Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2015, 7(4):322-325

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

An improved genetic algorithm to the job shop scheduling problem

Yu Yan-Fang and Ying Yue

Open and Distance Education Research Institute, Zhejiang Radio & Television University, Hangzhou, P. R. China

---

ABSTRACT

For a dynamic, changeable agile manufacturing system, a dynamic job shop scheduling approach is one of effective measures for production management. In this paper, an improved genetic algorithm is proposed to the job shop scheduling problem. The experimental results suggest that this improved genetic algorithm is correct, feasible and available. The data-driven optimization method is a new approach to study the agile manufacturing system.

Key words: Job shop scheduling, data-driven, genetic algorithm, dynamic optimization

---

INTRODUCTION

In recent years, more and more scholars have begun to focus on and study the agile manufacturing system oriented job shop scheduling and dynamic scheduling [1-2]. Chao-Yong Zhang et al. investigated the dynamic scheduling in the cases of delayed arrival, processing of raw materials and assembling, as well as adding the new jobs urgently [3]. Brank et al. studied the dynamic scheduling in the case of random arrival of jobs, and improved the scheduling performance through combining the early idle time [4]. Liu et al. shortened the idle time earned from production process by the job selection rule to improve the scheduling performance [5]. Adibi studied how to handle the dynamic events - the random arrival of jobs and machine error, and assessed the scheduling performance through minimizing the processing time and the job delay [6]. Considering adding new machines, Fattahi investigated a dynamic flexible job shop scheduling problem (JSSP) with changeable arrival time of new jobs and processing time, and measured the effectiveness of scheduling by minimizing time of completion [7].

With continuous upgrade and a wide application of internet of things, mass data of device, process and production reflecting running state of the agile manufacturing system have been collected and stored [8-9]. However, in practical optimization process, it is usually lack of an efficient statistic, analysis and evaluation of these data, so they cannot be converted to useful information to management departments.

AN IMPROVED GENETIC ALGORITHM

The Genetic algorithm derives from biogenetics and the natural law of survival of the fittest. It is an intelligent optimization method with an iterative process of ‘survival + detection’. Genetic algorithms express problem solving as the survival of the fittest of chromosomes, and converge to an individual the most adaptable to the environment finally by step iteration of chromosomes. In the optimization process of existing genetic algorithms, it often lacks an effective statistic, analysis and application of off-line and on-line data of optimization systems. In view of this, this paper attempts to design and implement a data-driven genetic algorithm for the dynamic JSSP. And its basic framework is as shown in Fig. 1.

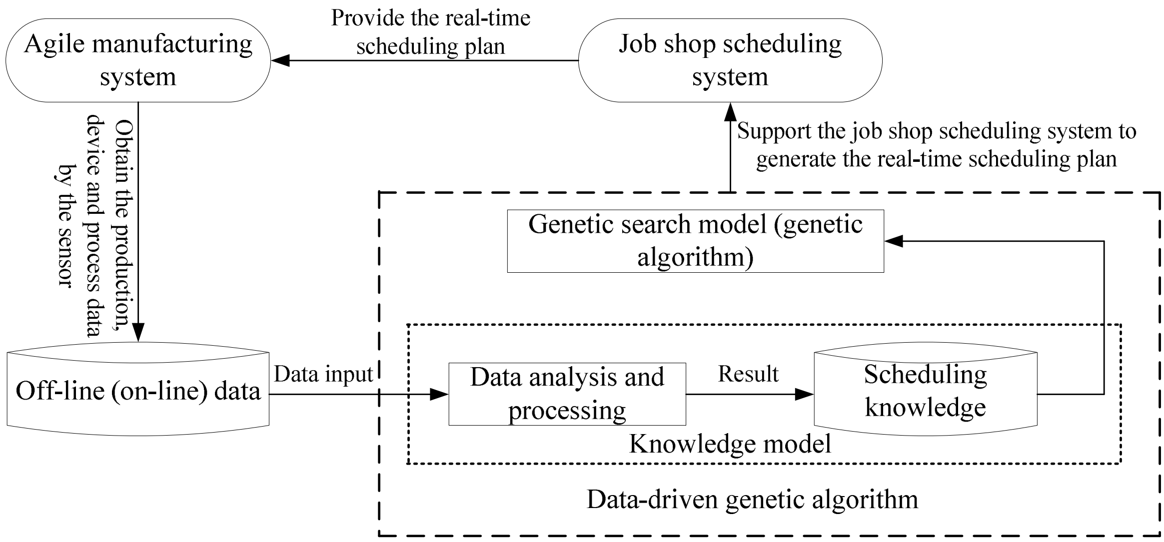

Fig. 1. The Framework of Improved Genetic Algorithm

In this paper, the data-driven genetic algorithm is defined as a hybrid genetic algorithm combining the genetic optimization model and the knowledge model. In this algorithm, the genetic optimization model searches the feasible space of the optimization problem according to the 'neighborhood search' strategy. While the knowledge model mines useful scheduling knowledge from mass off-line and on-line data, and then uses it to guide the follow-up optimization processes. The data-driven genetic algorithm is based on the genetic optimization model, and meanwhile, highlights the role of the knowledge model. It optimizes both models to improve the algorithm optimization efficiency.

With the continuous running of the production scheduling system, the off-line and on-line data of device, process and production reflecting the running state have been accumulated constantly. This paper mined the knowledge of parameter, operator and strategy from these off-line and on-line data, and then used it to guide the follow-up optimization processes of genetic algorithms.

(1) The three-stage division for the job shop scheduling plan. When solving the dynamic JSSP, the whole solving process should be planned reasonably to adapt the uncertainty and dynamics of the scheduling system. Considering the above, this paper divided the job shop scheduling plan into three stages: the executed plan, the plan to be executed and the one to be programmed (as shown in Fig. 2). The executed plan represents the scheduling plans executed by the scheduling system until now. And for such plans, they won't be changed basically. The plan to be executed refers to the scheduling plans that will be processed in the scheduling system. They have some dynamics and certainty, needing to be handled immediately by some selfreparing operations. The plan to be programmed refers to the scheduling plans that will be processed in the scheduling system after a longer follow-up time. They have some dynamics and uncertainty, needing to be handled immediately by some genetic algorithms. In the proposed algorithm, the strategy knowledge is used to guide the selfreparing operations, and the operator knowledge and parameter knowledge to guide the genetic optimization procedure. It is important to note that the reasonable division of the plan to be executed and the one to be programmed is usually determined by the historical data, user experience and expert knowledge. And this paper will divide these two plans based on the historical data.

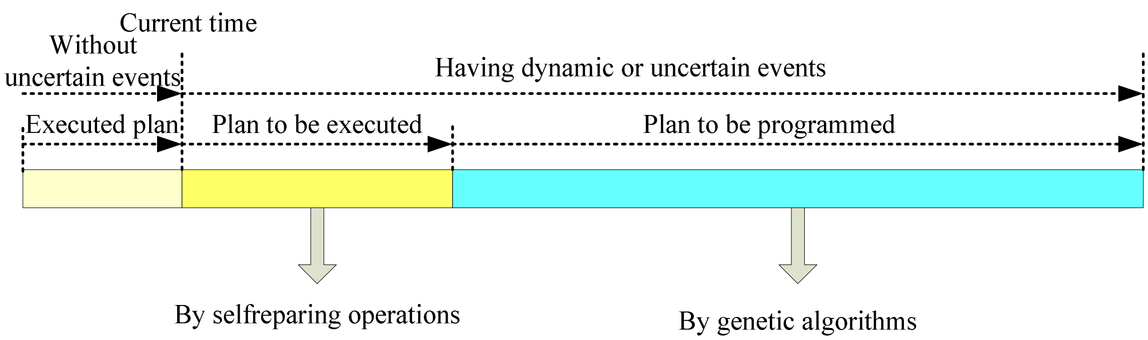

Fig. 2. The three-stage division for the scheduling plan

(2) The selfreparing operation based on strategy knowledge. When solving the dynamic JSSP, this paper mainly employs the following three scheduling strategies: give priority processing to the task with a higher priority (Str_1), the one with the shortest processing time (Str_2) and the one with the longest processing time (Str_3). Here suppose that the work performance of each scheduling strategy is ST_1, ST_2 and ST_3 respectively (at the initial phase, their work performance is initialized to 1). When implementing the selfreparing operation, this paper selects a scheduling strategy based on the following probability:

$$PR_i = \frac{ST_i}{\sum_{k=1}^{3} ST_k} \tag{1}$$

In which, PR_i denotes the selective probability of the i^{th} scheduling strategy.

(3) The genetic optimization operation based on parameter knowledge and operator knowledge. At the initial phase of data-driven genetic algorithm, the author used the orthogonal design method to generate 27 typical parameter combinations, and meanwhile, initialized the work performance of each set of parameters to 1. Before each iterative, the data-driven genetic algorithm adopts the roulette method to randomly select a set of parameters from 27 sets as the parameters for this iteration based on the work performance of parameter combinations. And the compute mode of selective probability of each set of parameters is as below:

$$PR_i = \frac{PA_i}{\sum_{k=1}^{27} PA_k} \tag{2}$$

In which, PR_i denotes the selective probability of the i^{th} parameter combination; PA_i is the work performance of the ith set of parameters.

In the data-driven genetic algorithm for the dynamic JSSP, the author employed five types of crossover operators and three types of mutation operators to implement its optimization process together. Here, suppose that the work performance of each crossover operator is OPX_i (i=1,2,…,5) and that of each mutation operator is OPM_i (i=1,2,3) respectively. Before each iterative, this algorithm will select a kind of crossover operator and mutation operator with the following probabilities respectively.

$$PRX_i = \frac{OPX_i}{\sum_{k=1}^{5} OPX_k} \tag{3}$$

$$PRM_i = \frac{OPM_i}{\sum_{k=1}^{3} OPM_k} \tag{4}$$

RESULTS

As shown in Table 1, five instances are used to test the feasibility and correctness of the proposed algorithm.

Table 1. Five instances of job shop scheduling

Name of instance	Number of jobs	Number of machines	Sum of processes
MK06	10	10	150
MK07	20	5	100
MK08	20	10	225
MK09	20	10	240
MK10	20	13	240

In order to simulate the uncertainty and dynamics in the real job shop scheduling system, this paper adopts the following way to add some uncertainty and dynamics to the above instances:

(1) Randomly add 5% new processes, and their processing time on the j^{th} machine is λT_{ij}. Here, λ is a random number in an interval [0.8, 1.2], T_{ij} is the average processing time for all the processes of the i^{th} task on the j^{th} machine.

(2) Randomly change the processing time of 5% existing processes, which is changed to λT_{ij}. Here, λ is a random number in an interval [0.8, 1.2], T_{ij} is the processing time for the current process of the i^{th} task on the j^{th} machine.

The author employed the Visual C++ language design and implemented the data-driven genetic algorithm for the dynamic JSSP. All experiments were completed in a desk computer with a CPU of Pentium Dual Core 2G and 2G memory. In order to assess the optimization performance of the proposed algorithm more objectively, this paper solved each instance 100 times by this genetic algorithm and the average value of experimental results was taken as the final result. The experimental results are shown in Table 2.

Table 2. Experimental results of different instances

Name of instance	Sum of processes	Variable quantity of Total processing time (%)	Variable quantity of Process processing plan (%)
MK06	150	8.13	6.72
MK07	100	6.28	5.51
MK08	225	9.27	7.32
MK09	240	11.05	8.15
MK10	240	9.82	7.18

It can be seen from Table 2, although some uncertainty and dynamics were added to these five instances respectively, the proposed algorithm can solve these uncertain events effectively. For the variable quantity of total processing time, that of the instance MK07 is minimum (6.28%), MK10 is maximum (11.05%), and their average value is 8.91%. For the variable quantity of process processing plan, that of MK07 is minimum (5.51%), MK09 is maximum (8.15%), and their average value is 6.98%. So from the above results, it is not hard to see whether the variable quantity of total processing time or that of process processing plan, the data-driven genetic algorithm can deal with the above dynamics and uncertainty by a smaller variable quantity (less than 10%). In general, the application examples have shown the correctness, feasibility and usability of the proposed algorithm.

CONCLUSION

In the modern dynamic job shop scheduling system, mass production, device and process data have been collected and stored. However, in the real optimization process, they cannot be converted into useful information for management departments. In view of this, a dynamic scheduling method based on the data-driven genetic algorithm was proposed. And the application examples have shown the correctness, feasibility and usability of the proposed algorithm. The data-driven optimization method created a new way to study the agile manufacturing system.

Acknowledgements
Supported by the education science and planning project of Zhejiang province and the science and technology project of Quzhou science and Technology Bureau.

REFERENCES

[1] OY Adisa, *Construction Management and Ec.*, **2010**, 28(5), 509-526
[2] JS Shane, *J. Manage. Eng.*, **2009**, 25(4), 221-229
[3] Lin C; Hu J; Kong XZ, *Chinese Journa*., **2012**, 35(1), 1-15
[4] L Dudek, *ACM Trans. Comput. Hum. Interact*., **2006**, 18(1), 1-30
[5] Li XQ, J. *Intell*., **2010**, 28(5), 763-767
[6] Pang QH, *Computer Engineering*, **2007**, 33(18), 59-61
[7] Zhou RG, *Computer Engineering and Applications*, **2007**, 43(31), 102-105
[8] Pang QH, *Computer Engineering*, **2007**, 33(18), 59-61
[9] Zhou RG, *Computer Engineering and Applications*, **2007**, 43(31), 102-105