Computers and Operations Research 88 (2017) 237–246

Contents lists available at ScienceDirect

Computers and Operations Research

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

The 2-stage assembly flowshop scheduling problem with total completion time: Efficient constructive heuristic and metaheuristic

Jose M. Framinan*, Paz Perez-Gonzalez

Industrial Management, School of Engineering, University of Seville, Ave. Descubrimientos s/n, Seville E41092, Spain

ARTICLE INFO

Article history:
Received 16 September 2016
Revised 23 June 2017
Accepted 17 July 2017
Available online 19 July 2017

Keywords:
Scheduling
Assembly
Completion time
Heuristics

ABSTRACT

In this paper, we address the 2-stage assembly scheduling problem where there are m machines in the first stage to manufacture the components of a product and one assembly station (machine) in the second stage. The objective considered is the minimisation of the total completion time. Since the NP-hard nature of this problem is well-established, most previous research has focused on finding approximate solutions in reasonable computation time. In our paper, we first review and derive a number of problem properties and, based on these ideas, we develop a constructive heuristic that outperforms the existing constructive heuristics for the problem, providing solutions almost in real-time. Finally, for the cases where extremely high-quality solutions are required, a variable local search algorithm is proposed. The computational experience carried out shows that the algorithm outperforms the best existing metaheuristic for the problem. As a summary, the heuristics presented in the paper substantially modify the state-of-the-art of the approximate methods for the 2-stage assembly scheduling problem with total completion time objective.

1. Introduction

Assembly scheduling refers to a branch of scheduling decisions in which parts/components/subsets of products or services must be first manufactured in parallel and later assembled in a final stage. Applications of assembly scheduling in industry and services have been reported in several works: Potts et al. (1995) describe the case of personal computer manufacturing where the different components of the computer are produced in the first stage to be later assembled in a second stage (a packaging station). Lee et al. (1993) describe the case of a fire engine assembly plant. In this case, the body and chassis of fire engines are produced in parallel, and assembled in a second stage. Finally, another application is presented by Al-Anzi and Allahverdi (2006a); 2006b) and Allahverdi and Al-Anzi (2006) in the area of distributed database systems.

Different objectives can be considered for the assembly scheduling problem. The first objective addressed in the literature is the minimisation of the makespan or maximum completion time of the set of jobs. This problem has been first tackled by Lee et al. (1993), and its NP-hardness in the strong sense (even when the first stage is composed of 2 machines in parallel) has been established by Potts et al. (1995). A number of efficient heuristics for the problem have been proposed by Sun et al. (2003). Regarding exact solutions, the best approach is the Branch & Bound algorithm proposed by Hariri and Potts (1997), which in many cases is able to schedule up to 8000 jobs in less than 100 s. Another well-studied objective is the minimisation of the sum of completion times of the jobs, which is also the aim of our paper and is discussed in detail below. Other objective considered in the literature is the maximum lateness (Al-Anzi and Allahverdi, 2006b; Allahverdi and Al-Anzi, 2006). Finally, additional constraints such as setup times (Al-Anzi and Allahverdi, 2007), more than one machine in the second stage (Al-Anzi and Allahverdi, 2013; Sung and Kim, 2008), or additional stages for the transportation of components (Koulamas and Kyparisis, 2001; Shoaardebili and Fattahi, 2015) have been also addressed.

As mentioned above, our paper is devoted to the 2-stage assembly scheduling problem with the minimisation of total completion time as objective, which can be denoted as $Am||\Sigma_j C_j$ according to the notation in Potts et al. (1995). Minimisation of the total completion time is an important scheduling objective since completion time can be viewed as a surrogate for the cycle time of the jobs, which in turns influences the inventory levels and the lead times that can be quoted by a company (Framinan et al., 2014). Also note that this problem has several connections to other scheduling problems: perhaps the most clear case is the 2-machine flowshop scheduling problem with total completion time as objective, which can be seen as a particular case of our problem when there is only one component to be manufactured. In turn, this NP-

* Corresponding author.
E-mail addresses: framinan@us.es, framinan@gmail.com (J.M. Framinan), pazperez@us.es (P. Perez-Gonzalez).

http://dx.doi.org/10.1016/j.cor.2017.07.012
0305-0548/

hard problem can be decomposed into consecutive single-machine scheduling problems, for which the Shortest Processing Time (SPT) rule provides the optimal solution, a property used by some of the heuristics for the $Am||\Sigma_j C_j$ problem. Another connection is with the customer order scheduling problem with total completion time as objective (see e.g. Framinan and Perez-Gonzalez, 2017; Leung et al., 2005), denoted as $PDm||\Sigma_j C_j$. In this problem, customer orders composed of a number of parts have to be manufactured in dedicated parallel machines. Clearly, $PDm||\Sigma_j C_j$ and $Am||\Sigma_j C_j$ problems are equivalent if the processing times of the jobs in the second (assembly) stage are zero, but, as we will show later, there is another less trivial relationship.

The first reference for total completion time minimisation is Tozkapan et al. (2003), where the authors address the problem (weighted minimisation) for the first time. They show that permutation sequences are optimal for this problem, and propose two heuristics, labelled TCK1 and TCK2 in the following. Al-Anzi and Allahverdi (2006a) also address this problem, stating some conditions that the processing times of an instance must fulfil to be optimally solved, and proposing both constructive heuristics and metaheuristics for the problem. Regarding the constructive heuristics, the computational experience carried out by these authors shows that two of them (the aforementioned TCK2 and a new proposal denoted as A1 in the following) are the most efficient heuristics for the problem, being around 8% with respect to the best known solutions while requiring a negligible computational effort. Regarding the metaheuristics proposed, it turns out that a Hybrid Tabu Search (HTS in the following) obtains the best results, being therefore the most efficient metaheuristic for the problem.

Note also that the problem under consideration can be regarded as a special case of the multi-machine assembly scheduling problem, where there are more than one machine in the second (assembly) stage. This problem has been addressed for the total completion time objective first by Sung and Kim (2008) when there are two assembly machines, and later generalised for $m \geq 2$ assembly machines by Al-Anzi and Allahverdi (2012). The most efficient approximate method for the problem is the Artificial Immune Intelligence (AIS) metaheuristic by Al-Anzi and Allahverdi (2013), as these authors conduct an exhaustive computational experience showing that AIS outperforms the rest of existing approximate methods. However, it is worth to note that the inclusion of more than one machine in the second stage may change the structure of solutions of the problem and therefore it remains uncertain whether efficient procedures for the multi-machine case are equally efficient when there is only one assembly machine.

Other related problem is that of the distributed two-stage assembly system, where the jobs have to be assigned to one of f factories each one consisting of a two-stage assembly system like the one treated in our research, and subsequently scheduled to minimise the total completion time. To the best of our knowledge, this problem has been addressed only by Xiong et al. (2014), also considering setup times. These authors propose the so-called ESPT constructive heuristic that could be potentially interesting for our problem and indeed, when there is only one factory and no setups are considered, it is equivalent to one of the already mentioned constructive heuristics by Al-Anzi and Allahverdi (2006a).

Finally, Al-Anzi and Allahverdi (2006a) study a number of theoretical properties for the problem under consideration. The work of these authors represents an important advance on analysing the problem, particularly on identifying distinct sub-cases depending on whether the first stage dominates the second, or vice versa. However, we will show in Section 2 that their results contain some flaws, and we provide a correct formulation. Also, based on the ideas regarding the predominance of one of the stages, we propose a constructive heuristic for the problem which turns out to be much more effective than existing constructive heuristics, i.e. the average error with respect to the best known solution is around five times smaller. Finally, we exploit some of the ideas used in the design of the HTS algorithm to propose a new local search algorithm for the problem which also favourably competes against HTS both in terms of quality of the solutions and in the computational effort required.

The remainder of the paper is as follows: in Section 2 the problem under consideration is formally stated and some properties are presented. Section 3 is devoted to first present the constructive heuristics available for the problem (Section 3.1), and second to discuss a new proposal (Section 3.2). In Section 4 we present a new metaheuristic for the problem, while Section 5 is devoted to the computational experiments. Finally, the main conclusions are discussed in Section 6.

2. Problem statement and properties

Formally stated, the problem under consideration consists of scheduling n jobs in a layout composed of two stages: In the first stage there are m machines in parallel, each one capable of processing one of the m components of the jobs. Let us denote by t_{ij} the processing time in machine i of the component of job j in this stage, or equivalently, t_{ij} is the processing time of the ith component of job j. It is convenient for us to denote the maximum and minimum of t_{ij}, i.e. $t_{\max} = \max_{\forall i,j} t_{ij}$, and $t_{min} = \min_{\forall i,j} t_{ij}$. The second stage consists of the assembly of the components, so operations in the second stage cannot start until the m components of the job have been completed. The processing time of job j in this assembly stage is denoted by p_j.

Given a permutation sequence, let us denote job $[j]$ as the job processed in order jth in the sequence. Furthermore, let $C_{[j]}$ be the completion time of job processed in order $[j]$. Clearly, the following recursive formula holds:

$$C_{[j]} = \max\left\{C_{[j-1]}; \max_{\forall i}\left\{\sum_{k=1}^{j} t_{i[k]}\right\}\right\} + p_{[j]} \quad (1)$$

with $C_{[0]} = 0$.

As mentioned in Section 1, a number of properties for the problem have been studied by Al-Anzi and Allahverdi (2006a). While this work is an important step towards the analysis of the problem, their results contain several flaws which also imply some changes in the tractable subcases. In the next theorems, we provide the correct formulation of the problem properties, point out the differences with the initial statements, and derive new properties.

Theorem 1. *If* $\max_{\forall j}\{p_j\} \leq t_{min}$, *then the total completion time of a sequence can be expressed as:*

$$\sum C_j = \sum_{j=1}^{n} \max_{\forall i}\left\{\sum_{k=1}^{j} t_{i[k]}\right\} + \sum_{j=1}^{n} p_j \quad (2)$$

Proof. From Eq. (1) it can be seen that $C_{[1]} = \max\{C_{[0]}; \max_{\forall i}\{t_{i[1]}\} + p_{[1]}\} = \max_{\forall i}\{t_{i[1]}\} + p_{[1]}$. Regarding the second job:

$$\begin{aligned} C_{[2]} &= \max\left\{C_{[1]}; \max_{\forall i}\left\{t_{i[1]} + t_{i[2]}\right\}\right\} + p_{[2]} \\ &= \max\left\{\max_{\forall i}\left\{t_{i[1]}\right\} + p_{[1]}; \max_{\forall i}\left\{t_{i[1]} + t_{i[2]}\right\}\right\} + p_{[2]} \\ &= \max\left\{\max_{\forall i}\left\{t_{i[1]} + p_{[1]}\right\}; \max_{\forall i}\left\{t_{i[1]} + t_{i[2]}\right\}\right\} + p_{[2]} \end{aligned}$$

Since $\max_{\forall j}\{p_j\} \leq t_{ik}$ for all i, k, we have:

$$C_{[2]} = \max_{\forall i}\{t_{i[1]} + t_{i[2]}\} + p_{[2]}$$