).

$$\vec{X}(t+1) = \vec{X}_p(t+1) - \vec{A} \cdot \vec{D} \tag{9}$$

$$\vec{D} = |2\vec{r_2} \cdot \vec{X}_p(t) - \vec{X}(t)| \tag{10}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \tag{11}$$

where X and X_p represent the positions of a grey wolf and its prey, t is the iteration number, D represents the distance between the prey and the grey wolf, vector A represents the simulated attack behavior of the grey wolf towards the prey, and a is the convergence factor.

3.2. **Pursuing and encircling.** GWO designates the best wolf as alpha, the second best as beta, the third best as delta, and the rest of the wolves are categorized as omega. This classification is based on the fact that it's easier to understand the potential optimal solution locations through Alpha, Beta, and Delta [21,22]. Therefore, the positions of the best three solutions obtained before the current iteration number t are recorded and retained, and all the positions of the individual wolves are updated using Equation (12)-Equation (15) as follows:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)\right| \tag{13}$$

$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)\right| \tag{14}$$

$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)\right| \tag{15}$$

The update of the grey wolf's positions can be represented as shown in Figure 1.

3.3. **Attacking.** The hunting phase of the wolf pack involves capturing prey, which corresponds to obtaining the optimal solution. The main control factor during this process is the linear decrease of the parameter 'a' in the equation [23]. By controlling the values of 'a' and 'A', the wolf pack can balance global and local search during the search for the optimal solution, effectively guiding the pack towards the best solution. The update equation for 'a' in the GWO algorithm is as follows:

$$a = 2 - 2t/\text{iterMax} \tag{16}$$

where t and iterMax are the number of iterations and the maximum number of iterations.

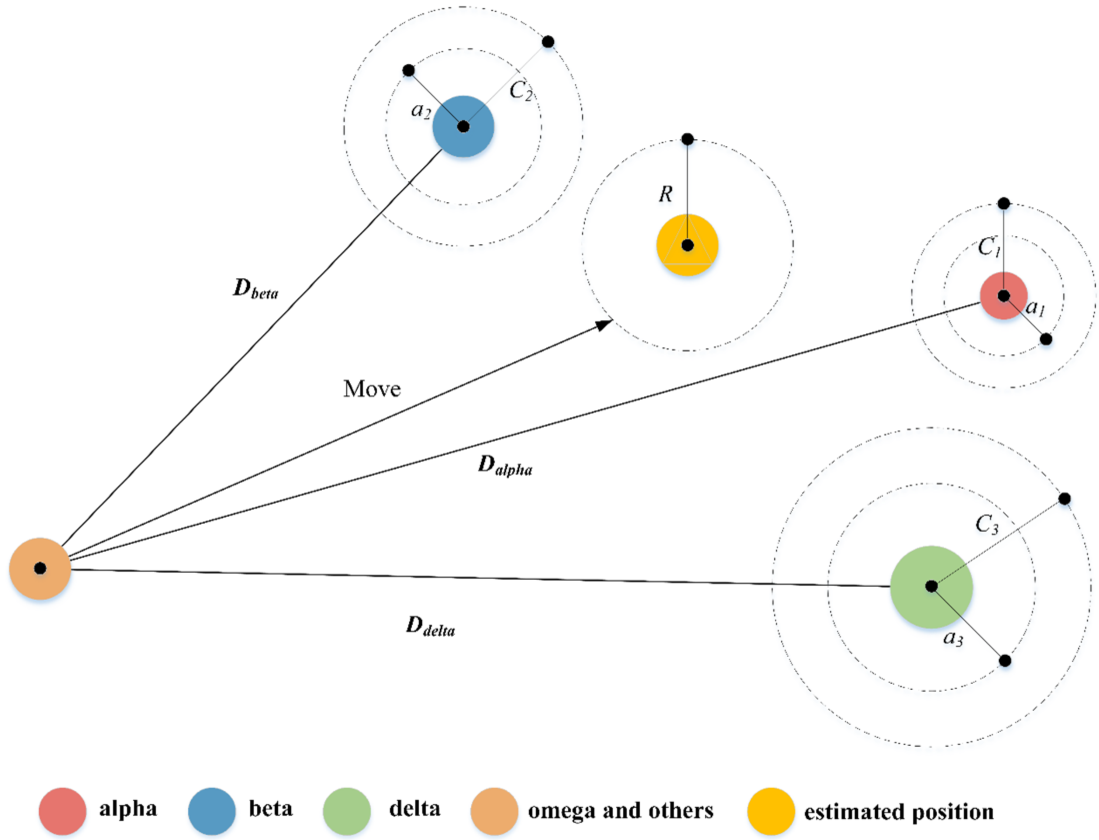

Figure 1. Grey wolf location update

4. **Discrete Grey Wolf Optimization Algorithm.** In logistics job scheduling optimization problems, the solution space is considered as a hypercube [24,25,26], while the GWO algorithm is specifically designed for continuous optimization problems. In the original GWO, the positions of the grey wolves are represented as real-valued vectors, and they are continuously updated based on the location and distance to the prey [27]. However, this mechanism is no longer suitable for the discrete domain. Therefore, we map the distance moved by individuals in the continuous space to the probability of changing their positions in the discrete space, thereby discretizing the GWO algorithm.

4.1. **Encoding Transformation Method.** In addressing numerical optimization problems, the GWO algorithm seeks the best solution by searching for the positions of grey wolf individuals in continuous space. However, in the discrete space, the way the positions of grey wolf individuals are updated differs [28]. To tackle this issue, it is necessary to map the coordinates of positions in continuous space to the discrete domain using a transfer function. The encoding transformation method typically utilizes a transformation function to map the position coordinates of each individual in the search space, in each dimension, to real values within the (0,1) range [29].

Let vector $\vec{X}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$ represent the position coordinates of the i-th grey wolf individual in an n-dimensional search space, and let $\vec{Y}_i = [y_{i1}, y_{i2}, \ldots, y_{in}]$ represent the binary vector obtained after encoding transformation of $\vec{X}_i$. Drawing from the approach proposed by Yue et al. [30], this paper introduces an encoding transformation method to convert the real-number position vector of grey wolf individuals in continuous space into an integer vector using Equation (17).

$$y_{ij} = \left\lfloor b_j * \frac{x_{ij} - lb}{ub - lb} + 0.5 \right\rfloor, \quad j = 1, 2, \ldots, n \tag{17}$$

where x_{ij} represents the coordinate of individual i in the j-th dimension, y_{ij} represents the value of the potential solution Y_i in the j-th dimension, b_j represents the total quantity of the j-th type of item.

4.2. **Initial population and individual evaluation.** Each grey wolf individual in the pack randomly generates an initial position coordinate vector X_i, which lies within the given search space between l_b and u_b. Through a transformation function, X_i is converted into the corresponding binary coordinate vector Y_i. Then, based on the objective function of the combinatorial optimization problem at hand, Y_i is used to compute the fitness value, fitness(Y_i), of the i-th grey wolf individual in the pack, thus evaluating all the grey wolves in the entire pack.

4.3. **Select and update.** By employing a greedy selection strategy, the algorithm reselects the top three individuals based on fitness ranking after each iteration and designates them as alpha, beta, and delta. For the logistics job scheduling problem, this paper determines their positions as follows: Y_{alpha} is the Y_i that maximizes $f(Y_i)$; Y_{beta} is the Y_{beta}, except for alpha, that maximizes $f(Y_i)$; and Y_{delta} is the Y_{delta}, except for alpha and beta, that maximizes $f(Y_i)$. Subsequently, the positions of X_{alpha}, X_{beta}, and X_{delta} are used to guide the entire pack for updates in the next generation.

4.4. **Greedy Strategy-Based Repair and Optimization Algorithm.** When the DGWO algorithm is used to solve logistics scheduling problems, it may produce infeasible solutions. Effectively handling these infeasible solutions is crucial. Common methods for dealing with infeasible solutions include penalty function methods, repair methods, and repair and optimization methods [17]. The repair and optimization method employed in this paper is as shown in Algorithm 1.

4.5. **Implementation of the DGWO Algorithm.** In the initialization phase of the DGWO algorithm, each individual's components in each dimension are randomly selected within the interval $[l_b, u_b]$. When solving the logistics job scheduling problem, the algorithm designed in this paper transforms the values of X into the corresponding potential solutions Y based on Equation (25). Then, individuals are repaired and optimized using the algorithm described in Section 3.4, and their fitness values are calculated. Finally, the top three individuals with the highest fitness values are selected to guide the evolution of the remaining individuals.

Let $L[1 \ldots n] \leftarrow \text{sort}((p_i/\omega_i | 1 \leq i \leq n))$ represent the array where n logistics jobs are sorted in p_i/ω_i descending order, and the pseudocode for the DGWO algorithm is described in Algorithm 2.

Algorithm 1 Repair and Optimization Algorithm

Input: potential solution $Y = [y_{i1}, y_{i2}, \ldots, y_{in}]$
Output: feasible solution $Y = [y_{i1}, y_{i2}, \ldots, y_{in}]$ and $f(Y)$
$R \leftarrow \sum_{j=1}^{n} w_j y_j$; $j \leftarrow n$
while $R > C$ **do**
  if $y_{H[j]} > 0$ **then**
    $y_{H[j]} \leftarrow y_{H[j]} - 1$; $R \leftarrow R - w_{H[j]}$
  else
    $j \leftarrow j - 1$
  end if
end while
$j \leftarrow 1$
while $j \leq n$ **do**
  if $y_{H[j+1]} < b$ **and** $R + w_{H[j+1]} \leq C$ **then**
    $y_{H[j]} \leftarrow y_{H[j]} + 1$; $R \leftarrow R + w_{H[j]}$
  else
    $j \leftarrow j + 1$
  end if
end while
$f(Y) \leftarrow \sum_{j=1}^{n} p_j y_j$
return $(Y, f(Y))$

Algorithm 2 DGWO

Input: size, mit, and P_c
Output: Y_a and $f(Y_a)$
$H[1 \ldots n] \leftarrow \text{QuickSort}\left(\left\{\frac{p_i}{\omega_i} | p_i \in P, \omega_i \in W, 1 \leq i \leq n\right\}\right)$
Generate initial population $X_i (i = 1 \ldots n)$ randomly and Y_i;
for $i \leftarrow 1$ to N **do** $(Y_i, f(Y_i)) \leftarrow \text{GROA}(Y_i, H[1 \ldots n])$;
end for
Find the α, β, δ positions based on fitness;
$t \leftarrow 0$;
while $t \leq$ MaxIter **do**
  $a \leftarrow 2 - 2 \times \frac{t}{\text{MaxIter}}$;
  for $i \leftarrow 1$ to N **do**
    Update position vector X_i by Equation (8);
    Update position vector Y_i by Equation (16);
    $(Y_i, f(Y_i)) \leftarrow \text{GROA}(Y_i, H[1 \ldots n])$;
    Crossover operate by probability P_i;
  end for
  Update the α, β, δ positions based on fitness;
  $t \leftarrow t + 1$;
end while
return $(Y_a, f(Y_a))$

5. **Experimental Results.** In this section, tests were conducted using the data from reference 11. The programming language used was Python. To evaluate the effectiveness of the algorithm proposed in this paper, the experimental results of the DGWO algorithm were compared with those of TTLS [25], MPCEA [11], and GA_El [4]. The parameter

settings for each algorithm can be found in Table 1, the definitions of the parameters for each algorithm are as given.

Table 1. Parameters of algorithms.

	Popsize	**Maxiter**	P_c	P_v	l_b	u_b
DGWO	30	1000	0.8	-	0	3
TTLS	50	2000	0.8	-	0	bj
MPCEA	50	2000	0.9	-	0	5
GA_ El	30	1000	0.8	0.001	0	1

To test the results of the four algorithms in logistics scheduling problems, this paper has compiled and compared the results of four algorithms after running 50 times on 10 different test instances. The results include the best solution (BEST), average solution (MEAN), standard deviation (STD), average number of iterations (MIT), average computation time (ACT), and the probability of obtaining the best solution (SR). The experimental results are presented in Table 2 and Table 3.

Table 2. Parameter Definition.

Parameters	**Numerical value**
popsize	the population size of the algorithm
maxIter	maximum number of iterations of the algorithm
P_c	cross probability of the algorithm
P_m	mutation probability of the algorithm
l_b	the lower bound of each one-dimensional component in an individual variabl
u_b	the upper bound of each one-dimensional component in an individual variabl

Table 3. Experimental Results of DGWO, TTLS, MPCEA, and GA_ El.

		pb1	**pb2**	**pb3**	**hp1**	**hp2**	**hp3**	**sen1**	**sen2**	**weing1**	**weing2**
DGWO	BEST	1118	2053	875	3118	2553	419	618	839	1845	882
	MEAN	1005	2037	859	2889	2331	401	597	815	1609	856
	MIT	145	225	327	240	164	254	153	308	150	276
	SR	1.00	1.00	1.00	0.94	1.00	1.00	1.00	1.00	1.00	1.00
TTLS	BEST	1095	1962	881	3002	2512	408	604	822	1830	884
	MEAN	997	1849	836	2734	2227	390	576	807	1554	851
	MIT	203	261	307	320	188	260	205	296	138	304
	SR	0.84	0.90	1.00	1.00	0.86	1.00	0.82	0.76	0.84	1.00
MPCEA	BEST	1083	2000	860	2957	2385	399	612	830	1685	869
	MEAN	1008	1776	834	2709	2064	376	583	804	1471	836
	MIT	197	245	358	312	182	269	145	302	146	283
	SR	1.00	0.90	1.00	0.86	1.00	0.84	0.92	0.86	1.00	0.96
GA_ El	BEST	1049	1900	843	2775	2362	405	598	798	1773	843
	MEAN	892	1553	790	2551	1895	392	564	770	1470	810
	MIT	210	281	410	325	223	297	196	406	135	384
	SR	0.62	1.00	0.70	0.86	0.66	0.90	0.68	0.74	0.82	0.78

It can be observed that DGWO achieved a success rate of 100% for all instances except ins4. TTLS and MPCEA reached a 100% success rate for four instances, while GA_El only achieved a 100% success rate for the ins2 instance. In terms of solving the optimization problem, for the pb3 and weing2 instances, the results obtained by DGWO and TTLS were similar. However, in the other eight instances, DGWO's optimal solutions were significantly better than the other three algorithms.

For the average solutions in each case, DGWO, TTLS, and MPCEA had relatively close values in the pb1 instance. In the weing2 instance, DGWO and TTLS algorithms had similar average solutions. In the remaining eight instances, DGWO's average solutions were superior to the other three algorithms.

To provide a more intuitive comparison of the convergence performance of the four algorithms, for six instances (pb1, hp1, sen1, sen2, weing1, and weing2), we present the average convergence curves for 50 experiments in Figure 2 to Figure 7.

In the pb1 and sen1 instances, DGWO outperforms the other three algorithms in terms of both finding the optimal solution and convergence. TTLS and MPCEA show similar performance and outperform the GA algorithm. In the hb1 instance, the four algorithms' effectiveness in finding the optimal solution is in the following order: DGWO, MPCEA, TTLS, GA_El. However, in terms of convergence, TTLS outperforms MPCEA. In the Sen2 instance, DGWO and TTLS exhibit similar performance, while MPCEA and GA algorithms perform similarly. For the Weing1 instance, after 200 iterations, all four algorithms converge and show similar convergence characteristics, but DGWO's optimal solution is somewhat superior to the other three algorithms. In the Weing2 instance, DGWO and TTLS algorithms perform similarly in both optimal solution search and convergence, while the GA algorithm outperforms MPCEA in finding the optimal solution.

Considering the results from all 10 instances and the convergence curves for the six instances (pb1, hp1, sen1, sen2, weing1, and weing2), it can be observed that DGWO exhibits superior overall performance compared to the other three algorithms, while TTLS and MPCEA demonstrate similar performance.

To further validate the robustness of the improved DGWO algorithm, the results of 50 experiments for each instance are compared with the average best solution for that instance, and the deviation results are calculated using Equation (18).

$$\text{dev} = \frac{|f_i - f_{\text{avg}}|}{f_i} \times 100\% \tag{18}$$

Where f_i represents the average best solution for the i-th instance, $i = 1, 2, \ldots, 10$. The deviation radar chart for the four algorithms across the ten instances is depicted in Figure 8.

From Figure 8, it is evident that the results for the ten instances corresponding to the four algorithms fluctuate within a certain range. DGWO exhibits the smallest fluctuation in its corresponding curve, while TTLS and MPCEA algorithms show comparable experimental deviation results. The deviation values for DGWO, TTLS, MPCEA, and GA_El algorithms fall within the ranges of 10%-18%, 15%-30%, 20%-35%, and 30%-45%, respectively. Therefore, in terms of robustness in problem-solving, the improved DGWO algorithm in this paper outperforms TTLS, MPCEA, and GA_El algorithms.

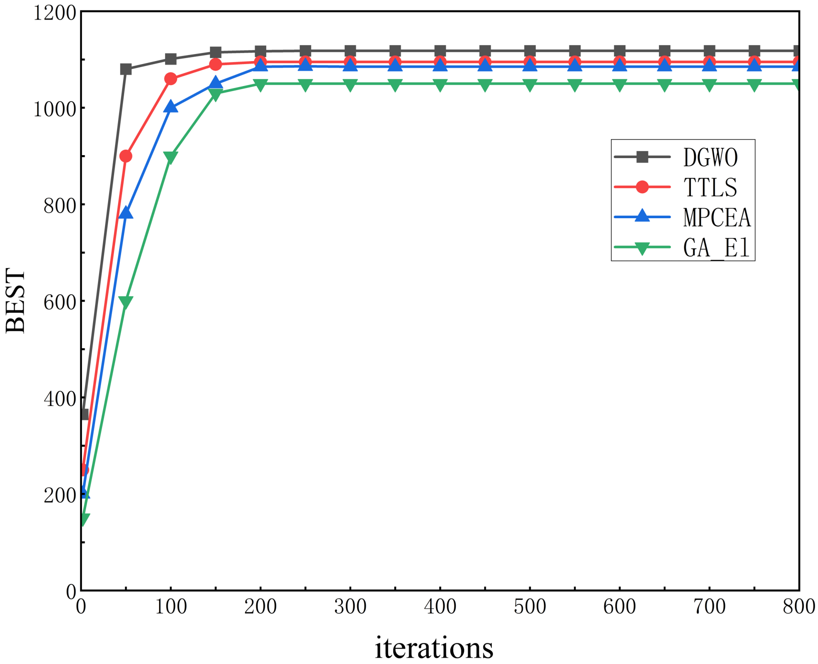

Figure 2. pb1 instance

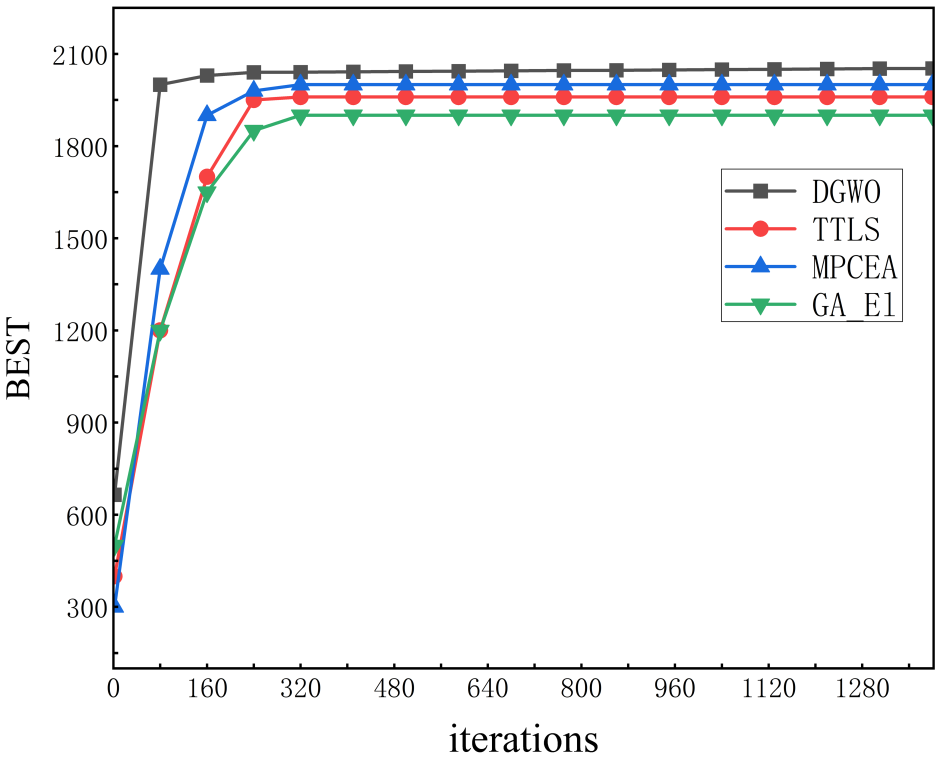

Figure 3. hb1 instance

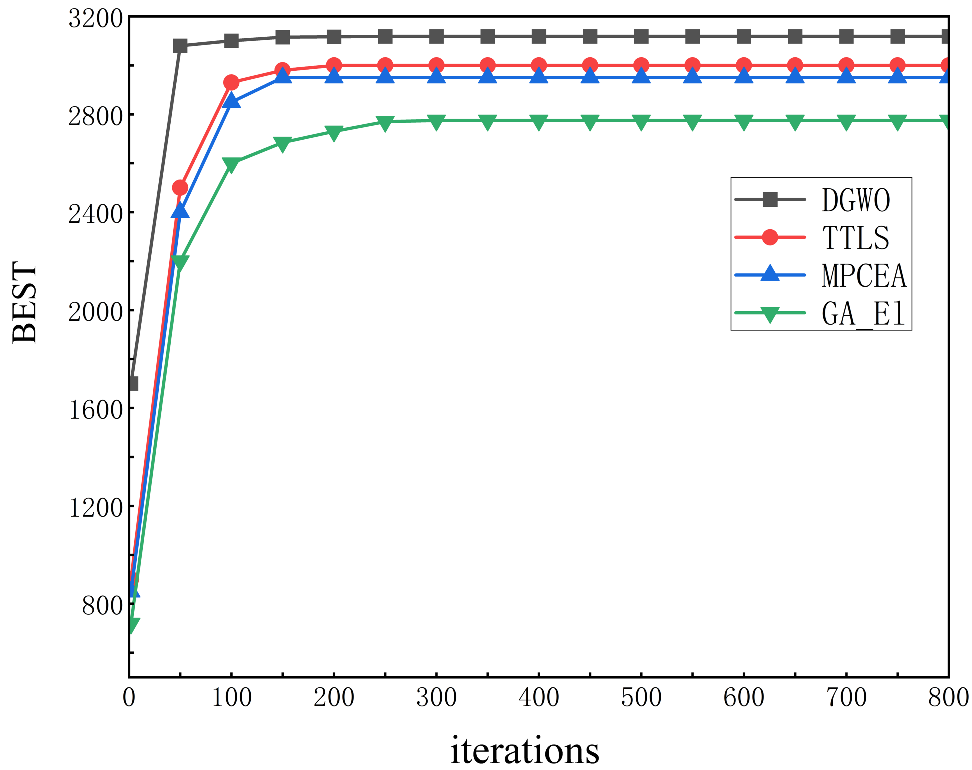

Figure 4. sen1 instance

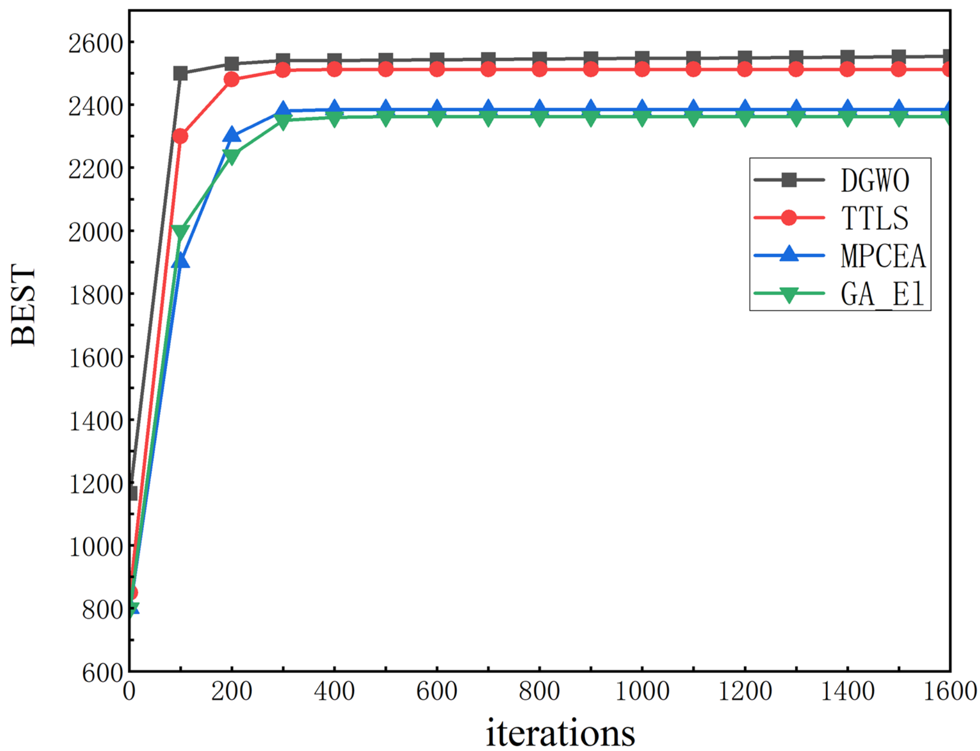

Figure 5. sen2 instance

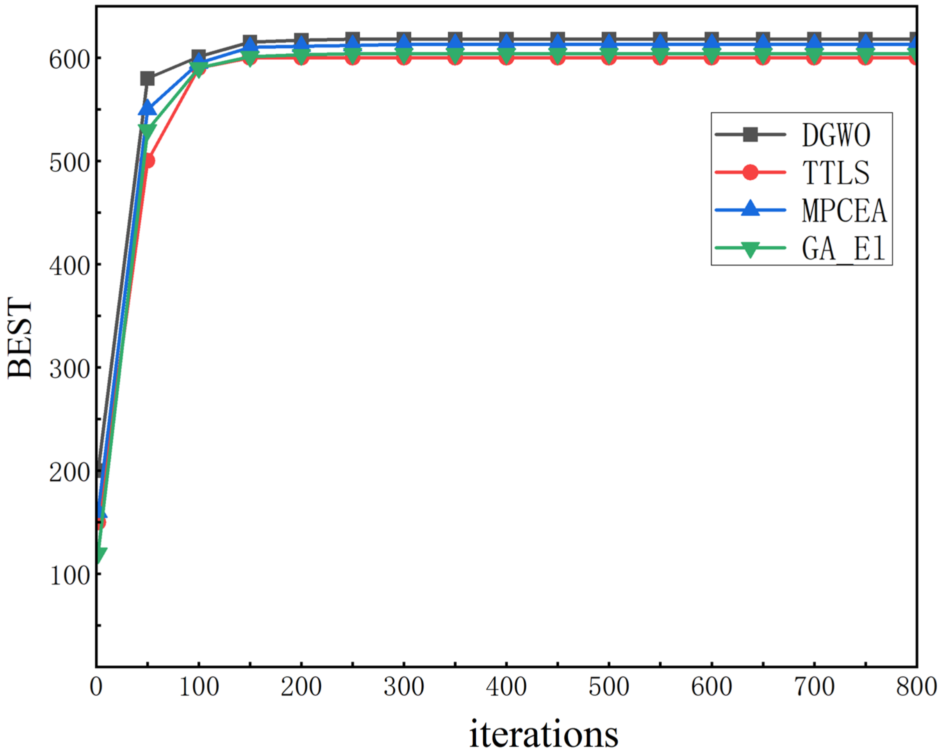

Figure 6. weing1 instance

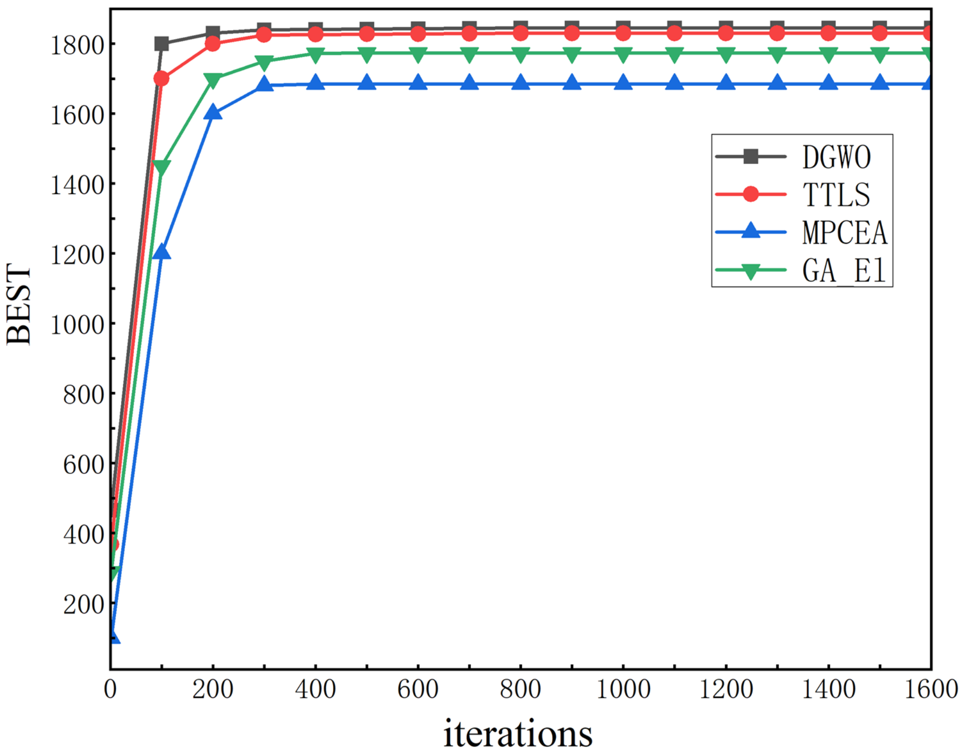

Figure 7. weing2 instance

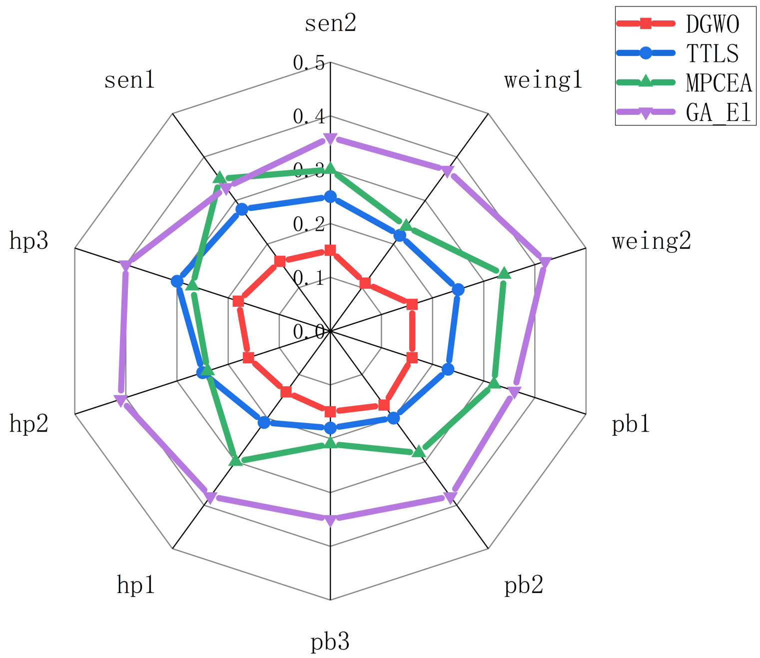

Figure 8. The deviation results for the ten instances

6. **Conclusions.** This paper addresses the intelligent scheduling problem in logistics operations and establishes a mathematical model for it. To tackle the issues of practical constraints and multi-objective optimization in logistics scheduling, this paper introduces the Grey Wolf Optimization (GWO) algorithm. Since the original GWO algorithm cannot directly address multi-objective logistics scheduling problems, this paper discretizes the GWO algorithm and proposes an improved Discrete Grey Wolf Optimization algorithm (DGWO). To test the results of the DGWO algorithm in logistics scheduling problems, the study uses 10 data instances from reference 11, and compares the DGWO algorithm with TTLS, MPCEA, and GA_El algorithms. The results indicate that the improved DGWO algorithm in this paper performs well in terms of solving the optimization problem, convergence, robustness, and other aspects. It can be effectively employed to address real-world logistics job scheduling problems.

REFERENCES

[1] Q. Liu, C. Zhang, K. Zhu, and Y. Rao, "Novel multi-objective resource allocation and activity scheduling for fourth party logistics," *Computers & Operations Research*, vol. 44, pp. 42-51, 2014.
[2] X. Feng, F. Chu, C. Chu, and Y. Huang, "Crowdsource-enabled integrated production and transportation scheduling for smart city logistics," *International Journal of Production Research*, vol. 59, no. 7, pp. 2157-2176, 2020.
[3] J. Delaram, and O. F. Valilai, "A Mathematical Model for Task Scheduling in Cloud Manufacturing Systems focusing on Global Logistics," *Procedia Manufacturing*, vol. 17, pp. 387-394, 2018.
[4] A. Elgendy, J. Yan, and M. Zhang, "Integrated Strategies to an Improved Genetic Algorithm for Allocating and Scheduling Multi-Task in Cloud Manufacturing Environment," *Procedia Manufacturing*, vol. 39, pp. 1872-1879, 2019.
[5] E. Taniguchi, and H. Shimamoto, "Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 3-4, pp. 235-250, 2004.
[6] A. Noroozi, M. M. Mazdeh, M. Heydari, and M. Rasti-Barzoki, "Coordinating order acceptance and integrated production-distribution scheduling with batch delivery considering Third Party Logistics distribution," *Journal of Manufacturing Systems*, vol. 46, pp. 29-45, 2018.

[7] M. Ji, J. Fang, W. Zhang, L. Liao, T. C. E. Cheng, and Y. Tan, “Logistics scheduling to minimize the sum of total weighted inventory cost and transport cost,” *Computers & Industrial Engineering*, vol. 120, pp. 206-215, 2018.

[8] Y. X. Chen, “Integrated Optimization Model for Production Planning and Scheduling with Logistics Constraints,” *International Journal of Simulation Modelling*, vol. 15, no. 4, pp. 711-720, 2016.

[9] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, “Selecting Hyper-Parameters of Gaussian Process Regression Based on Non-Inertial Particle Swarm Optimization in Internet of Things,” *IEEE Access*, vol. 7, pp. 59504-59513, 2019.

[10] C.-M. Chen, S. Lv, J. Ning, and J. M.-T. Wu, “A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem,” *Symmetry*, vol. 15, no. 1, 124, 2023.

[11] A. L. H. P. Shaik, M. K. Manoharan, A. K. Pani, R. R. Avala, and C.-M. Chen, “Gaussian Mutation–Spider Monkey Optimization (GM-SMO) Model for Remote Sensing Scene Classification,” *Remote Sensing*, vol. 14, no. 24, 6279, 2022.

[12] X. Qi, “Coordinated logistics scheduling for in-house production and outsourcing,” *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 188-192, 2008.

[13] H. Li, C. Wang, S. Jiang, S. Liu, Y. Rong, and X. Li, “The study of intelligent scheduling algorithm oriented to complex constraints and multi-process roller grinding workshop,” *Advances in Mechanical Engineering*, vol. 12, no. 11, 168781402097588, 2020.

[14] T.-Y. Wu, A. Shao, and J.-S. Pan, “CTOA: Toward a Chaotic-Based Tumbleweed Optimization Algorithm,” *Mathematics*, vol. 11, no. 10, 2339, 2023.

[15] T.-Y. Wu, H. Li, and S.-C. Chu, “CPPE: An Improved Phasmatodea Population Evolution Algorithm with Chaotic Maps,” *Mathematics*, vol. 11, no. 9, 1977, 2023.

[16] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, “Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction,” *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[17] J. Liu, C. Wu, J. Cao, X. Wang, and K. L. Teo, “A Binary differential search algorithm for the 0–1 multidimensional knapsack problem,” *Applied Mathematical Modelling*, vol. 40, no. 23-24, pp. 9788-9805, 2016.

[18] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, “Natural selection methods for Grey Wolf Optimizer,” *Expert Systems with Applications*, vol. 113, pp. 481-498, 2018.

[19] S. N. Makhadmeh, O. A. Alomari, S. Mirjalili, M. A. Al-Betar, and A. Elnagar, “Recent advances in multi-objective grey wolf optimizer, its versions and applications,” *Neural Computing and Applications*, vol. 34, no. 22, pp. 19723-19749, 2022.

[20] J. C. Yang, and W. Long, “Improved Grey Wolf Optimization Algorithm for Constrained Mechanical Design Problems,” *Applied Mechanics and Materials*, vol. 851, pp. 553-558, 2016.

[21] P. Hu, S. Chen, H. Huang, G. Zhang, and L. Liu, “Improved Alpha-Guided Grey Wolf Optimizer,” *IEEE Access*, vol. 7, pp. 5421-5437, 2019.

[22] N. Mittal, U. Singh, and B. S. Sohi, “Modified Grey Wolf Optimizer for Global Engineering Optimization,” *Applied Computational Intelligence and Soft Computing*, vol. 2016, pp. 1-16, 2016.

[23] B. BilalHabedalguni, and M. MalekBarhoush, “Distributed grey wolf optimizer for numerical optimization problems,” *Jordanian Journal of Computers and Information Technology*, vol. 4, no. 3, pp. 21, 2018.

[24] W. Yang, W. Li, Y. Cao, Y. Luo, and L. He, “An Information Theory Inspired Real-Time Self-Adaptive Scheduling for Production-Logistics Resources: Framework, Principle, and Implementation,” *Sensors*, vol. 20, no. 24, 7007, 2020.

[25] L. Zhou, L. Zhang, and Y. Fang, “Logistics service scheduling with manufacturing provider selection in cloud manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 65, 101914, 2020.

[26] K. T. Chaturvedi, M. Pandit, and L. Srivastava, “Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch,” *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 6, pp. 249-257, 2009.

[27] S. Mirjalili, I. Aljarah, M. Mafarja, A. A. Heidari, and H. Faris, ”Grey Wolf Optimizer: Theory, Literature Review, and Application in Computational Fluid Dynamics Problems,” *Nature-Inspired Optimizers*, Springer International Publishing, vol. 2019, pp. 87-105, 2019.

[28] G. Huang, Y. Cai, J. Liu, Y. Qi, and X. Liu, “A Novel Hybrid Discrete Grey Wolf Optimizer Algorithm for Multi-UAV Path Planning,” *Journal of Intelligent & Robotic Systems*, vol. 103, pp. 1-18, 2021.

[29] X. Kong, Y. Yao, W. Yang, Z. Yang, and J. Su, "Solving the Flexible Job Shop Scheduling Problem Using a Discrete Improved Grey Wolf Optimization Algorithm," *Machines*, vol. 10, no. 11, 1100, 2022.

[30] G. Yue, G. Tailai, and W. Dan, "Multi-layered coding-based study on optimization algorithms for automobile production logistics scheduling," *Technological Forecasting and Social Change*, vol. 170, 120889, 2021.