Int. J. Production Economics 141 (2013) 685–697

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Flexible job shop scheduling with due window—a two-pheromone ant colony approach

Rong-Hwa Huang[a,1], Chang-Lin Yang[a,*], Wei-Che Cheng[b,2]

[a] Department of Business Administration, Fu Jen Catholic University, 510, Zhongzheng Road, Xinzhuang District, New Taipei City, Taiwan, ROC
[b] Department of Industrial Management, National Taiwan University of Science and Technology, 106, no. 43, Keelung Road, Taipei, City, Taiwan, ROC

ARTICLE INFO

Article history:
Received 11 November 2011
Accepted 10 October 2012
Available online 26 October 2012

Keywords:
Flexible job shop scheduling
Parallel machine
Time window
Ant colony optimization
Two-pheromone ant colony optimization

ABSTRACT

Recently, the companies reduce the manufacturing costs and increase capacity efficiency in the competitive environment. Therefore, to balance workstation loading, the hybrid production system is necessary, so that, the flexible job shop system is the most common production system, and there are parallel machines in each workstation. In this study, the due window and the sequential dependent setup time of jobs are considered. To satisfy the customers' requirement, and reduce the cost of the storage costs at the same time, the sum of the earliness and tardiness costs is the objective. In this study, to improve the traditional ant colony system, we developed the two pheromone ant colony optimization (2PH-ACO) to approach the flexible job shop scheduling problem. Computational results indicate that 2PH-ACO performs better than ACO in terms of sum of earliness and tardiness time.

1. Introduction

Operation scheduling is designed to efficiently use production resources to explore how to organize work activities, including controlling inventories, reducing delays in parts delivery, reducing machine idle costs, and so on, to improve capacity and customer satisfaction. Among the various scheduling environments, flexible job shop scheduling with a work order to the different characteristic of the machine processing, in line with actual production conditions, this study focused on flexible job shop scheduling research. Generally, solutions for such scheduling problems can be divided into two types: the first approach comprises the optimal solution methods, including integer programming, dynamic programming, branch and bound, the enumeration method and so on. The advantage of optimal solution methods is that they guarantee the most accurate solution, but their disadvantage is that they are too time-consuming and impractical. The second approach involves meta-heuristic methods, such as the ant colony algorithm (ACO), genetic algorithm (GA), simulated annealing, tabu search, neural networks, etc.; the heuristic solution method has the advantage of being free of absolute limitations, but suffers weaknesses in unstable solution quality, and having only a limited ability to obtain accurate solutions. To date, no specific solution method has been obtained that is applicable to all scheduling problems. Regarding the flexible job shop problem (JSP) with due window, Sivrikaya-Serifoğlu and Ulusoy (1999) considered a parallel machine scheduling problem with a time window and sequence-dependent setup time, designed to minimize total tardiness cost. A genetic algorithm (GA) with crossover and non-crossover operators was utilized to solve the problem. The test result for 960 randomly generated samples demonstrates that the algorithm has good solving capability. Huang and Yang (2008) applied the ant colony optimization (ACO) algorithm to solve JSPs using the due date rather than a time point as a time window. The objective was to minimize total early and late penalties. Their test results indicated that the ACO algorithm is both effective and efficient for solving JSPs. Huang et al. (2010) solved the problem of a parallel machine production system with a common due date window using a simple algorithm based on the unique characteristics of early and tardy jobs. Data test results demonstrated that their algorithm rapidly obtained the best solution.

To make the scheduling obtained in the study more closely reflect actual production issues, job shop production is flexible, the workstation contains parallel machines, and the due window enhances customer satisfaction. In this study, the due window and the sequential dependent setup time of jobs are considered in the flexible job shop scheduling problem. To balance the workload on each workstation and improve the production efficiency, there are parallel machines in each workstation. This scheduling problem is a NP-hard. The ACO algorithm yields

* Corresponding author. Tel.: +886 2 29052657; fax: +886 2 29018475.
E-mail addresses: 026299@mail.fju.edu.tw (R.-H. Huang), 051125@mail.fju.edu.tw (C.-L. Yang), d9901004@mails.ntust.edu.tw (W.-C. Cheng).
[1] Tel.: +886 2 29052678; fax: +886 2 29018475.
[2] Tel.: +886 2 27396328; fax: +886 2 27376344.

0925-5273/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2012.10.011

excellent outcomes for such a scheduling problem (Hani et al., 2007; Huang and Liao, 2008). However, the ACO algorithm suffers limitations owing to the varying problem characteristics. In response to these limitations, numerous improvements to the ACO algorithm have been developed. Song et al. (2006) combined GA and ACO algorithms to solve a fuzzy JSP. Their test results demonstrated that GA-ACO has superior ability to solve GA and ACO alone. Furthermore, Zhao et al. (2008) proposed an improved ACO algorithm with an embedded GA to solve the traveling salesman problem (TSP). The GA was used to identify the most feasible and best solutions, and the ACO algorithm was then applied to identify the optimal solution. Experimental results showed that the proposed ACO algorithm obtained better solutions for benchmark instances within fewer iteration than existing ACO algorithms. Zhang et al. (2008) proposed a hybrid ACO with a Differential Evolution (DEACO) algorithm. In the DEACO algorithm, DE was applied to optimize the pheromone trail in the basic ACO model. The proposed algorithm was tested on the TSP. Experimental results demonstrated that the DEACO algorithm is a feasible and effective ACO model for solving complex optimization problems. Van Ast et al. (2009) introduced a fuzzy ACO algorithm for automated design of optimal control policies for continuous-state dynamic systems. The fuzzy ACO algorithm integrates the multi-agent optimization heuristic of the ACO algorithm with fuzzy partitioning of the system state space. Finally, a simulated control problem was solved to demonstrate the efficacy of the proposed algorithm.

This study proposes a modified ACO algorithm named two-pheromone ant colony optimization (2PH-ACO) which adds the second pheromone group to the ant system in order to solve the scheduling problem faster. This algorithm is expected to improve the solving ability of the traditional ant colony algorithm, and to provide an important tool for future use.

2. Problem definition

2.1. Definition

$$FJc|D_j=[D_j^e,D_j^l],s_{ijk}|W_1\sum_{j=1}^{n}E_j+W_2\sum_{j=1}^{n}T_j$$

This study considers n jobs, and uses the following symbols: FJc is the flexible job shop environment, $D_j=[D_j^e,D_j^l]$ denotes the due window, s_{ijk} represents the setup time at every workstation, $W_1\sum_{j=1}^{n}E_j+W_2\sum_{j=1}^{n}T_j$ is the regular measurement used to minimize the sum of weighted earliness and tardiness.

2.2. Notification

n	total number of jobs
m	total number of stations
J_j	the j-th job
S_i	the i-th station
g_i	the number of parallel machines in S_i
g_{ik}	the k-th machine in S_i
p_{ij}	the processing time of J_j in S_i
D_j^e	due date for the lower bound of J_j
D_j^l	due date for the upper bound of J_j
s_{ipj}	the setup time when J_p is processed before J_j in S_i
IST_{ijk}	the possible start time of J_j in S_i machine k
ST_{ijk}	the start time of J_j in S_i machine k
FT_{ijk}	the finishing time of J_j in S_i machine k
ST_{ij}	the start time of J_j in S_i
FT_{ij}	the finishing time of J_j in station S_i
C_j	the completion time of J_j
E_j	the earliness time of J_j
T_j	the tardiness time of J_j

$$V_{ijk}=\begin{cases}1, & \text{if } J_j \text{ is processed in } S_i \text{ machine } g_{ik}\\ 0, & \text{otherwise}\end{cases}$$

$$X_{ipjk}=\begin{cases}1, & \text{if } J_p \text{ is processed before } J_j \text{ in } S_i \text{ machine } g_{ik}\\ 0, & \text{if } J_p \text{ is processed after } J_j \text{ in } S_i \text{ machine } g_{ik}\end{cases}$$

$$R_{ijk}=\begin{cases}1, & \text{if } J_j \text{ is processed on machine } g_{ik} \text{ in one of combinations of } S_i\\ 0, & \text{otherwise}\end{cases}$$

W_1	the cost coefficient of earliness
W_2	the cost coefficient of tardiness

2.3. Integer programming model

Objective

$$\min W_1\sum_{j=1}^{n}E_j+W_2\sum_{j=1}^{n}T_j \tag{1}$$

Constraints

$$E_j=\max\{D_j^e-C_j,0\}\ \forall j=1,2\ldots,n \tag{2}$$

$$T_j=\max\{C_j-D_j^l,0\}\ \forall j=1,2\ldots,n \tag{3}$$

$$C_j=\max_{i=1}^{m}\{FT_{ij}\}\ \forall j=1,2\ldots,n \tag{4}$$

$$FT_{ij}=\max_{k=1}^{m}\{FT_{ijk}\}\ \forall j=1,2\ldots,n; i=1,2\ldots,m \tag{5}$$

$$FT_{ijk}=\left(IST_{ijk}+p_{ij}\right)\cdot R_{ijk}+s_{ipj}\cdot X_{ipjk}\quad \forall j=1,2\ldots,n;\ i=1,2\ldots,m;\ k=1,2\ldots,g_i \tag{6}$$

$$ST_{ij}=\max_{k=1}^{g_i}\{ST_{ijk}\}\quad \forall j=1,2\ldots,n;\ i=1,2\ldots,m \tag{7}$$

$$ST_{ijk}=IST_{ijk}\cdot R_{ijk}\quad \forall j=1,2\ldots,n;\ i=1,2\ldots,m;\ k=1,2\ldots,g_i \tag{8}$$

$$IST_{ijk}\geq 0\&IST_{ijk}\in \mathrm{N} \tag{9}$$

$$R_{ijk}=\sum_{t=1}^{d}V_{ijkt}\quad \forall j=1,2\ldots,n;\quad i=1,2\ldots,m;\quad k=1,2\ldots,g_i \tag{10}$$

$$FT_{fj}\leq ST_{ij}\quad S_f\prec S_i\quad \forall j=1,2\ldots,n \tag{11}$$

The objective and constraints are explained as follows.

Objective (1): This model is designed to minimize the sum of weighted earliness and tardiness.
Constraint (2): Earliness is the larger of $\left(D_j^e-C_j\right)$ and 0.
Constraint (3): Lateness is the larger of $\left(C_j-D_j^l\right)$ and 0.
Constraint (4): Completion time of each job is the maximum processing time on each machine.
Constraint (5): The finishing time of J_j in each workstation should be taken as the highest recorded finishing time.
Constraint (6): The finishing time of J_j in gik of S_i.
Constraint (7): J_j is the start time in each workstation should be taken as the highest recorded start time.
Constraint (8): J_j is processed in gik of S_i.
Constraint (9): IST_{ijk} is a number exceeding or equal to zero.