Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 7, No. 1, 2024

Flexible Job-shop Scheduling Optimization based on Improved Gray Wolf Algorithm

Minghong Xu *

School of Electronic and Information Engineering, Soochow University, Suzhou, Jiangsu, 215006, China

*** Corresponding Author Email:** xmh2028410256@163.com

Abstract: This article proposed an improved gray wolf optimizer to deal with the flexible job-shop scheduling problem. By using a random key for coding job positions and adopting a local search strategy, we achieve group reconstruction and updating with the help of three good fitness values of the population, hence continuously searching for the optimal solution. Simulation experiments were conducted on a standardized test case, demonstrating the effectiveness of this method.

Keywords: Flexible Job-shop Scheduling; GWO; Group Optimization.

1. Introduction

The flexible job-shop scheduling problem (FJSP), as a representative NP hard problem, is of great significance in production and processing processes. It is a popular research subject in the field of intelligent manufacturing. A series of optimization methods based on cluster optimization, ant colony optimization, artificial immunity, and particle swarm optimization have broad application prospects in industries such as electronics, component manufacturing, aerospace, etc. Zhang G uses an improved genetic algorithm to solve FJSP under multiple constraints [1]; Lu H proposes a new ant colony algorithm centered on distributed estimation of flexible job scheduling problem [2]; Zhao X proposes an improved ant algorithm to solve FJSP [3]; Zhang Y adopts the modified particle swarm optimization method for solving the sorting issue [4]. However, the methods discussed above are all based on specific mathematical models, and there are certain differences from the problems faced in reality. So, an effective intelligent algorithm is the key to solving combinatorial optimization problems. Seyedalis et al. introduced a new intelligent optimization method based on genetic networks [5], namely Gray Wolf Optimization (GWO). GWO is a new heuristic optimization method developed by imitating the hunting behavior mechanism of wolves, which is easy to implement and easy to implement. Unlike other genetic algorithms such as genetic algorithm, genetic algorithm, etc., this paper proposes a new algorithm - Improved Gray Wolf Optimization.

In response to the FJSP problem, this project plans to adopt an improved grey group optimization method based on global optimization and global optimal embedding, and apply it to FJSP, and prove its efficiency and feasibility through experiments.

2. Description of the Problem

(1) The set of workpieces to be processed is $\{J_{1,} J_2, J_3 \cdots J_n\}$, the set of machines tools $\{M_{1,}, M_2,\ M_3 \cdots M_m\ \}$that can be used in the workshop, n - the maximum number of workpieces, *m - the* maximum number of machines available;

(2) Process each workpiece in a different and fixed order.S_i —Number of work processes, $\{$i| i=1,2,3$\cdots n\}$;

(3) ST_{ijk} - starting time of machining of the no. j process of workpiece i on machine no. k, and C_{ijk} -completing time of machining of the no.j process of workpiece i on machine no. k, and F_{ijk} - duration of machining time of the no.j process of workpiece i on machine no. k, $\{$k| k=1,2,3$\cdots n\}$;

(4)T_k— The actual operating time of machine no. k.F_{max}-The final completing time for the completion of all processes for all workpieces.

The optimization of FJSP's operation time aims to be as close as possible to the real production situation, and achieving the optimal configuration between various equipment, workpieces, and processes so that F_{max}- The final completing time for the completion of all processes for all workpieces is minimized determines the machining time objective function:

$$\min F_{max} = \min\{\max T_k\} \tag{1}$$

3. Improved Gray Wolf Optimization (IGWO)

3.1. Gray Wolf Algorithm

GWO simulates the process of searching for prey by gray wolves, and its mechanisms are described below in terms of the principles of social hierarchy, tracking, and encircling and attacking prey, respectively.

(1) Social hierarchy: in formulating roles in the GWO, the community level of gray wolves needs to be formulated by noting the best-adapted, suboptimal, and third-best wolves in the pack as α, β and δ , and the remaining wolves are called ω.Therefore, the community level of gray wolves is ranked orderly from high to low: α, β, δ and ω . In the GWO algorithm, hunting (optimization) is mainly leaded by α, β, δ to lead the completion of the process. ω The wolves must obey the three wolves in front of them.

(2) Siege: During the hunting process, wolf packs must surround them in groups, and the mathematical model of their capture process is shown below:

$$\mathbf{D} = \left|\mathbf{C} \cdot \mathbf{X}_{\mathrm{p}}(t) - \mathbf{X}(t)\right| \tag{2}$$

$$\mathbf{X}_{\mathrm{p}}(t+1) = \mathbf{X}_{\mathrm{p}}(t) - \mathbf{A} \cdot \mathbf{D} \tag{3}$$

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a} \tag{4}$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2 \tag{5}$$

Here t is the current iteration number; A and C are coefficient vectors; X_p represents the position vector of the prey; *X* represents the position vector of the gray wolf. *a* decrease from 2 to 0 during the iteration.

(3) Hunting: Gray wolves are able to identify the their target and surround it, and their hunting behavior is usually accomplished by the guidance of α, β, δ . While in an abstract search space the gray wolf does not know the specific location of the optimal solution (prey). In a gesture to modify the hunting behavior of the gray wolf, we assume that α (the optimal candidate solution), the β and δ possess more spatial information about the potential prey. Therefore, during each iteration, the currently obtained best 3 wolves are saved (α, β, δ), based on their location information to update the other search agents (including the ω) locations, in this regard the following equation:

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}|, \mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}|, \mathbf{D}_\delta = |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}| \quad (6)$$

$$\mathbf{X}_1 = \mathbf{X}_a - \mathbf{A}_1 \cdot \mathbf{D}_a, \mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta, \mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta \quad (7)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1+\mathbf{X}_2+\mathbf{X}_3}{3} \quad (8)$$

(4)Attacking prey (development): in order to simulate the process of attacking prey, the value of A fluctuates by decreasing the value of a according to equation (4). That is, A is a random vector on the interval [-2a,2a], where a decreases linearly from 2 to 0 during the iteration process. If A is located in the region [-1,1], the next instant positioning of the search object may be located anywhere within that region.

Finding prey (searching): Gray wolves search for prey mainly based on the location information of α, β *and* δ . They first disperse among themselves to look for prey, then they gather together to attack the prey. For the dispersion modeling, a random value of A greater or less than 1 is adopted to keep its search agents away from the prey, in this way GWO is able to achieve a complete global search. Another search factor of the GWO algorithm is C. From equation (5), it can be seen that the C vector is a random value in the interval [0.2], and the factor gives a random weight to the prey in a gesture to increase randomly the value of the prey in the interval [0.2] ,so as to increase the influence of (|C|>1) prey or reduce that of (|C|<1) prey in equation (6). This can help the GWO to show a more stochastic behavior in the optimization, which is beneficial to avoid exhibiting local optimum. It should be pointed out that C is not linearly decreasing. Instead it is a random point during the iteration process for better search at the beginning and end of the iteration. This factor facilitates the algorithm to jump out of localization, especially during the process of the last iteration.

3.2. Designing of Improved GWO

3.2.1. Encoding and Decoding Methods

For the typical discrete optimization problem of FJSP, while traditional GWO methods are mostly suitable for continuous variable optimization problems and cannot be directly modified. As a result, designing an effective coding and decoding method is one of the keys to apply the GWO algorithm to FJSP. Here, this article uses the LOV (Largest Order Value) rule, which is based on random key encoding, is utilized to convert continuous positioning variables into discrete values of $X_i = \left[x_{i,1}, x_{i,2}, \cdots, x_{i,n}\right]$. This method does not require any modifications of the GWO algorithm to ensure its feasibility [6].

3.2.2. Initialize Population Generation

To guarantee the diversity of the initial population, a random method is used to generate all members of it, which is initialized by the following equation:

$$x_{i,j} = \text{lower}_j + \left(\text{upper}_j - \text{lower}_j\right) \times \text{random} \quad i = (1,2,\cdots,N), j = (1,2,\cdots,D) \quad (9)$$

Here, N and D represent the size and dimension (i.e., number of workpieces) of the population; $upper_j$ and $lower_j$ respectively represents the upper and lower solutions of the no.j element, which are 1 and 0 in the text.random is a random number on the interval [0,1].

3.2.3. Positional Fitness Function

The fitness function, also known as the evaluation function, is determined based on an objective function, which is used as a criterion for distinguishing between higher and lower levels of individual hierarchies in the wolf pack, and is the basis for determining the optimal top 3 levels of wolves. In the position update of the wolves, those with higher fitness (α, β, δ) are retained and guide the gray wolves with lower fitness (ω) or other wolves to search towards the prey.

In this case, the goal is achieving the shortest completing time, i.e., Makespan. Assuming C_{max}^k represents the ending time of the no.k scheduling method, then the fitness function is

$$\text{fitness}=1/C_{max}^k \quad (10)$$

For the flexible workshop scheduling problem of size n×m, it has n! workpiece processing orders. The task of scheduling is to select the optimal one from the n! orderings. Due to the large number of job schedules in batch manufacturing and the strong global optimization performance of GWO, it is suitable for solving large-scale scheduling problems.

3.2.4. Updating Mechanism

According to the above working principle in GWO, the tracking, encircling and attacking prey mechanism is used to update the wolf pack position information. As the iteration proceeds, the candidate wolves will move towards the targeted set (optimal solution) and launch an attack on it.

3.2.5. Local Search Strategy

The use of local optimization methods can effectively improve its convergence speed, and the performance depends on the neighborhood structure used [7]. For the FJSP problem, the use of the INSERT neighborhood structure enables a more thorough and efficient search. Therefore, in this article, we adopt insert local retrieval. And the designed insert local search flowchart is shown in Fig. 1

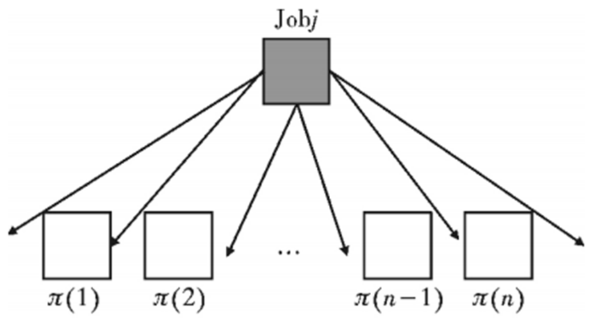

Fig 1. Insert operation

As shown in Fig. 1, for n documents corresponding to the ordering π, an attempt is made to insert artifact j into one of the (n+1) possible positions in the sequence, resulting in (n+1) possible candidate solutions. Among these candidate solutions, the ordering that corresponds to the minimum Makespan is taken as a newest ordering.

The flow pseudo-code of the IGWO is shown in Fig. 2.

```
Input Parameters(a);
wolf population X_i(i=1,···,N)←initialization(wolves_Size);
fitness(C_max)←evaluate(wolf population);
while (evaluation_number<max_evaluation_number)
    X_α,X_β,X_δ←select the first best three wolves(wolf population);
    new position of the wolf←update(the current wolf);
    update(a);
    fitness←evaluate(new position of the wolf);
    evaluation_number++;
    while(i<n)
        new position of the wolf←insert operation(X_α);
        evaluation_number++;i++;
        if new position of the wolf is better than that of X_α
            replacement(new position,X_α);
            break;
        end if
    end while
        evaluation_number++;
end while
X_α←select the best wolf(the entire wolves);
return X_α
```

Fig 2. Pseudo code of Improved GWO for FJSP

4. Experimentation and Analysis

By comparing with GA and standard GWO methods, the effectiveness of IGWO was confirmed, and the latest flow shop test set is used to minimize the maximization of the C_{max}[8]. As the goal, this paper takes the representative 12 test sets in this test set for testing. The algorithm parameters are set as follows: for GA, GWO and IGWO, a population of 100 gray wolves was selected, with a population size of 100 wolves and a fitness evaluation frequency of 100000. To ensure the stability of the algorithms, each algorithm is run independently on each test instance for 50 times.

Under the same experimental conditions, $\mathrm{RPD} = ({}_{\mathrm{Method}}\mathrm{Best}_{\mathrm{sol}})/\mathrm{Best}_{\mathrm{sol}} \times 100\%$ represents average relative error for each instance of the formula; $Best_{sol}$ represents the optimal result sought among all optimization algorithms;$Method_{sol}$ represents the optimal result obtained by a certain algorithm; its running simulation results are listed in Table 1:

Table 1. Comparisons between the algorithms

Sample	Scale	GA		GWO		IGWO	
		BS	RPD/%	BS	RPD/%	BS	RPD/%
VRF10_5_ 1	10X5	1 664	0	1 664	0	I 664	0
VRF20_ 10_ 1	20X10	2 293	23.1	2221	38.9	2221	20.5
VRF30_ 10_ 1	30X 10	2 937	28.1	2766	29.2	2 754	33.7
VRF40_ 10_ 1	40X10	3 643	14.5	3 393	19.3	3 353	24.5
VRF50_ 10_ 1	50X10	3 949	40.7	3 342	39.3	3 844	36.8
VRF60_ 10_ 1	60X 10	4 431	54.9	3 834	39.6	3 834	37.9
VRF100_ 20_ 1	100X 20	9 904	31.5	9727	29.2	9 679	28.7
VRF200_20_ 1	200X 20	15 835	21.1	15 644	20.0	15 573	17.1
VRF300_40_ 1	300X 40	21 545	32.2	21 473	13.8	21298	23.2
VRF500_ 40_ 1	500X40	32 807	19.7	32 546	16.3	32 340	22.3
VRF600_40_ 1	600X 40	38 104	24.4	37 886	17.1	33 780	15.7
VRF700_40_ 1	700X 40	43 598	16.7	43 448	11.6	43 248	11.8

Table 1 exhibits the performance statistics of the three algorithms in testing problems and compares them. Where BS denotes that the algorithm finds the best value in the corresponding instance in 50 runs, and RPD is the average relative deviation. The results demonstrate that IGWO is considerably better than the standard GWO and GA algorithms, indicating that IGWO has some advantages in dealing with FJSP.

5. Conclusion

In response to the flexible job-shop scheduling problem, we propose an improved Gray Wolf Optimization (IGWO) and introduce a local optimization search strategy. Experimental research was conducted to prove its feasibility and efficiency.

References

[1] Zhang, G., Hu, Y., Sun, J. (2020). Improved genetic algorithm for solving flexible job shop scheduling problem with multiple time constraints. Industrial Engineering, 19-25+48.

[2] Lu, H. (2019). Research on flexible job shop scheduling method based on distribution estimation-ant colony hybrid algorithm. Electromechanical Engineering, 366-375.

[3] Zhao, X., Wei, Y., Wang, K., et al. (2022). Study on flexible job shop scheduling problem with improved ant colony algorithm. Combined machine tools and automated machining technology, 02:15-22+52.

[4] Zhang, Y., Wang, G. (2020). Improving particle swarm algorithm to solve the replacement flow shop scheduling problem. Software, 173-186.

[5] Sevedali, E., Ruiz, R. (2014). Grey Wolf Optimizer. Advances in Engineering Software.

[6] James, C. Bean. (1994). Genetic Algorithms and Random Keys for Sequencing and Optimization. ORSA Journal of Computing, 6(2):86-112.

[7] Xie, F., Chen, X., Zhang, L. (2006) Realization of an effective dual-parameter sensor employing a single fibre Bragg grating structure. Optics and Lasers in Engineering, 1782-1789.

[8] Qu, P., Tang, X. (2022) Improved particle swarm algorithm for flexible shop scheduling problem. Mechanical Design and Manufacturing, 51-58.