Multi-Objective Scheduling Simulation of Adaptive Job Shop based on Modified SOMA Algorithm

Danding Jiang¹, Mingwei Wang², Ying Zhao³ and Tengyuan Jiang²

¹College of Systems Engineering, National University of Defense Technology, Changsha, China
²School of Mechanical Engineering, Northwestern Polytechnical University, Xi’an, China
³China Aerospace Electronic Technology Research Institute, Beijing, China

E-mail:jddname@aliyun.com

Abstract. In this paper, an effective modified Self-Organization Migrating Algorithm (SOMA) is proposed to solve multi-objective adaptive job shop scheduling with the criterion to minimize the processing cost, minimize makespan, minimize the total machine loads. The modified SOMA stresses the balance between global exploration and local exploitation by introduce adaptive step, and improved the population diversity in the process of individual migration by introduce quadratic interpolation, which effectively avoided the premature and improved the convergence of the SOMA. Finally, through the 4x6 job shop scheduling problem verify the performance of the modified SOMA algorithm, the computational results show that the proposed modified SOMA efficiently solves adaptive job shop scheduling.

1. Introduction

With the change of production model to small batch, multiple varieties and customization, the job shop is required to be able to adjust quickly according to the market demand in order to adapt to the fast tempo production mode, therefore, multi-objective adaptive job shop scheduling has attracted a lot of attention [1-7]. For instance, Ahmed Farouk etc. provide a nonlinear GA optimization model for allocation of resources and available time to tasks for scheduling project [1]. Imen Driss etc. propose a new genetic algorithm to solve FJSP to minimize makespan [2]. Miguel A etc. propose a new hierarchical heuristic algorithm which is an adaptation of the Newton’s method for continuous multi-objective unconstrained optimization problems for multi-objective FJSP [3]. Liu A J etc. established mathematical model according to the objective function of minimizing the tardiness penalty and manufacturing time, employed the hybrid and cycle driven rescheduling to modified genetic algorithm optimized the process route and processing sequences[4]. Piroozfard H, etc. used the modified multi-objective genetic algorithm to complete the job shop scheduling problem which use carbon emissions and total delay time as scheduling objective [5]. C Zhu etc. for optimized the scheduling stability and robustness of job shop, used the multi objective differential evolution algorithm to solve the scheduling of flexible job shop [6]. Hadi Mokhtari etc. aimed the target of makespan, system utilization and total energy cost, completed the scheduling problem by combined with enhanced evolutionary algorithms and global standards [7]. In conclusion, for balance multiple objectives to obtain the optimization, the multi objective need to transformed into single objective scheduling.
problem. Under the guidance of this method, overall consideration minimize the processing cost, minimize makespan, minimize the total machine loads, realization the self-adaptation of scheduling process. After the SOMA has been proposed to optimization problem, it widely used in many fields due to its strengths such as fast convergence speed, few adjustment parameters, and the ability of self-adaption [8]. Davendra D etc. have used the SOMA to solve the flow-shop scheduling with no-wait makespan [9]. The modified SOMA stresses the balance between global exploration and local exploitation by introduce adaptive step; improved the population diversity in the process of individual migration by introduce quadratic interpolation. Finally, the computational results show that the proposed modified SOMA efficiently solves job shop scheduling.

2. Adaptive Job Shop Scheduling

2.1. Job Shop Scheduling Description.
The job shop scheduling problem can be described as following: the jobs set \( J = \{ J_1, J_2, \cdots, J_n \} \), where \( n \) is the total number of workshop jobs; the machine set \( M = \{ M_1, M_2, \cdots, M_m \} \), where \( m \) is the total number of machine in the workshop; processes set \( p_{i,j} = \{ p_{i,1}, p_{i,2}, \cdots, p_{i,n} \} \) is the processes sequence of the job \( J_i \), where \( n_{bi} \) represents the number of processes in the job, \( p_{o,ij} \) represents the j-th process of job \( J_i \) is processed on machine \( k \).

2.2. Multi-Objective Optimization Model
Most of the literature use the makespan, cost as indicators to evaluate the scheduling scheme. Although these indicators can reflect the merits of the scheduling scheme at a large extent, but leave the machine load out of consideration. This paper analyzes the existing literature and considers the three goals of production cost, makespan and total machine loads to express the cost, time, and machine load of job shop respectively. The objective function is as follows:

- Minimization production cost
  \[
  \text{Cost} = \text{Cost}_{\text{ready}} + \text{Cost}_{\text{process}} = \sum_{i=1}^{n} \sum_{j=1}^{n_{bi}} \left( cr_{i,j}^k \cdot p_{o,ij}^k \cdot rt_{i,j}^k + cp_{i,j}^k \cdot p_{o,ij}^k \cdot ot_{i,j}^k \right) 
  \]
  Where \( \text{Cost} \) indicate the total cost of the job shop; \( \text{Cost}_{\text{ready}} \) indicate the preparation cost; \( \text{Cost}_{\text{process}} \) indicate the processing costs; \( cr_{i,j}^k \), \( cp_{i,j}^k \) are respectively indicates the preparation cost coefficient, the process cost coefficient of the j-th process of the job \( J_i \) on the machine \( k \); \( D_{i,j}^k \) indicates the deadline of the j-th process of the job \( J_i \) on the machine \( k \).

- Minimize the makespan
  \[
  T = \sum_{i=1}^{n} \sum_{j=1}^{n_{bi}} \sum_{k=1}^{m} (rt_{i,j}^k + ot_{i,j}^k) 
  \]
  Where \( T \) represents the total processing time of the job shop, \( rt_{i,j}^k \) represents the preparation time of the j-th process of the job \( J_i \) on the machine \( k \), \( ot_{i,j}^k \) represents the processing time of the j-th process of the job \( J_i \) on the machine \( k \).

- Minimize total machine loads
  \[
  \min W = \min \sum_{k=1}^{m} W_i 
  \]
  Where \( W \) is total machine loads, \( W_i \) is the load of machine \( k \), all of load is measured by machining time, which \( W_i \) can be description as follow:
\[ W_i = \sum_{j=1}^{n_i} \sum_{k=1}^{m_k} p_{k,j} \cdot t_{i,j}^k \]  

Where \( t_{i,j}^k \) is machining time of the \( j \)-th process of the job \( J_i \), on the machine \( k \), which consist of preparation time and processing time.

Subject to:
- Processing time is not negative
- the preparation time is before the processing time
- The processing of the next process can only be carried after the completion of the previous process
- A machine tool cannot perform two job simultaneously
- The job must be completed before the deadline

In order to achieve the balance among the various optimization objectives and obtain an approximate optimal scheme to the production process of the optimization process of job shop scheduling, therefore, integrated the above three optimization objectives form a single objective. Before integrating optimization objectives, needs to use the following formula to normalize the optimization objective:

\[ a = \frac{b_{\text{max}} - b}{b_{\text{max}} - b_{\text{min}}} \]  

Where \( a \) is the normalized result of the optimization objective, \( b_{\text{max}} \) represents the maximum value of the optimization objective, \( b_{\text{min}} \) represents the minimum value of the optimization goal, \( b \) represents the current value of the optimization function.

Based on the above three normalized optimization goals, weights are used to normalize them to a single scheduling objective, as follows:

\[ f = \omega_1 \cdot \overline{\text{Cost}} + \omega_2 \cdot \overline{T} + \omega_3 \cdot \overline{W_k} \]  

Where \( f \) represents the optimization objective, \( \overline{\text{Cost}} \) represents the normalized cost, \( \overline{T} \) represents the normalized makespan, \( \overline{W_k} \) represents the total machine loads, \( \omega_1, \omega_2, \omega_3 \) are weight factors, which are obtained through an adaptive scheduling strategy and expert experience, \( \omega_1 + \omega_2 + \omega_3 = 1 \), \( \omega_1 \geq 0, \omega_2 \geq 0, \omega_3 \geq 0 \).

3. Modified SOMA-Based Scheduling Algorithm

In order to modify the self-adaptive ability of SOMA algorithm, this paper combines adaptive step size with quadratic interpolation to improve the exploration and diversity of SOMA respectively.

3.1 Adaptive Step Setting

In the individual migration process of SOMA, step represents the size of the search range of the individual in the optimal scheme direction, and in the standard SOMA algorithm, the recommended value is 0.11. However, during the actual simulation application process, it can be found that in the early stage of the algorithm operation, a larger step size can be used to improve the exploration capability of the algorithm. In the later stage of operation, a smaller step length is used to enhance the local search ability of the algorithm, and the local search ability of the algorithm is effectively utilized while ensuring the improvement of the algorithm's convergence speed. Therefore, this paper adopts the linear decreasing step strategy proposed in literature [10], so that the initial step length is the maximum step size \( \text{Step}_0 \), the termination step size is the minimum step size \( \text{Step}_m \), and the step size is adaptively updated after each evaluation. The adaptive step length is defined as:
\[
Step = \text{Step}_{\text{max}} - \left( \frac{\text{Count}}{\text{Sum}_\text{Count}} \right) \cdot (\text{Step}_{\text{max}} - \text{Step}_{\text{min}}) \quad (7)
\]

Where \( \text{Count} \) represents the current time of evaluation, \( \text{Sum}_\text{Count} \) represents the maximum time of evaluation, and according to the experimental results in [10], \( \text{Step}_{\text{max}} \) is set to 0.7 and \( \text{Step}_{\text{min}} \) is set to 0.4.

3.2 Adaptive Individual Migration

Different from others evolutionary algorithms, the SOMA algorithm use a real number \( PRT \) between \([0, 1]\) as a perturbation to determine whether an individual migrate to a leader. The detailed introduction of \( PRT \) refer to the literature [8].

In order to improve the diversity of the population, refer to the literature [11] to introduce quadratic interpolation. When the perturbation matrix \( A \) is 0, the best individual (leader) and two random individuals in the population are selected, use the quadratic interpolation method generate a new individual. The formula is as follows:

\[
x = \frac{[(R_i^2 - R_i^1) \cdot f(R_1) + (R_i^2 - R_i^2) \cdot f(R_2) + (R_i^2 - R_i^3) \cdot f(R_3)]}{2[(R_i^2 - R_i^1) \cdot f(v_1) + (R_i^2 - R_i^2) \cdot f(v_2) + (R_i^2 - R_i^3) \cdot f(v_3)]} \quad (8)
\]

Where \( R_l \) represents the leader, \( R_i \) and \( R_j \) represent individuals randomly selected from the population. If the new individual is accepted when it is better than the original individual and is replaced with original individual.

4. Analysis of Example

A job shop is composed of 6 machines and it needs to produce 4 kinds of jobs. Every job need three processes, and each process can select and process on three different machine. The deadline of jobs are shown in table 1, the preparation and processing cost coefficient are shown in table 2, the preparation and processing time of every process are shown in table 3.

| Table 1. The deadline (hour) of jobs. | Table 2. The cost coefficient of preparation and processing ($). |
|--------------------------------------|---------------------------------------------------------------|
| job | 1 | 2 | 3 | 4 | 5 | 6 | machine | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|---|---|---|---|---|---|---------|---|---|---|---|---|---|
| Deadline | 19 | 15 | 19 | 20 |     |     | Preparation | 15 | 16 | 18 | 16 | 15 | 17 |
|       |     |     |     |     |     |     | Processing | 40 | 45 | 50 | 40 | 47 | 43 |

| Table 3. The preparation and processing time (hour) of jobs. |
|-------------------------------------------------------------|
| job | process | Machine |
|-----|---------|---------|
|     | 1 | 2 | 3 | 4 | 5 | 6 |     | 1 | 2 | 3 | 4 | 5 | 6 |
| 1   | 2,3 | --- | 2,5 | --- | 1,2 | --- |
| 2   | 2,4 | --- | --- | --- | 3,6 | 2,3 |
| 3   | --- | --- | 1,4 | --- | 2,7 | 3,4 |
| 1   | 2,2 | 2,3 | 2,3 | --- | --- | --- |
| 2   | --- | 1,3 | --- | 2,2 | 3,2 | --- |
| 3   | 1,1 | 2,4 | 3,5 | --- | --- | --- |
| 1   | 1,6 | --- | 2,5 | 2,4 | --- | --- |
| 3   | 2,6 | --- | 3,4 | --- | 3,5 | --- |
| 3   | 2,2 | --- | 2,3 | --- | 3,3 | --- |
| 2   | 2,6 | 2,5 | 3,7 | --- | --- | --- |
| 3   | --- | 2,4 | 1,3 | --- | 2,5 | --- |
| 3   | --- | 2,5 | 1,6 | --- | 2,7 | --- |
The basic parameters of this modified SOMA algorithm are as follows: Stepmax=0.7, Stepmin=0.4, PRT=0.1, PathLength=3. Through the simulation of the algorithm, the optimal value of the objective function is 0.9609, the cost is 1991 and the completion time of jobs are 15,11,17,18h. The total machine load is 61. The Gantt chart of its scheduling is shown in figure 1.

![Gantt chart](image)

**Figure 1.** The Gantt chart of 4*6 scheduling problem.

5. Conclusion
This paper put forward an efficient modified SOMA algorithm based on adaptive step and quadratic interpolation, for solving multi-objective job shop scheduling problem with the criterion to minimize the processing cost, minimize makespan, minimize the total machine loads. The performance of the algorithm is improved effectively by adaptive step and quadratic interpolation, which use the adaptive step improve the ability of the exploration and exploitation, use the quadratic interpolation to improve the diversity of population. Finally, check this algorithm with example of 4×6 job shop scheduling, and the computational results show that the proposed modified SOMA efficiently solves job shop scheduling. In the future work will use the modified SOMA to solve the large scale adaptive job shop scheduling and try to combine with other algorithms to improve the ability to solve problem.

6. Acknowledgments
This work was supported by The Marine Diesel Engine Cylinder Head Precision Processing Flexible Production Line Development and Application Project(Number: 2012ZX04011-041).

7. References
[1] Driss I, Mouss K N, Laggoun A. A new Genetic Algorithm for Flexible Job-Shop Scheduling Problems. Journal of Mechanical Science & Technology, 2015, 29(3):1273-1281.
[2] Pérez M A F, Raupp F M P. A Newton-based Heuristic Algorithm for Multi-objective Flexible Job-Shop Scheduling Problem. Journal of Intelligent Manufacturing, 2016, 27(2):409-416.
[3] Ahmed Farouk, Khaled El Kilany, Fady Safwat. Genetic Algorithm for Project Scheduling and Resource Allocation under Uncertainty. International journal of Mechanical Engineering and Robotics Research, Vol.2, No.3, July 2013
[4] Liu A J, Yang Y, Xing Q S, et al. Dynamic Scheduling on Multi-objective Flexible Job Shop. Computer Integrated Manufacturing Systems, 2011, 17(12):2629-2637.
[5] Piroozfar H, Wong K Y, Wong W P. Minimizing Total Carbon Footprint and Total Late Work Criterion in Flexible Job Shop Scheduling by Using an Improved Multi-Objective Genetic Algorithm. Resources Conservation & Recycling, 2016.
[6] C Zhu, W Qiu, C Zhang, et al. Multi-objective Flexible Job Shop Dynamic Scheduling Strategy aiming at Scheduling stability and robustness. China Mechanical Engineering, 2017.28(2):173-182.
[7] Mokhtari H, Hasani A. An Energy-Efficient Multi-Objective Optimization for Flexible Job-Shop Scheduling Problem. Computers & Chemical Engineering, 2017.
[8] Zelinka I. SOMA — Self-Organizing Migrating Algorithm, Self-Organizing Migrating Algorithm. Springer International Publishing, 2016:167-217.

[9] Davendra D, Zelinka I, Bialic-Davendra M, et al. Discrete Self-Organising Migrating Algorithm for flow-shop scheduling with no-wait makespan. American Institute of Physics Conference Series. American Institute of Physics, 2011:285-289.

[10] WENG Chunyi, LI Yuanxiang, WANG Lingling, et al. Modified self-organizing migrating algorithm with linear-digress step. Computer Engineering and Applications, 2011, 47 (18):26-28.

[11] Singh D, Agrawal S. Self-Organizing Migrating Algorithm with Quadratic Interpolation for Solving Large Scale Global Optimization Problems. Applied Soft Computing, 2016, 38:1040-1048.