Computers & Industrial Engineering 87 (2015) 30–39

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Efficient heuristic algorithms for the blocking flow shop scheduling problem with total flow time minimization ☆

Imma Ribas [a],*, Ramon Companys [b]

[a] Departament d'Organització d'Empreses, DOE – ETSEIB – Universitat Politècnica de Catalunya. BarcelonaTech, Avda. Diagonal, 647, 7th Floor, 08028 Barcelona, Spain

[b] EPSEB – Universitat Politècnica de Catalunya. BarcelonaTech, Av. Doctor Marañón, 44-50, 3rd Floor, 08028 Barcelona, Spain

ARTICLE INFO

Article history:
Received 25 February 2014
Received in revised form 23 December 2014
Accepted 9 April 2015
Available online 1 May 2015

Keywords:
Scheduling
Flow shop
Blocking
Total flow time
Heuristics

ABSTRACT

This paper proposes two constructive heuristics, i.e. HPF1 and HPF2, for the blocking flow shop problem in order to minimize the total flow time. They differ mainly in the criterion used to select the first job in the sequence since, as it is shown, its contribution to the total flow time is not negligible. Both procedures were combined with the insertion phase of NEH to improve the sequence. However, as the insertion procedure does not always improve the solution, in the resulting heuristics, named NHPF1 and NHPF2, the sequence was evaluated before and after the insertion to keep the best of both solutions. The structure of these heuristics was used in Greedy Randomized Adaptive Search Procedures (GRASP) with variable neighborhood search in the improvement phase to generate greedy randomized solutions. The performance of the constructive heuristics and of the proposed GRASPs was evaluated against other heuristics from the literature. Our computational analysis showed that the presented heuristics are very competitive and able to improve 68 out of 120 best known solutions of Taillard's instances for the blocking flow shop scheduling problem with the total flow time criterion.

1. Introduction

Many industrial systems can be modeled as a flow shop with zero capacity buffers between consecutive machines. In this type of production configuration, a machine can be blocked by the job it has processed if the next machine is not available. To avoid or minimize machine blocking and idle time, accurate scheduling is necessary. Examples of blocking flow shop scheduling can be found in the production of concrete blocks, where storage is not allowed in some stages of the manufacturing process (Grabowski & Pempera, 2000); in the iron and steel industry (Gong, Tang, & Duin, 2010); in the treatment of industrial waste and manufacture of metallic parts (Martinez, Dauzère-Pérès, Guéret, Mati, & Sauer, 2006); or in a robotic cell, where a job may block a machine while waiting for the robot to pick it up and move it to the next stage (Sethi, Sriskandarajah, Sorger, Blazewicz, & Kubiak, 1992).

This paper deals with the blocking flow shop scheduling problem to minimize the total flow time of jobs, denoted as Fm|block|ΣC_i according to the notation proposed by (Graham, Lawler, Lenstra, & Rinnooy Kan, 1979), if jobs and machines are available at instant zero, a hypothesis considered here. If jobs' release time are zero, this objective is equivalent to (total or average) flow time minimization, which, according to Rajendran (1993) and Framinan, Leisten, & Ruiz-Usano (2005), has been found to reduce the scheduling costs. Additionally, it has been found to be an important real-life objective in industries since it results in the even utilization of resources, even turn-over of finished jobs and reduced in-process inventory. Therefore, it is considered to be more relevant and meaningful for today's dynamic production environment (Liu & Reeves, 2001). Pan and Ruiz (2013) remark that the need to reduce Work In Process (WIP) or in-process inventory has fostered the study of the total flow time.

In the Fm|block|ΣC_i problem, n jobs have to be processed by m machines. All jobs follow the same route, implying that a job sequence determined for machine 1 is kept throughout the system. The processing time of job $i \in \{1, 2, \dots, n\}$ on machine j, $j \in \{1, 2, \dots, m\}$ is $p_{j,i} > 0$.

Although the blocking flow shop scheduling problem has not been as extensively studied as the permutation flow shop problem, the number of published papers concerning the former in order to minimize makespan has increased in recent years (Grabowski & Pempera, 2007; Liu, Wang, & Jin, 2008; Wang, Zhang, & Zheng, 2006; Qian et al., 2009; Wang et al., 2010; Davendra & Bialic-Davendra, 2013; Ribas, Companys, & Tort-Martorell, 2011).

☆ This manuscript was processed by Area Editor Hans Kellerer.

* Corresponding author. Fax: +34 93 401 60 54.
E-mail address: imma.ribas@upc.edu (I. Ribas).

http://dx.doi.org/10.1016/j.cie.2015.04.013
0360-8352/© 2015 Elsevier Ltd. All rights reserved.

However, little research has been done on total flow time criterion. From the best our knowledge, only Wang et al. (2010), who proposed a hybrid Harmony Search (HS) algorithm, Deng, Xu, and Gu (2012), who proposed a Discrete Artificial Bee Colony (DABC) algorithm and Moslehi and Khorasanian (2013), who presented a branch and bound algorithm that can be used in small instances, have addressed the Fm|block|ΣC_i problem. Therefore, it is interesting to intensify research to develop efficient heuristics for this problem, especially simple algorithms which are easy to adapt and implement in practical applications.

In this paper, we present two constructive procedures and two versions of an efficient Greedy Randomized Adaptive Search Procedure (GRASP) combined with variable neighborhood search in the improvement phase, which use the structure of these constructive heuristics to generate greedy randomized solutions. Computational evaluation against other algorithms from the literature has shown the effectiveness of the constructive heuristic and good performance of the proposed GRASP.

The rest of the paper is organized as follows. In Section 2, blocking flow shop scheduling is presented. Sections 3 and 4 describe the constructive procedures and the GRASP, respectively. Section 5 shows the computational evaluation of the algorithms and Section 6 concludes.

2. Problem definition

In the blocking flow shop problem, a set of n jobs must be processed by m machines in the same order, from the first machine to machine m. Each job i, $i \in \{1, 2, \ldots, n\}$ requires a fixed positive processing time $p_{j,i}$ on every machine j, $j \in \{1, 2, \ldots, m\}$. Jobs and machines are available from time zero onwards. Our objective is to find a job processing sequence that minimizes the total flow time. Fm|block|ΣC_i can be modeled with the following equations, where $[k]$ is the index of the job in the k-th position in the permutation, $e_{j,k}$ denotes the time at which the job $[k]$ starts to be processed by machine j and $c_{j,k}$ is the departure time of job $[k]$ from machine j. Note that if job $[k]$ can leave machine j when it is completed, which depends on the availability of machine $j+1$, then $c_{j,k}$ is not only the departure time but also the completion time of job $[k]$ on machine j:

$$e_{j,k} + p_{j,[k]} \leqslant c_{j,k} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{1}$$

$$e_{j,k} \geqslant c_{j,k-1} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{2}$$

$$e_{j,k} \geqslant c_{j-1,k} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{3}$$

$$c_{j,k} \geqslant c_{j+1,k-1} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{4}$$

$$TF = \sum_{k=1}^{n} c_{m,k} \tag{5}$$

with $c_{j,0} = 0 \; \forall j, \; c_{0,k} = 0, \; c_{m+1,k} = 0 \; \forall k$ being the initial conditions.

If Eqs. (2) and (3) are summarized as (6) and Eqs. (1) and (4) as (7), the schedule obtained is semi-active, which is interesting because an optimal solution can be found in the subset of the semi-active set of solutions.

$$e_{j,k} = \max\{c_{j,k-1}; c_{j-1,k}\} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{6}$$

$$c_{j,k} = \max\{e_{j,k} + p_{j,[k]}, c_{j+1,k-1}\} \quad j = 1,2,\ldots,m \; k = 1,2,\ldots,n \tag{7}$$

3. Constructive heuristics

The constructive heuristics proposed are based on the *profile fitting* (PF) procedure by McCormick, Pinedo, Shenker, and Wolf (1989) for solving sequencing problems in an assembly line with blocking to minimize the cycle time. The PF technique tries to sequence jobs in order to minimize machine timeout, which can be due to idle time, blocking time or the sum of both (see Fig. 1). This is an adequate objective for the blocking flow shop problem. The PF method, combined with the insertion phase of the heuristic NEH (Nawaz, Enscore, & Ham, 1983), was successfully used by Ronconi (2004) for scheduling jobs in a blocking flow shop to minimize makespan. Thus, this procedure considers the blocking constraint; however, it does not address total flow time minimization. To include this objective in the construction of sequences, two variants are proposed.

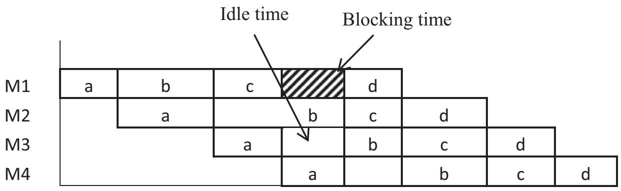

Fig. 1. Sequence for a 4-job, 4-machine blocking flow shop.

First, we describe the PF heuristic to make clear the procedure and its variants.

Let σ the partial sequence and $\sigma * i$ the partial sequence with job i added at the end of σ.

- Step 1: Select the job with the minimum sum of all operations of a job (P_i) and put it in the first position in sequence σ. Set $k = 1$. In case of ties, select the job with minimum $p_{1,i}$.
- Step 2: While $k < n$, calculate the machine timeout for each unscheduled job i when job i is added to the partial sequence σ, denoted as $\sigma * i$. Select the job that leads to the minimum timeout and add it to σ; $k = k + 1$. The timeout is calculated according to Eq. (8). In case of ties, select the job which leads to the partial sequence with minimum total flow time.

$$Ind0(i,k) = \sum_{j=1}^{m}(cj,k+1(\sigma * i) - cj,k(\sigma) - pj,i) \tag{8}$$

In order to consider the total flow time criterion during the scheduling of jobs, a new term was added to index (8) to measure the contribution of the evaluated job i to total flow time of the partial sequence. Therefore, the first heuristic proposed, named HPF1, can be defined as follows:

- Step 1: Select the job with minimum P_i and put it in the first position in sequence σ. Set $k = 1$. In case of ties, select the job with minimum $p_{1,i}$.
- Step 2: While $k < n$, calculate index (*ind1*) as in Eq. (9) for each unscheduled job i, where C_i is the completion time of job i. Select the job with minimum *ind1*. In case of ties, select the job which leads to the partial sequence with minimum total flow time.

$$ind1(i,k) = \mu \cdot \left(\sum_{j=1}^{m}(c_{j,k+1}(\sigma * i) - c_{j,k}(\sigma) - p_{j,i})\right) + (1-\mu)\cdot(C_i - C_{[k-1]}) \tag{9}$$

Both procedures, PF and HPF1, schedule job i with the minimum sum of processing time on machines m (P_i) in the first position, but this rule may not always be effective in minimizing the total flow time. Consider the following 3-job, 3-machine problem where jobs J1, J2 and J3 have the following processing time on each machine: J1 = (2,3,4), J2 = (3,2,4) and J3 = (4,3,2). The sum of processing times of each job is 9, but the total flow time of schedules {J1. J2, J3}, {J2, J1, J3} and {J3, J1, J2} is 38, 39 and 41, respectively. These differences are due to the front delay induced by the first job scheduled (grey parts in Fig. 2).