Grey wolf optimizer Based on Multi-Population Collaboration and Its Application in AGV Task Scheduling

Lijie Ma, Yaofa Li

***Abstract*— The real-time scheduling of automatic guided vehicles (AGV) in flexible manufacturing system (FMS) is observed to be highly critical and complex due to the dynamic variations of production requirements such as an imbalance of AGV loading, the high travel time of AGVs, variation in jobs, and AGV routes to name a few. The output from FMS considerably depends on the efficient scheduling of AGV in the FMS. This paper mainly studied intelligent logistics scheduling of automated guided vehicle(AGV) in job shop. AGV logistics scheduling optimization model was established to minimize the travel time of AGV and to reduce energy consumption of AGV. The multi-objective scheduling is carried out by the application of improved grey wolf optimizer (MPGGWO) with task sequencing as the constraint condition. Finally, the actual logistics scheduling of workshop was taken as example to verify the method proposed in this paper. The calculation results show that the AGV logistics scheduling model proposed can well simulate the AGV scheduling time and energy consumption, and the improved grey wolf optimizer (MPGGWO) presents a faster convergence speed and a better optimization ability.**

***Index Terms*—Automated guided vehicle, Energy consumption, Grey wolf optimization algorithm, Logistics scheduling.**

I. INTRODUCTION

Traditional manufacturing in the production, warehousing workshop in the production logistics and transportation methods, there are often high labor costs, operating time constraints, operational inefficiencies and other issues, in the global manufacturing industry competing to seize the high-end position in the value chain, through information technology, digitalization, intelligence and other technological means to promote the manufacturing industry to the development of intelligent manufacturing and to further advance the industrial technological change has been a general trend.

Manuscript received March 15, 2024

Lijie Ma, School of Computer Science and Technology, Tiangong University, Tianjin, China
Yaofa Li, School of Computer Science and Technology, Tiangong University, Tianjin, China

Automatic guided vehicles (AGVs) are one of the most representative logistics conveyor robots, and since their introduction in 1955[1], AGV systems consisting of multiple AGVs have been widely used in a variety of manufacturing systems and warehousing and distribution areas[2, 3]. The task of AGVs on the production floor is to transport finished bins to the inspection area. According to Gotting, more than 20,000 AGVs have been used in industrial applications, and the use of AGVs for transportation can reduce the labor intensity of workers, lower the labor cost of enterprises, and improve the productivity of the workshop[4]. How to perform path optimization for AGVs so that they can provide timely and effective scheduling according to the actual production situation in the workshop and also reduce the waste of energy consumption caused by untimely scheduling. However, how to design AGV scheduling algorithms that are more suitable for production workshops to achieve faster and more accurate transportation operations for production tasks is the great challenge facing the current transformation of the domestic manufacturing industry to intelligent manufacturing.

Currently, in the context of AGV task scheduling strategy in production plant, Umar et al[5]proposed a comprehensive hybrid genetic algorithm for optimization of various performance parameters of FMS. The FMS parameters such as AGV travel time, maximum range, job delay penalty cost and avoidance of AGV delays due to conflicts were optimized. Fazlollahtabar et al[6]solved the problem of scheduling multiple Automated Guided Vehicles (AGVs) in a manufacturing system by considering the arrival dates of AGVs required for material handling in each shop in a job shop arrangement. The applied algorithm minimizes the maximum completion time of the NP-hard combinatorial problem. Komaki and Kayvanfar[7]applied the Gray Wolf optimization algorithm to a two-stage assembly flow shop floor scheduling problem considering the release times of manufacturing and assembly operations. Xuesong Shao et al[8]proposed a multi-objective load task scheduling model for AGVs based on vehicle travel distance, task waiting time and handling task priority. The loading and unloading problem of FMS was solved by Singh and Khan[9]. The authors proposed an efficient analytical method for solving the loading and unloading problem. Similarly, Lu et al[10] applied the Gray Wolf optimization algorithm to welding operations to solve a multi-objective dynamic scheduling problem. In order to solve the real-time dynamic scheduling problem for welding operations with the objective of maximizing the completion time, the scheduling problem was formulated taking into account the job quality, machine reliability, and job delays, and controlling the process time, sequence dependency time, and job transfer time. The proper selection and application of material handling system equipment is a strategic decision[11]. Chanda and Chawla[12-15]applied the Modified Modal Particle Swarm Optimization (MMPSO) algorithm, Clone Selection (CS) algorithm, and Gray Wolf Optimizer (GWO) to synchronous scheduling of AGVs and AGV fleet size optimization in FMS. Angra et al[16] evaluated the performance of different priority scheduling rules when applied to multi-load AGVs with variable size FMS configurations. Han[17] et al. proposed a new realistic mixed-flow shop scheduling model that considers the potential impact of human factors. Experiments show that the model can better solve the practical problems in foundries and the scheduling scheme fully meets the delivery requirements. Wu[18] et al. proposed an improved non-dominated sorting genetic algorithm II based on similarity scheduling and demonstrated that the algorithm can effectively reduce the loading and unloading time of the workpiece while guaranteeing a certain maximum completion time.

From the literature studies, it is understood that there are few studies for the minimum waiting time and minimum travel time scheduling problems for multi-load AGVs, and the energy consumption of AGVs during the scheduling process is rarely considered. This paper focuses on the workshop AGV scheduling optimization problem, the goal is to reduce the response time and travel time of the AGV after the task occurs, and the introduction of green energy consumption indexes, the AGV response time, travel time, energy consumption of the three indexes will be weighted and unified, and put forward the workshop scheduling model of multiple AGVs, which is of great significance for the reduction of the production cost of the factory floor.

II. IMPROVEMENT OF THE GRAY WOLF OPTIMIZER

A. Grey Wolf Optimizer

The Gray wolf optimizer is a typical bio-heuristic algorithm proposed by Seyedali Mirjalili et al[19] in 2014, which is inspired by the two behaviors of social hierarchical stratification and group predation of prey in gray wolf packs. Gray wolves are considered to be top predators at the top of the food chain, and their social hierarchy is divided into four tiers: alpha, beta, delta and omega, the lower the tier, the greater the number of wolves. The position of alpha indicates the current optimal solution in the search space. beta and delta represent the second and third best solutions in the population. The alpha task is to guide the other wolves in their search, trying their best to bring the gray wolf population towards the global optimal solution. ω wolves guide the gray wolf population in a broader search by exploring unsearched areas, improving the algorithm's global search capabilities[20]. Through the interaction and cooperation among alpha, beta, delta and omega, the gray wolf optimizer is able to simulate the social behaviors and collaborations in the gray wolf population to find the optimal solution faster.

The hunting process of the gray wolf can be divided into three steps: searching for prey, encircling prey, and attacking prey.

The behavior of encircling prey during a hunt is defined as follows:

$$D = |C \cdot X_p(t) - X(t)| \tag{2.1}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{2.2}$$

Equation (2.1) represents the distance between an individual and its prey, Equation (2.2) is the position update formula for the gray wolf. Where t is the number of iterations, A and C are the coefficient vectors, X_p and X are the prey's position vector and the gray wolf's position vector, respectively. The formulas for A and C are as follows:

$$A = 2a \cdot r_1 - a \tag{2.3}$$

$$C = 2 \cdot r_2 \tag{2.4}$$

Where components of a are linearly decreased from 2 to 0 over the course of iterations and r_1 and r_2 are random vectors in [0,1].

Gray wolves are able to recognize the location of their prey and surround them. Once the gray wolves recognized the location of their prey, beta and delta, led by alpha, guided the pack to surround the prey, with each individual updating their position according to the formula.

$$\begin{aligned} D_\alpha &= |C_1 \cdot X_\alpha - X| \\ D_\beta &= |C_2 \cdot X_\beta - X| \\ D_\delta &= |C_3 \cdot X_\delta - X| \end{aligned} \tag{2.5}$$

Where $D_\alpha, D_\beta, D_\delta$ denote the distances between alpha, beta, delta and other gray wolf individuals respectively; X_α, X_β, X_δ represent the current position of alpha, beta, and delta; C_1, C_2, C_3 determined by Equation (2.4), X represents the current position of the individual gray wolf.

$$\begin{aligned} X_1 &= X_\alpha - A_1 \cdot D_\alpha \\ X_2 &= X_\beta - A_2 \cdot D_\beta \\ X_3 &= X_\delta - A_3 \cdot D_\delta \end{aligned} \tag{2.6}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{2.7}$$

Equation (2.6) defines the step size and direction of the individual omega approaching to alpha, beta, and delta, respectively, in the wolf pack, and Equation (2.7) is the updated location.

B. Improved Gray Wolf Optimizer

There is a strong correlation between the quality of the population in the initialization phase and the excellence of the algorithm. The standard gray wolf optimizer generally uses a random initialization method to generate the initial population in the initialization phase, which produces a poor diversity of the initial population and fails to achieve a uniform distribution of population individuals in the search space. Typical characteristics of chaotic mapping are randomness, ergodicity, regularity, etc., which can ensure population diversity and optimize the global search process[21]. Chaotic Tent mapping uses the Tent function as a mapping function, which under certain initial conditions and parameters can generate sequences with chaotic properties as an alternative to pseudo-random number generators, which usually give better results[22]. The chaotic Tent mapping is defined as follows:

$$x_{k+1} = \begin{cases} x_k/\beta, & x_k \epsilon (0,\beta) \\ (1-x_k)/(1-\beta), & x_k \epsilon (\beta, 1) \end{cases} \quad (2.8)$$

Assuming a population size of N, When $x\epsilon[0,1]$ and $x\epsilon[0,1]$, the system is in a chaotic state.

Genetic algorithms are a series of computational models developed by Holland[23, 24], which by applying the principle of survival of the fittest, are computational models that simulate the process of biological evolution based on the principles of natural selection and genetics of Darwin's theory of biological evolution, and are capable of searching for and determining the optimal solution in the evolutionary process. The crossover mutation operator is an operation in genetic algorithms used to introduce randomness and diversity into the evolutionary process, prompting the population to perform a comprehensive search in the solution space, thereby improving the algorithm's search performance. According to the model mentioned in this paper, the encoding of machine task sequences as chromosomes is used, and a chromosome is a sequence of task occurrences. For example, the chromosomes of an individual are coded as (1,3,2,4,7,8,9,6,5) indicating that the order of execution of the tasks is to perform the 1st occurring task first, then the 3rd occurring task, until all the tasks are performed. When the AGV actually performs a task, it needs to take into account the maximum number of loaded weights, for example, after completing 5 loading tasks, the AGV needs to deliver the goods to the unloading point X before going to perform the rest of the tasks. Then the actual traveling route of the AGV is (1,3,2,4,7, X, 8,9,6,5).

The crossover strategy used in this paper is two-point crossover, as shown in Fig. 1 crossover points are randomly set in the coding strings of two individuals paired with each other, and part of the chromosomes of the two individuals between the two crossover points set are exchanged with probability P_c. The mutation strategy employed in this paper is multi-point mutation, where each locus of the chromosome undergoes a mutation operation with a certain probability P_a:

$$P_a = 0.1 + l \cdot \frac{0.1}{Max_{iter}} \quad (2.9)$$

Where P_a is the probability of mutation, l is the current number of iterations, and Max_{iter} is the maximum number of iterations. As the number of iterations increases, the population tends to fall into a local optimum, and the probability of mutation is increased from 0.1 to 0.2 to help the population jump out of the local optimum.

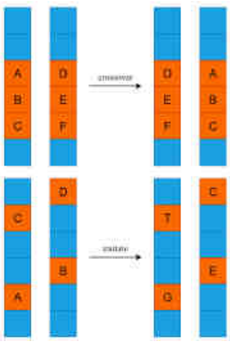


Figure1: Crossover and Mutation diagram

Existing research on multi population optimization has shown that multi population strategies can be easily integrated into various heuristic algorithms and usually perform better than single population optimization algorithms. The effectiveness of multi-population strategies lies in[25-27]: (1) dividing the entire population into multiple sub-populations allows for diversity maintenance as different sub-populations can explore different search spaces; (2) it enables searching in different regions, facilitating efficient identification of optimal solutions; (3) multi-population strategies can be easily integrated into various heuristic algorithms. In the Grey Wolf Algorithm, as individual directions are guided by the three alpha, beta, and delta wolves, if the alpha wolf gets trapped in a local optimum, it can lead to early convergence of the entire population and reduced diversity. Therefore, a multi-population strategy is employed to optimize the aforementioned problem.

The present study first initializes n*N gray wolf individuals (1 master population and n-1 slave populations) to enhance the algorithm's search capability, while employing a leader competition strategy to maintain information exchange among n populations. The article proposes a population scheme as shown in Fig.2: using a master-slave communication model to maintain algorithm synchronization, where the main group is associated with the subgroups, and each slave node independently executes the gray wolf algorithm, including searching for the leader individual of the population and updating individual positions.

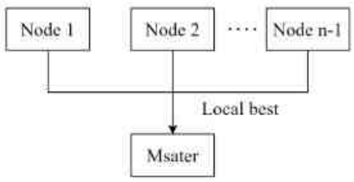


Figure2: The master-slave model

$$X_\alpha^M = Max(X_\alpha^1, X_\alpha^2, ..., X_\alpha^{n-1})$$
$$X_\beta^M = second\_Max(X_\alpha^1, X_\alpha^2, ..., X_\alpha^{n-1})$$
$$X_\delta^M = third\_Max(X_\alpha^1, X_\alpha^2, ..., X_\alpha^{n-1}) \quad (2.10)$$

Equation (2.10) is the calculation formula for the main group's leader position. where X_α^i represents the best individual position of slave population i, X_α^M represents the best individual position among all the slave nodes, X_β^M represents the second best individual position, and X_δ^M represents the third best individual position.

When all individuals in each node have completed their updates, each node sends the best individual of its respective population to the master node. The master node then selects the top three individuals from all the received individuals as the leader wolves of the main group, enabling dynamic updates of the master population's leader. The multi-population strategy allows the main group to benefit from the search experiences of other subgroups, helping to avoid getting trapped in local optima and increasing the likelihood of discovering the global optimal solution. The pseudo-code of MPGGWO is shown in Fig.2.

Algorithm 1: Pseudo-code of MPGGWO

Input: Max_{iter}: number of iterations, n: number of population

Initialize the gray wolf population $X_i (i = 1,2,\ldots,n)$ based on chaotic tent map
The population is divided into three subpopulations
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent in all populations
X_β=the second best search agent in all populations
X_δ=the third best search agent in all populations
while(t < Max_{iter})
    for each search agent
        update the position of the current search agent by equation
        maintain the better solution by crossover and mutation
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, X_δ
    t=t+1
end while
return X_α

Figure2: Pseudo-code for the MPGGWO algorithm

III. AGV TASK SCHEDULING MODEL

A. Description of the problem

Fig.3 depicts the layout of machine positions in the workshop, with each machine accompanied by a buffer area. There are a total of 30 task points represented by black circles, and the black straight line represents the path for the Automated Guided Vehicle (AGV) to travel. The black circular markers denote the task points, while the black ring-shaped circles indicate the unloading points. The AGV starts its operation at the unloading point. Upon receiving a scheduling task, it selects a feasible path to move towards the task point and load the goods. When the cargo reaches a certain threshold, denoted as "k," the AGV returns to the unloading point to unload the goods before proceeding with the next task. This process is repeated until all tasks are completed. The objective of this study is to find a reasonable task sequence that minimizes the waiting time for AGV responses and reduces energy consumption as much as possible.

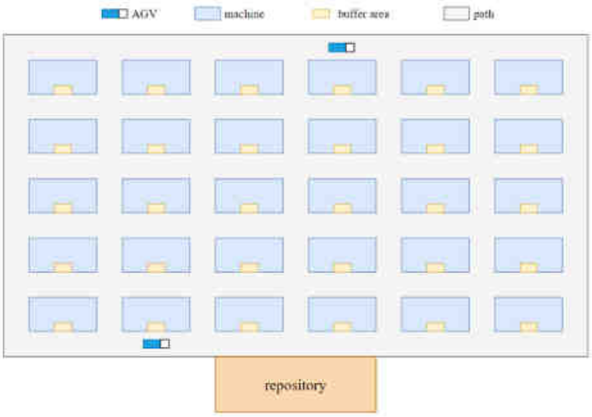


Figure3: Task point distribution of workshop

The provisions for modeling are as follows:

(1) Work of AGV: Deliver products from the task point to the unloading point for unloading.

(2) All production facilities have sufficient raw materials, and no lack of raw materials will occur.

(3) All production equipment will process products without interruption due to malfunction.

(4) The maximum load of the AGV cannot exceed k products.

(5) The loading time and unloading time of the product are fixed.

(6) The AGV receives a loading task and performs only one loading task at the mission point.

(7) All production equipment and AGVs are operational at the start of moment zero.

(8) The AGV maintains a constant speed and its travel time is only related to the path length and the number of curves it passes through.

The model parameters are shown in Table I.

Table I Symbol meaning table

symbol	meaning
T_i	Moment of occurrence of task i
Z	Loading time
V_z	Straight-line travel speed of AGV
V_w	Curve travel speed of AGV
$t_{0,i}$	Travel time from unloading point to task point i
$t_{i,j}$	Travel time from task point i to task point j
t_i	Moment of completion of task i
D_j	AGV pending response time for task j
S_w	Curve travel distance of AGV
S_z	Straight-line travel distance of AGV
P_r	Operating power of AGV
P_s	Standby power of AGV
T_r	AGV running time
T_s	AGV standby time
T_c	Time cost
E_c	Energy cost
F_i	AGV scheduling cost

B. Description of the AGV Task Scheduling Model

In the logistics scheduling of workshop, the main factors affecting the scheduling time of AGVs are the traveling time of AGVs between task points and the standby response time after receiving a scheduling task. AGVs are not running every moment during the whole scheduling period, and if there is no new task to schedule AGVs will remain in standby at the previous task point. The operation power of AGV is much larger than the standby power, therefore, finding an optimal task sequence to make the operation time of AGV and the energy consumption of AGV operation less is the key of this paper. Considering the above factors, the scheduling model is established as follows:

$$F_t = \sum_{i=1}^{n} F_i \tag{3.1}$$

$$F = u_1 * T_c + u_2 * E_c \tag{3.2}$$

$$T_c = \sum_{i=1}^{n-1} \sum_{j=2}^{n} t_{ij} + \sum_{j=2}^{n} D_j \tag{3.3}$$

$$t_{ij} = \begin{cases} \dfrac{S_w}{V_w} + \dfrac{S_z}{V_z} & i \neq ka\,(a=1,2,3\ldots) \\ t_{ij} = t_{i,0} + t_{0,j} & i = ka\,(a=1,2,3\ldots) \end{cases} \tag{3.4}$$

$$t_j = T_j + D_j + t_{ij} + Z \quad (3.5)$$

$$D_j = \begin{cases} t_i - T_j & t_i \geq T_j \\ 0 & t_i < T_j \end{cases} \quad (3.6)$$

$$T_r = \sum_{i=1}^{n-1} \sum_{j=2}^{n} t_{ij} \quad (3.7)$$

$$T_s = T_n - t_1 - T_r \quad (3.8)$$

$$E_c = T_r * P_r + T_s * P_s \quad (3.9)$$

Where u_1 and u_2 are weight coefficients. P_r represents the operational power of the vehicle. Equation (3.4) represents the travel time of the vehicle from task point i to j, If the vehicle reaches its maximum load capacity when completing task i, it needs to return to the unloading point to unload the goods before proceeding to execute task j. Equation (3.5) represents the completion time of task j. Equation (3.6) represents the waiting time for task j, which is the time that has elapsed since task i was completed until task j occurs. If task j occurs after task i has been completed, the waiting time for task j is considered to be 0. Equation (3.7) represents the total running time of the vehicle. Equation (3.8) represents the total idle time of the vehicle. Equation (3.9) represents the objective function for the vehicle's energy consumption.

Equation (3.2) represents the total objective function for a single vehicle, converting the multi-objective model of AGV scheduling in a production workshop to a single-objective model. The Analytic Hierarchy Process (AHP) is used for assigning values. The relative importance scores for the three factors affecting the objective function, namely AGV response time, running time, and energy consumption, were determined through expert assessment to determine their relative weights. The consistency-checked weight coefficients obtained are shown in Table II. The total objective function for n AGV vehicles is given by Equation (3.1).

Table II Weight evaluation table

u_1	u_2
0.7703	0.2297
77.03%	22.97%

IV. EXPERIMENTAL SIMULATION AND ANALYSIS

A. Simulation Experiment Environment

The simulation environment of this paper is AMD Ryzen 7 6800H with Radeon Graphics CPU, 3.20 GHz running memory 32GB, operating system Windows11, programming environment MatlabR2022b.

B. Comparison Algorithms and Parameter Settings

In order to verify the effectiveness and superiority of the MPGGWO algorithm mentioned in this paper for solving the AGV task scheduling, this paper uses three different sets of task data as shown in Table III, Table IV and Table V, and seeks for the optimal scheduling result using Equation (3.1) as the objective function. The GWO algorithm, GGWO algorithm, MPGWO algorithm and MPGGWO algorithm mentioned in this paper are compared and analyzed. The specific parameters of each algorithm are shown in Table VI.

Table III Task1 data sheet

Task number	Machine number	T_i	Task number	Machine number	T_i
1	10	16	9	15	284
2	3	67	10	2	311
3	7	96	11	4	351
4	14	149	12	11	370
5	6	181	13	13	410
6	12	199	14	8	418
7	1	207	15	5	479
8	9	238			

Table IV Task2 data sheet

Task number	Machine number	T_i	Task number	Machine number	T_i
1	19	37	13	11	633
2	12	89	14	5	657

3	21	155	15	22	686
4	18	172	16	17	741
5	8	202	17	3	766
6	2	221	18	16	821
7	9	256	19	1	841
8	4	307	20	10	884
9	6	399	21	15	903
10	13	426	22	20	947
11	24	522	23	7	956
12	23	574	24	14	992

Table V Task3 data sheet

Task number	Machin e number	T_i	Task number	Machine number	T_i
1	25	5	16	21	373
2	4	7	17	28	444
3	13	13	18	18	489
4	20	79	19	15	522
5	30	102	20	26	532
6	2	117	21	22	550
7	23	161	22	24	618
8	3	191	23	10	695
9	14	269	24	17	749
10	12	274	25	9	754
11	7	293	26	6	760
12	1	308	27	19	817
13	11	312	28	8	844
14	29	331	29	27	875
15	16	360	30	5	961

Table VI Algorithm parameter settings

Algorithm	Parameter settings
GWO	$N = 81$, $Max_{iter} = 100$, $a = 2 - 2 * l/Max_{iter}$, r_1 and r_2 are the random number of the interval [0,1]
GGWO	$N = 81$, $Max_{iter} = 100$, $a = 2 - 2 * l/Max_{iter}$, r_1 and r_2 are the random number of the interval [0,1], $P_a = 0.1 + l * 0.1/Max_{iter}$, $P_c = 0.9$
MPGWO	$N = 81$, $Max_{iter} = 100$, $a = 2 - 2 * l/Max_{iter}$, r_1 and

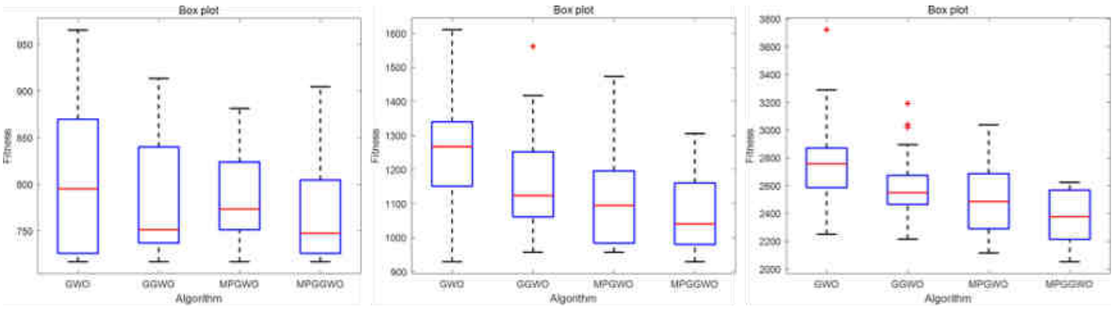
	r_2 are the random number of the interval $[0,1]$, $n = 4$
MPGGWO	$N = 81$, $Max_{iter} = 100$, $a = 2 - 2 * l/Max_{iter}$, r_1 and r_2 are the random number of the interval $[0,1]$, $P_a = 0.1 + l * 0.1/Max_{iter}$, $P_c = 0.9$, $n = 4$

C. Experimental Results and Analysis

Table VII shows the optimization results of the four algorithms under different tasks, and 30 sets of data were used to compare the superiority of the algorithms respectively. When the problem size is simple, the improvement of the improved algorithms is not obvious. When the complexity of the problem increases, the improved algorithm MPGGWO has a more prominent optimization searching ability, and the effect is improved by 14.14% and 13.08% in Task 2 and Task 3, respectively.

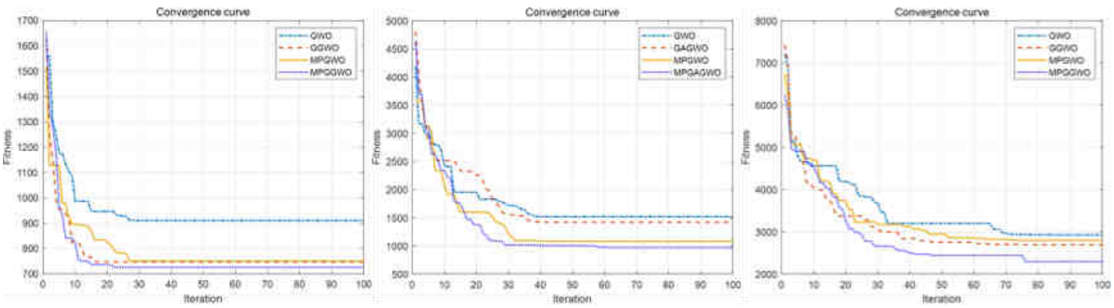
Table VII The result of the comparison of MPGGWO with GWO, GGWO and MPGWO

Task	Index	GWO	GGWO	MPGWO	MPGGWO
Task1	Best	716.9905	716.9905	716.9905	716.9905
	Avg	808.7612	767.6604	798.8993	754.0125
	Worst	993.1095	911.3037	889.8246	900.7682
Task2	Best	929.3597	957.1789	957.1789	929.3597
	Avg	1251.0901	1161.051	1103.7189	1074.7294
	Worst	1610.9565	1561.7177	1473.703	1305.5817
Task3	Best	2250.2054	2214.8273	2116.3107	2052.9883
	Avg	2752.4221	2601.6227	2504.3359	2392.0506
	Worst	3722.8201	3190.496	3038.1803	2623.3984



(a) (b) (c)

Figure4： Box plot of various algorithms on task1, task2 and task3



(a) (b) (c)

Figure5： Convergence curves of various algorithms on task1, task2 and task3

Fig.4 shows the box plot comparison of the four algorithms under different tasks, and the results of 30 sets of experiments were used to analyze the experimental data of the MPGGWO algorithm, which is more stable and has not appeared dirty data, and has achieved a more satisfactory result.

Fig.5 shows the convergence images of the four algorithms under different task sizes. By analyzing the convergence graph, it can be seen that the traditional GWO algorithm has the problem of slow convergence in planning the optimal task sequence, and the MPGGWO algorithm proposed in this paper further improves the convergence speed of the global search at the initial stage, and the optimal solution of multiple experiments is also more stable. The MPGGWO algorithm effectively improves the deficiencies of the traditional GWO algorithm in optimizing the solution, and improves the overall convergence performance and local search performance of the algorithm. It can be seen that the MPGGWO algorithm has better comprehensive optimization ability in the AGV task scheduling problem, which helps to reduce the waiting response time of the machine in the workshop and the energy consumption of the AGV operation.

V. CONCLUSIONS

In this paper, an optimization model combining AGV pending response time and energy consumption is established for the characteristics of the workshop AGV scheduling problem, for which an improved gray wolf algorithm, MPGGWO, is proposed, which mainly improves the traditional grey wolf optimizer from the initialization of the population, the introduction of the Crossover and mutation and the multi-population collaboration. To address the defects of the traditional GWO algorithm that the initialized population is not uniformly distributed, this paper introduces the chaotic Tent mapping to initialize the gray wolf population to increase the randomness and diversity of the population, to expand the search space, to accelerate the convergence speed, and to improve the robustness of the algorithm. Aiming at the traditional grey wolf optimizer, which is easy to fall into the local optimal solution problem, this paper proposes the elite retention and multiple population synergy strategies, so that the algorithm can jump out of the local optimum in the iterative process. Simulation experiments show that under three groups of simulation experimental scenarios with different complexity, the MPGGWO algorithm in this paper has certain superiority in terms of optimality seeking ability and stability in AGV task scheduling. It has greater practical significance for solving the workshop AGV scheduling problem.

ACKNOWLEDGMENT

The authors would like to thank the Professor Liang for supporting the research work.

REFERENCES

[1] MŸLLER T. Automated Guided Vehicles. IFS (Publications) Ltd [Z]. Springer-Verlag. 1983

[2] FRAGAPANE G, DE KOSTER R, SGARBOSSA F, et al. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda [J]. 2021, 294(2): 405-26.

[3] KABIR Q S, SUZUKI Y J C, ENGINEERING I. Increasing manufacturing flexibility through battery management of automated guided vehicles [J]. 2018, 117: 225-36.

[4] GOTTING H J P T I. Automation and Steering of Vehicles in ports [J]. 2000, 10: 101-11.

[5] UMAR U A, ARIFFIN M, ISMAIL N, et al. Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment [J]. 2015, 81: 2123-41.

[6] FAZLOLLAHTABAR H, SAIDI-MEHRABAD M, BALAKRISHNAN J J R, et al. Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms [J]. 2015, 72: 131-8.

[7] KOMAKI G, KAYVANFAR V J J O C S. Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time [J]. 2015, 8: 109-20.

[8] XUESONG S, YUXIANG G, RUIPENG S. Multi-target Compounded AGV Scheduling System Modeling and Application in Electric Power Metering Calibration [J]. Electric Power Engineering Technology, 2016(35(05)): 24-7.

[9] SINGH R, KHAN B J I J O I E C. Meta-hierarchical-heuristic-mathematical-model of loading problems in flexible manufacturing system for development of an intelligent approach [J]. 2016, 7(2): 177-90.

[10] LU C, GAO L, LI X, et al. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry [J]. 2017, 57: 61-79.

[11] SEN K, GHOSH S, SARKAR B J J O T I O E S C. Comparison of customer preference for bulk material handling equipment through fuzzy-AHP approach [J]. 2017, 98: 367-77.

[12] CHAWLA V, CHANDA A, ANGRA S J I J O D, et al. Sustainable multi-objective scheduling for automatic guided vehicle and flexible manufacturing system by a grey wolf optimization algorithm [J]. 2018, 2(1): 27-40.

[13] CHAWLA V, CHANDA A, ANGRA S J J O P M. Scheduling of multi load AGVs in FMS by modified memetic particle swarm optimization algorithm [J]. 2018, 3(1): 39-54.

[14] CHAWLA V, CHANDA A, ANGRA S J M S L. Automatic guided vehicles fleet size optimization for flexible manufacturing system by grey wolf optimization algorithm [J]. 2018, 8(2): 79-90.

[15] CHAWLA V K, CHANDA A K, ANGRA S J J O T I O E S C. A clonal selection algorithm for minimizing distance travel and back tracking of automatic guided vehicles in flexible manufacturing system [J]. 2019, 100: 401-10.

[16] ANGRA S, CHANDA A, CHAWLA V J M S L. Comparison and evaluation of job selection dispatching rules for integrated scheduling of multi-load automatic guided vehicles serving in variable sized flexible manufacturing system layouts: A simulation study [J]. 2018, 8(4): 187-200.

[17] HAN W, DENG Q, GONG G, et al. Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint [J]. 2021, 168: 114282.

[18] WU X, PENG J, XIAO X, et al. An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading [J]. 2021, 32: 707-28.

[19] MIRJALILI S, MIRJALILI S M, LEWIS A J A I E S. Grey wolf optimizer [J]. 2014, 69: 46-61.

[20] MIRJALILI S, SAREMI S, MIRJALILI S M, et al. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization [J]. 2016, 47: 106-19.

[21] YOU Y, WANG S A, SHENG W. New chaos optimization algorithm with applications [J]. Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University, 2003, 37(1): 69-72.

[22] SHANG L, LI H, HOU Y, et al. Short-Term Power Load Forecasting Based on Feature Selection and Optimized Extreme Learning Machine [J]. Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University, 2022, 56(4): 165-75.

[23] GOLDBERG D J R M. Genetic Algorithms in Search, Optimization and Machine Learning,. Addition-Westly [J]. 1989.

[24] SAMPSON J R. Adaptation in natural and artificial systems (John H. Holland) [Z]. Society for Industrial and Applied Mathematics. 1976

[25] LI C, NGUYEN T T, YANG M, et al. Multi-population methods in unconstrained continuous dynamic environments: The challenges [J]. 2015, 296: 95-118.

[26] NSEEF S K, ABDULLAH S, TURKY A, et al. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems [J]. 2016, 104: 14-23.

[27] WU G, MALLIPEDDI R, SUGANTHAN P N, et al. Differential evolution with multi-population based ensemble of mutation strategies [J]. 2016, 329: 329-45.