Cost Optimization Problem of Hybrid Flow-shop Based on PSO Algorithm

HAN Zhonghua\textsuperscript{1, a}, MA Xiaofu\textsuperscript{2, b}, YAO Lili\textsuperscript{3, c} and SHI Haibo\textsuperscript{4, d}

\textsuperscript{1} Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China
\textsuperscript{2} Graduate School, Chinese Academy of Sciences, Beijing, China
\textsuperscript{3} Department of Computer Science and Engineering, Tongji University, Shanghai, China
\textsuperscript{a} hanzhonghua@sia.cn, \textsuperscript{b} maxiaofu2008@gmail.com, \textsuperscript{c} yaolili@sia.cn, \textsuperscript{d} hbshi@sia.cn

Keywords: Hybrid flow-shop scheduling problem (HFSP). Particle swarm optimization (PSO). Genetic algorithm (GA). Cost optimization.

Abstract. A PSO-algorithm-based job scheduling method that takes production cost as optimization object is presented in this paper. The cost optimization model of HFSP, in which production cost is considered as an optimal factor, is constructed. PSO is used to take global optimization, make the production task assignment and find which machine the jobs should be assigned at each stage, which is also called the process route of the job. After that the local assignment rules are used to determine the job’s starting time and processing sequence at each stage. The total production cost converted by time-based scheduling results is comprehensively considering the processing cost, waiting costs, and the products storage costs. The numerical results show the effectiveness of the algorithm after comparing between multi-group programs.

Introduction

In the practical production, there exist variable processing modes and multiple process routines for jobs. And thus, there exists at least one work shop which has the classical feather of HFSP. Therefore, it is of practical significance to study the cost optimization problem in HFSP work shop. Neural network, simulated annealing, and genetic algorithm are studied by many scholars to solve the cost optimization problem in HFSP\textsuperscript{[1,2]}.

Particle swarm optimization (PSO) is an evolutionary algorithm, presented by Kennedy and Eberhart\textsuperscript{[3]}. Due to its easy implementation and quick convergence, the PSO algorithm has gained much attention and a wide range of successful applications. In\textsuperscript{[4]}, the author adopts a hybrid PSO algorithm to solve the total weighted tardiness problem of the single machine; in article\textsuperscript{[5]}, PSO algorithm is used to minimize the makespan and total flow-time in the permutation flow-shop sequencing problem.\textsuperscript{[6]} uses a hybrid discrete PSO algorithm to solve flow shop scheduling problems. In this paper, PSO algorithm is used to solve the cost optimization problem of HFSP.

Formulation of HFSP

The HFSP scheduling optimization model is established from viewpoint of the production cost. In this model, the production materials consumption represented by processing cost, the equipments waiting to make process consumption which is represented by waiting cost and the storage cost consumed by products which are working in process (WIP) and waiting for being processed at the next stage storage are overall considered. The scheduling aim is to minimize the product cost.

A Model Parameters.
\begin{itemize}
\item \(n\) – the number of job waiting for being processed;
\item \(m\) – the number of stages which each job must go through;
\item \(p\) – the position of job in the queue;
\item \(k\) – the machine on which the job is being operated;
\item \(M_j\) – the number of parallel machines at stage \(j, j=1,2,\ldots, m\);
\end{itemize}
\( j-k \)–the \( k \)th machine at stage \( j \), \( j=1,2,\ldots,m \), \( k=1,2,\ldots,M_j \);
\( \text{P}_{ij} \)–the processing time when job \( i \) is processed at stage \( j \);
\( \text{S}_{ij} \)–the time when job \( i \) starts to be processed at stage \( j \);
\( \text{C}_{ij} \)–the time when job \( i \) is finished at stage \( j \);

**Production Cost Optimization Mode of HFSP.** On condition of satisfying HFSP constraints, we add the computational formula of processing cost, waiting cost and WIP storage cost between stages, which all adapt to the stage and machine features of HFSP. And the cost optimization model of HFSP, which aims to minimize the production cost, is constructed.

• **Processing Cost**

\( \text{C}_{\text{cost}}^\text{p} \), the total processing cost of \( n \) jobs through \( m \) stages by sequence is calculated as:

\[
\text{C}_{\text{cost}}^\text{p} = \sum_{j=1}^{m} \sum_{i=1}^{n} \left( \text{P}_{ij} \cdot \sum_{k=1}^{M_j} \text{Y}_{ijk} \cdot F_{j-k} \right) \tag{1}
\]

where \( F_{j-k} \) is the processing cost rate for machine \( k \) at stage \( j \).

The processing time when job \( i \) is processed at stage \( j \) is \( \text{P}_{ij} \cdot \sum_{k=1}^{M_j} \text{Y}_{ijk} \cdot F_{j-k} \). Represents when job \( i \) is processed at stage \( j \), only one of the parallel machines \( M_j \) can be selected to operate. Then we multiply the processing cost rate by the processing time of this stage to get the processing cost of job \( i \) at stage \( j \).

• **Waiting Cost**

Waiting cost \( \text{C}_{\text{cost}}^\text{s} \) can be defined as when processing resources are waiting jobs, a certain amount of cost must be consumed to ensure the ready state, the total waiting cost \( \text{C}_{\text{cost}}^\text{s} \) for all machines of \( m \) stages is calculated as:

\[
\text{C}_{\text{cost}}^\text{s} = \sum_{j=1}^{m} \sum_{k=1}^{M_j} \text{T}_{j-k}^\text{s} \cdot F_{j-k} \tag{2}
\]

where \( T_{j-k}^\text{s} \) is the sum of the waiting time between every two processing tasks through the process, where machine \( j-k \) starts from the first task until all allocation processing tasks are completed. \( F_{j-k} \) is the waiting cost of machine \( j-k \).

The sum of machine \( j-k \) waiting time is calculated as:

\[
\text{T}_{j-k}^\text{s} = \sum_{i=2}^{n} \left( \text{S}_{ij} - \text{C}_{i-1,j} \right), \text{S}_{ij} \geq \text{C}_{i-1,j} \tag{3}
\]

where there are \( n \) (\( n \leq n \)) jobs being operated on machine \( j-k \) at stage \( j \), \( S_{ij} \) is the starting time of current job \( i \) on machine \( j-k \) at stage \( j \), and \( C_{i-1,j} \) is the completion time of the previous operated job on machine \( j-k \) at stage \( j \).

• **Work In Process (WIP) Storage Cost**

The total storage cost \( \text{C}_{\text{cost}}^\text{wip} \) of \( n \) jobs through \( m \) stages by sequence is calculated as:

\[
\text{C}_{\text{cost}}^\text{wip} = \sum_{j=2}^{m} \left( \sum_{i=1}^{n} \left( \text{S}_{ij} - \text{C}_{i,j-1} \right) \cdot F_j \right) \tag{4}
\]

where \( F_j \) is the storage cost rate of stage \( j \), \( \text{S}_{ij} - \text{C}_{i,j-1} \) represents the storage time between stages, where \( j > 2 \).

• **Scheduling Optimization Objective**

The minimum completion time of \( n \) jobs completing all \( m \) stages in HFSP is calculated as:

\[
f_{\text{min}} = \min \max \left\{ \text{C}_m \right\} \tag{5}
\]
The total production cost of \( n \) jobs completing all \( m \) stages in HFSP is calculated as:

\[
C_{\text{ost}} = C_{\text{out}} + C_{\text{ost}} + C_{\text{wip}}
\]  

(6)

The cost scheduling optimization objective of the HFSP is shown as formula (7). This objective is to minimize the total cost production of \( n \) jobs completing all \( m \) stages in HFSP.

\[
\min C_{\text{ost}}
\]  

(7)

**Constructing Simulation Data**

The simulation of the job scheduling of HFSP contains 8 jobs to be processed and 6 stages, and shows the effectiveness of the PSO based production cost optimization problem for HFSP. The number of the machine at each stage is 3, 2, 3, 3, 2 and 2, respectively. Table.1 shows the processing time of each job at each stage. In order to verify the algorithm's versatility, the processing time of each job at a certain stage is generated randomly between 30 and 60, and the minimum processing time unit is the minute.

In order to compare the difference of the total production cost between the two optimal results, where one is obtained by optimizing the cost, and the other is obtained by optimizing the processing time, we assume that processing cost rate \( F_{j-k}^v \), waiting cost rate \( F_{j-k}^s \) and WIP storage cost rate \( F_j^w \) are the random number between 0.5 and 1, and they reach the second place after decimal point. The processing cost rate \( F_{j-k}^v \) and waiting cost rate \( F_{j-k}^s \) directly relate to the machine of each stage, as shown in Table 1. The WIP storage cost rate \( F_j^w \) relates to the stage of WIP, as shown in Table 2.

| Stage | Machine | \( F_{j-k}^v \) | \( F_{j-k}^s \) | Stage | Machine | \( F_{j-k}^v \) | \( F_{j-k}^s \) | Stage | Machine | \( F_{j-k}^v \) | \( F_{j-k}^s \) |
|-------|---------|----------------|----------------|-------|---------|----------------|----------------|-------|---------|----------------|----------------|
| Stage1 | Machine 1 | 0.98 | 0.73 | Stage3 | Machine 1 | 0.90 | 0.57 | Stage4 | Machine 3 | 0.64 | 0.73 |
| Stage1 | Machine 2 | 0.50 | 0.92 | Stage3 | Machine 2 | 0.88 | 0.73 | Stage5 | Machine 1 | 0.72 | 0.91 |
| Stage1 | Machine 3 | 0.89 | 0.67 | Stage3 | Machine 3 | 0.52 | 0.80 | Stage5 | Machine 2 | 0.91 | 0.59 |
| Stage2 | Machine 1 | 0.78 | 0.59 | Stage4 | Machine 1 | 0.60 | 0.90 | Stage6 | Machine 1 | 0.64 | 0.71 |
| Stage2 | Machine 2 | 0.70 | 0.97 | Stage4 | Machine 2 | 0.78 | 0.95 | Stage6 | Machine 1 | 0.73 | 0.52 |

**Table 2 The waiting cost rate**

| Stage1 | Stage2 | Stage3 | Stage4 | Stage5 | Stage6 |
|--------|--------|--------|--------|--------|--------|
| \( F_j^w \) | 0.74 | 0.87 | 0.68 | 0.87 | 0.86 | 0.66 |

**PSO Algorithm Simulation Model of the Global Assignment**

- The Design of Initial Population. Considering the characteristic of HFSP, which is multi-stage and multi-job, we propose a real coding based on matrix in this paper. Each particle consists of \( m \) parts, each part represents one stage, and each part includes \( n \) subparts. So each particle is a \( m \times n \) real series. For the problem with 6 stages and 8 jobs, each particle is a \( 6 \times 8 \) real series, and the initial population is a matrix in which the elements are selected randomly in the corresponding ranges.
- The local assignment rules. In the process of solving the cost optimization problem of HFSP, PSO is used to make a global assignment. Thus, we can know the process route of the job. Local assignment rules are also necessary and should be determined by the practical production. In this paper, jobs are handled based on the FIFO rules.

**Computation Result**

We select 7 sets of scheduling schemes from the simulation data to compare and analyze. Table.3 listed the parameters of 7 schemes, the parameters including the initial population size \( NP \), the largest training algebra \( C_{\text{max}} \), genetic algorithm crossover rate \( P_c \) and mutation rate \( P_m \), the inertia weight \( w \) and acceleration constants \( c_1 \) and \( c_2 \). Scheme 1 is the project of obtaining production cost using initiative forward scheduling algorithm; Scheme 2 is the scheduling project of obtaining the minimum time
period using genetic algorithm; Scheme 3 is the scheduling project of obtaining the minimum total production cost using genetic algorithm; Scheme 4 is the scheduling project of obtaining the minimum time period using PSO algorithm; Scheme 5, 6, and 7 are the scheduling projects of obtaining the minimum total production cost using PSO algorithm. Through changing the values of $w, c_1$ and $c_2$, the impact of these parameters on scheduling results can be got. Scheme 2 and 4 select formula (5) as the scheduling objective function; Scheme 3, 5, 6, 7 select formula (7) as the scheduling objective function.

Table 3 The parameters setting of 7 sets of schemes

| Parameters Setting | NP | c | $P_c$ | $P_m$ | $w$ | $c_1$ | $c_2$ | Objective function |
|-------------------|----|---|------|------|----|------|------|-------------------|
| Scheme 1          | 1  | 1 |      |      |    |      |      |                   |
| Scheme 2          | 30 | 2000 | 0.6 | 0.2  |    |      |      | formula (12)      |
| Scheme 3          | 30 | 2000 | 0.6 | 0.2  |    |      |      | formula (14)      |
| Scheme 4          | 30 | 2000 | 0.35| 0.729| 2.187| formula (12) |
| Scheme 5          | 30 | 2000 | 0.35| 0.729| 2.187| formula (14) |
| Scheme 6          | 30 | 2000 | 0.71| 0.759| 2.187| formula (14) |
| Scheme 7          | 30 | 2000 | 0.35| 1.5  | 1.5 | formula (14) |

Table 4 The completion time and costs of seven schemes

| Scheme | $f_{min}$ | $C_{on}^\sigma$ | $C_{on}^\epsilon$ | $C_{on}^{sw}$ | $C_{on}$ |
|--------|-----------|----------------|----------------|-------------|---------|
| Scheme 1 | 374 | 1419.26 | 168.69 | 208.67 | 1796.62 |
| Scheme 2 | 389 | 1418.94 | 69.63 | 309.86 | 1798.45 |
| Scheme 3 | 425 | 1296.94 | 151.93 | 211.35 | 1660.22 |
| Scheme 4 | 357 | 1300.94 | 215.19 | 189.25 | 1705.38 |
| Scheme 5 | 414 | 1372.1 | 97.21 | 67.57 | 1536.88 |
| Scheme 6 | 390 | 1336.63 | 167.75 | 71.74 | 1576.12 |
| Scheme 7 | 398 | 1337.61 | 36.76 | 176.06 | 1550.43 |

Fig 1 The relationships between penalties sum of the total production cost and training iterations of Scheme 3, 5, 6, 7

From Fig. 1 we can see that the penalties sums of PSO based Scheme 5, 6, 7 and GA based Scheme 3 are reduced as the training iterations increases, which proves that both PSO and GA approaches the optimal value continuously. When iterations become larger, the change of GA’s penalties slows down but PSO maintains the convergent trend, which indicates that PSO algorithm has faster convergence rate. As shown in Figure 1, all schemes keep a downtrend within 600 iterations, the penalties sum of PSO based Scheme 5 still keeps a downtrend between 600 and 1500 iterations, while the downtrend of Scheme 3 slows down obviously after 360 iterations. So, Scheme 5 is closer to the optimal solution than Scheme 3. From the curves of Scheme 5, 6, 7 in Fig. 1 and the data in Table 4, we could find that the inertia weigh $w$ and acceleration constants $c_1$ and $c_2$ affects the results a lot. If we select the appropriate $c_1$, $c_2$ and $w$, the PSO algorithm could achieve better results.

In order to analyze and testify the effectiveness of PSO based cost optimization scheduling problem deeply, when the termination conditions of the largest training iteration $G_{max} = 2000$, 20 simulations are carried out. Thus, the average result can be got, and the completion time and production cost comparison table about 7 sets of schemes can be constructed, which is shown in Table

---

**Advanced Materials Research Vols. 532-533**

---

1619
Because the forward scheduling algorithm is a preceding algorithm, we can see the completion time $f_{\min}$ and the total production cost $C_{tot}$ of Scheme 1 are smaller. But compared with the Scheme 4 and Scheme 5, the total completion time $f_{\min} = 374$ in Scheme 1 is longer than that of Scheme 4 which uses PSO algorithm; the total production cost $C_{tot} = 1796.62$ in Scheme 1 is larger than that of Scheme 5 which also uses PSO algorithm. Therefore, good results are not easy to be obtained using the initiative forward scheduling algorithm.

Scheme 2 and Scheme 4 are both using formula (5) as objective function. Except the waiting cost $C_{w}$, other results of Scheme 4 show the superiority to that of Scheme 2, and Scheme 4 has the minimum total completion time of all jobs, which reflects that better results are easier to be obtained using PSO algorithm other than GA to solve the maximum and minimum problems.

From the table 6, we can see the total production cost of Scheme 5 based on PSO algorithm is minimum. Compared with the GA based Scheme 3 which uses the same objective function (7), the processing cost $C_{tot}^3$ of Scheme 3 is 75.16 less than Scheme 5, but the total waiting cost $C_{w}$ and the total storage cost $C_{w}$ of Scheme 5 is 54.72 and 143.78 less than those of Scheme 3, respectively. Considering all the three costs, the total production cost $C_{tot}$ of Scheme 5 is 123.34 less than that of Scheme 3, which means the cost of Scheme 5 reduces 7.43% of the cost Scheme 3. The reduction of the total waiting cost shows that the idle time of the equipments decreases, and the utilization rate of equipments increases. The reduction of the total storage cost shows that the storage cost of WIP decreases. In a word, the total production cost is an evaluation indicator of comprehensive consideration of all kinds of costs. And Simulation results show that after setting the correct parameters, PSO algorithm can solve the cost optimization scheduling problem more effectively.

Summary

In this paper, HFSP is studied, PSO algorithm is proposed to solve production cost optimization problems, and the HFSP production cost optimization model based on PSO algorithm is constructed. This proposed method can better minimize the production cost of jobs; thereby a better scheduling result can be obtained. The computational result demonstrates the effectiveness of the algorithm to solve such problems.

Acknowledgement

This work was supported by the National High Technology Research and Development Program “863”-Program Foundation of China under Grant No. 2007AA040702.

References

[1]. JIA Z., GONG L., The Study on Rate of Nonlinear Direct Cost Based on Neural Network[C]// 2009 International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, vol.3, 2009:342-345.
[2]. SHI Y., LI H., LU N., Quality Reliability-Cost Optimization of Construction Project based on Genetic Algorithm[C]// International Conference on Engineering Computation, Hongkong, 2009:159-162.
[3]. KENNEDY J, EBERHART R C. Particle swarm optimization[C]// Proceedings of IEEE Conference on Neural Network. Piscataway, N.J., USA: IEEE Press, 1995:1942-1948.
[4]. TASGETIREN M F, LIANG Y C, SEVKLI M, et al. Particle swarm optimization and differential evolution for the single machine total weighted tardiness problem[J], International Journal of Production Research, 2006, 44(22): 4737-4754.
[5]. TASGETIREN M F, LIANG Y C, SEVKLI M, et al. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem [J], European Journal of Operational Research, 2007, 177(3): 1930-1947.
[6]. CHANDRASEKARAN S, PONNAMBALAM S G, SURESH R K, et al, A hybrid discrete particle swarm optimization algorithm to solve flow shop scheduling problems[C]// IEEE Conference on Cybernetics and Intelligent Systems, Bangkok, 2006:1-6.