OPTIMIZATION OF ASSEMBLY LINE SCHEDULING BASED ON CROSSOVER OPTIMIZATION ALGORITHM

Xiang YANG

Information Construction Management Office, Nanjing University of Finance & Economics, Nanjing 210023, China.
E-mail: yang_xiang@163.com

ABSTRACT*: Due to ongoing refinement and complexity of flow-shop machining processes in the production workshop, the study of the assembly line scheduling optimization has aroused common concern since it has a high academic value and a good application prospect in the engineering field. This paper establishes a mathematical model for shop floor assembly scheduling optimization with the maximum completion time of assembly line as the objective function, where, a cross optimization operator is introduced to propose a cross optimization algorithm used to optimize the production line scheduling. Then the MATLAB simulation test is cited to determine the values of key parameters for the crossover algorithm. Comparing with the traditional two algorithms, it is proved that the crossover optimization algorithm is more superior and feasible. The study method proposed in this paper yields fruits for assemble line scheduling problem in production workshops, and has a practical application value and significance.*
KEYWORDS*: assembly line scheduling, crossover optimization algorithm, IBOF algorithm, robustness.*

1   INTRODUCTION

With the evolution of refinement and complexity of flow-shop machining process in the production workshop, the assembly line scheduling optimization has become a global concern. It has a high academic value and good application prospect in the engineering field. The study argues that there are one-quarter of production machining, product assembly, and information service systems that can be abstracted into a shop floor assembly line model (Zheng & Yamashiro, 2010). Workshop scheduling refers to the process in which *n* types of different products and components are organized onto *m* types of machines for production. Similarly, the Flow-shop Scheduling Problem (FSP) means that each product is produced in the similar conditions, namely, all products are machined on individual machines routed identically; further, if each product is machined in the same order on each machine, it is called a replacement line scheduling problem.

In fact, the assembly line scheduling problem in production workshops is an NP-Hard problem. It features large calculation scale and high complexity. For this reason, there are many different algorithms available for solving and calculating FSP. As computers and artificial intelligence have sprung up, FSPs can be calculated using intelligent optimization algorithms. For example, a hybrid genetic algorithm has been proposed and improved based on traditional genetic algorithm (Chiang et al., 2011; Li and Liu, 2018; Somashekhara et al., 2019). The tabu algorithm is used to solve the FSPs to obtain better results than various heuristic algorithms (Wang & Wang, 2012); with the maximum production time as the objective function, the FSPs can be solved with the ant colony optimization algorithm (Chung & Choi, 2013); and with the particle swarm optimization, from which the results can be available to outperform that of Rajendran et al. (Jing et al., 2011); or using the discrete difference and an iterative greedy algorithm to improve the Taillard calculation example (Chou, 2013). There are also many scholars who have optimized the production assemble line scheduling by using the simulated annealing algorithm, leapfrog algorithm and artificial bee colony algorithm, etc.

Given the above, this paper establishes a mathematical model with the maximum completion time of the assembly line as the objective function, hereby to propose a crossover algorithm to optimize the FSPs in the production workshop. Then the values for key parameters of crossover algorithm are determined by means of the simulation test or other methods. It is proved by comparison with the traditional algorithms that the proposed algorithm is more superior and feasible in the field.

2 FLOW-SHOP SCHEDULING OPTIMIZATION MODEL

The FSPs in the production workshop can be abstracted into a mathematical problem in order to make the study meaningful for guiding field production: n types of different products are machined on m types of different machines. The maximum production time is chosen as the production performance indicator, and the following five assumptions are made for machining processes:

(1) Each product has a similar machining process, and the sequences for producing one type of products on one machine are also identical;

(2) There is no buffer time between two adjacent products produced on one machine;

(3) It is required to determinate the production sequence of the products so as to make the production indicators optimal;

(4) Each product is machined only once on one machine;

(5) Only one piece of product is machined on each machine at a time, and certain process of one product is processed only on one machine.

For the FSPs in the production workshop, it is not allowed for any machine to have idle time during the production.
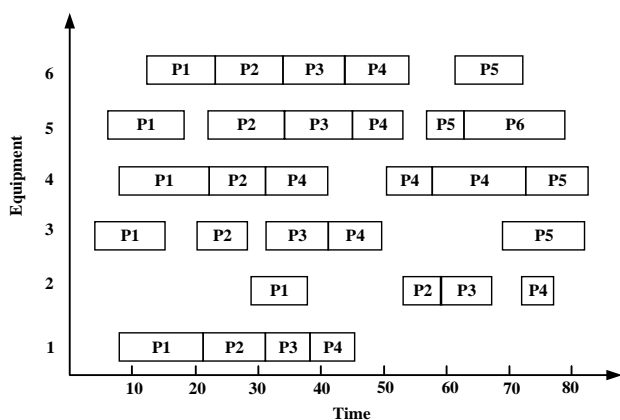

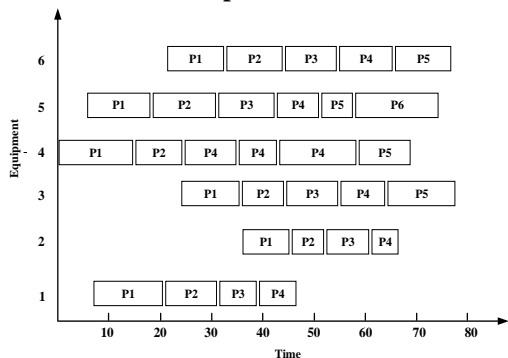**Fig. 1 Gantt chart of replacement line scheduling problem**

Fig. 2 Gantt chart of zero idle line scheduling problem

As shown in fig. 1 and fig. 2, there are Gantt charts separately representing the replacement line scheduling problem and the zero idle scheduling problem. In contrast, it is found that the machine in Fig. 1 has free time in the production process, while the machine in Fig. 2 works continuously. It is thus possible that the zero idle line scheduling is significantly better than the replacement line scheduling. For example, in FIG. 1, when the product 1 arrives and the machine 2 is idle, then the machine 2 starts to process the product 1. When the process 2 of the product 1 is ended, the process 2 of the product 2 does not come to an end, so that the process 2 of the product 1 cannot be operated. Hence, there is an idle time between machining processes of products 1 and 2 on the machine 2; when product 1 arrives at machine 2 in Fig. 2, the production is not required immediately until there is a right time in order to ensure the machines can continuously operate. How to determine the appropriate start time will be explained in detail in the calculation example.

With the maximum production time as the objective function, the flow-shop scheduling optimization model for production workshop is established as follows:

Objective function:
$$\min(C_{max}) = \min\left(\max_{k\in\{1,2,\dots,n\}}(C_{k,m})\right) \ (1)$$

Constraint conditions:
$$\sum_{i=1}^{n} X_{i,k} = 1, i \in \{1,2,\dots,n\} \ (2)$$
$$\sum_{k=1}^{n} X_{i,k} = 1, k \in \{1,2,\dots,n\} \ (3)$$
$$C_{1,1} = \sum_{i=1}^{n} X_{i,1} \times t_{i,1} \ (4)$$
$$C_{k+1,j} = C_{k,j} + \sum_{i=1}^{n} X_{i,k+1} \times t_{i,j}, k \in \{1,2,\dots,n-1\}, j \in \{1,2,\dots,m\} \ (5)$$
$$C_{k,j+1} \geq C_{k,j} + \sum_{i=1}^{n} X_{i,k} \times t_{i,j+1}, k \in \{1,2,\dots,n\}, j \in \{1,2,\dots,m-1\} \ (6)$$
$$C_{k,j} \geq 0, k \in \{1,2,\dots,n\} \ and \ j \in \{1,2,\dots,m\} \ (7)$$
$$X_{i,k} = \begin{cases} 1, \text{If the workpiece } J_i \text{ is precepted at K} \\ \quad \text{in the sequence R} \\ 0, \text{other cases} \\ (i,k \in \{1,2,\dots,n\}) \end{cases}$$
$$(8)$$

Where, n -product quantity; m -machine quantity; J_i-set of products $i = (1,2,\dots,n)$; M_j-set of machines $j = (1,2,\dots,n)$; $t_{i,j}$ - Production time of product J_i on machine M_j; $C_{k,j}$- Completion time of the product k produced on machine j; R- Sequence of products; $X_{i,k}$ −decision variable.

Formula (1) represents the maximum production time as the objective function of the FSPs: with the maximum production time as the indicator, the optimization objective is to find a minimum time series that can maximize the production time;

In the constraints, Formulas (2) and (3) represent that each product in the production workshop must appear one on the assembly line; formula (4) represents the time required for the first machine to finish the first process; Formula (5) represents the precedence relationship between adjacent two products k and k+1 when they are produced on one machine, that is to say, one machine should be allowed to produce one product at a certain moment. If an equal sign is used, it is specified that there may be no time interval between two adjacent processes on machine j, that is, the specified machine can work without interruption; Formula (6) guarantees that each machine only processes one product at a time, and a certain process of one product is performed only on one machine; Formula (7) gives constrains that the production time for each process must be non-negative.

3 OPTIMIZATION OF FLOW-SHOP SCHEDULING BASED ON CROSSOVER OPTIMIZATION ALGORITHM

3.1 Design of crossover optimization algorithm

A crossover optimization algorithm is designed to solve the FSPs in production workshop. It adopts the path coding method to encode the products and required machine on the assembly line. It is known from previous studies that the standard heuristic algorithm enables global search, tropism to find out a global optimal solution. However badly, the labeling algorithm runs too slowly during calculation. In this paper, the machining process is encoded based on the path, and the cross-optimal operator is used to calculate and optimize FSPs with tropism; in order to accelerate the calculation of the algorithm, a hybrid computing strategy is proposed herein.

The crossover optimization algorithm first needs to be initialized. In order to test the robustness of the crossover algorithm designed herein, the NEH heuristic algorithm is used to initialize it and form an initial set to calculate the optimal model solution as the initial solution used for subsequent optimization and calculation. The rest of the initial solutions generate randomly. According to the above ideas, it is found that the initial solution available by the NEH heuristic algorithm is indeed good, and the overall computing speed of algorithm is also improved.

The procedure of algorithm is given as follows:

Step1: Perform a tropism operation. Determine specific position of individuals, $\theta_{pbest} = \{\theta^i_{pbest}, i = 1,2,\cdots,S\}$.

Step2: Filter the optimal individual θ_{pbest} in the set.

Step3: Perform crossover optimization. All individuals in the set are queried, and each individual θ^i_{pbest} is intersected with the optimal individual θ_{pbest} by partial mapping crossover to obtain a new individual θ^i_{new}. There should be S individuals in the final set.

Step4: Cross-replacement. If the new individual θ^i_{new} is better than θ^i_{pbest}, let $\theta^i_{pbest} = \theta^i_{new}$. And keep looping.

Table 1. List of initialized parameters

Parameter	Meaning
S	Population size
N_s	Maximum number of forward processes in the same direction
N_c	Maximum number of iterations required for tropism operations
N_{re}	Maximum number of repetitive iterations
N_{ed}	Maximum number of migrated iterations
P_{ed}	Probability of migration

The crossover operation controls the generation and disappearance of the individuals via the crossover probability P_{ed}. This crossover can avoid the defect that the algorithm may generate a local optimal solution, and make it easier to obtain a global optimal solution that satisfies the constraints. In the heuristic algorithm, the crossover probability is constant, that is, the probabilities that individuals in the set intersects remain consistent, which may substantially cause the case that the better solution that partially closes to the optimal solution will be abandoned. In this paper, P^i_{v-ed} is defined as the adaptive crossover probability, and all individuals in the set are intersected according to the calculation method of Formula (9).

$$P^i_{v-ed} = \left(1 - \frac{H^i - H^{min}}{H^{max} - H^{min}} \times \frac{f^i - f^{min}}{f^{max} - f^{min}}\right) \times P_{ed}$$
(9)

Where, H^i—the maturity of individual i; H^{max} and H^{min}—the maximum maturity and minimum maturity in the set; f^i—fitness of individual i; f^{max} and f^{min} - maximum fitness and minimum fitness in the set. The set judges the probability that the individual intersects according to H^i and f^i, that is, the individuals with better H^i and f^i are mature, and the probability they may intersect is lower, and vice versa.

Migration is operated as follows:

Core: Given individual i, the value of its indicator P^i_{v-ed} is calculated.

Judgment: judge according to the condition $rand() < P_{v-ed}^i$. If the inequality is true in the judgment, the current individual is replaced with the generated new one; if the inequality is not true, it remains unchanged; but let $i = i + 1$, calculate P_{v-ed}^i of the next individual.

End: The loop ends when $i = S$, at this time, the algorithm has finished all the products needed to be produced.

3.2 Algorithm process

The algorithm process is constructed according to the above model and the proposed cross-optimal algorithm as follows:

Fig.3 Algorithm process

4 SIMULATION TEST AND ANALYSIS OF RESULTS

The CAR class example is cited to test the above crossover optimization algorithm. Individual variables and parameters in the example are first encoded before the calculation. Here we use the MATLAB2016.

The IBOF algorithm is initialized by creating a random number; IBOF_NEH first solves a problem suboptimal solution by the NEH to be the initial value in order to generate a random number. To test the robustness of the initial values of the algorithm,

it can be compared to the initial values available by the above two algorithms.

Table 2. Example variable codes

Factor	Meaning
$n \times m$	The product of the number of products and equipment, the size of the example to be solved
C_{NEH}^*	The optimal solution obtained by the heuristic algorithm (NEH)
C_{IBOF}^*	The optimal solution obtained by IBOF
$C_{IBOF\_NEH}^*$	The optimal solution obtained by IBOF_NEH

Table 3. Setting of algorithm parameters

Factor	Value
S	40
N_N	6
N_C	10
N_{re}	3
N_{ed}	2
P_{ed}	0.2

The above example is repeated for 20 times, and the results from the operation are statistically analyzed. Among them, Hit-goal - the ratio of the number of the optimal solution obtained to the times of the example run; RE - the error between the calculated value and the optimal solution every time it runs; ARE - average error; SRE - minimum error; BRE - maximum error; Time - the total running time (s) of the program.

Comparing relative errors of the three algorithms NEH, IBOF and IBOF_NEH, it is obvious that the maximum error of the NEH algorithm is 5.5 %, while the IBOF_NEH algorithm is only 0.99 %, so that the improved crossover algorithm has an obvious effects on the FSPs in the production workshop.

In the end, we draw the following conclusions: the crossover optimization algorithm is better for solving the FSPs in the production workshop. It well fits the bill for the production workshop. It is proved by comparing the solutions from the algorithms that the IBOF_NEH algorithm features the initial value with good robustness and higher stability.

Table 4. Comparison of the results from three algorithms

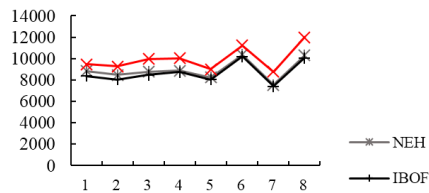| NO.		1	2	3	4	5	6	7	8
$n \times m$		6*11	4*16	5*13	4*14	7*10	8*7	7*9	10*8
C^*		7364	7037	7459	7737	7029	9199	6404	9057
NEH	RE	0.0562	0.0478	0.0124	0.0701	0.0345	0.0123	0.0097	0.378
IBOF	ARE	0.0004	0	0.0009	0.0002	0	0	0.0002	0.0001
	SRE	0	0	0	0	0	0	0	0
	BRE	0.0123	0.0109	0	0	0.0188	0	0	0.0102
IBOF_NEH	ARE	0.0015	0.0045	0	0	0	0	0.0028	0
	SRE	0	0	0	0	0	0	0	0
	BRE	0.0145	0	0.0007	0.0008	0.0001	0	0	0

Fig. 4 Evolution trend of solutions for 3 algorithms

5   CONCLUSION

Taking the assembly line scheduling problem in production workshop as the study object, this paper explores how to improve the workshop production efficiency under the influence of ongoing refinement and complexity of the flow machining processes in the shop floor. Here come specific conclusions as follows:

(1) Taking the maximum completion time of assembly line as objective function, and the time simultaneity and sequence as the constraints, we establish the mathematical model for assembly line scheduling optimization in production workshop.

(2) Based on the standard heuristic algorithm, a cross-optimal operator and spread probability are introduced to design the algorithm, thus propose to optimize the crossover optimization algorithm to optimize the assembly line scheduling problem in the production workshop.

(3) The MATLAB simulation test is conducted to determine the value of the key parameters for the crossover algorithm. Comparing with the traditional two algorithms, it is proved that the crossover optimization algorithm is more superior and feasible.

6   REFERENCES

►Allahverdi, A., Ng, C. T., Cheng, T. C. E., Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3), 985-1032.

►Bozorgirad, M. A., Logendran, R. (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. The International Journal of Advanced Manufacturing Technology, 83(5-8), 1135-1151.

►Chiang, T. C., Cheng, H. C., Fu, L. C. (2011). Nnma: an effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. Expert Systems with Applications, 38(5), 5986-5999.

►Chou, F. D. (2013). Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks. International Journal of Production Economics, 141(1), 137-145.

►Chung, D. Y., Choi, B. C. (2013). Outsourcing and scheduling for two-machine ordered flow shop scheduling problems. European Journal of Operational Research, 226(1), 46-52.

►Jamin, S., Danzig, P. B., Shenker, S. J., Zhang, L. (1997). A measurement-based admission control algorithm for integrated service packet networks. IEEE/ACM Transactions on Networking, 5(1), 56-70.

►Jing, C., Huang, W., Tang, G. (2011). Minimizing total completion time for re-entrant flow shop scheduling problems. Theoretical Computer Science, 412(48), 6712-6719.

►Li, G., Wang, X. Y., Wang, J. B., Sun, L. Y. (2013). Worst case analysis of flow shop scheduling problems with a time-dependent learning effect. International Journal of Production Economics, 142(1), 98-104.

►Li, L. X., Gao, J., Liu, Y. F. (2018). Opti-SW: An improved gene sequence alignment algorithm, Ingénierie des Systèmes d'Information, 23(6), 73-85.

►Ramanan, T. R., Sridharan, R., Shashikant, K. S., Haq, A. N. (2011). An artificial neural network based heuristic for flow shop scheduling problems. Journal of Intelligent Manufacturing, 22(2), 279-288.

►Schrage, I. L. (1965). Application of the branch and bound technique to some flow-shop scheduling problems. Operations Research, 13(3), 400-412.

►Somashekhara, S.C.H., Setty, A.K.Y., Sridharmurthy, S.M., Adiga, P., Mahabaleshwar, U.S., Lorenzini, G. (2019). Makespan reduction using dynamic job sequencing combined with buffer optimization applying genetic algorithm in a manufacturing system, Mathematical Modelling of Engineering Problems, 6(1), 29-37.

►Vanchipura, R., Sridharan, R. (2013). Development and analysis of constructive heuristic algorithms for flow shop scheduling problems with sequence-dependent setup times. International Journal of Advanced Manufacturing Technology, 67(5-8), 1337-1353.

►Wang, J. B., Wang, M. Z. (2012). Worst-case analysis for flow shop scheduling problems with an exponential learning effect. Journal of the Operational Research Society, 63(1), 130-137.

►Zheng, T., Yamashiro, M. (2010). Solving flow shop scheduling problems by quantum differential evolutionary algorithm. International Journal of Advanced Manufacturing Technology, 49(5-8), 643-662.