A Grey Wolf Optimization Algorithm for Stochastic Multi-Objective Disassembly Line Balancing Problem

Hanshuo Hui, Fuwei Wang, Weimin Wang, Ze Zhang, Meiqin Zuo and Fengting Li

October 8, 2021

A Grey Wolf Optimization Algorithm for Stochastic Multi-objective Disassembly Line Balancing Problem

HanShuo Hui[1], FuWei Wang[1], WeiMin Wang[2],Ze Zhang[2], MeiQin Zuo[3], FengTing Li[3]
College of Computer and Communication Engineering, Liaoning Petrochemical University, Fushun, 113001, China
754858783@qq.com, wangfw4048@163.com, willing50@qq.com, 846780589@qq.com, 1397034443@qq.com, 1401442170@qq.com

***Abstract*—Outdated waste products need to obtain high profits in a short period of time in the process of recycling and disassembly. Uncertainty factor of product wear aging and disassembly sequence increase the risk of demolition failure, which involves the disassembly line balance problem(DLBP). This paper set up a model for maximizing disassembly profit and minimizing failure based on the AND/OR graph. For this purpose, we present a random multi-objective discrete Grey Wolf optimization algorithm. By radio example experiments, we found that the algorithm was superior to NSGA-II and MOEA/D, which proved the effectiveness and feasibility of the method.**

***Keywords*—Disassembly line balancing problem, disassembly failure cost, disassembly profit, disassembly time, multi-objective optimization.**

I. INTRODUCTION

In recent years, the world economy and science technology develop at a faster speed. As a result, the life cycle of daily product and industrial equipment is gradually shortened. For industrial equipment, each update iteration is a not small cost. Therefore, the recycling products can reduce the expenditure of enterprises and factories on the purchase of products and equipment, and also reduce the increase of product waste, which also plays an important role in protecting our ecological environment. So, recycling of products can reduce costs for businesses and factories. At the same time, it also reduces the production of waste. It also plays a right role in protecting the ecological environment.

The pivotal step in product recycling is disassembly. In the process of disassembly, there is a DLBP. Some scholars have studied DLBP[1-3]. Guo *et al*[4] proposed a dictionary multi-objective scatter search (SS) algorithm to optimize disassembly time, energy consumption and profit. Fu *et al*[5] put a fruit fly optimization algorithm to solve stochastic multi-objective problem. Zhao *et al*[6] built a mixed integer linear program to describe the problem, and propose a memetic algorithm to obtain the Pareto solution in order to reduce the number of workstations and the total setting time. Qin *et a*l[7] consider resource constraints and failures factor and propose a discrete migratory bird optimize algorithm. Bentaha *et a*l[8] takes the optimization of income as the goal, establishes a stochastic programming model and an accurate solution method. Tan *et al*[9] proposed a two-stage method based on spitting search(SS) and mixed integer programming(MIP) to optimize the disassembly completion time and energy consumption. Tian *et al*[10] proposes some chance constrained disassembly cost programming models from the perspective of random planning and proposes two hybrid intelligent algorithms to solve the model. There are some failure factors and constraint of priority disassembly sequence in disassembly process, thus produced Stochastic Multi-objective Disassembly sequencing line balancing Problem (SM-DSBP). We propose a method to maximize the disassembly profit and minimize disassembly time. Then we design a Stochastic Multi-objective-discrete Grey Wolf Optimization(SMGWO) algorithm to solve SM-DSBP. Compare with the existing research, this paper makes the following improvements:

1) It propose SM-DSBP based on an AND/OR. Under the constraints of the priority of disassembly task and the

cost of disassembly failure, we establish a stochastic programming model to optimize disassembly profit and time.

2) To solve the problems in this study, the Grey Wolf algorithm was improved. The Monte Carlo stochastic simulation method was combined with the PPX[11] operator and the PBM operator, and the PPX operator was improved to some extent.

3) Through the disassembly experiment of the radio, we select two classic task sequence optimization algorithms NSGA-II[12], MOEA/D[13] to compare with SMGWO. Then, the results of inverted generational distance(IGD) are compared to verify the validity of SMGWO.

The rest of this paper is divided into four parts: the second part describes the problem, proposes the SMGWO algorithm for three parts, the fourth part is the experiment and results, and the last part is the conclusion of this paper.

II. Problem description

A. Problem description

We list a module graph of teakettle in Fig.1 to help us better describe the problem. Based on the disassembly priority of each module in Fig.1, we construct the AND/OR module of teakettle in Fig.2.

Let's take an example from Fig.2, we identify these modules by integer with angle brackets. Then,we use integer with round brackets to indicate dismantle task index. Modules connect via a directed edge with a task index. In Fig.2, we can decompose <2>ABCEFG to get <5>BCEFG and <12>A by task (3). That is, module <2> is the parent of module <5> and <12>, module <5> and <12> is the son of module <2>. Therefore, there is an AND relationship between module <5> and module <12>. If the task (3) has been performed, it can not continue to perform the task (4). Therefore, there is an OR relationship between task(3) and task(4). On the basis of Fig.2, we can aim at DLBP to establish the following two matrices:

1) A task priority matrix $A = [a_{ij}]$:

$$a_{ij} = \begin{cases} 1, \text{if task } j \text{ is performed task } i; \\ -1, \text{if task } i \text{ and } j \text{ conflict with each other;} \\ 0, \text{otherwise;} \end{cases}$$

2) A matrix associated with task and module $B = [b_{ni}]$:

$$b_{ni} = \begin{cases} 1, \text{if subassembly } n \text{ is obainted by task } i; \\ -1, \text{if subassembly } n \text{ is disassembled by task } i; \\ 0, \text{otherwise;} \end{cases}$$

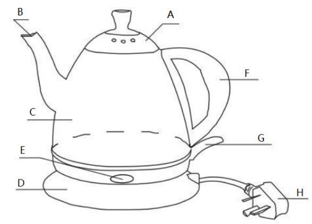

Fig.1. Module graph of a teakettle

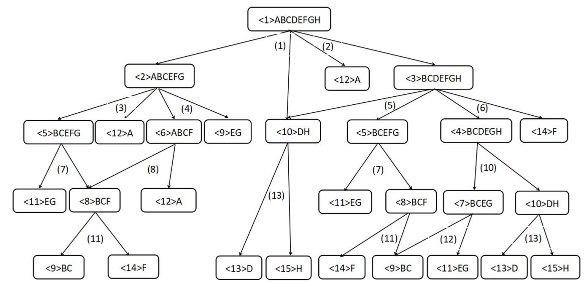

Fig.2. The AND/OR graph of teakettle

B. Notation definition

1) n = 1,2,..,N index of module, N is module max number.

2) i,j = 1,2,..,I index of task, I is task max number.

3) k = 1,2,3,4,...,K index of workstation,K is the max number of workstation.

4) p_n: the profit from dismantling module n.

5) t_i^d: the running time of task i.

6) t_{ij}^s: the set time that task j runs after task i.

7) q_i^d: the cost per second of running task i.

8) q_{ij}^s: the cost per second that task j runs after task i.

9) q_k: the cost of a workstation for one cycle.

10) T_c: the time of a workstation for one cycle.

11) r_{ij}: the probability of failure that task j runs after task i.

12) F_c: the max cost of dismantling failure.

13) υ: the total cost of failure to perform the dismantling task is less than the minimum probability of the expected acceptable maximum cost of failure.

14) A: the priority matrix of task.

15) B: the association matrix of tasks and modules.

16) a_{ij}: the priority relationship between tasks i and j in A

17) b_{ni}: the disassembly relationship between module n and taks j in B

Decision variable:

1) x_i:if task i is executed,$x_i = 1$,otherwise $x_i = 0$.

2) y_{ij}:if task j is executed after task i,$y_{ij} = 1$otherwise $y_{ij} = 0$.

3) z_{ik}:if workstation k perform task i,$z_{ik}=1$,otherwise $z_{ik}= 0$.

4) u_k:if workstation k is working, $u_k = 1$ otherwise $u_k = 0$.

C. Mathematical Model:

$$\max f_1 = E\left\{\sum_{i=1}^{I}\sum_{n=1}^{N} b_{ni} p_n x_i - \sum_{i=1}^{I} t_i^d q_i^d x_i - \sum_{i=1}^{I}\sum_{j=1}^{I} t_{ij}^s q_{ij}^s y_{ij} - \sum_{k=1}^{K} q_k u_k\right\}$$
(1)

$$\min f_2 = E\left(\sum_{i=1}^{I} t_i^d x_i + \sum_{i=1}^{I}\sum_{j=1}^{I} t_{ij}^s y_{ij}\right) \quad (2)$$

$$\sum_{i=1}^{I} x_i \geq 1 \quad (3)$$

$$x_j = \sum_{i=1}^{I} y_{ij} \leq 1, j = 1,2,3..,I \quad (4)$$

$$x_i = \sum_{k=1}^{K} z_{ik} \leq 1, k = 1,2,...,K \quad (5)$$

$$u_k - \sum_{i=1}^{I} x_i z_{ik} \geq 0, k = 1,2,...,K \quad (6)$$

$$a_{ij} - y_{ij} \geq 0, \quad i,j = 1,2,...,J \quad (7)$$

$$z_{jk} - \left(s_{ij} + \sum_{k=1}^{K} z_{ik}\right) \geq 0, i,j = 1,2,..,I, k = 1,2,..,K \quad (8)$$

$$E\left(\sum_{i=0}^{I}\sum_{j=1}^{I} (x_j z_{jk} t_j^d + x_i y_{ij} z_{ik} t_{ij}^s)\right) \leq T_c, k = 1,2,..,K \quad (9)$$

$$\upsilon \leq \Pr\left(F_c \geq \sum_{k=1}^{K}\sum_{i=0}^{I}\sum_{j=1}^{I} (t_j^d q_j^d x_j + t_{ij}^s q_{ij}^s y_{ij}) z_{jk} q_k r_{ij}\right) \quad (10)$$

$$x_i, y_{ij}, z_{ik}, u_k \in \{0,1\}, i,j = 1,2,..,I, k = 1,2..,K \quad (11)$$

The objective (1) represents the max profit of product disassembly. The objective (2) represents the min time of product disassembly. (3) represents must be a task to execute. (4) represents that task j cannot be executed repeatedly. (5) represents that task i cannot be duplicated on the workbench. (6) represents that if workstation k is already on, a task must be executed. (7) represents that the execution order of task i,j must satisfy the condition in *A*. (8) represents that workstations can be assigned to the next task only after the previous task has completed execution. (9) represents that the working hours of the workstation must be within the one cycle time. (10)represents the probability constraint that the failure cost of performing the disassembly task is less than the acceptable failure cost of disassembly. (11)constrains the index range of the task and workstation, then gives two optional values of the decision variables.

III. OPTIMIZED GRAY WOLF ARITHMETIC

The basic Gray Wolf algorithm divides the wolf pack into four classes alpha, beta, delta and omega. Each wolf has its own duties. Organization and discipline are far more important than strength in wolf pack, so the first level alpha wolf is the manager in wolf pack. In the second level, Beta wolves obey Alpha wolves and give missions to the other wolves, acting as a minister in the pack. The three level is delta wolves, they obey the orders of alpha and beta. The lowest omega wolves obey the other levels, but without omega wolves, the pack will fight among itself. The basic Grey Wolf algorithm has a good advantage in finding the sequence combination problem. On this basis, we combined with the stochastic simulation to optimize it to solve the SM-DSBP in this paper.

A. Initial Population:

To solve the problem studied, we design an individual into two parts. The first part $\pi'=\{\pi'_1,\pi'_2,..,\pi'_I\}$ is the execution sequence of the disassembly task, π'_i represents the sequence number of the task. The second part $\pi''=\{\pi''_1,\pi''_2,..,\pi''_I\}$ indicates whether to execute the relative task in π', if $\pi''_i = 1$, then perform the task π'_i,else if $\pi''_i = 0$, do not perform. After generating new individuals, we need to run a Monte Carlo simulation to evaluate the random simulation and then update all the generated non-dominated solutions using the Pareto idea.

B. Solving Process:

In the basic idea of gray Wolf algorithm, a parameter α is set to control global and local search, $\alpha = 1- \lambda_c / \lambda_m$, λ_c and λ_m represent the current and overall evaluation index, respectively. In order to choose better solution in DLBP, we apply the rank and crowding distance methods[14] to SMGWO. In order to optimize the generated individual, we use the priority-preserving crossover operator PPX and the location-based variation PBM, PPX and PBM are given in Algorithms 1 and 2, respectively.

Algorithm 1: PPX.

Input: $\pi_\alpha=(\pi_\alpha^{'},\pi_\alpha^{''})$,$\pi_\beta=(\pi_\beta^{'},\pi_\beta^{''})$**and** $\pi_\chi=(\pi_\chi^{'},\pi_\chi^{''})$

Output:$\pi_\delta=(\pi_\delta^{'},\pi_\delta^{''})$

Begin

Generate two integer number $\lambda = 0$ and $\mu = I$-1.

for(i = 1 to I)

if($\pi_\chi^{'}(i) = 0$)

$\pi_\delta^{'}(\lambda) = \pi_\alpha^{'}(0)$, $\pi_\delta^{''}(\lambda) = \pi_\alpha^{''}(\lambda+I-1)$, λ++

Delete $\pi_\alpha^{'}(0)$ **from** $\pi_\alpha^{'}$ **and** $\pi_\beta^{'}$

else if($\pi_\chi^{'}(i)$=1)

$\pi_\delta^{'}(\mu) = \pi_\beta^{'}(I-1)$, $\pi_\delta^{''}(\mu) = \pi_\beta^{''}(\mu+I-1)$, μ--

Delete $\pi_\beta^{'}(I-1)$ **from** $\pi_\alpha^{'}$ **and** $\pi_\beta^{'}$

end if

end for

End.

Algorithm 2: PBM.

Input: $\pi_\gamma=(\pi_\gamma^{'},\pi_\gamma^{''})$

Output:$\pi_\gamma=(\pi_\gamma^{'},\pi_\gamma^{''})$

Begin

Generate a random integer number r_f on an interval[1,3] then generate three random integer number pos_1, pos_2 and pos_3 on an interval[0, I].

if(r_f= 1)

swap $\pi_\gamma^{'}(pos_1)$ and $\pi_\gamma^{'}(pos_2)$

else if(r_f=2)

if($\pi_\gamma^{''}(pos_3)$=0) $\pi_\gamma^{''}(pos_3) = 1$

else $\pi_\gamma^{''}(pos_3) = 0$

end if

else if(r_f=3)

swap $\pi_\gamma^{'}(pos_1)$ and $\pi_\gamma^{'}(pos_2)$

if($\pi_\gamma^{''}(pos_3)$=0) $\pi_\gamma^{''}(pos_3) = 1$

else $\pi_\gamma^{''}(pos_3) = 0$

end if

end if

End.

C. Stochastic Simulation Evaluation:

Monte Carlo simulation method[15] is a method for statistical experiments and random sampling. In this study, the disassembly time t_i^d and setting time t_{ij}^s are different between tasks, which are randomness and uncertainty. So we need to use stochastic simulation to evaluate objective function. We use N and F_c to identify the time and failure cost, and give a simulation process in Algorithm 3.

Algorithm 3: The Stochastic simulation.

Input:a original solution λ

Output:a new solution λ

Begin

for (i = 1 to N)

Generate a sample based on the corresponding random distribution.The sample contains t_i^d and t_{ij}^s.Based on this sample to calculate f_1 , f_2 and failure cost.

end for

$N^{'} = [\upsilon \cdot N]$,$N_m$= $N^{'}$-th largest failure cost sample .

if (N_m's failure cost < F_c)

λ = objective values of N.

Return new solution λ

else

Return null.

End.

IV. EXPERIMENT

A. Case study

In this paper, we choose a radio example and two classical algorithms NSGA-II, MOEA/D to testify the feasibility of the algorithm SMGWO. In Fig.5 we show a radio AND/OR graph, NSGA-II can rapidly find Pareto solution and maintain population diversity. MOEA/D has faster convergence and lower calculation complex rate. ALL algorithms are implemented in VisualStudio 2019 and runs on an Intel(R) Core(TM) i7 CPU (2.6GHz/8.00GB RAM) PC windows 10 operating system. This case has 29 tasks and 29 modules,for SM-DLPB, we set the parameter as:F_c=60,υ=0.95,q_k= 70,T_c= 80.

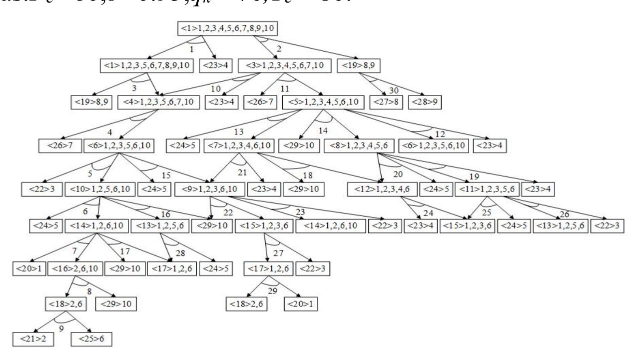

Fig.5. The AND/OR graph of radio[16]

We run each of these three algorithms twenty times and select twelve Pareto non-dominant solutions to list in TABLE I. f_1 represents disassembly profit, f_2 represents disassembly time, F_c represents disassembly failure cost.

TABLE I. THE DISASSEMBLY TASK SEQUENCE AND OBJECTIVE FUNCTION OF THE RADIO ARE SOLVED BY SMGWO

	Disassembly task sequence	f_1	f_2	F_c
1	1 3 4 15 23 17 29	567.91	84.921	35.92
2	1 3 4 15 22 27 29	644.32	108.69	35.46
3	1 3 4 30 15 22 27 29	709.39	130.74	42.34
4	1 3 4 15 22 30 27 29 9	756.89	141.09	41.85
5	1 3 4 15 23 17 30 29	610.23	101.90	38.90
6	2 11 13 21 22 30 27 29 9	782.93	167.90	37.89
7	2 11 13 21 22 27 30 29 9	813.26	171.66	42.30
8	2 11 13 21 22 27 29	725.19	135.69	31.33
9	2 11 14 19 26 28 30 29	705.68	124.70	36.31
10	2 11 14 19 26 30 28 29	679.17	122.70	33.01
11	2 11 13 21 30 22 27 29	790.38	153.92	38.77
12	2 11 14 19 26 28 29	604.55	107.76	36.77

To evaluate the overall superiority of the three algorithms,we choose following three evaluation indexes:

1) Inverted Generational Distance (IGD): IGD[17] is often used to evaluate the convergence and distribution of the algorithm. The value calculated by IGD is smaller, the overall property of arithmetic better. The IGD calculation results are given in Table II.

2) Pareto solution number: It helps us to compare the distribution of the Pareto solutions obtained by the three algorithms. We show the number of Pareto solution for three algorithms in Fig.6.

3) Generate a single Pareto Solution CPU Running time: We use it to compare the efficiency of the Pareto solutions produced by the three algorithms. We show the CPU running times of algorithms to generate a single Pareto solution in Fig.7.

B. Analysis of experimental results

1) According to the data in Table.II, the mean and std value of SMGWO is better than NSGA-II and MOEA/D. This show that SMGWO has better convergence and distribution.

2) In Fig.6, we can find that the number of Pareto optimal solutions of SMGWO is significantly more than that of NSGA-II and MOEA/D, this indicates that the Pareto solution obtained by SMGWO has superior distribution, the decision-makers can make more choices.

3) In Fig.7, generate a single solution CPU running time of SMGWO is close to NSGA-II and MOEA/D in the SM-DSBP.

TABLE II. COMPARISON OF THREE ALGORITHMS VIA IGD-METRICBY

Algorithm	IGD-metric	
	mean	std
SMGWO	0.0347	0.000089
NSGA-II	0.0475	0.000124
MOEA/D	0.0653	0.000645

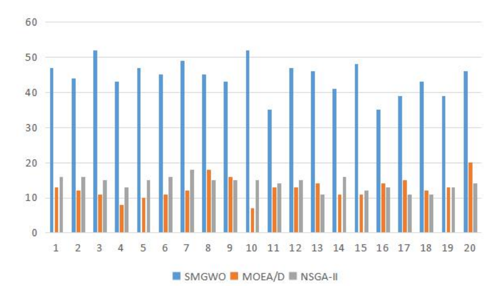

Fig.6. Solutions number of three agorithms at each run

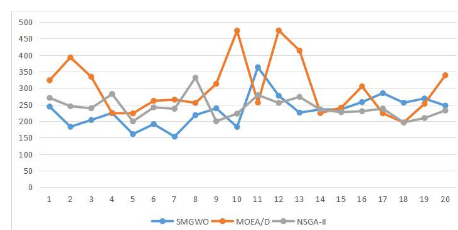

Fig.7. CPU running time of three agorithms generate a single Pareto solution(Unit:ms)

V. CONCLUSION

This study takes into account uncertain disassembly failure factors such as worn and aging of products. We propose a DLBP model to optimize the total disassembly profit and time. Then, an SMGWO algorithm based on AND/OR graph and stochastic simulation is designed to solve this problem. Compare the experiment results of the three algorithms that SMGWO is superior to NSGA-II and MOEA/D in SM-DSBP.In future work, we will study other types of disassembly line problems[18-23] and propose more advantageous algorithm models to help us solve them.

REFERENCES

[1] X. W. Guo, M. C. Zhou, F. Alsokhiry, and K. Sedraoui, “Disassembly sequence planning: a survey,” *IEEE/CAA J. Autom. Sinica*, 2020. DOI: 10.1109/JAS.2020.1003515

[2] X. W. Guo, S. X. Liu, M .C. Zhou, L. Qi, “Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets,” *IEEE Trans. Cyber.*, vol. 46, no. 11, pp. 2435-2446, 2016.

[3] X. W.Guo, M. C. Zhou, S. X. Liu, L. Qi, “Multi-resource Constrained Selective Disassembly with Maximal Profit and Minimal Energy Consumption”, *IEEE Trans. Autom. Sci. Eng.*, 2020, doi: 10.1109/TASE.2020.2992220.

[4] X. W. Guo,M. C. Zhou, S. X. Liu, L. Qi, “Lexicographic Mult iobjective Scatter Search for the Optimization of Sequence-Depe ndent Selective Disassembly Subject to Multiresource Constraint s” *IEEE Transactions on Cybernetics*., vol.50, no.7, pp.3307-33 17.2020.

[5] Y. P. Fu, M. C. Zhou, X. W. Guo, L. Qi, Stochastic multi-objective integrated disassembly-reprocessing-reassembly scheduling via fruit fly optimization algorithm, *Journal Clean Product*, 2020.

[6] Z. Y. Zhao, S.X. Liu, M.C. Zhou and A. Abusorrah, "Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem," *IEEE/CAA Journal of Automatica Sinica*, doi: 10.1109/JAS.2020.1003539.

[7] G. B. Qin, X.W.Guo, M.C.Zhou, S.X.Liu, L.Qi, “Multi-objectiv e discrete migratory bird optimizer for stochastic disassembly li ne balancing problem” 2020.*IEEE.International.Conference.on.Sys tems,Man.and.Cybernetics(SMC)*2020,doi:10.1109/SMC42975.2020. 9283371

[8] M. L. Bentaha, A. Dolgui, O. Battaïa, R. J. Riggs, J. Hu, “Profitoriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty,” *Int. J. Prod.* Res., 2018, doi: 10.1080/00207543.2017.1418987.

[9] Y. Y. Tan, M. C. Zhou, Y. Zhang, X. W. Guo, Liang Qi, Yanhong Wang, “Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting,”*IEEE Trans. Autom. Sci. Eng., Accept,* 2020.

[10] G.D.Tian, M.C.Zhou, J.W.Chu, “A chance constrained programming approach to determine the optimal disassembly sequence,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 4, pp. 1004-1013, 2013.

[11] E. Kongar and S. M. Gupta, “Disassembly sequencing using genetic algorithm,” *Int. J. Adv. Manuf. Tech.*, vol. 30, no. 5-6, pp. 497-506, 2006.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evolut. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.

[13] Q. F. Zhang and H. Li, “MOEA/D: A multiobjective evolutionary algorithm based on decomposition,” *IEEE Trans. Evolut. Comput.*, vol. 11, no. 6, pp. 712-731, 2007.

[14] H. T. Geng, H. J. Li, Y, G, Zhao, Z. P. Chen, “Improved NSG A-II algorithm based on adaptive hybrid non-dominated individ ual sorting strategy,”*J.Comput.Appl.*,vol.36,no.5,pp.1319-1324+134 0,2016.

[15] Y. P. Fu, H. F. Wang, M. Huang,“Integrated scheduling for a d istributed manufacturing system: a stochastic multi-objective mo del,”*Enterp.Inform.Syst.*,2018,doi:10.1080/17517575.2018.1545160.

[16] F. T. Altekin, L. Kandiller, N. E. Ozdemirel, “Profit-oriented-dis assembly-line balancing,” *Int. J. Prod. Res.*,vol.46,no.10,pp.2675 –2693, 2008.

[17] H. T. Geng, H. J. Li, Y, G, Zhao, Z. P. Chen, “Improved NSGA-II algorithm.based.on.adaptive.hybrid.non-dominated.individual.sortin g strategy,”*J.Comput.Appl.*,vol.36,no.5, pp.1319-1324+1340, 2016.

[18] J.Zhao,S.X.Liu,M.C.Zhou,X.W.Guo,and.L.Qi,“Modified.cuckoo.sea rch.algorithm.to.solve.economic.power.dispatch.optimization.proble ms,” *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp. 794-806, 2018.

[19] JZ Li, GJ Liu, CG Yan and CJ Jiang, “Robust Learning to Rank Based On Portfolio Theory and AMOSA Algorithm[J]” *IEEE Transaction on Systems, Man and Cybernetics: System*, 2017,47(6):1007-1018

[20] SQ Wang, Aorigele, GJ Liu, SC Gao, “A hybrid discrete imperialist competition algorithm for fuzzy job-shop scheduling problems[J]” *IEEE Access*, 2016, 4: 9320-9331

[21] X. Zuo, et al., “Optimizing Hospital Emergency Department Layout via Multi-objective Tabu Search,” *IEEE Transactions on Automation Science and Engineering*, 16(3), pp. 1137-1147, July 2019;

[22] “Multi-model based PSO method for burden distribution matrix optimization with expected burden distribution output behaviors,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1506-1512, Nov. 2019;

[23] Zhu, H., et al. “Dandelion algorithm with probability-based mutation.” *IEEE Access*, 7, 97974-97985, 2019.