Computers & Industrial Engineering 57 (2009) 641–651

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

An evolutionary algorithm for assembly job shop with part sharing

Felix T.S. Chan [c], T.C. Wong [b,*], L.Y. Chan [a]

[a] Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Pokfulam Road, Hong Kong
[b] Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[c] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

ARTICLE INFO

Article history:
Received 21 April 2008
Received in revised form 24 November 2008
Accepted 24 November 2008
Available online 3 December 2008

Keywords:
Assembly job shop
Lot streaming
Part sharing
Dispatching rules
Genetic algorithm

ABSTRACT

Assembly job shop problem (AJSP) is an extension of classical job shop problem (JSP). AJSP first starts with a JSP and appends an assembly stage after job completion. Lot Streaming (LS) technique is defined as the process of splitting lots into sub-lots such that successive operation can be overlapped. In this paper, the previous study of LS to AJSP is extended by allowing part sharing among distinct products. In addition to the use of simple dispatching rules (SDRs), an evolutionary approach with genetic algorithm (GA) is proposed to solve the research problem. A number of test problems were conducted to examine the performance of the proposed algorithm. Computational results suggested that the proposed algorithm can outperform the previous one, and can work well with respect to the objective function. Also, the inherent conflicting relationship between the primary objective and the system measurements can be addressed.

1. Introduction

For classical job shop problem (JSP), there are m machines and n jobs. Each machine can process only one operation. A job (or lot), which is defined as a batch of identical items, should be processed on all machines until all of its operations are completed. Also, each job can only visit each machine once and the processing sequence of jobs should be strictly followed. Lot streaming (LS) technique depicts a process of splitting lots into sub-lots such that successive operations of the same lot can be processed in parallel on different machines. Over the past few decades, there has been an increasing emphasis on the application of LS to JSP, and the results are promising. In this paper, assembly job shop problem (AJSP), which is an extension of JSP, is investigated. AJSP first starts with JSP and appends an assembly stage after job completion. Therefore, the completed jobs (after the JSP stage) must be assembled if they belong to the bill-of-material (BOM) of the same product. The product assembly can start once all jobs of the same BOM are completed or available after the JSP stage. In the previous study (Chan, Wong, & Chan, 2008), part sharing is not allowed such that completed jobs from distinct BOMs cannot be assembled. To allow part sharing, jobs are classified in two types: Unique and Standard. Unique job type is specific to only one product and only standard job type can be shared among distinct products. Suppose the BOM of Product 1 or P1 contains Job 1 or J1 (unique type) and J2 (standard type one) while the BOM of P2 includes J3 (unique type) and J4 (standard type one). Since both J2 and J4 are of the same standard job type, J2 can substitute J4 for the assembly of P2 or J4 can replace J2 for the assembly of P1. In this connection, the assembly of each product may involve jobs from other BOMs. Obviously, part sharing has enhanced the complexity of the research problem. To solve this problem efficiently, a genetic algorithm (GA) approach is developed with dedicated crossover and mutation operators.

The paper is structured as follows: the literature review is discussed in the next section. In Section 3, the problem background and formulations are presented. Section 4 depicts the development of the proposed approach. Computation results are reported in Section 5. Section 6 discusses the results, and concludes the paper together with future research works.

2. Literature review

Firstly, introduced by Reiter (1966), LS technique is a methodology to split a job into a number of smaller sub-jobs such that successive operations of the same job can be processed in parallel. Thus, the lead time of the whole job can be possibly shortened. Prior to job splitting, the nature of job size and the sub-job type should be defined. In general, the job size can be discrete or continuous. Discrete job size means a job contains an integer number of identical items. Continuous job size can be a real number. Also, the sub-lot type can be variable or consistent. Variable type means that the sub-job size may vary between successive machines. Consistent type restricts that the sub-job size is fixed. Over the past few years, LS has been prevalently applied to Flow Shop Problem

* Corresponding author.
E-mail addresses: mffchan@inet.polyu.edu.hk (F.T.S. Chan), andywtc@graduate.hku.hk (T.C. Wong), plychan@hku.hk (L.Y. Chan).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.11.017

(FSP) (Chen & Steiner, 2003; Kalir & Sarin, 2001a, 2001b; Kumar, Bagchi, & Sriskandarajah, 2000; Liu, 2008; Marimuthu, Ponnambalam, & Jawahar, 2008; Martin, 2006; Smunt, Buss, & Kropp, 1996; Yoon & Ventura, 2002a, 2002b) which only requires one route for all jobs. Thus, the optimum makespan may be obtained by allowing maximum operation overlapping. In other words, it implies that jobs may be split into single-unit sub-jobs. In reality, this is often infeasible due to various practical constraints such as limited material handling capacity. Nevertheless, this "one-route-for-all" feature has enabled LS to work its very best in FSP. In contrast, LS seems not very promising in JSP and Open Shop Problem (OSP). Nevertheless, some studies about LS to JSP (e.g. Chan et al., 2008; Dauzère-pérès & Lasserre, 1997; Jeong, Park, & Leachman, 1999; Smunt et al., 1996) and OSP (e.g. Şen & Benli, 1999) can be found. Recently, the application of LS has been extended to parallel machine scheduling problem (PMSP) (Kim, Shim, Kim, Choi, & Yoon, 2004; Tahar, Yalaoui, Chu, & Amodeo, 2006; Yalaoui & Chu, 2006). For PMSP, jobs are processed on only one machine given that all machines are functionally identical. According to Trietsch and Baker (1993), LS strategy can be classified into four types: (I) equal size sub-jobs without intermittent idling means that jobs are split into sub-jobs with even size, and are processed on the same machine continuously, (II) equal size sub-jobs with intermittent idling that allows idle time between sub-jobs on the same machine, (III) varied size sub-jobs without intermittent idling means that jobs are split into sub-jobs of uneven size, and there is no idle time between sub-jobs on the same machine, and (IV) varied size sub-jobs with intermittent idling. A more comprehensive review on LS strategy can be found in Chang and Chiu (2005).

In general, assembly is the process of constructing a final product from its root components. If part sharing is not considered, components can be assembled only if they belong to the same BOM. In contrast, components from different BOMs may be assembled if they are of the same standard job type. The complexity of a product mainly depends on the number of its components and the assembly levels. A typical product structure is defined as the BOM. By appending assembly stage to JSP, the problem is known as AJSP. Over the past few years, many research works have been dedicated to AJSP (Gravel, Price, & Gagne, 2000; Guide, Srivastava, & Kraus, 2000; Guo, Wong, Leung, Fan, & Chan, 2006; Mckoy & Egbelu, 1998; Mckoy & Egbelu, 1999; Mohanasundaram, Natarajan, Viswanathkumar, Radhakrishnan, & Rajendran, 2002; Thiagarajan and Rajendran, 2003). Guide et al. (2000) discussed about the priority scheduling polices (dispatching rules) in repair shop with no spares. In fact, the repair shop can be regarded as AJSP based on our definition. They further classified a 3-level component matching as: serial number specific, common and the mix of them. Serial number specific means that each component is unique to one product only. Common level allows all components to be shared among all products. The last level is the mix of the previous two levels. In this paper, a 4-level part sharing ratio is introduced such that all levels of component matching can be examined and controlled. However, LS is ignored. Guo et al. (2006) developed a universal mathematical model with genetic optimization process to an industrial case study. Significant improvements have been observed but LS again is not considered. McKoy and Egbelu (1998) presented a 12-step heuristic to minimize the production flow time for AJSP. Comparisons were made between their proposed heuristic and the mixed integer linear program (MILP) on some test problems. The results suggested MILP performs better in terms of solution quality. However, MILP requires substantial computational time to obtain the optimality. Thus, their proposed heuristic is relatively useful for solving big problems with less computing time. However, LS is clearly not considered. Thiagarajan and Rajendran (2003) proposed 10 dispatching rules to solve AJSP in which jobs may carry different holding and tardiness costs. In their study, a 3-level product structure was used to present the product complexity. Also, they incorporated those penalties into the proposed dispatching rules. The experimental results indicated some rules perform better with respect to some weighted objectives (costs). They have successfully dealt with the fact that jobs may carry different penalty costs but failed to consider the impact of LS. In reality, those costs are usually given and fixed. Therefore, more attention must be given to LS. Similarly, Mohanasundaram et al. (2002) proposed new dispatching rules to dynamic AJSP for minimizing flow time based and due date based measures. The simulation results suggested that the new rules can outperform the benchmarked rules without LS. In fact, the use of dispatching rules is not always promising if the problem is complex. In this regard, we proposed a GA-based approach, which is the modified version of the previous algorithm (Chan et al., 2008), to solve AJSP with LS technique. For the rest of the paper, the previous algorithm and the newly proposed algorithm are denoted as PAL and NAL, respectively.

3. Problem background

3.1. Notations

p	total type of products
m	total number of machines
n	total number of lots
n^*	total number of sub-lots
P_h	product h
DM_h	demand of product h
DD_h	due date of product h
CN_h	total number of components of product h
At_h	assembly time of product h
Ct_h	delivery time of product h
M_i	machine i
CF_i	current fixture type on M_i
L_{hj}	lot j of product h
$\mathrm{WIP}_i(t)$	work-in-process inventory of M_i at time t
Q_{hj}	lot size of L_{hj}
F_{hj}	fixture type of L_{hj}
MS_{hjk}	machine for kth operation of L_{hj}
Pt_{hjk}	processing time of kth operation of L_{hj}
SU_{hjk}	setup time of kth operation of L_{hj}
St_{hjk}	start time of kth operation of L_{hj}
Ct_{hjk}	completion time of kth operation of L_{hj}
SN_{hj}	sub-lot number of L_{hj}
L_{hjs}	sth sub-lot of L_{hj}
Q_{hjs}	lot size of L_{hjs}
St_{hjsk}	start time of kth operation of L_{hjs}
Ct_{hjsk}	completion time of kth operation of L_{hjs}
mc_i	machining cost of M_i per hour
lc_h	late cost of P_h per unit per hour
ic_{hj}	inventory cost of L_{hj} per unit per hour
S_{Makespan}	one of the system measurements, makespan
$S_{\mathrm{Inventory}}$	one of the system measurements, inventory cost
S_{Setup}	one of the system measurements, setup cost
S_{WIP}	one of the system measurements, work-in-process cost.

3.2. Problem formulation

In this paper, an AJSP contains a Work Station, an Inventory Station and an Assembly Station as presented in Fig. 1. The Work Station contains m distinct machines, each represents one operation. During each planning period, there are p product types. Each product P_h contains $CN_h \in [2, 10]$ distinct components where $h = 1 \ldots p$,