Int j simul model 17 (2018) 3, 512-520
ISSN 1726-4529 Original scientific paper

# SIMULATION OF FLOW LINE SCHEDULING OF PRODUCTION ENTERPRISES BASED ON IMPROVED ARTIFICIAL FISH SWARM ALGORITHM

Lin, C.[*] & Yang, B.[**,#]

[*] Research Center for Enterprise Management, Chongqing Technology and Business University, Chongqing 400067, China

[**] School of Management, Chongqing Technology and Business University, Chongqing 400067, China

E-Mail: stockyangb@sina.com ([#] Corresponding author)

**Abstract**

This paper aims to develop a desirable solution for non-blocking flow line problem under lean production mode. For this purpose, the artificial fish swarm algorithm (IAFSA) was improved by the probability coding of polar coordinates for solving the flow line scheduling under lean production mode. The improvement can prevent the algorithm from falling into the local optimum trap. The convergence of the proposed algorithm was verified by the popular test function Ackley. Finally, the IAFSA and another two popular swarm intelligence algorithms were applied to solve the test function of flow line scheduling. The comparison reveals the advantages of the IAFSA over the classic AFSA and the classic particle swarm algorithm in convergence speed and solution accuracy. The research findings lay a solid basis for the scheduling of actual hybrid flow lines and the preparation of scheduling plans in flow line production enterprises.

(Received, processed and accepted by the Chinese Representative Office.)

**Key Words:** Flow Line Scheduling, Improved Artificial Fish Swarm Algorithm (IAFSA), Lean Production Mode

## 1. INTRODUCTION

Since the reform and opening up, manufacturing has become a pillar industry in China thanks to cheap and efficient labour, turning the country into the workshop of the world. Against this backdrop, it is very meaningful to explore the management and operation of the manufacturing industry, especially those in lean production flow lines. In traditional flow line scheduling, the job sequence can be adjusted randomly, and the waiting time of jobs on the machines is infinite, i.e. there is no limit on the size of the buffer zone.

With the diversification of production enterprises and jobs, an increasing number of scholars have probed into complex flow line problems. For instance, Abadi et al. [1] described a flow line model with blocking, which is in line with the actual flow lines in production enterprises; in this model, a job has to wait if the next machine is busy; otherwise, it can go directly to the next machine. Aldowaisan and Allahverdi [2] developed a non-blocking continuous flow line model with no buffer zone, i.e. a job enters the next machine immediately. Papers [3-7] also investigate non-blocking flow line problems. Paper [8] targets the model in which the flow line rhythm is adjusted independently of the sequence of job placement. Papers [9-13] examine the model in which the flow job rhythm is adjusted independently of the sequence of job processing.

In light of the above, this paper studies the non-blocking flow line problem under lean production mode. Invented by Toyota, the lean production mode has proliferated from car making to product R&D, after-sales service and other links in the production cycle [14, 15]. Here, an advanced swarm intelligence algorithm is selected to solve the traditional flow line scheduling problem, aiming to realize efficient management of the scheduling process [16-21].

https://doi.org/10.2507/IJSIMM17(3)CO11

The remainder of this paper is organized as follows: The first chapter introduces the research background, significance and purpose, as well as relevant studies; the second chapter describes the flow line problem with a mathematical model; the third chapter improves the classical artificial fish swarm algorithm (AFSA) based on probabilities and polar coordinates; the fourth chapter tests the improved AFSA (IAFSA) and conducts the numerical simulation of the flow line problem; the fifth chapter wraps up this paper with some valuable conclusions.

## 2. PROBLEM DESCRIPTION

Featuring by synchronous operation, the flow line production mode is a production management method proposed by Johnson in the 1950s. The jobs are arranged in sequence on the flow line for sequential processing. The whole process consists of multiple phases. In each phase, a job can only be processed on the same machine. With the complication of the manufacturing process, the scale and difficulty of flow line operations are on the rise. Currently, hybrid flow line scheduling has been highlighted by many scholars. The hybrid flow lines fall into parallel flow line and common flow line. This paper focuses on the following non-blocking parallel flow line scheduling: $i$ jobs ($i = 1, 2, \dots, n$) are processed for time $t_{ij}$ on $j$ machines ($j = 1, 2, \dots, n$) at a fixed interval of $T$; among them, job $i$ is processed for time $y_{ij}$ ($y_{ij} = |t_{ij} - x_{ij}|$) on machine $j$ and another machine, with $x_{ij}$ being the pre-set fixed processing time of job $i$ on machine $j$. The $x_{ij}$ can be expressed as the piecewise function below:

$$x_{ij} = \begin{cases} T & y_{(i-1)j} = 0 \\ T + \left|y_{(i-1)j}\right| & -T < y_{(i-1)j} < 0 \\ T - y_{(i-1)j} & 0 < y_{(i-1)j} \le T \\ 2T & y_{(i-1)j} < -T \\ 0 & T < y_{(i-1)j} \end{cases} \tag{1}$$

If $y_{(i-1)j} = 0$, $x_{ij} = T$, indicating that the processing time of job $i$ on machine $j$ and that of job $i$-1 on machine $j$ are equal to the processing interval; if $-T < y_{(i-1)j} < 0$, $x_{ij} = T + |\, y_{(i-1)j}\,|$, indicating that job $i$-1 is suitable for further processing when its processing time on machine $j$ is smaller than or equal to the processing interval; if $0 < y_{(i-1)j} \le T$, $x_{ij} = T - |\, y_{(i-1)j}\,|$, indicating that job $i$-1 should be processed across two machines when its processing time on machine $j$ is greater than the processing interval; if $y_{(i-1)j} < -T$, $x_{ij} = 2T$, indicating that the processing time of job $i$ on machine $j$ is $2T$ when job $i$-1 is processed across two machines.

## 3. IAFSA

To overcome the defects of the traditional optimization algorithms for solving the flow line scheduling under the lean production mode, this chapter introduces the principle and idea of the classic AFSA and modifies the algorithm by improving the coding method. Specifically, the probability coding of polar coordinates was introduced to the coding of the flow line scheduling. The improved coding method greatly expands the search range. Coupled with the probability coding of polar coordinates, the IAFSA can avoid the local optimum trap and realize fine local search of extremums. Finally, the classical test function was employed to test the performance of the IAFSA.

### 3.1 Classic AFSA

The AFSA is an optimization algorithm inspired by the behaviours of fish swarm. The key of the algorithm lies in simulating the foraging, scattering and rear-end collision. The search for the optimal solution mimics the foraging behaviour of the fish swarm. Below is a mathematical description of the AFSA.

Suppose that there are $n$ artificial fish in a $k$-dimensional target search space. The $k$ states of artificial fish $n$ can be expressed as a vector $X_n = (x_{n1}, x_{n2}, \ldots, x_{nk})$, i.e. the position of artificial fish $n$ in the $k$-dimensional space. The food concentration at the current location of the artificial fish is expressed as the objective function $Y{=}F(X)$. This concentration can be obtained by substituting the $k$ states of artificial fish $n$, i.e. the position of artificial fish $n$ in the $k$-dimensional space into the objective function. The priority of the position is positively correlated with the food concentration.

Let $d_{ij} = \|X_i - X_j\|$ be the distance between individual artificial fish, $S$ be the sensing radius of individual artificial fish, $Step$ be the moving step length of the artificial fish, and $\delta$ be the crowding factor. The greater the sensing radius, the more coordinated the fish swarm, and the closer the connections between individual fish. In this case, the algorithm, despite high complexity, can converge rapidly to the global optimal solution. If the moving step length is short, the algorithm may consume a long time to converge to the global optimal solution and may easily fall into local optimum; if the moving step length is long, the global optimal solution can be identified at the cost of fine local search. The crowding factor is positively correlated with the swarm density in the search process [22, 23].

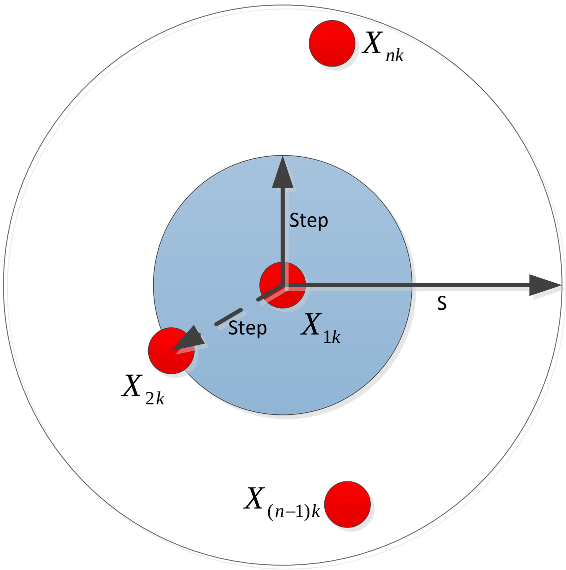

Figure 1: Iterative evolution of artificial fish swarm.

(1) Foraging behaviour

The foraging behaviour of artificial fish is an abstraction of the foraging process of actual fish swarm. To find food, the actual fish swarm moves to the unvisited area via random swimming and pinpoints the possible position of food due to visual and olfactory guidance. The two behaviours of the fish swarm were simulated by the random moving step length $Step$ and the visual radius $S$. The former allows individual artificial fish to perform global search in random swimming, and the latter ensures that the individual artificial fish detects the food area within its visual range. The foraging behaviour of the artificial fish swarm can be expressed as follows:

$$X_j = X_i + rand \cdot S \tag{2}$$

$$X_{inew} = X_i + \frac{X_j - X_i}{\left\| X_j - X_i \right\|} \cdot step \cdot S \tag{3}$$

$$X_{inew} = X_i + step \cdot rand \tag{4}$$

The above equations describe the foraging behaviour of artificial fish swarm. The $X_i$ is the current position of an individual artificial fish in the multi-dimensional space. When it discovers a new position $X_j$ within its visual radius $S$, the individual artificial fish will compare the food concentrations between the new position and the local position. If $F(X_j) > F(X_i)$, the individual artificial fish will move one step towards $X_j$. After several attempts, the individual artificial fish will move a random step from the current position if the new position has a lower food concentration than the current position.

(2) Scattering behaviour

The scattering behaviour refers to that the fish swarm tends to avoid overcrowding, a cause of the decline of food concertation. This behaviour can be described as follows:

$$S = \left\{ X_i \middle| \left\| X_j - X_i \right\| \leq S \right\} \quad (j = 1, 2, \ldots, i-1, i+1, \ldots, N) \tag{5}$$

$$X_c = \frac{\sum S}{N_f} \tag{6}$$

$$\frac{Y_c}{N_f} > \delta Y_i \tag{7}$$

$$X_{inew} = X_i + \frac{X_c - X_i}{\left\| X_c - X_i \right\|} \cdot step \cdot rand \tag{8}$$

The above equations describe the scattering behaviour of artificial fish swarm. After finding a swarm with high food concentration within the visual range $S$, an individual in the current artificial swarm will compute the central position $X_c$ of the multi-dimensional space, Eq. (6). If the crowding factor $\delta$ is as defined, the individual artificial fish should move a step towards the central position, Eq. (8). Otherwise, the fish should follow the model for the foraging behaviour.

(3) Rear collision behaviour

The rear collision behaviour is abstracted from the tendency that an individual in the current swarm approaches an individual in a nearby swarm with high food concentration. This behaviour can be described as follows:

$$\frac{Y_{\text{best}}}{N_f} > \delta Y_i \tag{9}$$

$$X_{inew} = X_i + \frac{X_{\text{best}} - X_i}{\left\| X_{\text{best}} - X_i \right\|} \cdot step \cdot rand \tag{10}$$

The above equations describe the rear collision behaviour of artificial fish swarm. After finding an individual in a nearby swarm with high food concentration, the artificial fish should move a step towards the nearby individual if the nearby swarm satisfies the crowding factor $\delta$. This behaviour reflects the fine local search of the AFSA.

### 3.2 Improvement of the AFSA based on probability coding of polar coordinates

The AFSA, a parallel search algorithm, mainly optimizes the multi-dimensional discrete nonlinear problem with multiple extremums. Over the years, the AFSA has been improved in many different ways. Papers [24-26] introduce Gaussian variation factor and simulated annealing algorithm to enhance the low accuracy of the parallel search of the AFSA. Papers [27-30] accelerate the convergence of the AFSA through integration of the principles of genetic algorithm, tabu search algorithm, and other intelligent search algorithms.

In this research, the AFSA is improved by modifying the coding method. The swarm intelligence algorithms cannot be directly applied to the solution of specific problems. The problems must be properly coded in a format solvable by swarm intelligence algorithms. The common coding approaches include real number continuous coding, discrete coding, and continuous variable and discrete variable mixed coding. Here, the probability coding of polar coordinates is implemented to the AFSA. By this coding method, the individuals in the fish swarm after $t$ iterations can be expressed as $Q(t) = [|p_1^t(\rho_1^t, \varphi_1^t)|\ldots|p_k^t(\rho_k^t, \varphi_k^t)|\ldots|p_f^t(\rho_f^t, \varphi_f^t)|]$. There are $f$ components of the swarm individuals. Based on quantum bit coding, the proposed coding method can be expressed as:

$$x_{nk} = \begin{cases} 1 & rand \le \varphi_k^t / 2\pi \\ 0 & rand > \varphi_k^t / 2\pi \end{cases} \tag{11}$$

## 4. PERFORMANCE ANALYSIS

### 4.1 Performance analysis of the test function

The IAFSA was validated through the Ackley function test. The Ackley is a popular 3D test functions with multiple extremums:

$$\begin{aligned} &\max_{(x_1,x_2,\ldots,x_n)} f(x_1,x_2,\ldots,x_n) = a \times e^{-bx\sqrt{\sum_{i=1}^{n} x_i^2 / n}} + e^{-bx\sqrt{c \times x_i}} - a - e \\ &s.t. \\ &\qquad -1 \le x_i \le 1,\ i = 1,2,\ldots,n \end{aligned} \tag{12}$$

Fig. 2 shows the image of the function at $a = 25$, $b = 0.4$, $c = 2\pi$ and $n = 2$. It can be seen that the function has multiple peaks and valleys and maximizes at the origin (0, 0).

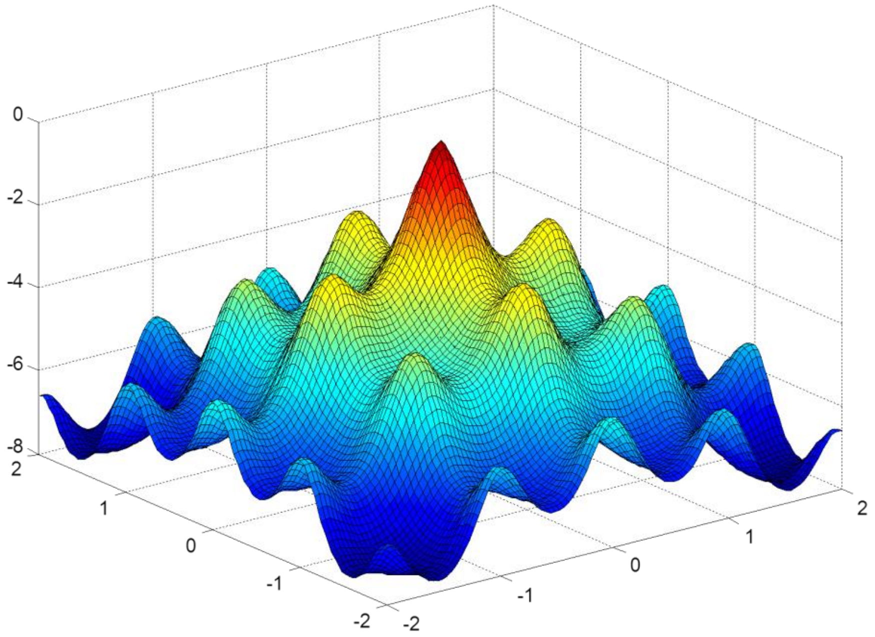

Figure 2: Image of Ackley function.

The Ackley function was solved by the IAFSA, i.e. the AFSA with probability coding of polar coordinates, aiming to verify the convergence and accuracy of the IAFSA. The parameters were configured as follows: swarm size $N = 40$, maximum number of iterations $Num = 300$, visual radius $S = 12$ and crowding factor $\delta = 0.218$. The fitness evolution curve of the Ackley function was obtained through 300 iterations by the IAFSA. As shown in Fig. 3, the IAFSA converged rapidly before the 100th iteration and basically stabilized after that. In general, the IAFSA is able to converge to the global optimal solution, despite a few fluctuations about the solution under the disturbance from the other individuals in the swarm.

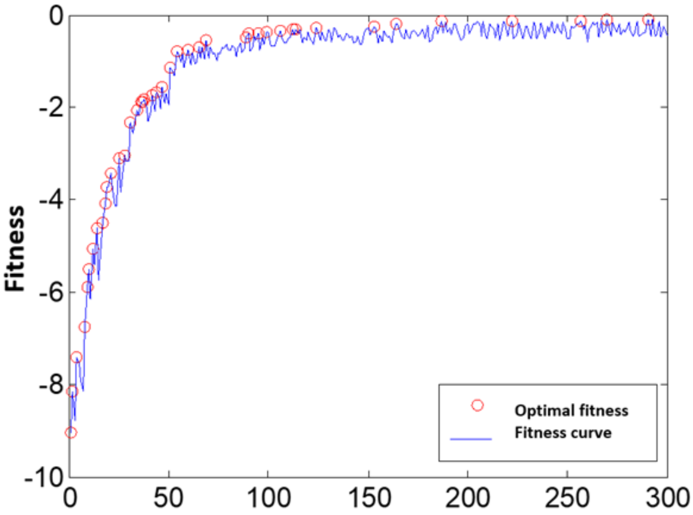

Figure 3: Fitness evolution curve of the Ackley function.

## 4.2 Comparative test

Three swarm intelligence parallel search algorithms, namely, the classic AFSA, the IAFSA and the classic particle swarm algorithm, were applied to solve the test function of flow line problem. The parameters of the classic AFSA and the IAFSA were configured as follows: swarm size $N = 30$, maximum number of iterations $Num = 500$, visual radius $S = 5$ and crowding factor $\delta = 0.218$. The parameters of the classic particle swarm algorithm were configured as follows: maximum number of iterations $Num = 500$, individual learning factor $C_1 = 0.2$, social learning factor $C_2 = 0.5$, and iterative weight $K = 10$. The evolutionary convergence graph after 500 iterations is shown in Fig. 4.

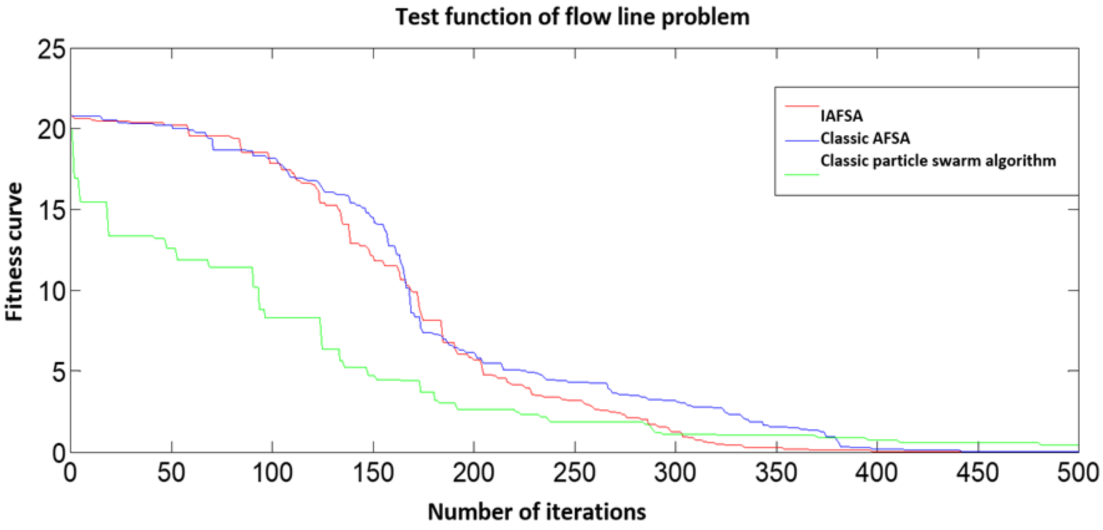

Figure 4: Comparison between three popular algorithms.

As shown in Fig. 4, the IAFSA was similar to the classic AFSA in convergence speed, convergence process and solution accuracy, and the convergence curves of these two algorithms were basically coincident. Nevertheless, the IAFSA converged much faster towards the global optimal solution than the classic AFSA. By contrast, the classic particle swarm algorithm converged rapidly at the beginning but jumped into the local optimum trap in the later stage, failing to find the global optimal solution.

## 5. CONCLUSIONS

This paper firstly introduces the research background of flow line scheduling problems under the lean production mode and describes the problem details with mathematical models. Then, the classic AFSA was introduced, together with the research status and modifications. After that, the coding method of the optimization problem was improved by the probability coding of polar coordinates, thus enhancing the performance of the classic AFSA. The improvement can prevent the algorithm from falling into the local optimum trap. The convergence of the proposed algorithm was verified by the popular test function Ackley. The results show that the IAFSA can achieve fast convergence and accurate solution. Finally, the IAFSA and another two popular swarm intelligence algorithms were applied to solve the test function of flow line scheduling. The comparison reveals the advantages of the IAFSA over the classic AFSA and the classic particle swarm algorithm in convergence speed and solution accuracy. The research findings lay a theoretical basis for the solution of flow line scheduling.

### ACKNOWLEDGEMENT

The authors acknowledge funding from the Ministry of education of Humanities and Social Science Project of China (Project No. 18YJC630087) and the National Social Science Foundation of China (Project No. 16XJY014), as well as the contributions from all partners of the mentioned projects. Besides, Yang Bai is the corresponding author and can be contacted at: stockyangb@sina.com.

### REFERENCES

[1] Abadi, I. N. K.; Hall, N. G.; Sriskandarajah, C. (2000). Minimizing cycle time in a blocking flowshop, *Operations Research*, Vol. 48, No. 1, 177-180, doi:10.1287/opre.48.1.177.12451

[2] Aldowaisan, T.; Allahverdi, A. (1998). Total flowtime in no-wait flowshops with separated setup times, *Computers & Operations Research*, Vol. 25, No. 9, 757-765, doi:10.1016/S0305-0548(98)00002-1

[3] Stafford Jr, E. F.; Tseng, F. T. (2002). Two models for a family of flowshop sequencing problems, *European Journal of Operational Research*, Vol. 142, No. 2, 282-293, doi:10.1016/S0377-2217(01)00320-4

[4] Allahverdi, A.; Aldowaisan, T. (2004). No-wait flowshops with bicriteria of makespan and maximum lateness, *European Journal of Operational Research*, Vol. 152, No. 1, 132-147, doi:10.1016/S0377-2217(02)00646-X

[5] Cheng, T. C. E.; Gupta, J. N. D.; Wang, G. (2000). A review of flowshop scheduling research with setup times, *Production & Operations Management*, Vol. 9, No. 3, 262-282, doi:10.1111/j.1937-5956.2000.tb00137.x

[6] Nagano, M. S.; Araújo, D. C. (2014). New Heuristics for the no-wait flowshop with sequence-dependent setup times problem, *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 36, No. 1, 139-151, doi:10.1007/s40430-013-0064-4

[7] Hatami, S.; Ruiz, R.; Andrés-Romano, C. (2015). Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times, *International Journal of Production Economics*, Vol. 169, 76-88, doi:10.1016/j.ijpe.2015.07.027

[8] Chang, J.-L.; Shao, H.-H. (2006). Scheduling a three-machine no-wait flowshop with separated setup time, *Journal of Harbin Institute of Technology*, Vol. 7, No. 2, 206-210

[9] Parthasarathy, S.; Rajendran, C. (1997). A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs – a case study, *Production Planning & Control*, Vol. 8, No. 5, 475-483, doi:10.1080/095372897235055

[10] Cheng, T. C. E.; Lin, B. M. T.; Toker, A. (2000). Makespan minimization in the two-machine flowshop batch scheduling problem, *Naval Research Logistics*, Vol. 47, No. 2, 128-144

[11] Gong, H.; Tang, L.; Duin, C. W. (2010). A two-stage flow shop scheduling problem on a batching machine and a discrete machine with blocking and shared setup times, *Computers & Operations Research*, Vol. 37, No. 5, 960-969, doi:10.1016/j.cor.2009.08.001

[12] Logendran, R.; Deszoeke, P.; Barnard, F. (2006). Sequence-dependent group scheduling problems in flexible flow shops, *International Journal of Production Economics*, Vol. 102, No. 1, 66-86, doi:10.1016/j.ijpe.2005.02.006

[13] Hwang, H.; Sun, J. U. (1998). Production sequencing problem with re-entrant work flows and sequence dependent setup times, *International Journal of Production Research*, Vol. 36, No. 9, 2435-2450, doi:10.1080/002075498192616

[14] Zhou, Y.-H.; He, Z.; Gao, X.-F. (2006). Comparing and integrating Lean production and Six Sigma, *Industrial Engineering Journal*, Vol. 9, No. 6, 1-4

[15] Zhang, Y. (2017). Comparison and Integration of Lean Production and Six Sigma Management, *Modern Business*, Vol. 17, No. 6, 102-103, doi:10.14097/j.cnki.5392/2017.06.053

[16] Song, C. L. (2011). *Research on production scheduling problem and its intelligent optimization algorithm*, Dalian University of Technology, Dalian City

[17] Wu, S. S. (2011). *Research on intelligent production scheduling method and its application for complex manufacturing system*, Donghua University, Shanghai

[18] Liu, A. J. (2011). *Study on operation scheduling in discrete manufacturing workshop under multi-production mode*, Chongqing University, Chongqing

[19] Tang, M.; Gong, D.; Liu, S.; Zhang, H. (2016). Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem, *Advances in Production Engineering & Management*, Vol. 11, No. 4, 299-310, doi:10.14743/apem2016.4.228

[20] Huang, Y. J. (2012). *Workshop scheduling problem based on target cascade and intelligent optimization algorithm*, South China University of Technology, Guangzhou

[21] Camaraza-Medina, Y.; Khandy, N. H.; Carlson, K. M.; Cruz-Fonticiella, O. M.; Garcia-Morales, O. F.; Reyes-Cabrera, D. (2018). Evaluation of condensation heat transfer in air-cooled condenser by dominant flow criteria, *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 2, 76-82, doi:10.18280/mmep.050204

[22] Liu, S.; Dai, H.; Gan, M. (2018). Information-decomposition-model-based missing value estimation for not missing at random dataset, *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 1, 85-95, doi:10.1007/s13042-015-0354-5

[23] Javanmard, A.; Montanari, A. (2014). Hypothesis testing in high-dimensional regression under the Gaussian random design model: Asymptotic theory, *IEEE Transactions on Information Theory*, Vol. 60, No. 10, 6522-6554, doi:10.1109/TIT.2014.2343629

[24] Li, X.-L.; Qian, J.-X. (2003). Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques, *Journal of Circuits and Systems*, Vol. 2003, No. 1, 1-6, doi:10.3969/j.issn.1007-0249.2003.01.001

[25] Zhang, M.-F.; Shao, C.; Gan, Y.; Li, M.-J. (2006). Hybrid artificial fish swarm optimization algorithm based on mutation operator and simulated annealing, *ACTA Electronica Sinica*, Vol. 2006, No. 8, 1381-1385, doi:10.3321/j.issn:0372-2112.2006.08.005

[26] Qu, L.-D.; He, D.-X. (2008). Artificial fish-school algorithm based on hybrid mutation operators, *Computer Engineering and Applications*, Vol. 2008, No. 35, 50-52, doi:10.3778/j.issn.1002-8331.2008.35.015

[27] Yu, Y.; Yin, Z.-F.; Tian, Y.-F. (2007). Multiuser detector based on adaptive artificial fish school algorithm, *Journal of Electronics & Information Technology*, Vol. 8, No. 1, 121-124

[28] Li, L.; Chi, S.-C.; Lin, G. (2006). Tabu fish swarm algorithm and its application to slope stability analysis, *Engineering Mechanics*, Vol. 23, No. 3, 6-10, doi:10.3969/j.issn.1000-4750.2006.03.002

[29] Liu, B.; Zhou, Y.-Q. (2008). Artificial fish swarm optimization algorithm based on genetic algorithm, *Computer Engineering and Design*, Vol. 29, No. 22, 5827-5829

[30] Li, R.; Liao, R.-Q. (2018). Research on estimation of optical fiber probe gas holdup based on the adaptive weighted data fusion algorithm, *International Journal of Heat and Technology*, Vol. 36, No. 2, 543-547, doi:10.18280/ijht.360217