

A MULTI-OBJECTIVE LOT-STREAMING OPTIMIZATION SCHEDULING MODEL CONSIDERING THE BLOCKING EFFECT

Li, Y. G.^{*,**,#} & Zhang, M. S.^{*,**}

^{*}Hunan University, Changsha 410082, China

^{**}Southwest Minzu University, Chengdu 610041, China

E-Mail: 20100068@swun.cn ([#] Corresponding author), zms@swun.cn

Abstract

In view of such difficulties as multi-objective, multiple constraints and the uncertainty of calculating parameters in product line scheduling, the multi-objective batch scheduling model considering the blocking effect was constructed. The feasibility and superiority of the algorithm in this paper are verified through carrying out a comparison between the model in this paper and the traditional assembly line scheduling. The research conclusions are as follows. The probability distribution model is established using discrete product coding way to enhance the quality of initial population. The application of restarting mechanism can improve the diversity of population. The blocking constraints occurring in the processing of adjacent machines are considered, and multiple lot-streaming flowshop scheduling model considering the blocking effect is taken into consideration. The solutions to the problem of obstruction are sought by using hybrid artificial swarm method. The global convergence of the model is enhanced utilizing the Pareto local search method. The original cross operator is improved with the information of non-dominated solutions and new generations are generated. The established model is compared with the traditional flow shop model. It is verified that the algorithm in this paper has fairly high global convergence and can generate high quality solution.

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

Key Words: Production Line, Optimal Scheduling, Multiple Target, Improved NSGA-II Algorithm, Blocking Effect

1. INTRODUCTION

Batch scheduling is a kind of new product line scheduling model. The products are divided into several sub-products to form a batch. The sub-products will be assembled into a complete product eventually after being processed by a machine. The batch processing of the assembly line can improve the processing efficiency of the production line, and shorten the idle time of the machine, thus bringing more profits to the enterprise.

Generally, there is only one objective function in the traditional batch scheduling model, such as the shortest production time, the lowest processing cost, the longest time to complete ahead of schedule, etc. In addition, there are multiple constraint parameters during the scheduling process. In the increasingly fierce market competition, the production line processing and scheduling have been unable to meet the company's earning requirement when they only satisfy a kind of target function. Therefore, the multi-objective product scheduling has become the current research hotspot. In the process of product processing, the blocking effect of adjacent machines also becomes another difficulty in restraining the working efficiency of the assembly line. The blocking effect occurs when the current machine has finished processing and the following machine has not finished the processing task. Because there is no buffer room, the first machine can only idle, thus reducing the processing efficiency.

Aiming at the above problems, many scholars have carried out relevant researches and established multiple optimal scheduling models, mainly including single-machine scheduling,

assembly line scheduling, job scheduling, open scheduling and parallel machine scheduling [1-7]. If according to the difference of constraint conditions in the scheduling process, the models proposed by researchers can be divided into assembly line displacement scheduling, zero wait assembly line scheduling, zero-leisure free assembly line scheduling, zero buffer-zone assembly line scheduling, lot-streaming flowshop scheduling and hybrid flow shop scheduling problem [8-11]. Meanwhile, in view of the diversification and uncertainty of the objective function, corresponding solutions mainly cover stochastic optimization algorithm, fuzzy optimization method and interval optimization method [12-15].

In view of such difficulties as multi objectives, multiple constraints and the uncertainty of calculating parameters in product line scheduling, the multi-objective batch scheduling model considering the blocking effect was constructed, which significantly improved the product processing efficiency. The feasibility and superiority of the algorithm in this paper are verified through carrying out a comparison between the model in this paper and the traditional assembly line scheduling.

2. IMPROVED MULTI-OBJECTIVE LOT-STREAMING FLOWSHOP SCHEDULING ALGORITHM

2.1 Description of relevant problems

NSGA (Improved Non-dominated Sorting Genetic Algorithm) was put forward in the 1990s. It selects population individual in accordance with the non-domination ordering relation, carries out cross and aberrance calculation on the individuals, and finally generates high quality species. NSGA-II is an improved quicksort algorithm established on the basis of NSGA. Its core idea is to modify the construction strategy and solution set distribution of NSGA to significantly reduce the calculation complexity and improve the distributivity and retention of high quality individual. The multiple optimization scheduling function is generally considered to be minimum in NSGA-II algorithm:

$$\min f(\phi) = [f_1(\phi), f_2(\phi), \dots, f_n(\phi)] \quad (1)$$

ϕ is the q -dimensional decision variable.

NAGA-II optimizes n objective functions together, and each objective function cannot meet the optimal requirements at the same time. Due to the above defects of the algorithm, it is aimless and non-directional in terms of optimizing objective function and the offspring quality is not high either.

In the machining process of the production line, generally, there is no buffer zone for storing products between adjacent machines. After the upstream machines complete processing, if the downstream machines are still working, the products cannot enter the next step of processing immediately, which will cause the product quality problem. Therefore, the product congestion problem inhibiting the assembly line was put forward.

It was assumed that the product series is $\delta = \{\delta(1), \delta(2), \dots, \delta(n)\}$. After the upstream machines complete processing, if the downstream machines are being occupied, it is deemed that the assembly line is blocked. If there are n products on the assembly line in total, and the number of processing machine is m , then there are $Q_{j,k}$ processing steps in total. A machine can only process one product at a time, so the makespan of δ is:

$$\left\{ \begin{array}{l} S_{\delta(1),0} = 0 \\ D_{\delta(1),t} = D_{1,t-1} + p_{\delta(1),t} \\ S_{\delta(j),0} = p_{\delta(j-1),1} \\ D_{\delta(j),t} = \max \{ D_{\delta(j),t-1} + p_{\delta(j),t}, D_{\delta(j-1),t+1} \} \\ D_{\delta(j),m} = D_{\delta(j),m-1} + p_{\delta(j),m} \\ C_{\max}(\pi) = D_{\delta(n),m} \end{array} \right. \quad (2)$$

$p_{\delta,t}$ and $D_{\delta,t}$ are the processing time and time of completion of product δ on the machine t . The maximum completion time in the sequence is $D_{\delta,t}$ of the last processing product on the last machine, which can be expressed as follows:

$$C_{\max}(\delta) = D_{\delta(n),m} \quad (3)$$

The scheduling blocking the assembly line is illustrated with 4 products and 3 machines. The Gantt chart of the scheduling strategy is shown in Fig. 1. The ordinate is the machine number. It can be seen from the figure that after $\delta(2)$ is processed on the first machine, since the second machine is still processing $\delta(1)$, $\delta(2)$ can only wait on machine 1. In addition, the processing of $\delta(3)$ has to be delayed. It creates a vicious cycle. Meanwhile, since $\delta(3)$ was processed for a short period of time on machine 1, and $\delta(2)$ was processed for a longer period of time on machine 2, blocking of $\delta(3)$ occurred, which postponed the processing of $\delta(4)$. Therefore, the traditional scheduling method must be optimized to reduce the overall processing time of the assembly line.

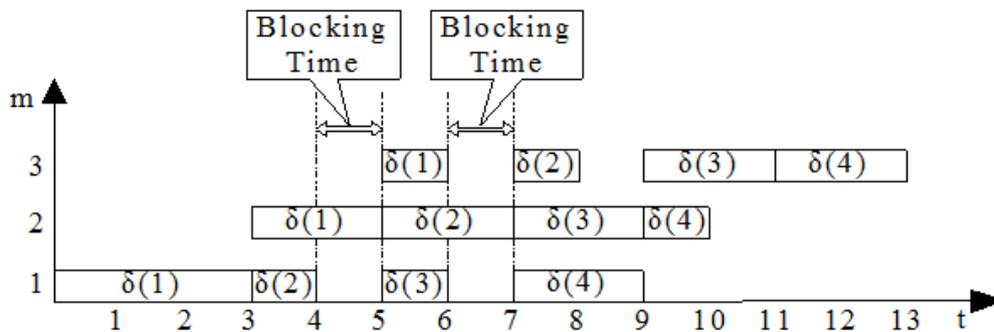


Figure 1: Gantt figure of a blocking flow shop scheduling problem.

2.2 Multi-objective lot-streaming flowshop scheduling mathematical model

The processed product in the lot-streaming flowshop scheduling can be evenly broken up into multiple sub-products. Assume that $\delta(j)$ are several sub-batches, then we have:

$$\delta(j) = \{ \delta(j)^1, \delta(j)^2, \dots, \delta(j)^{l_{\delta(j)}} \} \quad (4)$$

Let the batch processing sequence of the sub-products be the same and the processing of the sub-product of a certain product can't be started until the processing of all the sub-products of the previous product is completed. The planned processing time, delivery date and the number of decomposed sub-batches are already known, then the objective function of the scheduling sequence is as follows:

$$\left\{ \begin{array}{l} S_{\delta(j),t,e} = \max \{ C_{\delta(j),t-1,e}, C_{\delta(j),t,e-1} \} \\ C_{\delta(j),t,e} = S_{\delta(j),t,e} + p_{\delta(j),t} \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} \min f_1(\delta) = C_{\delta(n),m,l_\delta(n)} \\ f_2(\delta) = \sum_{j=1}^n C_{\delta(j),m,l_\delta(j)} \\ f_3(\delta) = \sum_{t=1}^m \left(C_{\delta(n),t,l_\delta(n)} - \sum_{j=1}^n \sum_{l=1}^{l_\delta(j)} P_{\delta(j),t} \right) \\ f_4(\delta) = \min \left(\sum_{j=1}^n \max(0, d_j - C_{\delta(j),m,l_\delta(j)}) \right) \end{array} \right. \quad (6)$$

where, f_1, f_2, f_3 and f_4 are the maximum product completion time, the total flow time of product, the total waiting time of the machine in the assembly line and the delayed completion time respectively. $S_{\delta(j),t,e}$ and $C_{\delta(j),t,e}$ are the start time and time of completion of the e^{th} sub-batch of the j^{th} product on machine t ; $p_{\delta(j),t}$ is the processing time of the j^{th} product; d_j is the delivery date of the product. The contrast Gantt chart of the improved multi-objective lot-streaming flowshop scheduling process and the traditional assembly line scheduling is shown in Fig. 2. It can be seen from the figure that the time to process $\delta(1)$ and $\delta(2)$ in order using traditional assembly line scheduling algorithm reaches 32, while it only takes 26 to process $\delta(1)$ and $\delta(2)$ in order using improved algorithm, which significantly reduced the product's maximum completion time and the idle time of the machine and further improved the production efficiency.

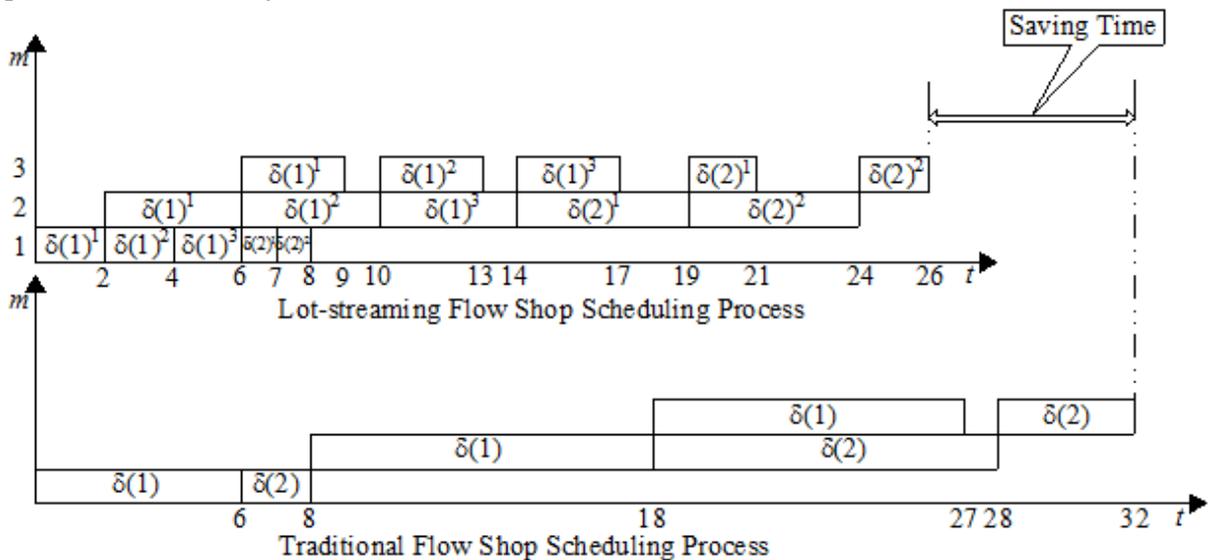


Figure 2: Gantt figure of lot-streaming and traditional flow shop scheduling process.

In the traditional NSGA-II optimal scheduling algorithm, the aimlessness and directionlessness of crossover operator and mutation operator are the main factors limiting the offspring to produce high quality populations, and the global convergence rate also declined accordingly. In view of this problem, this paper improved the estimation and insertion exchange method of the crossover operator and mutation operator and inset the new program into NSGA-II algorithm.

f_1, f_2, f_3 and f_4 in Eq. (5) were optimized based on NEH heuristic algorithm and four initial solutions were obtained. The EDA algorithm was implemented using pc probability and the commutating operator was figured out using 1-pc probability and corresponding offspring individual was generated, which formed the candidate population. The probability statistics and modelling of candidate population are carried out and a certain number of new solutions are generated. The construction of the population and probability model are as follows:

a) Build candidate population. The non-dominated solutions generated from each generation are taken as the candidate solutions. If the candidate solution is not larger than the population quantity, the domination solutions are all put into the candidate population, and the rest solutions will be selected according to the tournament method.

b) Establish probability distribution models. Two matrixes of ρ_{ij} and $\beta_{j,j}$ were obtained according to the position of the candidate population product.

$$[\rho_{i,j}]_{n \times n} = \begin{bmatrix} \rho_{1,1} & \rho_{1,2} & \cdots & \rho_{1,n} \\ \rho_{2,1} & \rho_{2,2} & \cdots & \rho_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ \rho_{n,1} & \rho_{n,2} & \cdots & \rho_{n,n} \end{bmatrix} \quad [\beta_{j',j}]_{n \times n} = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,n} \\ \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ \beta_{n,1} & \beta_{n,2} & \cdots & \beta_{n,n} \end{bmatrix} \quad (7)$$

ρ_{ij} refers to the frequency of the j^{th} product appearing on the position of i . $\beta_{j,j}$ refers to the number of no-waiting times of product j , which directly follows product j_1 . The candidate population $[\xi_{ij}]$ was established using ρ_{ij} and $\beta_{j,j}$. The calculation method of the component ξ_{ij} inside the population is:

$$\xi_{i,j} = \begin{cases} \frac{\rho_{i,j}}{\sum_{t \in \mu(i)} \rho_{i,t}} & i=1 \\ \frac{\rho_{i,j}}{\sum_{t \in \mu(i)} \rho_{i,t}} + \frac{\beta_{j',j}}{\sum_{t \in \mu(i)} \beta_{j',t}}}{2} & i=2,3,\dots,n \end{cases} \quad (8)$$

c) Seek new solution to the progeny population. Choose a solution from the parent population and figure out the probability of the product appearing at the position of i according to Eq. (8). Select the largest product and put it in the waiting sequence. Multiple iterations were carried out according to this method until the new solution is the completed product sequence. The optimization scheduling solution and the establishment process of the population and probability model were embedded in the original NSGA-II algorithm, then the optimization algorithm ONSGA-II was obtained. The steps of multiple-objective lot-streaming flowshop scheduling are shown in Fig. 3 and the specific steps are as follows:

1) Initialize the calculating parameter and population and calculate the objective function value in Eq. (6).

2) Calculate the inserted mutation operator based on 1-pc algorithm and generate progeny population. Calculate new solutions based on Eqs. (7) and (8).

3) Merge the selected progeny population and parental population and screen parental population using Pareto method. At the same time, each non-domination solution obtained through each iteration is saved.

4) Calculate the population diversity after each iteration. If the diversity is smaller than the pre-set threshold value, the restart strategy shall be activated to initialize the population and step 1, 2 and 3 shall be repeatedly implemented. It shall be determined whether the new solution obtained satisfies the terminal condition.

The ONSGA-II improved algorithm proposed in this paper uses EDA algorithm to replace the crossing algorithm in NSGA-II and embeds restart mechanism, which improves the calculation efficiency of the new algorithm and dramatically enhances the solution quality and population diversity.

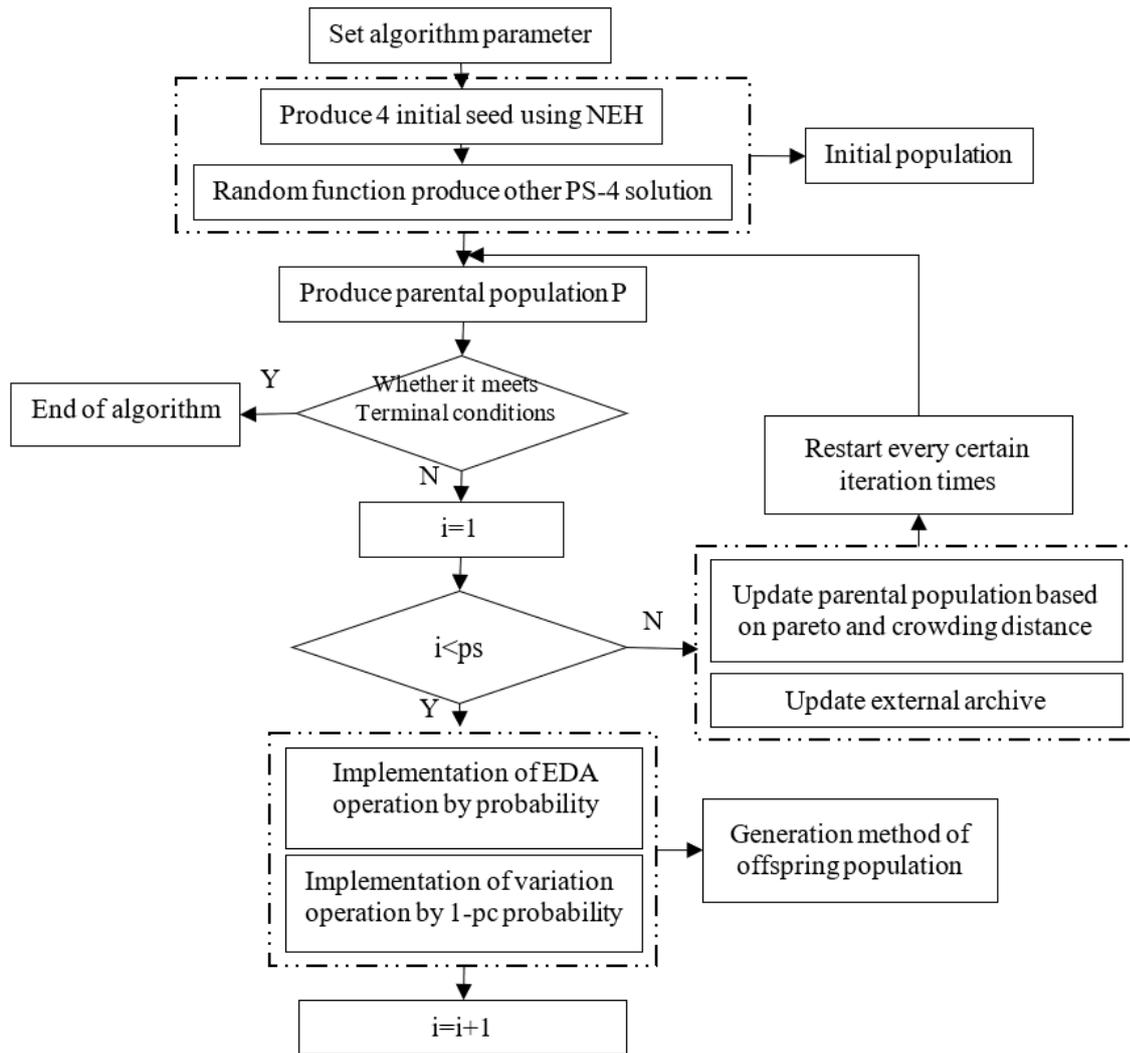


Figure 3: Flowchart of the optimal operation in multi-objective batch assembly line.

2.3 Blocking Constraint-based Multi-objective Lot-streaming Flowshop Scheduling

The above analyses the optimal scheduling when the buffer zone between adjacent machines was not considered. It is necessary to add blocking constraints to the established model because the pending processing of product frequently appears during the actual processing of product.

The blocking constraint-based multi-objective lot-streaming flowshop scheduling can be described as follows: the product series $\delta(j)$ adopts the same processing route. $\delta(j)$ was divided into multiple sub-batches. The request is the same as the previous one: the processing sequence of the sub-products is the same and the processing of the sub-product of a certain product can't be started until the processing of all the sub-products of the previous product is completed.

On the basis of the mathematical model in the previous section, blocking constraint was added additionally. The objective functions of the maximum time of completion and the completion time ahead of schedule are as follows:

$$\min f_1 = C_{\delta(n),m,J_{\delta(n)}} \quad (9)$$

$$f_2 = \sum_{j=1}^n \max\left(0, d_j - C_{\delta(j),m,J_{\delta(j)}}\right) \quad (10)$$

$$\begin{cases} S_{\delta(j),m,e} = \max \{ C_{\delta(j),m-1,e}, C_{\delta(j),m,e-1} \} & j=1,2,\dots,n \\ C_{\delta(j),m,e} = S_{\delta(j),m,e} + P_{\delta(j),m} & e=2,3,\dots,l_{\delta(j)} \end{cases} \quad (11)$$

Multiple initial solutions were generated using the heuristic algorithm. MME heuristic algorithm can improve the quality of initial solution in a short time. The constant insert operation makes the output value contain the value minimizing the target function.

A new solution based on mutation operator was calculated using hybrid artificial swarm algorithm and the function expression of the new solution was further figured out. The observing swarm locally disturbed the selected solution using Pareto local search method. The scouting swarm reconstructed partial not upgraded solutions. Finally, the external reserves were updated. When the algorithm satisfies the terminal conditions, the calculation ends. When the blocking constraint is embedded in the model, the population evolution can be led, thus improving the solution precision.

3. SIMULATION RESULTS AND EXAMPLE VERIFICATION

3.1 Performance analysis of multi-objective lot-streaming flowshop scheduling

The multi-objective lot-streaming flowshop scheduling algorithm (ONSGA-II) proposed in this paper was compared with the traditional algorithms of DHS, NSGA-II and TA. Coding was carried out based on C++6.0. The operating system is Windows 10. The following indexes were adopted to evaluate the performance of each algorithm:

Number of non-dominated solution $N(S_i)$:

$$N(S_i) = |x' \in S_i | \exists x' \prec \forall y, y \in S \cap x \neq y | \quad S = \bigcup_{i=1,2,3,4} S_i \quad (12)$$

The larger $N(S_i)$ is, the more non-dominated solutions there are.

The ratio of non-dominated solutions $R(S_i)$:

$$R(S_i) = \frac{N(S_i)}{|S_i|} \quad (13)$$

$R(S_i)$ can evaluate the quality of the non-dominated solution sets. The larger $R(S_i)$ is, the better the quality of non-dominated solution is.

The distance $D(S_i)$ from the referential solution is:

$$D(S_i) = \frac{1}{|S^*|} \sum_{y \in S^*} d_y(S_i) \quad (14)$$

$$d_y(S_i) = \min_{x \in S_i} \left\{ \sqrt{\sum_{k=1}^4 \left(\frac{f_k(x) - f_k(y)}{f_k^{\max}(\bullet) - f_k^{\min}(\bullet)} \right)^2} \right\} \quad (15)$$

$D(S_i)$ is mainly to evaluate the convergence of the algorithm; $f_k(x)$ and $f_k(y)$ are the k^{th} objective function of the independent variable. The larger $D(S_i)$ is, the farther the solution is from the centre of the optimal solution.

The $N(S_i)$, $R(S_i)$ and $D(S_i)$ values of such three algorithms as ONSGA-II, DHS and TA were figured out in Table I. It can be seen from the table that for the value of $N(S_i)$, the average value of ONSGA-II algorithm reached 82.58, and DHS, 9.17. The average value of $N(S_i)$ of TA algorithm is only 0.59, suggesting that more solutions were obtained using ONSGA-II algorithm. Most of the solutions were non-dominated ones, which are superior to other algorithms, and proves the superiority of ONSGA-II. The value of $R(S_i)$ obtained using the ONSGA-II algorithm is mostly larger than that of DHS algorithm, with the average

reaching 89 %, i.e. high-quality non-dominated solution can be obtained using ONSGA-II algorithm. For $R(S_i)$ value, the average of ONSGA-II algorithm is 0.03, which is much smaller than that of DHS algorithm and TA algorithm. The superiority of ONSGA-II algorithm could be learned through comparison.

Table I: $N(S_i)$, $R(S_i)$ and $D(S_i)$ obtained by ONSGA-II, DHS and TA.

Instance	$N(S_i)$			$R(S_i)$			$D(S_i)$		
	ONSGA-II	DHS	TA	ONSGA-II	DHS	TA	ONSGA-II	DHS	TA
10×5	28	37	0	69	62	0	0.02	0.11	0.33
10×15	29	24	0	98	69	0	0.04	0.02	0.61
30×10	99	9	0	100	48	0	0.04	0.93	3.66
30×20	100	7	0	99	43	0	0.05	0.88	4.56
50×10	81	5	0	81	77	0	0.01	0.23	3.45
50×15	100	8	0	92	34	0	0.01	0.06	1.55
70×5	96	4	0	100	68	0	0.06	0.17	0.84
70×15	98	2	0	99	22	0	0.01	0.16	4.72
90×10	99	5	0	97	98	0	0.03	0.10	1.73
90×15	100	4	2	100	98	1	0.01	0.11	1.38
110×10	62	3	3	62	77	5	0.09	0.06	1.41
110×20	99	6	4	100	98	4	0.05	0.24	1.18

The adjusted goodness of fit index is as follows:

$$SC(S_i) = \sqrt{\frac{1}{|S_i|-1} \sum_{j=1}^{|S_i|} (\bar{d} - d_j)^2} \tag{16}$$

$SC(S_i)$ was used to evaluate the distribution feature of the non-dominated solution. When $SC(S_i) = 0$, the non-dominated solutions spread evenly. The larger $SC(S_i)$ is, the larger the distribution discrepancy of the non-dominated solution is.

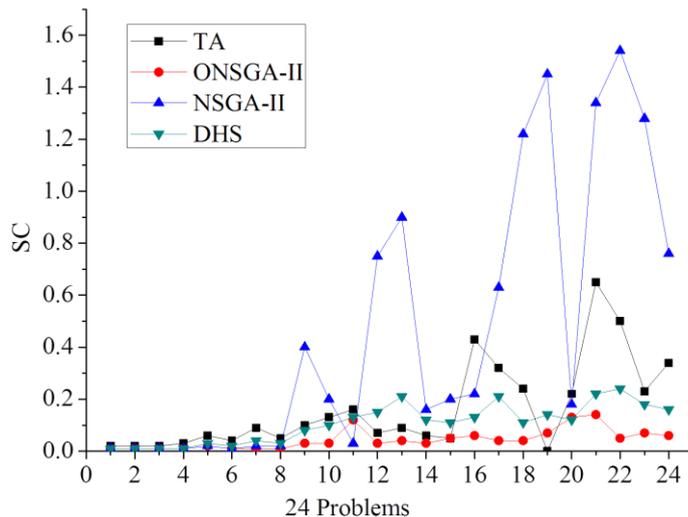


Figure 4: Scatter $SC(S_i)$ of ONSGA-II, NSGA-II, DHS and TA algorithms.

Fig. 4 shows the distribution of $SC(S_i)$ of ONSGA-II, NSGA-II, DHS and TA algorithms. It can be seen from the figure that in ONSGA-II algorithm, the values of $SC(S_i)$ at different distances are all fairly small and evenly distributed, without singular points. The distribution uniformity of $SC(S_i)$ in DHS algorithm is fairly sound, which is better than that of TA algorithm. The distribution uniformity of $SC(S_i)$ in NSGA-II algorithm is the worst, and the singular points are the most.

3.2 Multi-objective flowshop scheduling example analysis

The optimal flowshop scheduling in this paper is analysed with the practical production of the solar cell. The processing of battery in the production line mainly includes battery panel welding, backfaces cascading, lamination laying, structural assembly, performance test, packaging, etc. Various types of batteries are produced in the assembly line. Therefore, the objective function and the scheduling through the assembly line make the product completion time the shortest, meeting the requirement of the delivery quality and date.

The battery assembly line is optimized based on the optimal scheduling steps. The mathematical model of the optimal objective is:

$$\min f_1 = C \max(\delta) = D_{\delta(e),m} \tag{17}$$

$$f_2 = T \max(\delta) = \sum_{j=1}^n \max(0, D_{\delta(l_s),m} - d_j) \tag{18}$$

f_1 and f_2 are the longest time of completion and delayed completion dates of the battery pack assembly line. Fig. 5 shows the multi-objective lot-streaming flowshop optimal scheduling when there is a blockage in the processing by three machines. It can be seen from this figure that when the first set of machine is blocked, it is possible to make the assembly line scheduling more efficiently by adjusting the delay time of the second machine and breaking up the processing time.

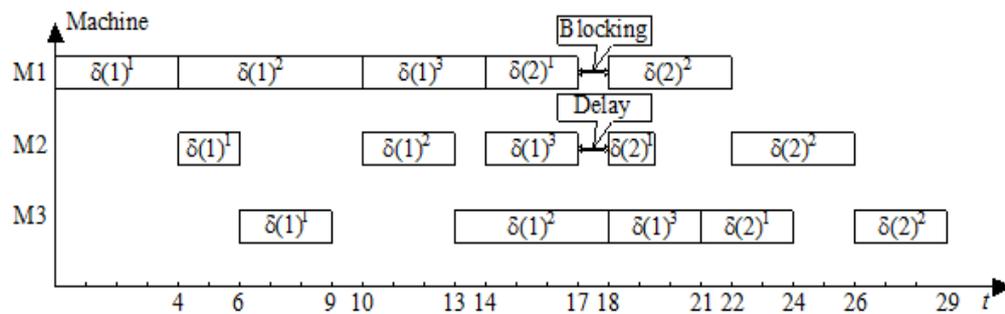


Figure 5: Gantt figure of lot-streaming flow shop scheduling problem.

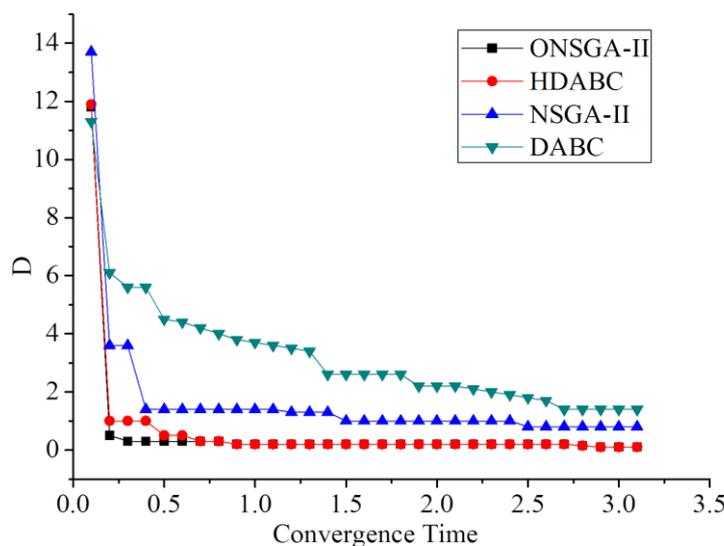


Figure 6: Convergence curves of four different algorithms.

Fig. 6 presents the iteration evolution curve chart of such four algorithms as ONSGA-II, DABC, NAGA-II and HDABC. The vertical coordinate represents the variation of D index in

the algorithms. It can be seen from the figure that as a whole, the evolution curve of the four algorithms decreases with time until it goes to a steady state. In the four evolution curves, the value of D in ONSGA-II algorithm is the minimum, the value of D in HDABC algorithm is relatively larger, and the value of D in NSGA-II algorithm is larger than that of HDABC algorithm. The value of D in DABC algorithm is the largest. The convergence rate of NSGA-II algorithm and DABC algorithm is relatively slow, because the genetic operators of NSGA-II have randomness and blindness in the scheduling process and DABC algorithm does not have an excellent offspring individual retention mechanism.

4. CONCLUSION

In view of such difficulties as multi-objective, multiple constraints and the uncertainty of calculating parameters in product line scheduling, the multi-objective lot-streaming flowshop scheduling model considering the blocking effect is constructed, which significantly improves the product processing efficiency. The feasibility and superiority of the algorithm in this paper are verified through carrying out a comparison between the model in this paper and the traditional flowshop scheduling. The research results of this paper are as follows:

(1) The traditional NSGA-II model is improved. The probability distribution model is established using discrete product coding way to enhance the quality of initial population. The information of non-dominated solutions enables each iteration to generate offspring individual. The restart threshold value was set. The diversity indexes after multiple iterations are still smaller than the threshold value. The application of restarting mechanism can improve the diversity of population.

(2) The blocking constraints occurring in the processing of adjacent machines are considered, and multiple lot-streaming flowshop scheduling model considering the blocking effect are taken into consideration. The solutions to the problem of obstruction are sought by using hybrid artificial swarm method. The global convergence of the model is enhanced utilizing the Pareto local search method. The original cross operator is improved with the information of non-dominated solutions and new generations are generated.

(3) The established model is compared with the traditional flowshop model. It is verified that the algorithm in this paper has fairly high global convergence and can generate high quality solutions through comparing the quantity of non-dominated solutions, ratio of non-dominated solutions, the distance from the referential solution set and distribution index of the established model and traditional model.

ACKNOWLEDGEMENT

This paper was supported by Youth Project of the National Social Science Fund of China, in the year of 2013 (13CGL133), National Natural Science Fund of China, in the year of 2017 (71774134) and Fundamental Research Funds for the Central Universities of Southwest Minzu University (2017NZYQN18).

REFERENCES

- [1] Kir, S.; Yazgan, H. R. (2016). A sequence dependent single machine scheduling problem with fuzzy axiomatic design for the penalty costs, *Computers & Industrial Engineering*, Vol. 92, 95-104, doi:[10.1016/j.cie.2015.12.012](https://doi.org/10.1016/j.cie.2015.12.012)
- [2] Nie, L.; Shao, X.; Gao, L.; Li, W. (2010). Evolving scheduling rules with gene expression programming for dynamic single-machine scheduling problems, *The International Journal of Advanced Manufacturing Technology*, Vol. 50, No. 5-8, 729-747, doi:[10.1007/s00170-010-2518-5](https://doi.org/10.1007/s00170-010-2518-5)

- [3] Dimopoulos, C.; Zalzal, A. M. S. (2001). Investigating the use of genetic programming for a classic one-machine scheduling problem, *Advances in Engineering Software*, Vol. 32, No. 6, 489-498, doi:[10.1016/s0965-9978\(00\)00109-5](https://doi.org/10.1016/s0965-9978(00)00109-5)
- [4] Huang, X. R.; Dai, W.; Du, B. G. (2016). Resource-constrained project scheduling problem for large complex equipment: a hybrid approach using Pareto genetic algorithm and interval-valued intuitionistic fuzzy sets, *Academic Journal of Manufacturing Engineering*, Vol. 14, No. 1, 12-21
- [5] Yu, G. D.; Yang, Y.; Zhao, X.; Li, G. (2014). Multi-objective rescheduling model for product collaborative design considering disturbance, *International Journal of Simulation Modelling*, Vol. 13, No. 4, 472-484, doi:[10.2507/IJSIMM13\(4\)CO17](https://doi.org/10.2507/IJSIMM13(4)CO17)
- [6] 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](https://doi.org/10.14743/apem2016.4.228)
- [7] Crăciunean, D. C.; Crăciunean, V. (2016). Membrane modelling for batch manufacturing scheduling, *Academic Journal of Manufacturing Engineering*, Vol. 14, No. 3, 36-45
- [8] Laha, D.; Chakravorty, A. (2011). A new heuristic for minimizing total completion time objective in permutation flow shop scheduling, *The International Journal of Advanced Manufacturing Technology*, Vol. 53, No. 9-12, 1189-1197, doi:[10.1007/s00170-010-2895-9](https://doi.org/10.1007/s00170-010-2895-9)
- [9] Shen, J. N.; Wang, L.; Wang, S. Y. (2015). A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion, *Knowledge-Based Systems*, Vol. 74, 167-175, doi:[10.1016/j.knosys.2014.11.016](https://doi.org/10.1016/j.knosys.2014.11.016)
- [10] Saidi-Mehrabad, M.; Fattahi, P. (2007). Flexible job shop scheduling with tabu search algorithms, *The International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 5-6, 563-570, doi:[10.1007/s00170-005-0375-4](https://doi.org/10.1007/s00170-005-0375-4)
- [11] Ho, N. B.; Tay, J. C.; Lai, E. M.-K. (2007). An effective architecture for learning and evolving flexible job-shop schedules, *European Journal of Operational Research*, Vol. 179, No. 2, 316-333, doi:[10.1016/j.ejor.2006.04.007](https://doi.org/10.1016/j.ejor.2006.04.007)
- [12] Ahmadizar, F.; Ghazanfari, M.; Ghomi, S. M. T. F. (2010). Group shops scheduling with makespan criterion subject to random release dates and processing times, *Computers & Operations Research*, Vol. 37, No. 1, 152-162, doi:[10.1016/j.cor.2009.04.002](https://doi.org/10.1016/j.cor.2009.04.002)
- [13] Lin, J. T.; Chen, C.-M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing, *Simulation Modelling Practice and Theory*, Vol. 51, 100-114, doi:[10.1016/j.simpat.2014.10.008](https://doi.org/10.1016/j.simpat.2014.10.008)
- [14] Juan, A. A.; Barrios, B. B.; Vallada, E.; Riera, D.; Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times, *Simulation Modelling Practice and Theory*, Vol. 46, No. 4, 101-117, doi:[10.1016/j.simpat.2014.02.005](https://doi.org/10.1016/j.simpat.2014.02.005)
- [15] Wen, F. H.; Gong, X.; Cai, S. H. (2016). Forecasting the volatility of crude oil futures using HAR-type models with structural breaks, *Energy Economics*, Vol. 59, 400-413, doi:[10.1016/j.eneco.2016.07.014](https://doi.org/10.1016/j.eneco.2016.07.014)