Intelligent Scheduling of Distributed Displacement Pipeline Based on Hybrid Discrete Drosophila Optimization Algorithm

Pan Yuxia(✉) and Xie Guang

School of Information and Intelligence Engineering, University of Sanya, Sanya 572022, HaiNan, China
panyuxia12123@yeah.net

Absrtact. In order to solve the problem of long scheduling time and low efficiency of interval number representation scheduling method, a distributed replacement Pipeline Intelligent Scheduling Based on hybrid discrete Drosophila optimization algorithm is proposed. According to the distributed permutation pipeline scheduling problem, the coding method based on operation is adopted to make the algorithm suitable for solving the scheduling problem. The hybrid discrete Drosophila optimization algorithm is used to solve the batch pipeline scheduling problem with the maximum completion time as the goal. In order to balance the local search ability of the algorithm, the evolutionary mechanism is combined with cooperative learning among groups. Build a mathematical model to achieve efficient scheduling in the maximum completion time. The simulation results show that the scheduling time of this method is short, and the overall scheduling efficiency is higher than 80%, which has good scheduling effect.

Keywords: Hybrid discrete Drosophila · Optimization algorithm · Distributed displacement pipeline · Intelligent scheduling

1 Introduction

In the production and operation process of enterprises, there are often various uncertain factors, which lead to the original scheduling plan can not be executed normally. Therefore, the production scheduling problem in uncertain environment has a strong application value, which attracts much attention. Distributed permutation assembly line scheduling problem is widely used in metallurgy, plastics, chemical industry and other industrial production, which has important theoretical research background and practical application value [1, 2]. With the increasingly fierce competition in the market and the diversification of customer demand, multi variety small and medium-sized batch production mode occupies a dominant position [3]. Therefore, the research on batch pipeline scheduling problem has become the focus of academic and engineering circles. Each factory is a permutation flow shop. The processing time of distributed permutation assembly line is usually a definite value, but it is difficult to determine the processing

W. Fu et al. (Eds.): ICMTEL 2021, LNICST 387, pp. 77–90, 2021.
https://doi.org/10.1007/978-3-030-82562-1_7

time accurately in actual production. In the past, interval number was used to represent the uncertainty of process time, and then extended to generate interval number distributed permutation pipeline scheduling problem. When the processing time of an operation was interval number, jobs were allocated to multiple workshops reasonably, and then the scheduling index was optimized by reasonable sequencing [4]. Although the scheduling effect of this method is good, it is affected by the complexity problems such as large-scale, strong coupling and uncertainty, resulting in long scheduling time. Based on this problem, an intelligent scheduling method of distributed displacement pipeline based on hybrid discrete drosophila optimization algorithm is proposed. Drosophila optimization algorithm originated from the simulation of Drosophila foraging behavior, and has been successfully applied to solve mathematical function extremum, automatic picking scheduling problem, etc. its innovation lies in the use of hybrid discrete Drosophila optimization algorithm to solve the batch flow line scheduling problem with the goal of maximum completion time. Through multi operation collaborative search, the effective solution effect is achieved.

2 Distributed Permutation Pipeline Scheduling Problem

Distributed permutation pipeline scheduling is to study the processing process of a certain number $n(n > F)$ of jobs in $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$ number of displacement pipeline F factories with limited buffers. There is a job to be processed in the same factory, and each factory contains m machines. During the processing, the jobs should pass through each machine in turn, and the processing time of each job on each machine is greater than 0. Once the jobs are allocated to After a factory, it can not be assigned to other factories, and all operations of workpieces can only be completed in this factory. In addition, it is agreed that all workpieces are independent and processing can be started at 0:00. In the factory, once the order of the work pieces is determined, it will not change. In the buffer area, the work pieces follow the principle of first in first out. Preemption is not allowed. Once the operation starts, it must be completed and cannot be interrupted. The setting time of the machine and the moving time between operations are ignored [5–9].

In petrochemical production, semiconductor manufacturing and other production processes, the capacity of intermediate storage is often limited, most of which can be modeled as pfssp with buffer. More and more attention has been paid to permutation flow shop scheduling with finite buffers. With the development of globalization, cooperation between factories and annexation between enterprises have been very common, and distributed manufacturing mode is becoming more and more popular [10–15]. As an important part of intelligent factory, intelligent production should coordinate production, scheduling, logistics and management. Production scheduling is an important part of manufacturing industry, and logistics is also an essential link, which directly affects the efficiency and competitiveness of enterprises. In the actual production and transportation, for example, a certain kind of parts in Apple products should be purchased from various suppliers around the world and then transported to the distribution center; Taobao, Jingdong and other major stores order a certain kind of product from all over the world and then transport them to the distribution warehouse [16–20].

3 Local Search Based on Hybrid Discrete Fly Optimization Algorithm

The optimization algorithm of Drosophila mainly includes two parts: olfactory search and visual search. Olfactory search is divergent search, visual search is greedy iteration. The population of flies evolves through continuous iteration of two links. Because of the small control parameters, it is easy to implement.

3.1 Coding and Decoding

The hybrid discrete fly optimization algorithm is used, which is one of the individual representatives in the population $n \times m$. For the scale of the problem $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$, the solution is a coding arrangement. For example $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\} = \{3, 5, 4, 1, 2\}$, the first piece 3 is processed, followed by workpiece 5, and so on, and the last one is workpiece 2 [21–25].

One sort decoding is divided into two steps. The first step is to assign all the work pieces to the factory, generate a feasible scheduling scheme in each factory, and the second step is to transport the finished workpiece to the distribution center [26–31]. Here, a sort or solution is called an individual, and a sort in a factory is called a sub individual or individual sub sort. The first distribution rules are designed to arrive at the distribution center, so that the processing order (sub order) of the workpiece in the factory is consistent with that of the individual workpiece. In order to overcome the inconsistency between individual sorting and sub ordering caused by local search, we designed the mapping rule of reverse first arrival to distribution center, which makes probability model learn the information of work-piece sorting in excellent individuals more accurately. In the second stage, after the workpiece processing is completed, according to the weight of the workpiece and the load weight of the transport vehicle, the workpiece is reasonably allocated to the transport vehicle. In order to make the vehicle load as many workpieces as possible, a reasonable vehicle loading rules are designed. Assign the workpiece to each factory in sequence. For the work pieces to be allocated in the sequence, place them at the end of each factory respectively, calculate the completion time of the current factory and assign them to the factory that finished the work piece the earliest.

3.2 Population Initialization

The initial population includes indi_ Num individuals, one of which is determined by the heuristic method neh2_ The other individuals were randomly generated. Neh2_ IM is proposed to improve neh2 by idpfsp, and its steps are shown in Fig. 1.

As can be seen from Fig. 1, all jobs are arranged in descending order according to their total processing time, and the jobs are allocated in order of arrangement. The current jobs to be assigned are assigned to each position of each factory, and the sequence that makes the current minimum is reserved. Select the previous workpiece in the current position and assign it to each position of each factory, and keep the sequence that makes the current minimum. If there is no improved minimum sequence, the next job in the current position is selected and assigned to each position of each factory, and the current

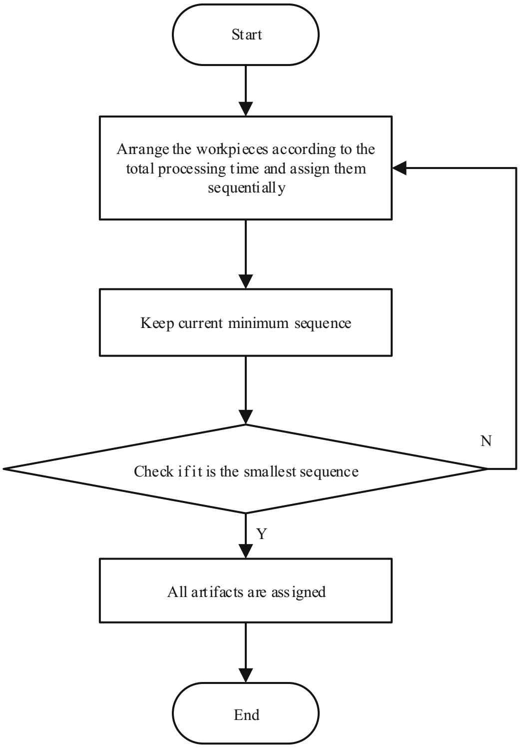

Fig. 1. Population initialization process

minimum sequence is retained until all jobs are assigned. The construction of individuals with good quality and the generation of initial population can guarantee both quality and diversity.

3.3 Sampling to Generate New Species Group

Each individual of the new population is generated by sampling the updated probability distribution model. Generally, roulette or tournament mechanism will be used in sampling, which will waste a lot of time. Half of the population individuals adopt direct sampling method, that is $P(gen+1)$, the job number on the i position in the individual ranking is directly according to the column with the largest value in each row element $p_i(gen+1) = \{p_{i,1}(gen+1), p_{i,2}(gen+1), \ldots, p_{i,n}(gen+1)\}$ in the probability matrix; the other half of the population individuals use roulette sampling. The process of generating new individual sequences by direct sampling is described as follows:

Step 1: order i = 1;
Step 2: read the i value of row 1 from the probability distribution matrix P(gen + 1)

$$p_i(gen+1) = \{p_{i,1}(gen+1), p_{i,2}(gen+1), \ldots, p_{i,n}(gen+1)\} \tag{1}$$

Step 3: compare the size of each element $P_i(gen+1)$ in to find the column where the largest element j is located;
Step 4: set the job number j on the position in the individual sorting as i, and then set the column $P(gen+1)$ in the probability distribution model j to 0, and $i = i+1$, then turn to step 2. Until the job number is selected at all positions in the individual sort.

3.4 Dual Mode Local Search Based on Handover Mechanism

In order to enhance the ability of local search, local search based on switching mechanism is designed for the optimal solution and suboptimal solution: if the optimal solution (or sub optimal solution continuous generation search) $k_{\max}$ in the population does not improve the performance, the next generation uses the sub optimal solution (or optimal solution) to search.

Local search is divided into two modes: L_{S1} and L_{s2}. L_{S1} uses a variety of local search operations based on the key plant and the factory with the minimum completion time to strengthen the fine search, including F_ S_{wap}, $S_{wap\_}$ F_k, Insert_ F_K, the first three operations are the same as the enhanced search phase in the population coordination stage_ F_K is executed as follows: two non adjacent jobs are randomly selected from the key plant, and then the sequence of jobs between them is reversed, and then the optimal reservation is achieved. The search depth of L_{S1} mode is controlled by parameters.

The steps of L_{s2} mode are as follows:

Step 1: randomly select the workpiece from the key factory and assign it to each position in the key factory, and keep the minimum sequence and position;
Step 2: randomly select the previous job or the next one in the current position of the workpiece, and insert it all the same to keep the smallest sequence.

The search depth ln of L_{s2} mode is controlled by adaptive parameters, and each search ensures that the selected jobs are different. The adaptive parameters ln are as follows:

$$\ln = \left\lceil \min\left(jn_{F_k}, 200 \times f/n\right) \right\rceil \tag{2}$$

In formula (2), jn_{F_k} denotes the maximum number of workpieces in the factory, n epresents the number of workpieces, f represents the factory, and $\lceil\rceil$ indicates rounding up.

In the process of local search execution, the LS1 search is performed for the individuals to be searched (the first generation adopts the optimal solution). If the individual's performance is improved, the next generation performs L_{s2} search, otherwise the next generation continues to perform L_{S1} search.

4 Design of Intelligent Scheduling Scheme for 3-distributed Permutation Pipeline

Through local search based on hybrid discrete Drosophila optimization algorithm, the parameters needed for intelligent scheduling of distributed displacement pipeline are obtained. All jobs in the distributed displacement pipeline are independent of each other and can be processed at zero time, and the machines are continuously available, that is, without considering the factors such as machine failure; at the same time, a machine can only process one workpiece, and a workpiece will not be processed by multiple machines at the same time; each workpiece can be assigned to any factory, and once the factory allocation is determined, it can not be changed. The sequence of workpieces processed on each machine in the same factory is the same, and the process to be processed for each workpiece is also the same; the time of moving the workpiece on the machine and the setting time of the machine are ignored.

4.1 Optimal Estimation Algorithm

According to the search results, design the optimization estimation process, as shown in Fig. 2.

The overall steps of optimization estimation are as follows:

Step 1: let *genMax* represent the maximum evolution algebra, set the population size *popsize*, buffer size *buffer_size*, learning rate α and the number of good individuals selected to update the probabilistic model N. When executing $FindBestN_{insert}\left(\pi_i^k(gen)\right)$, the continuous cumulative value *Insert_count* of high quality solution is not improved, which is the threshold value *Insert_count* of;
Step 2: initialize.
Let $gen = 0$, $Insert\_count\ = 0$, initialize the probability matrix $P(0)$ so that the value of each element is $1/n$. The 0 generation individuals $\pi_i(0)$ for $i = 1, \ldots,$ popsize are directly sampled from the initial probability model, and the subsequence of each individual is generated by allocation rules $\pi_i^k(0)$. According to the evaluation value of each individual, the best individual subsequence $\pi_{best}^k(0)$ in the current population is used to update the best individual subsequence π_{gbest}^k in the current population;
Step 3: using the N best previous $\pi(gen)$ individual update matrix $p(gen)$ of the old population, directly sampling the updated probability matrix $p(gen + 1)$ to generate the new species group with the size of popsize, and generate the sub ranking of each individual through the allocation rules $\pi_i^k(gen + 1)$. According to the evaluation value of each individual subsequence, the best individual subsequence $\pi_{best}^k(gen + 1)$ in the population is used to update the individual subsequence with the best history;
Step 4: implement mutation operator based on the best individual subsequence generated $\pi_{best}^k(gen + 1)$ by step 3 to generate individual subsequence int*erchange*;
Step 5: $\pi_{best}^k(gen + 1)$ map back to individual sorting $\pi_{best}(gen + 1)$;
Step 6: if gen < genMax, skip to step 2;
Step 7: output π_{gbest}^k.

4.2 Intelligent Scheduling Model of Distributed Permutation Pipeline

In order to reduce the waiting time of the machine tool, each workpiece is divided into several small batches according to the principle of equal distribution. After the small batch processing of each workpiece is completed, it can be transferred to the next

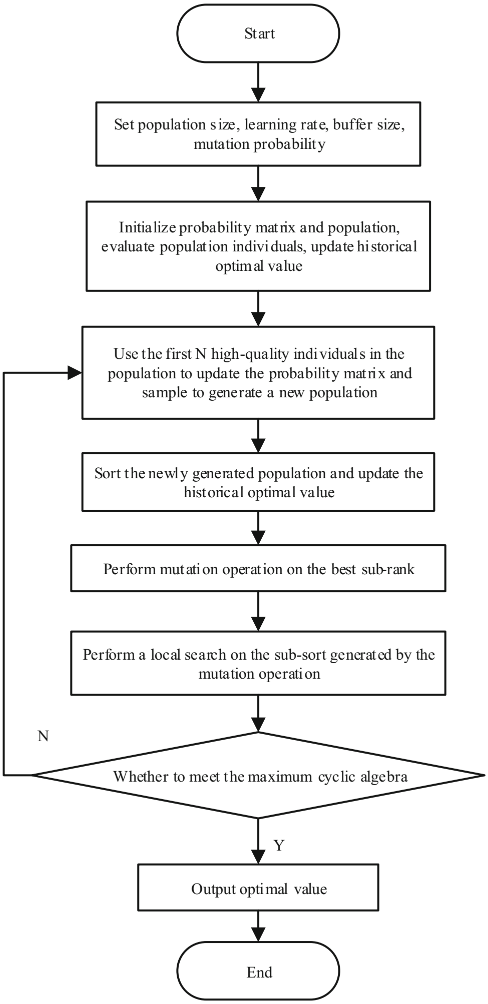

Fig. 2. Optimization estimation process

machine tool for processing, and the processing sequence of all small batches is the same. Suppose that the workpiece is processed in the order of machine tool 1 to, given a process $\pi = (\delta_1, \delta_2, \cdots, \delta_n)$, the workpiece j is divided into s_j small batches, which It_{ij} is the processing time of the small batch j of workpiece on the machine tool i, the completion time ct_{ijk} of the k small batch of the workpiece j on the machine tool i, and the completion time c_j of the workpiece, then the mathematical scaling model j is constructed as shown in Fig. 3.

Based on this model, the scheduling process of distributed permutation pipeline in maximum completion time is designed. Each step of the mathematical model in Fig. 3 represents: the completion time of the first small batch of the first workpiece on the first machine tool I; the completion time of the first small batch of the first workpiece on

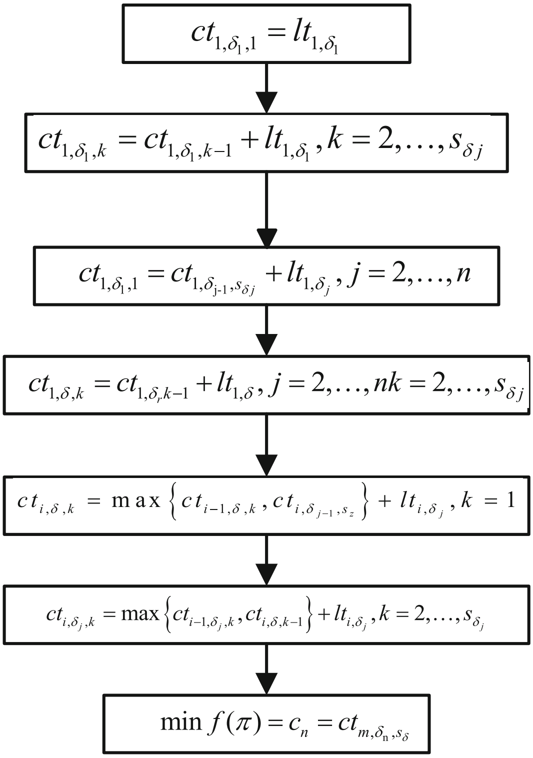

Fig. 3. Scheduling model

the first machine tool; the completion time of the first small batch of the first workpiece on the first machine tool; the completion time of the first small batch of the first job on the first machine tool; the completion time of the first small batch of the first job on the first machine tool; the completion time of the first small batch of the first workpiece on the first machine tool; the completion time of the first small batch of the first workpiece on the first machine; The completion time of the kth small batch of the *j*-th job on the first machine tool; the completion time of the first small batch of the *j*-th workpiece on the first machine tool; the completion time of the *k*-small batch of the *j*-th job on the i-machine tool; the goal of scheduling is to minimize the maximum completion time. According to the time of each step, the corresponding scheduling steps are completed to achieve efficient scheduling in the maximum completion time.

5 Simulation Experiment

5.1 Experimental Settings

Intel Core i5 processor, which runs on a 3.2 GHz CPU, is programmed in C +. The IDPFSP test suite is obtained by extending the DPFSP standard test suite.There are 12 kinds of combinations in the test set $n \times m$: $\{(20, 50, 100) \times 5\}$, $\{(20, 50, 100, 200) \times 10\}$, {(20, 50, 100, 200, 500) × 20}, each of which generates 10 different examples, and then considers 6 kinds of factory numbers, that is f = {2, 3, 4, 5, 6, 7}, all the problems determine the processing time.

5.2 Parameter Settings

Experimental design was used to investigate the influence of parameters on the performance of the algorithm. Ra14_18 was used to test the algorithm. 4 horizontal values were set for each parameter, as shown in Table 1.

Table 1. Parameter level values

Parameter	Level			
	1	2	3	4
Indi_num	100	150	200	250
SN	1	2	3	4
Is	100	150	200	250
kmax	15	20	25	30

According to the orthogonal table, each group of parameter combinations is run independently for 10 times, and the total evaluation of 500000 times is taken as the termination criterion. The average performance of the algorithm is taken as the response value. The results are shown in Table 2.

Table 2. Orthogonal table of parameters

Number	parameter				Response value
	Indi_num	SN	Is	k_{max}	
1	1	1	1	1	660.5
2	1	2	2	2	655.1
3	1	3	3	3	656.3
4	1	4	4	4	658.5
5	2	1	2	3	652.3
6	2	2	1	4	651.7
7	2	3	4	1	654.7
8	2	4	3	2	652.8
9	3	1	3	4	651.0

(*continued*)

Table 2. (*continued*)

Number	parameter				Response value
	Indi_num	SN	Is	k_{max}	
10	3	2	4	3	651.8
11	3	3	1	2	652.0
12	3	4	2	1	655.1
13	4	1	4	2	653.2
14	4	2	3	1	651.0
15	4	3	2	4	651.8
16	4	4	1	3	658.7

The average response value of each parameter at each level is calculated, and then the influence level of parameters on performance is determined. The influence trend of each parameter on performance is shown in Fig. 4.

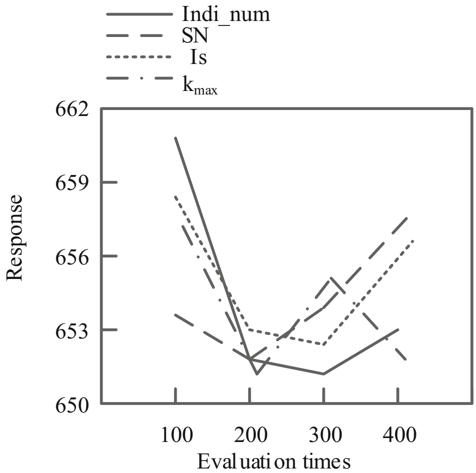

Fig. 4. Trends in the impact of parameters on performance

As shown in Fig. 4, India _ num has a greater impact on performance, followed by SN, Is, and kmax.

5.3 Simulation Results and Analysis

The interval number representation scheduling method and the hybrid discrete Drosophila optimal scheduling method are used to compare the scheduling time and efficiency. The results are shown as follows.

(1) Scheduling time
The scheduling time of different methods is compared and the results are shown in Tables 3 and 4.

Table 3. Interval number represents scheduling time of scheduling method

Scheduling node	1	2	3	4	5	6	7	8	9
1	0	0	0	3	3	4	5	5	5
2	0	0	0	4	3	5	6	6	3
3	0	0	0	5	2	5	5	5	3
4	0	0	0	4	2	4	6	5	4
5	0	0	0	3	2	5	4	6	3
6	0	0	0	4	0	3	3	4	5
7	0	0	0	3	3	4	5	5	3
8	0	0	0	3	2	5	4	4	4
9	0	0	0	4	2	1	4	4	2

Table 4. Scheduling time based on hybrid discrete Drosophila optimization scheduling method

Scheduling node	1	2	3	4	5	6	7	8	9
1	0	0	0	1	2	4	5	5	1
2	0	0	0	3	3	5	6	6	3
3	0	0	0	4	2	5	5	6	3
4	0	0	0	0	2	4	5	5	4
5	0	0	0	0	0	3	4	6	3
6	0	0	0	4	0	3	3	4	5
7	0	0	0	3	3	4	4	5	3
8	0	0	0	1	2	5	3	4	4
9	0	0	0	0	1	1	4	4	1

It can be seen from the above comparison results that the scheduling time of the scheduling method represented by interval number is different from the expected time, while the scheduling method based on hybrid discrete Drosophila is consistent with the expected time, and the error is 0.

(2) Dispatching efficiency
The results of comparative analysis of scheduling efficiency of different methods are shown in Fig. 5.

It can be seen from Fig. 5 that the overall scheduling efficiency of the scheduling method represented by interval number is less than 40%, while the overall scheduling

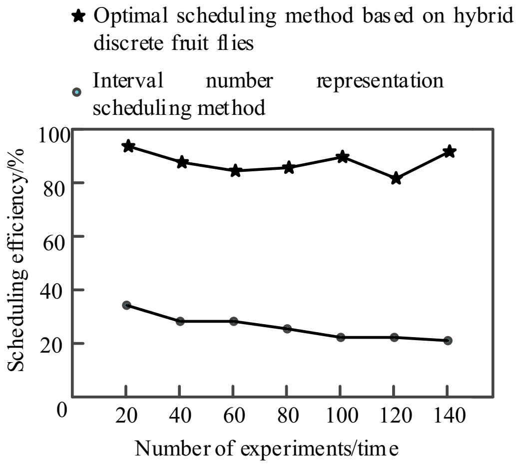

Fig. 5. Comparison results of scheduling efficiency of different methods

efficiency of the scheduling method based on hybrid discrete Drosophila optimization is higher than 80%, which has good scheduling effect.

6 Conclusion

A hybrid discrete Drosophila optimization scheduling algorithm for interval number distribution permutation pipeline problem is proposed. The effectiveness and efficiency of the algorithm are verified by a large number of simulation examples and data analysis. The innovation of the algorithm lies in the combination of improved heuristic rules and random methods to initialize the population, and a multi operation cooperation link based on probability selection mechanism is designed to give full play to the performance of each operator.

Future research work will focus on the extension of other uncertain problems, optimization of other scheduling indicators, design of adaptive algorithm and collaborative optimization of multiple scheduling objectives.

References

1. Liao, Q., Zhang, H., Xu, N., et al.: A MILP model based on flowrate database for detailed scheduling of a multi-product pipeline with multiple pump stations. Comput. Chem. Eng. **117**(2), 63–81 (2018)
2. Asl, N.B., Mirhassani, S.A.: Benders decomposition with integer sub-problem applied to pipeline scheduling problem under flow rate uncertainty. Comput. Chem. Eng. **123**(6), 222–235 (2019)
3. Qin, H., Chen, W., Cao, B., Zeng, M., Li, J., Peng, Y.: DIPS: dual-interface dual-pipeline scheduling for energy-efficient multihop communications in IoT. IEEE Internet Things J. **6**(1), 718–733 (2019). https://doi.org/10.1109/JIOT.2018.2855695

4. Chang, X., Xu, X., Yang, D.: Pipeline scheduling based on constructive interference in strip wireless sensor networks. Comput. Mater. Continua **64**(1), 193–206 (2020)
5. Moradi, S., Mirhassani, S.A., Hooshmand, F.: Efficient decomposition-based algorithm to solve long-term pipeline scheduling problem. Petrol. Sci. **16**(5), 1159–1175 (2019)
6. Amine, A., Mouhoub, M., Ait Mohamed, O., et al.: Optimal Scheduling of Multiproduct Pipeline System Using MILP Continuous Approach. In: IFIP Advances in Information and Communication Technology Computational Intelligence and its Applications, vol. 522 (2018). https://doi.org/10.1007/978-3-319-89743-1(36):411-420
7. Fu, W., Liu, S., Srivastava, G.: Optimization of big data scheduling in social networks. Entropy **21**(9), 902 (2019)
8. Liu, S., Li, Z., Zhang, Y., et al.: Introduction of key problems in long-distance learning and training. Mob. Netw. Appl. **24**(1), 1–4 (2019)
9. Liu, S., Liu, D., Srivastava, G., et al.: Overview and methods of correlation filter algorithms in object tracking. Comp. Intell. Syst. (2020). https://doi.org/10.1007/s40747-020-00161-4
10. Krishnadas, G., Kiprakis, A., Sciubba, E.: A machine learning pipeline for demand response capacity scheduling. Energies **13**(7), 1848 (2020)
11. Qiu, S., Wang, S., Xiao, C., Ge, S.: Assessment of microalgae as a new feeding additive for fruit fly Drosophila melanogaster. Sci. Total Environ. **667**, 455–463 (2019). https://doi.org/10.1016/j.scitotenv.2019.02.414
12. Yang, X., Han, Y., Mu, Y., et al.: Multigenerational effects of cadmium on the lifespan and fertility of Drosophila melanogaster. Chemosphere **245**(Apr), 125533.1-125533.7 (2020)
13. Gärtner, S., Hundertmark, T., Nolte, H., Theofel, I., Eren-Ghiani, Z., Tetzner, C., Duchow, T., Rathke, C., Krüger, M., Renkawitz, R.: Stage-specific testes proteomics of Drosophila melanogaster identifies essential proteins for male fertility. Eur. J. Cell Biol. **98**(2–4), 103–115 (2019). https://doi.org/10.1016/j.ejcb.2019.01.001
14. Hsieh, Fu-Shiung., Guo, Yi-Hong.: A discrete cooperatively coevolving particle swarm optimization algorithm for combinatorial double auctions. Appl. Intell. **49**(11), 3845–3863 (2019). https://doi.org/10.1007/s10489-019-01556-8
15. Lakshman, A.A., et al.: Selection for timing of eclosion results in co-evolution of temperature responsiveness in drosophila melanogaster. J. Biol. Rhyth. **34**(6), 596–609 (2019)
16. Qiu, B., Guo, J., Li, X., et al.: Discrete-time advanced zeroing neurodynamic algorithm applied to future equality-constrained nonlinear optimization with various noises. IEEE Trans. Cybern. (99), 1-14 (2020)
17. Wu, Q., Zhang, R.: Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts. IEEE Trans. Commun. **68**(3), 1838–1851 (2020)
18. Shao, Z., Pi, D., Shao, W.: A novel multi-objective discrete water wave optimization for solving multi-objective blocking flow-shop scheduling problem. **165**(FEB.1), 110–131 (2019)
19. Zhang, J., You, K., Basar, T.: Distributed discrete-time optimization in multiagent networks using only sign of relative state. IEEE Trans. Autom. Control **64**(6), 2352–2367 (2019)
20. Teng, Y., Yang, L., Song, X., et al.: An augmented Lagrangian proximal alternating method for sparse discrete optimization problems. Numer. Algor. **83**(3), 833–866 (2020)
21. Li, Y., Yang, W., He, P., et al.: Design and management of a distributed hybrid energy system through smart contract and blockchain. Appl. Energy **248**(15), 390–405 (2019)
22. Spencer, A.A.M.S., Luciano, S., Mario, M.: Analysis and design of high-efficiency hybrid high step-up DC-DC converter for distributed PV generation systems. IEEE Trans. Ind. Electron. (5), 1 (2018)
23. Zhang, L., Liu, W., Qi, B.: Innovation design and optimization management of a new drive system for plug-in hybrid electric vehicles. Energy **186**, 115823.1-115823.19 (2019). https://doi.org/10.1016/j.energy.2019.07.153
24. Zkik, K., Hajji, S.E., Orhanou, G.: Design and implementation of a new security plane for hybrid distributed SDNs. J. Commun. **14**(1), 26–32 (2019)

25. Han, X., Dong, Y., Yue, L., Quanxi, X.: State transition simulated annealing algorithm for discrete-continuous optimization problems. IEEE Access **7**, 44391–44403 (2019). https://doi.org/10.1109/ACCESS.2019.2908961
26. Kamalakis, T., Dogkas, L., Simou, F.: Optimization of a discrete multi-tone visible light communication system using a mixed-integer genetic algorithm. Optics Commun. **485**(1), 126741 (2020)
27. Wang, L., Guohua, W., Gao, L.: Thematic issue on "advanced intelligent scheduling algorithms for smart manufacturing systems." Memetic Comput. **11**(4), 333–334 (2019). https://doi.org/10.1007/s12293-019-00297-y
28. Kamalakis, T., Dogkas, L., Simou, F.: Optimization of a discrete multi-tone visible light communication system using a mixed-integer genetic algorithm. Optics Commun. **485**(8), 126741 (2020)
29. Rui, L., Qin, Y., Li, B., et al.: Context-based intelligent scheduling and knowledge push algorithms for ar-assist communication network maintenance. Comput. Model. Eng. Sci. **118**(2), 291–315 (2019)
30. Bruballa, E., Wong, A., Rexachs, D., et al.: An intelligent scheduling of non-critical patients admission for emergency department. IEEE Access (99), 1 (2019)
31. Yuan, L.: Scheduling analysis of intelligent machining system based on combined weights. IOP Conf. Ser. Mater. Sci. Eng. **493**(1), 12146 (2019)