Computers & Industrial Engineering 99 (2016) 202–209

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing makespan for solving the distributed no-wait flowshop scheduling problem

Shih-Wei Lin [a,b], Kuo-Ching Ying [c,*]

[a] Department of Information Management, Chang Gung University, Taoyuan, Taiwan
[b] Stroke Center and Department of Neurology, Linkou Chang Gung Memorial Hospital, Taoyuan, Taiwan
[c] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 28 August 2015
Received in revised form 11 April 2016
Accepted 27 July 2016
Available online 28 July 2016

Keywords:
Scheduling
Distributed no-wait flowshop
Iterated cocktail greedy algorithm

ABSTRACT

This paper presents the distributed no-wait flowshop scheduling problem (DNFSP), which is the first attempt in the literature to solve this key problem faced by the manufacturing industry. A mixed integer programming (MIP) mathematical model and an iterated cocktail greedy (ICG) algorithm are developed for solving this problem of how to minimize the makespan among multiple plants. The ICG algorithm presented herein is an enhanced version of the iterated greedy algorithm, and it includes two self-tuning mechanisms and a cocktail destruction mechanism. Exhaustive computational experiments and statistical analyses show that the proposed ICG algorithm is a highly efficient approach that provides a practical means for solving the challenging DNFSP.

1. Introduction

Because of the competitive trend of globalization, distributed multisite production systems are widely used in the manufacturing industry. Although various types of distributed production systems have been broadly applied in diverse industries, the scheduling problems associated with these systems have been analyzed theoretically to a lesser extent, as compared with the classical scheduling problem. Conversely, numerous variants of the flowshop scheduling problem (FSP) have been studied (Yenisey & Yagmahan, 2014) due to the nature of distinct industrial processes. This study addresses a key branch of FSPs, the distributed no-wait FSP (DNFSP), which is a crucial aspect of distributed scheduling problems that is applied in myriad businesses, including the chemical, plastic, metal, electronic, food-processing, and pharmaceutical industries.

DNFSP contains n jobs in the set $N = \{1, \ldots, n\}$ that must be assigned to one factory out of f identical factories in the set $F = \{1, \ldots, f\}$, in which each factory contains the same m machines in the set $M = \{1, \ldots, m\}$ that must be set up in series. The jobs are processed using the same route, and no interruption is permitted, either on or between any two consecutive machines in an assigned factory in the route. The production sequence (or *permutation*), in which the jobs assigned to a given factory go through the first machine, is maintained throughout the factory. All the jobs are available for processing at the beginning of the planning horizon (i.e., at time zero), and the processing time required for Job j ($j \in N$) on Machine i ($i \in M$) is $p_{j,i}$, which is identical for every factory. The objective is to allocate jobs to a set of factories and determine the corresponding production sequences in each factory, thereby minimizing the maximal completion time of the last jobs among all factories (i.e., the makespan), which is the most-studied optimization criterion. Following the conventional three-field notation established by Pinedo (2012), the addressed problem can be designated as $DF_m|prmu, nwt|C_{\max}$.

If the number of factories equals one, or if all the jobs are assigned to a single factory, then the $DF_m|prmu, nwt|C_{\max}$ problem reduces to a corresponding no-wait FSP (i.e., $F_m|prmu, nwt|C_{\max}$), which is strongly NP-hard when the number of machines is more than two (Rock, 1984). Therefore, we readily conclude that the $DF_m|prmu, nwt|C_{\max}$ problem is also strongly NP-hard and cannot be easily solved using a traditional mathematical model. To obtain high-quality solutions quickly and with acceptable memory usage, herein we develop an iterated cocktail greedy (ICG) algorithm that includes two self-tuning mechanisms and a cocktail destruction mechanism for solving the $DF_m|prmu, nwt|C_{\max}$ problem.

The remainder of this paper is organized as follows. After a brief introduction, Section 2 reviews the literature on DNFSP. Section 3 presents a mixed integer programming (MIP) mathematical model. Section 4 offers the newly developed ICG algorithm. Section 5 reports the results of the computational and statistical evaluation

* Corresponding author.
E-mail address: kcying@ntut.edu.tw (K.-C. Ying).

http://dx.doi.org/10.1016/j.cie.2016.07.027
0360-8352/

conducted on two benchmark problem sets of instances. Finally, Section 6 brings together our conclusions and recommendations for a possible extension of this work in future research.

2. Literature review

In order to enhance the reliability and usage of resources in multisite production systems, researchers and industrialists have recently increased their focus on developing effective and efficient optimization algorithms designed for solving distributed scheduling problems. Existing algorithms on distributed scheduling can be broadly classified into three types: exact methods, agent-based algorithms, and heuristic-based algorithms. Because most cases of distributed scheduling problems are considered to be NP-hard, the exact methods (Hammami & Frien, 2013; Naderi & Ruiz, 2010; Thomas, Singh, Krishnamoorthy, & Venkateswaran, 2013) in the literature cannot typically be used to attain an optimal solution for a practical-sized distributed scheduling problem within a reasonable computational time. Consequently, studies on this topic have recently concentrated on developing agent-based algorithms and heuristic-based algorithms.

The agent-based algorithm executes distributed scheduling problems through negotiation and coordination between all agents, in which each agent is responsible for searching a single factory's schedule to determine a global solution (Chan & Chan, 2004). A few effective and efficient agent-based algorithms have been proposed for solving various distributed scheduling problems in recent decades. For instance, Lim and Tan (2013) developed a multi-agent system that integrates process planning and production-scheduling activities across multisite manufacturing facilities to optimize resource usage. By actively negotiating between agents, these agent-based algorithms obtain favorable global schedules for distinct distributed scheduling problems. The computational and statistical results of the aforementioned studies indicate that these agent-based algorithms are able to efficiently handle certain complex distributed scheduling problems.

Apart from developing agent-based algorithms, effective and efficient heuristic-based algorithms must also be developed for addressing the distributed scheduling problem. Noteworthy heuristic-based algorithms developed for this problem to date include constructive heuristics (Gao & Chen, 2011a, 2011b; Naderi & Ruiz, 2010; Ruiz & Naderi, 2009), variable neighborhood descent heuristics (Ruiz & Naderi, 2009), simulated annealing algorithms (DiNatale & Stankovic, 1995), genetic algorithms (Gao & Chen, 2011a, 2011b), Tabu search algorithms (Gao, Chen, & Deng, 2013), neural networks (Jia, Fuh, Nee, & Zheng, 2002), hybrid algorithms (Chan, Prakash, Ma, & Wong, 2013), iterated greedy (IG) algorithms (Lin, Ying, & Huang, 2013), scatter search (SS) algorithm (Naderi & Ruiz, 2014), and bounded-search iterated greedy (BSIG) algorithm (Fernandez-Viagas & Framinan, 2015). During the past decade, researchers and industrialists have confirmed the effectiveness and efficiency of heuristic-based algorithms. The IG and SS algorithms presented by Lin et al. (2013) and Naderi and Ruiz (2014) are recognized as the most efficient and effective approaches among existing heuristic-based algorithms developed for solving the distributed FSP. Moreover, a review of the articles cited in this paragraph indicates that because of their polynomial-time complexity, the agent-based algorithms and the heuristic-based algorithms are more practical than the exact methods for solving large distributed scheduling problems, such as those that appear in numerous real-life situations.

Numerous heuristic-based algorithms have been proposed for solving the no-wait FSP (NFSP), because of the problem's theoretical significance and application in diverse industries. The heuristic-based algorithms available for tackling NFSP can be classified into two main categories: constructive heuristics and meta-heuristics. Certain efficient constructive heuristic algorithms have been proposed for solving NFSPs with respect to the makespan criterion (Framinan & Nagano, 2008; Laha & Chakraborty, 2009; Li, Wang, & Wu, 2008; Sapkal & Laha, 2013). Moreover, several noteworthy meta-heuristic algorithms have also been set up for minimizing makespan in NFSPs, including simulated annealing algorithms (Aldowaisan & Allahverdi, 2003), genetic algorithms (Li et al., 2008), Tabu search algorithms (Grabowski & Pempera, 2005), particle swarm optimization algorithms (Pan, Tasgetiren, & Liang, 2008), differential evolution algorithms (Qian, Wang, Hu, Huang, & Wang, 2009), and hybrid meta-heuristic algorithms (Samarghandi & ElMekkawy, 2012; Tseng & Lin, 2010). Related methods and their applications in NFSP have also been comprehensively reviewed elsewhere (Hall & Sriskandarayah, 1996). Although NFSPs have been widely studied during the past decades, literature searches indicate that no study has been conducted to date on the $DF_m|prmu, nwt|C_{\max}$ problem. Therefore, in this study we formulate an MIP mathematical model for solving the $DF_m|prmu, nwt|C_{\max}$ problem. Because of this problem's computational complexity, we also develop an ICG algorithm that is extremely efficient and effective when used for solving large benchmark instances of this problem.

3. MIP mathematical model

This section presents an MIP mathematical model for solving the $DF_m|prmu, nwt|C_{\max}$ problem. This MIP formulation is similar to a model introduced by Naderi and Ruiz (2010), who addressed the distributed permutation flowshop scheduling problem. To simplify the exposition of the mathematical model, the following notations are defined.

- n: Number of jobs.
- m: Number of machines.
- f: Number of factories.
- i: Index of machines, $i \in \{1, 2, \ldots, m\}$.
- j: Index of jobs, $j \in \{1, 2, \ldots, n\}$.
- k: Index of job positions in a given sequence, $k \in \{1, 2, \ldots, n\}$.
- l: Index of factories, $l \in \{1, 2, \ldots, f\}$.
- $p_{j,i}$: Processing time of Job j at Machine i.

The MIP mathematical model involves the following decision variables.

- $X_{j,k,l} = \begin{cases} 1, & \text{if Job } j \text{ occupies Position } k \text{ in Factory } l \\ 0, & \text{otherwise} \end{cases}$.
- $C_{k,i,l}$ = Completion time of the job in Position k on Machine i of Factory l.
- $C_{\max}$ = Maximal completion time among all the factories.

The objective function is:

$$\text{Min } C_{\max} \tag{1}$$

The constraints are:

$$C_{k,i,l} = C_{k,i-1,l} + \sum_{j=1}^{n} X_{j,k,l} \cdot p_{j,i}, \ \forall k, i > 1, l, \tag{2}$$

$$C_{k,i,l} \geqslant \sum_{j=1}^{n} X_{j,k,l} \cdot p_{j,i}, \ \forall k, i = 1, l, \tag{3}$$

$$C_{k,i,l} \geqslant C_{k-1,i,l} + \sum_{j=1}^{n} X_{j,k,l} \cdot p_{j,i}, \ \forall k > 1, i, l, \tag{4}$$