International Journal of Hybrid Information Technology
Vol. 9, No. 5 (2016), pp. 303-310
http://dx.doi.org/10.14257/ijhit.2016.9.5.25

Flexible Workshop Scheduling Optimization Based On Multi-agent Technology

Jiang Xuesong[1], Tao Sun[2], Tao Qiaoyun[3] and Jian Wang[4]

[1,2,3] Qilu University of Technology
Jinan, China
[4]Shandong College of Information
Jinan, china
jxs@qlu.edu.cn, suntao0906@163.com, tqy198909@163.com,
wj7059830570@sina.com

Abstract

Considering the complexity of flexible workshop scheduling, combined with plant production process characteristics and constraints, we constructed a multi-agent system model to solve multi-objective flexible workshop scheduling problems. This paper proposed an algorithm which was a combination of the ant colony algorithm and Q-learning algorithm. This paper also analyzed and implemented how to solve the workshop scheduling optimization problem. Finally, this paper proved the validity of methods to solve the multi-objective flexible workshop scheduling optimization problems with examples on JADE platform.

Keywords*: flexible workshop scheduling, multi-agent, multi-objective, ant colony algorithm, Q-learning algorithm, JADE, optimization*

1. Introduction

The flexible workshop scheduling problem is a core part of business management, which is a problem of assembles optimization, a simplified model of production scheduling problem and a typical NP-complete problem [1]. In a manufacturing environment, workshop scheduling is a core model of organizing production, adapting to environmental changes both inside and outside and external collaboration. In the actual dynamic workshop, the occurrence of random events will bring a lot of uncertainties. In order to solve these problems, we need to have the workshop scheduling system more flexible to meet the demands of diversity, openness to accept new features functions and intelligence for autonomous production planning. In this regard, scholars put forward various optimization methods, such as genetic algorithms, ant colony algorithm, particle swarm optimization, taboo search algorithms, multi-agent algorithms and their combinations which were in good application [2-3]. SCCheng [4] proposed solving workshop scheduling issues of describing static and dynamic production environment using genetic algorithm. L.De Giovanni and F.Pezzella [5] proposed a FMS scheduling and re-scheduling algorithm. Quanyong Ju [6] proposed a multi-population particle swarm hybrid algorithm with tendentious search to improve search efficiency and quality, combining the advantages of multi-swarm particle swarm search and genetic algorithm. It was proposed an optimization goal with time, cost and quality integrated, and designed a hybrid particle swarm optimization with flexible workshop scheduling optimization as model and optimal target as calculation method [7]. This paper proposed building a goal combining ant colony intelligence and Q-learning algorithm with rational allocation of resources, effective use, the highest efficiency and the lowest cost based on multi-agent flexible workshop scheduling model.

ISSN: 1738-9968 IJHIT

2. Multi-objective Flexible Workshop Scheduling

2.1 Problem Description

The flexible job-shop scheduling considers n jobs to be processed on m machines, where each job i consists of a sequence of n_i operations V_{ij}, j=1, 2, …, n_i. For the flexible job-shop scheduling, it needs to determine both the assignment of machines and the sequence of operations on all the machines to optimize multiple scheduling objectives under the conditions meet the constraints.

Given the flexible workshop scheduling problem, some assumptions are made as follows: All jobs can be started at time 0.All machines are available at time 0.Each machine can process only one job at a time. Each job can be processed by only one machine at a time. Each machine cannot be interrupted before it finishes the job's work. The process of an operation cannot be interrupted once started. Machine breakdown does not occur, which means all machines are continuously available throughout the production stage.Job transportation time among machines is not considered.

2.2. Mathematical Model

In this paper, the following three objectives which include the minimum processing cost, the shortest processing time and the highest rate of qualified products are to be optimized. Mathematically, the corresponding optimization model is described as follows:

$$f_1(\mathrm{x}) = \min \sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{ijk} t_{ijk} \qquad (1)$$

$$f_2(\mathrm{x}) = \min \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{n_i} c_{ijk} x_{ijk} = \min \sum_{m=1}^{M} \sum_{i=1}^{N} \left(\sum_{j=1}^{n_i} E_{ijk} x_{ijk} + \sum_{j=1}^{n_i} V_{ijk} x_{ijk}\right) \qquad (2)$$

$$f_3(\mathrm{x}) = \max \sum_{m=1}^{M} \sum_{i=1}^{N} h_{ij} \qquad (3)$$

$$\sum_{k=1}^{M} S_{ijk} x_{ijk} \geq \sum_{k=1}^{M} \left[(\mathrm{S}_{i(\mathrm{j}-1)\mathrm{k}} \mathrm{t}_{i(\mathrm{j}-1)\mathrm{k}}) \right] x_{i(\mathrm{j}-1)\mathrm{k}} \qquad (4)$$

$$X_{ijk} = \begin{cases} 1, & if\ V_{ij}\ is\ done\ on\ machine\ k \\ 0, & elsewhere \end{cases} \qquad (5)$$

$$R_{ijmnq} \begin{cases} 1, & if\ V_{ij}\ is\ done\ on\ machine\ i\ in\ priority\ m \\ 0, & elsewhere \end{cases} \qquad (6)$$

We use the following notations:

n	Number of jobs	v_{ij}	The j operation of job i ;
m	Number of machines	h_{ij}	Product performance indicators
X_{ijk}	The operation j of job i is assigned to machines k	t_{ijk}	Completion time of the operation j in job i by machine k
E_{ijk}	Machine cost of operation v_{ij} on machine k	S_{ijk}	Start time of operation v_{ij} on machine k
C_{ijk}	the i step of the operation on the j path of the machine k processing costs;	R_{ijmnq}	Sequence between operation i on the machine m and operation j on the machine n

2.3. Restrictions

Time constraints: the adjacent processes of the same job should be started chronologically according to technological requirements.

Resource constraints: you must complete the current task before starting the next one on the same machine. No machine can process two jobs in both same and different processes synchronously.

Other constraints: any process can only be processed on a machine at a fixed time.

3. Workshop Scheduling Multi-agent Mode

Multi-agent system is composed of many independent and coordinating agents, each of which has different solutions and functions. These agents communicate, cooperate and solve complex problems according to pre-appointed protocols [8].Through the collaboration among agents, machines have intelligence, and the automation and optimization of job scheduling will be implemented. In the workshop scheduling, Multi-agent system mainly consists of by a global agent, management agent, machine intelligence agent and the job agent. Each agent is dynamically scheduling the workshop through pre-appointed protocols of unified communications, and makes decisions by tender, negotiation, *etc*. Using the multi-agent architecture, the system will not entirely collapse as a result of some part error, which is helpful to improve the stability of the system. Moreover, implementing distributed decision-manufacturing system, the system has strong robustness and scalability [9]. The Figure of the multi-agent system dynamic scheduling model is as follows.

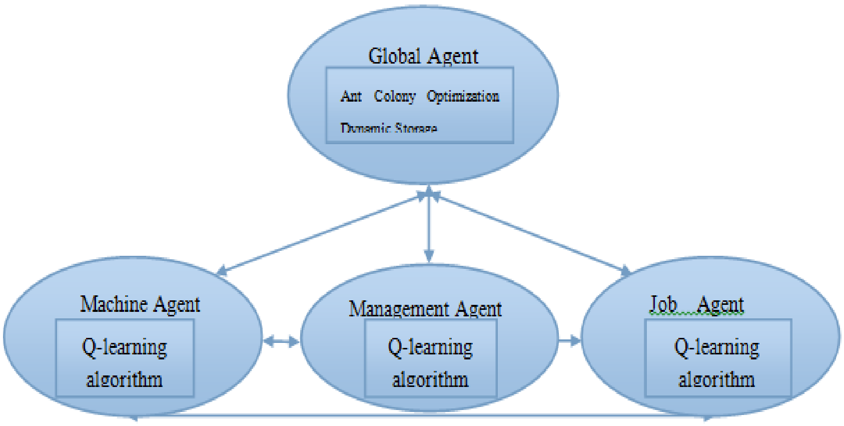

Figure 1 Multi-Agent System Dynamic Scheduling Model

Scheduling system works as follows:

step1: Global agent builds participants assembles of this schedule, and creates a temporary multi-agent system;

step2: the job agent and the machine intelligence agent invite bids by CNP, and determine the initial scheduling scheme. Each job agent submits their scheduling result to the management agent;

step3:the management agent sends the initial scheduling scheme to all machine intelligence agents, then each machine intelligence agent has the scheme locally optimized through Q- learning algorithm and sends the best solution back to the global agent;

step4:the global agent makes a global optimization and gains the optimal scheduling scheme by calling the ant colony optimization algorithm.

4. Q-Learning Algorithm

Q-learning is one of the main algorithms, which was firstly proposed by Watkins in his doctoral dissertation in 1989 [10]. A key assumption, which Q- learning is based on, is that the interaction between the intelligent agent and the environment can be regarded as a Markov decision process (MDP), which means the current state and the selected action of the intelligent agent determine a fixed state transition probability distribution and the next state, and obtain an instant reply. The goal of Q- learning is to find a strategy to make the largest profits in the future.

4.1 Algorithm Description

In Q-learning, Q (s_t, a_t) represents an action of at according to some kind of strategy in a state of st. Each Q (s, a) corresponds to a Q value, which decides to choose an action during the learning process. Q value is the total profits, if the current relevant action is implemented according to some kind of strategy. The optimal Q value is the total profits when the current relevant action is implemented according to some kind of strategy. Its definition is as follows:

$$Q(s_t,a_t)=Q(s_t,a_t)+a_t[r(s_t,a_t)+\gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \qquad (7)$$

In this formula, a_t represents the learning factor; γ is the discount factor; $(\mathrm{s}_t, \mathrm{a}_t)$ means the state - action pair at t time; s_{t+1} is the state at t+1 time.

Q learning chooses the optimal strategy according to the formula below:

$$\pi^*(\mathrm{s}_t) = \arg\max_{a_t} Q(\mathrm{s}_t, \mathrm{a}_t) \tag{8}$$

As can be seen by the formula above, the optimal strategy when Q learning chooses the action is helpful to make the best cumulative profits. According to the formula, the machine intelligent agent and the job agent take actions, and acquire the optimal action steps through value analysis.

4.2. Algorithm Steps

The algorithm steps of the machine intelligent body are as follows:

```
Initialize Q(s,a)  arbitrarily;
Repeat (for each episode);
Initialize s;
Repeat (for each step of episode);
Choose a from s using policy derived from  Q(e.g., -greedy);
Take action a,observe r,s'
        Q(s,a) ⟵ $Q(s_t,a_t)+a_t[r(s_t,a_t)+\gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]$
      s ⟵ s';
Until s is terminal;
```

5. Global Ant Colony Optimization Algorithm

Although Q- learning algorithm can achieve a local optimization to certain extent, it does not represent a global optimization. This paper introduces ant colony optimization algorithm to optimize the overall agent. In ant colony algorithm, the solution to each optimization problem is to search an ant in space. The ants update rules through the pheromone and update the optimal solution through the state transition rules in each iteration.

5.1. Algorithm Description

Ant colony algorithm, firstly proposed by MDorigo [11], an Italian scholar, is a new heuristic optimization algorithm inspired by ant behavior in nature, which can be applied to the swarm intelligence algorithm of a variety of optimization assembles problems.

Ant colony algorithm simulates the mechanism of ants' finding the shortest path from the nest to food sources and the path back to the nest in nature. Ants communicate with each other through the pheromone. When moving, ants disseminate the pheromone on the path, through which ants can communicate with each other. They do not communicate with each other individually; however they just do so by changing their common environment. The individual impacts the behavior of others by changing their environment, which forms a positive feedback mechanism. The shorter length the road is, the more ants pass, the more pheromones are gathered. Ants tend to choose the path with more pheromones, through which they eventually find the shortest path over a period time.

This algorithm adds an external set of BP (t). When many ants go into BP (t) at the same time, in order to distinguish the pheromone increment of each ant, Θ(t),which is the minimum distance between the ith ant which is the last one going into BP (t) and the objective function value of the solution of BP (t), is used as the pheromone released at the ith ant's position.

$$\theta(\mathrm{t}) = \min \sqrt{\sum_{i=1}^{n} (\mathrm{f}_i(\mathrm{x}) - \mathrm{f}_i(\mathrm{x}_v))^2} \quad \mathrm{x}_v \in BP(\mathrm{t}) \qquad (9)$$

Therefore, the definition of ant colony algorithm pheromone is updated as follows:

$$\tau_i(\mathrm{t}+1) = \begin{cases} \rho\tau_i(\mathrm{t}) + \theta(\mathrm{t}), \mathrm{x} \in BP(\mathrm{t}+1) \\ \quad \rho\tau_i(\mathrm{t}), \text{Otherwise} \end{cases} \qquad (10)$$

ρ is represents the evaporation rate.

The formula of the state transition rule of ants is as follows:

$$P_{ij}(\mathrm{t}) = \begin{cases} \dfrac{\left[\tau_{ij}(\mathrm{t})\right]^{\alpha} (\eta_{ij})^{-\beta d_{ij}}}{\sum_{s \in J_k} \left[\tau_{is}(\mathrm{t})\right]^{\alpha} (\eta_{is})^{-\beta d_{ij}}}, (d_{ij} < 0 \cap i \neq j) \cup (d_{ij} = 0 \cap i = 1) \\ \qquad 0, Otherwise \end{cases} \qquad (11)$$

$d_{ij} = f(x_i) - f(x_j)$ and $d_{ij} < 0 \wedge i \neq j \cup d_{ij} = 0 \wedge i = j$. α and β indicate the emphasis degree of the amount of information on the node and the heuristic information respectively. η_{ij} indicates the visibility of information, which takes the reciprocal of the objective function increment and remains unchanged during the process of finding solutions. Probability formula ensures that ant i moves to the area of ant j to obtain a better solution according to the probability, or stays in situ.

5.2. Algorithm Steps

When the global agent receives information from each machine intelligence agent, it calls ant colony algorithm from the library to make global optimization. The process is as follows:

Step 1: the initial ACO comes from the encoding information fed back by machine intelligence agents. Initialization algorithm parameters include the number of ants (k), iterations (N), initial pheromone (τ_0), all processes collection ants will visit (G_k), the next process collection needing to be visited (S_k), the processes collection ants have passed (J_k). Every ant is randomly assigned to G_k . In this case the number of iterations n = 0;

Step 2: choose search paths. Ant k (k = 1, 2...) chooses the next process according to transition probability equation (9) in collection S_k .

Step 3: update the processes collection. Update BP (t). After ant k chooses a process according to step 2, it is added into J_k . At the same time, this process is removed from G_k and S_k , and S_k is updated. If this process is not the last one, the subsequent processes are added into S_k . The above process is repeated till G_k is empty.

Step 4: update the pheromone. Update the pheromone using equation (9).

Step 5: repeat from step 2 to step 5, start the next generation of ants search, and look for the global optimization and the iterative optimal ants till the termination condition is met.

6. Example Applications

6.1. Example Overview

Taking a lifting equipment painting workshop of a heavy machinery group as an example, this paper makes a flexible workshop scheduling multi-objective optimization. This workshop has primary equipments necessarily needed for painting process, which includes pretreatment device, cathode electrophoresis equipment, dryers, grinding machines, painting robot, online testing equipment, *etc.* T_{ijk} , C_{ijk} and h_i can be obtained from the record data of the job process and status on a production line of a workshop. After calculation, 6 jobs, 10 devices, and the flexible workshop scheduling problem of a number of processes of each job are obtained.

6.2. Experimental Environment

Intel (R) Core (TM) i5-34700 CPU @ 3.20GHz, 4.00GB RAMS, Windows 7 OS. This experiment uses JADE to develop multi-intelligence agent system and programs in Eclipse.

6.3 Computational Results

After several experiments, the optimum parameters by using Q learning algorithm and the global intelligence agent are listed below.

Table 1. The Best Parameters

Q-Learning Algorithm	Ant Colony Algorithm
a =0.3	a =0.4
γ =0.9	β =0.7

In order to verify the advantage of the algorithm to deal with the problem of multi-objective flexible workshop scheduling optimization, this paper made comparisons using used a few algorithms under the condition of same parameters. Table 2 lists the comparison result of several algorithms' running 50 times respectively. This result shows that this algorithm is better to solve the problem of multi-objective flexible workshop scheduling problem with better performance in processing time and cost.

Table 2. Comparison between the Proposed Algorithm and Other Algorithms

Optimization goal	Herein algorithm		Q-learning algorithm		Ant colony algorithm		Genetic - Particle Swarm	
	Optimal solution	average solution	Optimal solution	average solution	Optimal solution	average solution	Optimal solution	average solution
Processing costs	95	257	112	297	127	322	101	296
Processing time	206	312	256	332	277	374	227	330
Finished pass rate	89	74	79	69	75	64	84	72

7. Summary

Compared to multi-agent technology, this paper constructed multi-agent workshop scheduling system model for solving workshop scheduling problems. Meanwhile, the multi-objective mathematical model is established. Q-learning algorithm was used to achieve a local optimum and ant colony algorithm was used as a global optimum in machine intelligence agents and job agents. The experimental results show that the combination of Q-learning algorithm and ant colony algorithm can effectively solve workshop scheduling problems.

Acknowledgement

Thanks Project supported by the projects of Key research and development program of Shandong Province (No. 2015GGX106003), Project supported by the projects of Shandong Provincial Natural Science Foundation, China (No. ZR2014FQ021), a Project of Shandong Province Higher Educational Science and Technology Program (No.J15LN03).

References

[1] H. Ting, C. Hai, “Workshop Production Scheduling Problem”, Journal of Mechanical Engineering, vol. 36, no. 5, **(2000)**, pp. 97-102

[2] A. N. Haq, T. Karthikeyan and M. Dinesh, “Scheduling Decisions in FMS Using A Heuristic Approach”, International Journal of Advanced Manufacturing Technology, vol. 22, no. 5-6, **(2013)**, pp. 374-379.

[3] K.-Y. Chen and C.-J. Chen, “Applying Multi-Agent Technique in Multi-Section Flexible Manufacturing System”, Expert Systems with Applications, **(2010)**, vil. 37, no. 11, **(2010)**, pp. 7310–7318.

[4] S.C. Cheng, D. F. Shiau and Y. M. Huang,“Dynamic Hard-Real-Time Scheduling using Genetic Algorithm for Multiprocessor Task With Resource and Timing Constraints”, Expert Systems with Applications, vol. 36, no. 1, **(2009)**, pp. 852–860.

[5] L.D Giovanni and F. Pezzella, “An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling Problem”, European Journal of Operational Research, vol. 200, no.(2), **(2010)**, pp. 395-408.

[6] T. L. Lin, S. J Horng and T. W. Kao, Tzong, *et al*, “An Efficient Job-Shop Scheduling Algorithm Based on Particle Swarm Optimization”, Expert Systems with Applications, vol. 37, no. 3, **(2014)**, pp. 2629–2636.

[7] C.J.C.H. Watkins, “Learning from Delayed Rewards”, Ph D Thesis Cambridge University Psychology Department, **(1989)**.

[8] Z. Yong, D. Xing and W. Xiaojuan, *etc*, “Improved Non - Dominated Sorting Genetic Algorithm for Multi-Objective Flexible Job Shop Scheduling”, Journal of Mechanical Engineering, vol. 46, no. 11, **(2010)**, pp. 156-164

[9] P. Brucker and R. Schile, “Job-Shop Scheduling with Multipurpose Machines”, Computing, vol. 45, no. 4, **(1990)**, pp. 369-375.

[10] A. Motaghedi, K. Sabri-Laghare and M. Heydari, “Solving Flexible Job Shop Scheduling Problem With Multi Objectives”, International Journal of Industrial Engineering and Production Research, vol. 21, **(2010)**, pp. 197-209.

[11] Q. Zhang, H. Manier and A. Manier, “A Genetic Algorithm with Tabu Search Procedure for Flexible Job Shop Scheduling with Transportation Constants and Bounded Proceding Times”, Computers and Operations Research, vol. 39, **(2012)**, pp. 1713-1723.

[12] J.Q Li, Q.K Pan and K.Z. Gao, “Pareto-Based Discrete Artificial Bee Colony Algorithm for Multi-Objective Flexible Job Shop Scheduling Problems”, International Journal Adv Manu Tech., vol. 55, no. 9–12, **(2011)**, pp. 1159–1169**.**

[13] Zhang deposit, Z. Pi-e and X. –D. Wu, “Colony and Particle Swarm Optimization Algorithm-Based Solution to Multi-Objective Flexible Scheduling Problems”, Journal of Computer Applications, vol. 27, no. 4, **(2011)**, pp. 936-938.

[14] J. Li and P. Feng, “Improved Ant Colony Algorithm for Solving Multi-Objective Optimization Problem”, Southern Yangtze University: Natural Science, vol. 12, no. 4, **(2013)**, pp. 394-398.

[15] “Shi Marching Focus of Army, Tao Delivery under Penalty Pareto Multi-Objective Flexible Job Shop Scheduling Optimization”, Journal of Mechanical Engineering, vol. 48, no. 12, **(2012)**, pp. 184-192.