Available online at www.sciencedirect.com

ScienceDirect

Robotics and Computer-Integrated Manufacturing 24 (2008) 321–331

Robotics and Computer-Integrated Manufacturing

www.elsevier.com/locate/rcim

Lot streaming for product assembly in job shop environment

F.T.S. Chan*, T.C. Wong, L.Y. Chan

Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, HKSAR, China

Received 26 May 2006; received in revised form 21 September 2006; accepted 4 January 2007

Abstract

Assembly job shop scheduling problem (AJSP) is an extension of classical job shop scheduling problem (JSP). AJSP starts with JSP and appends an assembly stage to the completed jobs. Lot streaming (LS) technique is a process of splitting jobs into smaller sub-jobs such that successive operations can be overlapped. This paper combines, for the first time, LS and AJSP, extending LS applicability to both machining and assembly. To solve this complex problem, an efficient algorithm is proposed using genetic algorithms and simple dispatching rules. Experimental results suggest that equal size LS outperforms varied size LS with respect to the objective function.

Keywords: Assembly job shop; Lot streaming; Genetic algorithms; Dispatching rules

1. Introduction

In this paper, a job or a lot is defined as a batch of identical items or components. In order to complete a job, all of its operations should be processed on the machines. If there is assembly stage, only the completed jobs from the same bill-of-material (BOM) can be assembled for the final product. If no assembly is available, the completed jobs should be stored at the inventory. The assembly of the final product and each subassembly in the same BOM can start only when all of its components are completed. Classical job shop scheduling problem (JSP) is one of the most well-known scheduling issues and it assumes that there is no assembly stage after the job completion. One common objective of most of the available JSP models is the minimization of lateness which is defined as the penalty for completing jobs beyond its due dates. Lot streaming (LS) technique which allows splitting of jobs into sub-jobs can improve shop floor performance. As a result, lead time can be shortened and more jobs or sub-jobs may meet its due dates. In the current study, assembly job shop scheduling problem (AJSP) which appends an assembly stage to JSP has been studied. For the first time, we attempt to extend the application of LS to AJSP. To justify this study, the research objectives now become the minimization of the delay cost of the final products and the storage cost of the completed jobs and sub-jobs at the inventory. In addition to job shop features, the assembly stage should be solved in AJSP which can be regarded as an advanced version of JSP. Given the demand of a product, the relative job demands that supply the same product must be determined. If necessary, Manufacturing Resource Planning software packages can help to determine the batch sizes of components over a certain planning period. However, it is commonly assumed that the lot size is a constant, i.e. lot splitting is not allowed (e.g. [1]). Hence, we argue that the significance of LS must not be underrated in the shop floor level. For example, if a lot is composed of 4 identical items or components, there are at least 4 ways to split it: (1) {1,1,1,1} means 4 sub-jobs of size 1, (2) {1,1,2}, (3) {2,2} or {1,3}, and (4) no splitting. Hence, we have total 5 sub-job combinations. Since the combination increases significantly with the lot size and the number of jobs, it is necessary to develop an efficient algorithm. The determination of sub-job combinations is not the end of the story. After splitting jobs into sub-jobs, we need to solve AJSP which is also NP-hard. In this paper, an efficient algorithm has been proposed to solve this complex problem using genetic algorithms (GAs) and simple dispatching rules (SDRs).

*Corresponding author. Tel.: +852 2859 7059; fax: +852 2858 6535.
E-mail address: ftschan@hkucc.hku.hk (F.T.S. Chan).

0736-5845/$ - see front matter © 2007 Published by Elsevier Ltd.
doi:10.1016/j.rcim.2007.01.001

2. Literature review

LS technique which was first introduced by Reiter [2] is a methodology to split a job into smaller sub-jobs such that successive operations of the same job can be overlapped. 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 (FSP) [3–5] which only allows one route for all jobs. Thus, the maximum operation overlapping can derive the optimum makespan. It implies that jobs are split into single-unit sub-jobs. In reality, this is often infeasible due to various practical constraints. 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). Even so, some studies about LS to JSP [6,7] and OSP [8] can be found. According to Trietsch and Baker [9], LS can be classified into 4 types: (I) equal size (ES) 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) ES sub-jobs with intermittent idling that allows idle time between sub-jobs on the same machine, (III) varied size (VS) 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) VS sub-jobs with intermittent idling. For a comprehensive review on LS, please refer to Chang and Chiu [10].

Assembly is usually defined as the process to construct a final product from its components. One basic rule is that components or items can be assembled only if they belong to the same BOM. The complexity of a product mainly depends on the number of its components and the assembly levels. Furthermore, each component must be dedicated to only one product. A typical product structure with 4 assembly levels is presented in Fig. 1. The top level is Product 1 (P1). The second level contains Assembly 1 (A1), Assembly 2 (A2), and Component 2 (C2). A1 is the assembly of A3 and C2. C4 and C5 are assembled for A3 and so on. To study AJSP, there are mainly two streams. The first stream only allows machining to the components but not the assemblies. The second stream allows both machining and assembly to the assemblies. Similar to the current study, Kim and Kim [11] have studied AJSP with no machining for the assemblies, i.e. the first stream. They have compared two evolutionary algorithms for minimizing the earliness penalty of sub-jobs and the tardiness penalty of the final products. For simplicity, lot splitting is neglected. McKoy and Egbelu [12] have presented a 12-step heuristic to minimize the production flow time for AJSP. Comparisons are made between their proposed heuristic and the mixed integer linear program (MILP) on some test problems. The results have suggested MILP performs better in terms of solution quality. However, MILP requires substantial computational time to obtain the optimality. Thus, their proposed algorithm is more preferable if the problem size is big and computational time is relatively significant. However, LS is clearly not considered. Thiagarajan and Rajendran [13] have proposed 10 dispatching rules to solve AJSP with jobs having different holding and tardiness costs. In their study, a 3-level product structure is used to define the product complexity. They have incorporated those penalties into the proposed dispatching rules. The experiment results reveal that 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 lot splitting. In reality, those costs are usually given and fixed. Therefore, more attention must be given to lot splitting.

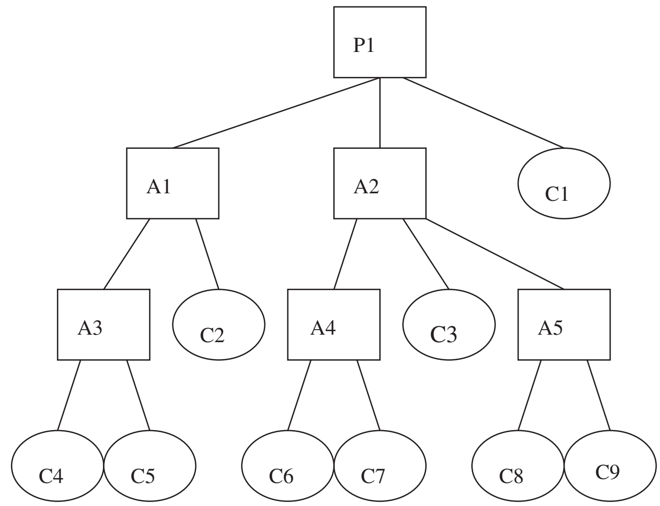

Fig. 1. A typical product structure.

Guide et al. [14] have discussed the priority scheduling policies (or SDRs) in repair shop with no spares. In their paper, the repair shop contains a disassembly area, a repair area and an assembly area. All components of the disassembled products from the disassembly area are processed or repaired through a fixed sequence of operations in the repair area which contains a number of work centres. Then all repaired components are re-assembled in the assembly area. Since each component has its unique and fixed operation sequence, this shop actually can be regarded as AJSP. In addition to different product structures, they have further classified the component matching into three levels, i.e. serial number specific, common, and the mix of them. They have reported that SDRs may outperform complex optimization methods