Expert Systems with Applications 41 (2014) 7754–7763

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Modeling and heuristics for scheduling of distributed job shops

B. Naderi [a,b], A. Azab [a,*]

[a] Product Lifecycle Management Research Lab, Department of Industrial and Manufacturing Systems Engineering, Faculty of Engineering, University of Windsor, Windsor, Canada
[b] Department of Industrial Engineering, Faculty of Engineering, University of Kharazmi, Karaj, Iran

ARTICLE INFO

Article history:
Available online 20 June 2014

Keywords:
Distributed job shop scheduling
Mathematical modeling
Greedy heuristics

ABSTRACT

This paper deals with the problem of distributed job shop scheduling in which the classical single-facility job shop is extended to the multi-facility one. The mathematical formulation of the problem is comprehensively discussed. Two different mixed integer linear programming models in form of sequence and position based variables are proposed. Using commercial software of CPLEX, the small sized problems are optimally solved. To solve large sized problems, besides adapting three well-known heuristics, three greedy heuristics are developed. The basic idea behind the developed heuristics is to iteratively insert operations (one at each iteration) into a sequence to build up a complete permutation of operations. The permutation scheme, although having several advantages, suffers from redundancy which is having many different permutations representing the same schedule. The issue is analyzed to recognize the redundant permutation. That improves efficiency of heuristics. Comprehensive experiments are conducted to evaluate the performance of the two models and the six heuristics. The results show sequence based model and greedy heuristics equipped with redundancy exclusion are effective for the problem.

1. Introduction

In the classical job shop, it is assumed that there is a single-facility with a set of m machines. There is also a set of n job and each job has its own process route among the machines. The problem is to schedule jobs on machines so as to minimize an objective which is always time-related. The most frequently used objective is makespan, i.e., the maximum completion time of jobs. Commonly, it is assumed that jobs are independent. Machines and jobs are all continuously available. The setup times can be ignored or included into processing times. A job can be processed by at most one machine at a time and a machine can process at most one job at a time.

Currently, we are facing quite some change with the structure and topology of small and midsize enterprises especially when it comes to the manufacturing world and metal cutting. Job shops are constantly facing more challenges and an increased need to reduce on both their costs and time-to-market. This has enforced a new decentralized scheme for the world of job shops, where shops in the high-wage developed countries started to establish branches for themselves in the low-wage developing world. This new paradigm of distributed manufacturing is increasingly replacing the traditional centralized single-facility one. It brings in advantages such as low production costs, production flexibility and enhanced overall manufacturing capabilities. Manufacturers become closer to both suppliers and customers, and thus are more able to adhere to local regulations and be more responsive to market changes. In DJS, we have a set of identical f facilities each of which consists of m machines. The problem of DJS is more complicated since two decisions have to be taken; first, the allocation of jobs to facilities and then production scheduling of jobs. It is additionally assumed that the job crossing is not allowed since it is very likely uneconomical or technologically difficult and impractical to transport a work-in-process job from one facility to anther for its remaining operations. In distributed scheduling, makespan minimization becomes the minimization of maximum makespan among facilities.

It is key when studying an optimization problem to develop its mathematical formulation. Besides being able to solve relatively small-to-midsize instances of the problem exact for optimality, however, we get also to rigorously and accurately define the problem. The problem is comprehensively formulates using two mixed integer linear programming (MILP) models. The first one treats the problem as a sequencing decision while the second oy67ne treats it as a positioning one. Computational complexities for both models have been presented and compared.

Due to inherent NP hardness of DSJ, the mathematical models are only capable of solving small to midrange problems for optimality. Hence, three well established heuristics are being adapted to the problem at hand; these are shortest processing time first (SPT), longest processing time first (LPT) and longest remaining

* Corresponding author. Tel.: +1 5192533000x4958; fax: +1 5199737062.
E-mail addresses: azab@uwindsor.ca, azab.ahmed@gmail.com (A. Azab).

http://dx.doi.org/10.1016/j.eswa.2014.06.023
0957-4174/

processing time (LRPT). Also, three greedy heuristics have been deployed. The algorithms are greedy since at each step, several alternatives are generated and the best one is being selected. The permutation encoding scheme is used to represent solutions. Although this encoding scheme has several advantages such as simplicity of encoding, less exhaustive computations, ease of adjustment to operators, and so forth, it suffers from a serious shortcoming, which is redundancy. That is, many different encoded solutions represent the same schedule. However, the algorithms have been treated to recognize and discard redundant permutations. Several numerical experiments are conducted to evaluate the performance of the two models and the six developed algorithms/heuristics as well as the efficiency of the developed redundancy mitigation mechanism. The results show sequence based model and greedy heuristics are effective for the problem.

The rest of the paper is organized as follows. Section 2 reviews the related literature. Section 3 proposes the mathematical models. Section 4 presents the adapted heuristics and proposed greedy algorithms. Section 5 conducts the numerical experiments. Finally, Section 6 concludes the paper and suggests few future research directions.

2. Literature review

In spite of the growing role of distributed and globalized manufacturing, the main focus in the literature on production scheduling is still so far on single-facility manufacturing. As of that dearth of the literature, Jia, Fuh, Nee, and Zhang (2002), Jia, Nee, Fuh, and Zhang (2003) study the distributed production scheduling problem and propose a genetic algorithm. Later on, in another paper, Jia, Fuh, Nee, and Zhang (2007) renew the previous genetic algorithm for the problem. This algorithm is a rather standard genetic algorithm that suffers from many serious shortcomings. For example, all individuals of a generation are of the same job-facility assignment. From a generation to another, all the job-facility assignment is changed together. Chan, Chung, and Chan (2005) propose a genetic algorithm for distributed flexible manufacturing systems. Chan, Chung, Chan, Finke, and Tiwari (2006) adapt this genetic algorithm for the same problem, however, considering this time with maintenance. Moreover, a memetic algorithm is applied to the same problem by Yadollahi and Rahmani (2009). Wang and Shen (2007) write a book about distributed manufacturing with a focus on planning and manufacturing problems rather than production scheduling. Behnamian and Fatemi Ghom (2013) consider distributed parallel machine problems and propose a genetic algorithm hybridized with local search.

Regarding distributed flow shop scheduling, more papers have been published. The first attempt to formulate this problem is done by Naderi and Ruiz (2010) where they develop six different mathematical models. They also propose different heuristics based on two job-facility assignment rules. After the pioneering paper of Naderi (2010), different metaheuristics have been applied to the problem. This is to include electromagnetism-like mechanism algorithm by Liu and Gao (2010), knowledge-based genetic algorithm by Gao, Chen, and Liu (2012), hybrid genetic algorithm by Gao and Chen (2011a), NEH based heuristic by Gao and Chen (2011b), variable neighborhood descent algorithm by Gao, Chen, Deng, and Liu (2012), tabu search by Gao, Chen, and Deng (2013), modified iterated greedy algorithm by Lin, Ying, and Huang (2013), estimation of distribution algorithm by Wang, Wang, Liu, and Xu (2013).

To the top of authors' knowledge, there is no paper directly studying the distributed job shop problem. Obviously, this paper represents the first attempt at mathematically modeling DJS allowing for a precise characterization of the problem. Apart from the mathematical modeling, some algorithms are proposed to effectively solve the problem.

3. Developed mathematical models

This section mathematically models the problem of scheduling distributed job shops by two different mixed integer linear programs. The application of integer programming models in solving scheduling problems starts with the early model of Wagner (1959). Yet, with earlier limitations on computing power and the lack of commercial software, the research progresses on this field were quite slow during the second half the past century. But, due to recent leaps in computing capacity and advent of specialized software, the MILP model and solution development is improving day after day. Even if this idea is accepted that mathematical models cannot be efficient solution algorithms, they are the first natural way to approach scheduling problems (Pan, 1997). It is the first natural step to describe a scheduling problem. Furthermore, mathematical models are along with in many solution methods such as branch and bound, dynamic programming and branch and price. The more efficient MILP models and its mechanism and conceptual workings, the better solution is obtained as results.

The following parameters and indices are used in both developed models:

- n the number of jobs $j, k = \{0, 1, 2, \ldots, n\}$, where job 0 is a dummy one to aid with defining and identifying the first job to be processed on a machine
- m the number of machine $i, l = \{1, 2, \ldots, m\}$
- f the number of facilities $r = \{1, 2, \ldots, f\}$
- $p_{j,i}$ the processing time of job j on machine i
- $a_{j,i,l}$ 1 if machine i is used immediately after machine l in the processing route of job j and 0 otherwise
- M a large positive number

3.1. Operation-sequence based model

To formulate the problem, the first model views the problem as a sequencing decision. The decision variables are as such.

- $X_{k,j,i,r}$ binary variable taking value 1 if job j is processed immediately after job k on machine i in facility r, and 0 otherwise (where $j \neq k$)
- $S_{j,i}$ continuous variable for starting time of operation of job j on machine i

The MILP model is as follows.

$$\textit{Minimize } C_{max} \tag{1}$$

Subject to:

$$\sum_{k=0,k\neq j}^{n}\sum_{r=1}^{f} X_{k,j,1,r} = 1 \quad \forall_j \tag{2}$$

$$\sum_{k=0,k\neq j}^{n} X_{k,j,i,r} = \sum_{k=0,k\neq j}^{n} X_{k,j,1,r} \quad \forall_{j,i>1,r} \tag{3}$$

$$\sum_{j=1,j\neq k}^{n} X_{k,j,i,r} \leqslant \sum_{j=0,j\neq k}^{n} X_{j,k,1,r} \quad \forall_{k>0,i,r} \tag{4}$$

$$\sum_{j=1}^{n} X_{0,j,i,r} = 1 \quad \forall_{i,r} \tag{5}$$

$$\sum_{r=1}^{n} (X_{k,j,i,r} + X_{j,k,i,r}) \leqslant 1 \quad \forall_{j<n,k>j,i} \tag{6}$$