Computers & Industrial Engineering 122 (2018) 149–160

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Lower bound development in a flow shop electronic assembly problem with carryover sequence-dependent setup time

M.T. Yazdani Sabouni, Rasaratnam Logendran*

School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331-2407, USA

ARTICLE INFO

Keywords:
PCB group-scheduling
Branch-and-price
Mathematical programming
Lower bound

ABSTRACT

A flow shop group-scheduling problem in the assembly of printed circuit boards (PCBs) is addressed in this paper. We have developed search algorithms yet their quality cannot be attested to when optimal solutions or lower bounds are unavailable. A very effective and efficient lower-bounding mechanism based on the underlying concepts of column generation and branch-and-price is developed. The problem remains so complex even after decomposition. Thus, optimal properties and strategies are developed to facilitate efficiently solving the sub problems. Accompanied by an experimental design and statistical analysis, comprehensive computational tests for a wide range of problems are carried out. The findings suggest that the lower bound and search algorithms are very effective even for large-size problem instances.

1. Introduction

In this research, a mixed-integer linear programing (MILP) model has been formulated for the Printed Circuit Board (PCB) assembly problem on flow shop machines. The model formulated integrates machine setup times, namely the internal setup, with that of the external setup, which is the time spent on performing kitting operations of components. ILOG CPLEX (2013) as a strong mathematical optimization software failed to identify good quality solutions or even valid lower bounds after letting the model run for a large amount of computation time of up to *three days*. In order to identify implementable solutions for industry-size problems, we developed fast heuristic and metaheuristic algorithms in Yazdani Sabouni and Logendran (2013a). This research was continued in Yazdani Sabouni and Logendran (2013b) by developing a lower bounding mechanism based on the concepts of Column Generation (CG) and Branch-and-Price (B&P) to quantify the quality of solutions identified by the proposed search algorithms. CG decomposes the original mathematical formulation into a master problem and sub problem(s). The most difficult constraints which do not have a block diagonal structure are kept in the master problem and the rest of the constraints are included in the sub problems. Iteratively, the master and sub problems are solved to optimality to provide a lower bound for the original problem. If solving the sub problems is computationally difficult, any lower bound on the optimal solution of the sub problems can be used to establish a lower bound for the original problem. In this research, strictly following the traditional CG approach leads to sub problems that remain unsolvable. Sub problems in this research represent single machines, thus for the three-machine flow shop problem addressed, we have sub problem 1 (SP(1)), SP(2) and SP(3). To overcome the difficulty of solving SP(1), an optimal strategy is developed in Yazdani Sabouni and Logendran (2014a). The general idea to facilitate solving SP(2) along with an example is demonstrated in Yazdani Sabouni and Logendran (2014b).

In order to give different importance to the producer's and customers' interests, the objective function is considered to be the weighted linearized form of the two criteria, namely weighted total flow time and weighted total tardiness, which enables evaluating different trade-offs between the producer and customers, respectively. The problem formulated here concerns with the evaluation of weighted total flow time on the first two machines and evaluation of both weighted total flow time and weighted total tardiness on the third machine. The two-machine flow shop scheduling problem with total flow time minimization (Garey, Johnson, & Sethi, 1976) and single-machine problem with total tardiness minimization (Du & Leung, 1990) have been proven to be strongly *NP*-hard. This establishes the premise that the flow shop problem with the objective of weighted sum of total flow time and total tardiness investigated in this research is also strongly *NP*-hard.

A setup operation is needed to remove the components required in the previous assembly and replace them with the components required in the current assembly. To minimize the number of unnecessary setup operations because of frequent changes, boards requiring similar components are grouped and, thus, the problem is commonly referred to as *group-scheduling*. There is an enormous number of research efforts around group-scheduling. We reviewed some of the more related ones

* Corresponding author.
E-mail address: Logen.Logendran@oregonstate.edu (R. Logendran).

https://doi.org/10.1016/j.cie.2018.05.033
Received 18 December 2017; Received in revised form 15 April 2018; Accepted 21 May 2018
Available online 31 May 2018
0360-8352/

to the problem of our interest. Li, Xiaoping, and Gupta (2015) minimized total weighted flowtime as well as total tardiness. They proposed a hybrid harmony search algorithm and compared it against several meta-heuristic algorithms. Minimization of total completion time in a flexible flowshop sequence-dependent group scheduling problem is addressed in Keshavarz, Salmasi, and Varmazyar (2015). They developed a linear mathematical model and applied a Memetic algorithm. They compared their algorithm against a lower bounding method based on B&P algorithm. Costa, Cappadonna, and Fichera (2014) minimized total flowtime in a flowshop sequence-dependent group scheduling problem. They proposed a hybrid metaheuristic algorithm that combines the features of Genetic algorithm and Biased Random Sampling. Lu and Logendran (2013) addressed minimization of sum of total weighted completion time and total weighted tardiness. They introduced an MILP model and compared ten heuristic algorithms. Consequently, a TS algorithm with two initial solution finding mechanisms was found to have the best performance.

SP(3) in this research is by nature more complex than SP(1) and SP(2) because of having the total tardiness criteria, which would imply the inapplicability of the optimal properties developed for SP(1) and SP(2) for SP(3). However, SP(3) can be simplified by finding the optimal arrangements of boards for a number of groups (not all the groups), which is presented in this paper. These groups are identified by the works of two algorithms that are also developed in this paper. The proofs for all of the theorems and the properties used for SP(3) and for some properties/theorems previously given in Yazdani Sabouni and Logendran (2014a, 2014b) are fully described in this paper.

The highlights of this paper essentially are:

1. The original mathematical model and the one previously used in the B&P algorithm have considerably a large number of integer variables. In this paper the number of variables (integer or non-integer) is kept at the minimum, which enhances the computational efficiency of solving the sub problems, thus resulting in the identification of tight lower bounds in a shorter time.
2. Development of an efficient search algorithm named CFIM2 (Cycle Forward Improving Moves), which is an improvement over the FIEI (Forward Imploring Exchanges/Inserts) and the Tabu Search (TS) algorithms developed in Yazdani Sabouni and Logendran (2013a).
3. Development of a lower bounding mechanism based on the B&P methodology.
4. Sub problems simplification is fully accomplished in this paper by simplifying SP(3) as a closure to the research reported in Yazdani Sabouni and Logendran (2014a, 2014b).
5. A comprehensive computational study with a large number of problem instances is also provided to test the lower bounds against the search algorithms. This study is also supported by an experimental design along with statistical analysis to compare the performance of the search algorithms for different problem types.

There is very little research in flow shop and job shop scheduling that employed CG. In reviewing some of these papers, we considered the strategy they used to decompose the original problem. Bulbul, Kaminsky, and Yano (2004) developed heuristics for the m-machine flow shop problem and approximately solved the problem by column generation. They decomposed the original problem and assigned each machine to an independent sub problem. This strategy is also used in Kirchner, Gebauer, and Lübbecke (2014) where the problem is decomposed into single machine sub problems in job shop scheduling with the sum of completion times. Keshavarz and Salmasi (2014) considered minimizing total completion time in a permutation flowshop group scheduling problem. They proposed a hybrid Genetic algorithm and established a lower bounding mechanism based on B&P. They showed that their lower bound has better performance than the previous methods reported in the literature. Gelogullari and Logendran (2010) addressed the *carryover sequence-dependent* (CSD) setup time in an m-machine flow shop group-scheduling problem with a single objective of minimization of total flow time and provided lower bounds using a B&P algorithm. However, their work is fundamentally different from our research with respect to two major factors: (1) Existence of kitting operation (Section 1.2) and this being integrated with the assembly and setup operations, and (2) Minimization of total flow time and total tardiness simultaneously. Since minimizing the total tardiness on a single machine itself is strongly NP-hard, minimizing it along with total flow time on flow shop machines is even more complicated. Thus, the approaches in Gelogullari and Logendran (2010) for a single objective function of minimizing total flow time are inapplicable to our problem.

The paper continues in Sections 1.1 and 1.2 by providing explanations of setup time and kitting operation. A mathematical programming model is provided in Section 2. Several heuristics are discussed in Section 3. Section 4 develops an efficient decomposition approach to determine lower bounds for the problem. The computational study, experimental design and statistical analysis are provided in Section 5. Conclusions and future research are given in Section 6.

1.1. Carryover sequence-dependent setup time

The CSD setup accurately describes the application of PCB assembly and is an advanced form of the conventional sequence-dependent setup. In this setup time, the dependency is on *all* of the previous groups in the sequence and not necessarily on the immediately preceding group. The example below simply describes the assembly. Consider there are three groups G_g, $g = 1, 2, 3$ and two sequences $S_1 = R, G_1, G_3, G_2$ and $S_2 = R, G_2, G_1, G_3$, where R is the reference group, the set of components already on the machines from previous day (shift). Table 1 presents the component requirements on the feeders. The CSD setup between G_1 and G_3 is the sum of the setup time per feeder changes from G_1 to G_3. To change from R to G_1, one setup is needed on feeder 2 to remove component 05 existing due to R and placing 06 needed by G_1. No setup is needed on feeder 5 since 07 is shared by R and G_1. Doing so will obtain the total number of setup changes from R to G_1 evaluated as 4 ($4 = 1 + 0 + 0 + 1 + 1 + 1$) and the other feeders not shared by R and G_1 remain unchanged. Similarly, the number of setup changes from $(R - G_1)$ to G_3 is evaluated as 5 ($5 = 1 + 1 + 1 + 0 + 0 + 1 + 1$). However, the number of setup changes from G_1 to G_3 in S_2 is 4. This implies that different number of setup changes exist between G_1 and G_3 in S_1 and S_2 (5 vs. 4) although G_1 is immediately before G_3 in both S_1 and S_2.

The sequence-independent setup times are studied in Schaller (2001) and Choi and Kim (2009). The sequence-dependent cases can be found in Schaller, Gupta, and Vakharia (2000), Strusevich (2000) and Eom, Shin, Kwun, Shim, and Kim (2002). A setup strategy called the decompose and sequence (DAS) method proposed in McGinnis et al. (1992) considers only the feeders not shared by the next board group

Table 1
Feeder configuration for single machine.

Reference configuration	Feeder	Configuration required per group in S_1			Configuration required per group in S_2		
		G_1	G_3	G_2	G_2	G_1	G_3
	1		04	04	04		04
05	2	06	17	10	10	06	17
03	3		01				01
	4			03	03		
07	5	07	07			07	07
15	6	15				15	
	7	14		14	14	14	
16	8	09	09			09	09
	9		16	13	13		16
12	10	11	02			11	02