Computers & Operations Research 40 (2013) 117–128

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Invited Review

A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime

Quan-Ke Pan [a,b], Rubén Ruiz [c,*]

[a] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, PR China
[b] College of Computer Science, Liaocheng University, Liaocheng 252059, PR China
[c] Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Ciudad Politécnica de la Innovación, Edifico 8G, Acc. B. Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

A R T I C L E I N F O

Available online 19 June 2012

Keywords:
Scheduling
Flowshop
Flowtime
Heuristics

A B S T R A C T

In recent years, a large number of heuristics have been proposed for the minimization of the total or mean flowtime/completion time of the well-known permutation flowshop scheduling problem. Although some literature reviews and comparisons have been made, they do not include the latest available heuristics and results are hard to compare as no common benchmarks and computing platforms have been employed. Furthermore, existing partial comparisons lack the application of powerful statistical tools. The result is that it is not clear which heuristics, especially among the recent ones, are the best. This paper presents a comprehensive review and computational evaluation as well as a statistical assessment of 22 existing heuristics. From the knowledge obtained after such a detailed comparison, five new heuristics are presented. Careful designs of experiments and analyses of variance (ANOVA) techniques are applied to guarantee sound conclusions. The comparison results identify the best existing methods and show that the five newly presented heuristics are competitive or better than the best performing ones in the literature for the permutation flowshop problem with the total completion time criterion.

Contents

* Corresponding author. Tel.: +34 96 387 70 07x74946; fax: +34 96 387 74 99.
E-mail addresses: panquanke@gmail.com (Q.-K. Pan), rruiz@eio.upv.es (R. Ruiz).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.05.018

1. Introduction

A flowshop is a common layout in production shops where m continuously available machines are disposed in series. Each machine is a production stage and products must visit all machines in order. Scheduling in a flowshop entails the production of n known jobs from a set $J=\{1, 2,..., n\}$. All the n jobs follow the same order of visitation to the machines. This order is, without loss of generality, machine 1, machine 2 and so un until machine m. Each job requires a given known, deterministic and non-negative processing time at each machine, denoted as $p_{i,j}$, $j\in J$, $i=1, 2,\ldots, m$. The flowshop scheduling problem or FSP in short is a theoretical version of reality and several simplifying assumptions apply: all jobs are independent and available for processing at time 0; machines are continuously available; each job is either waiting for processing or being processed by a machine at any given time; machines can only process one job at a time, etc. A complete list of these assumptions is detailed, for example, in Baker [3]. A solution for the FSP is a production sequence or schedule for all jobs which aims at optimizing a given criterion. Most optimization criteria in scheduling are based on the completion times of the jobs or C_j. The time at which a given job finishes processing at a given machine is denoted as $C_{i,j}$ and therefore, $C_{m,j}=C_j$. The most common and widely studied optimization criterion in the flowshop problem is the makespan or $C_{\max}$ minimization. Minimizing makespan is important in situations where a batch of jobs is received and it is required to be completed as soon as possible. For example, a multi-item order submitted by a single customer which needs to be delivered at the earliest possible time. The makespan criterion also increases the utilization of machines. The paper of Johnson [20] is recognized as the pioneering work for the FSP where the specific cases of two and three machines were studied with the objective of makespan minimization. Since then, the FSP has attracted considerable attention from researchers and hundreds of papers have been published in scheduling and related journals. The vast majority of research on flowshop scheduling deals with makespan minimization and several survey papers have been published like those of Framinan et al. [7], Ruiz and Maroto [39], Hejazi and Saghafian [16] and Gupta and Stafford [15].

As of late, there has been an increasing interest in other objective functions. Sometimes each job is needed as soon as it is completed. Similarly, the need to reduce Work In Process (WIP) or in-process inventory has fostered the study of the total flowtime, also referred to as total completion time. When all jobs are available for processing at time 0 (i.e., no release times) the flowtime of a job is equal to its completion time and hence, the total flowtime is equal to $\sum_{j=1}^{n} C_j$. Flowtime minimization leads to a more stable utilization of machines. The FSP with a total flowtime minimization objective was initially classified as $n/m/F/\sum C_j$ following the four parameter notation A/B/C/D of Conway et al. [5]. Later, it has been denoted as $F//\sum C_j$ using the three field notation $\alpha/\beta/\gamma$ of Graham et al. [13]. In the most general setting, the FSP has a search space of $(n!)^m$ sequences. However, the majority of the published research deals with a more restricted version, the so called permutation flowshop scheduling problem of PFSP in which job passing is not allowed and all machines follow the same sequence of jobs. In this case, the search space reduces to $n!$ sequences. The PFSP is classified as $n/m/P/\sum C_j$ or $F/prmu/\sum C_j$ according to Pinedo [34]. We will refer to this last problem with flowtime objective as PFSP-TFT in short. The PFSP-TFT was demonstrated to be NP-Hard in the strong sense for two or more machines by Gonzalez and Sahni [12].

Initial efforts focused on the development of exact implicit enumeration techniques and on approximate approaches to obtain good (but not necessarily optimal) solutions. These solution techniques can be broadly classified into two groups referred to as heuristics and metaheuristics, respectively. Some initial heuristics for the PFSP were introduced by Campbell et al. [4], Gupta [14] and Miyazaki et al. [29], to name just a few. Metaheuristics include many different approaches, like genetic algorithms [42], simulated annealing [44], differential evolution [33] and many others. A metaheuristic method usually obtains better solutions than heuristic algorithms but normally at the cost of significantly added CPU time. Heuristics typically need no more than a few seconds whereas metaheuristics might take several minutes. This is problematic, especially if there are real time requirements or large scale problems [25]. Furthermore, effective and efficient heuristics are still needed in metaheuristic methods for the initial seed sequence. As a result, heuristics are still essential in the scheduling community.

This paper focuses on heuristics for the PFSP-TFT. The flowshop literature already contains some reviews such as Framinan et al. [11]. However, there is room for improvement: comparisons have been performed among no more than a few heuristics; the latest heuristics have not been compared; no common data sets have been used and available results cannot be easily generalized or are not even reproducible; existing comparisons have not carried out comprehensive statistical testing. For all these reasons, we provide an up to date comprehensive review and evaluation of the existing heuristics. From the knowledge obtained after such evaluation we also present five heuristics for the problem under consideration. In total we compare 27 heuristics, which are put through comprehensive computational and statistical testing. The benchmark of choice is given by Taillard [41]. Our results attest to the fact that the five presented heuristics outperform all heuristics proposed up to date.

The rest of the paper is organized as follows: in Section 2, the most well-known heuristics for the PFSP-TFT are reviewed. Section 3 presents the five new heuristics in detail. A comprehensive comparison of the various heuristics is given in Section 4. Finally, we conclude the paper in Section 5.

2. Heuristics for the flowshop scheduling problem

Framinan et al. [11] divided the existing heuristics into two groups: simple and composite methods. A heuristic commonly consists of one or more of three typical phases, namely index development, solution construction, and solution improvement. According to Framinan et al. [11], the method is regarded as composite if it employs a simple heuristic for one or more of the three above-mentioned phases [11]. Conversely, it is regarded as a simple method if no phase contains a heuristic. This distinction is sometimes not easy to apply for some methods but it represents a simple framework. Our literature review is therefore divided between simple and composite heuristics.

2.1. Simple heuristics

The CDS heuristic introduced by Campbell et al. [4] is a simple heuristic for the PFSP. It is basically an extension of the algorithm of Johnson [20]. The CDS creates m^{-1} problems with of two “virtual”