European Journal of Operational Research 240 (2015) 666–677

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

New hard benchmark for flowshop scheduling problems minimising makespan

Eva Vallada [a], Rubén Ruiz [a,*], Jose M. Framinan [b]

[a] Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Ciudad Politécnica de la Innovación, Edificio 8G, Acc. B. Universitat Politécnica de València, Camino de Vera s/n, 46021 València, Spain
[b] Industrial Management School of Engineering, University of Seville, Av.Descubrimientos s/n, 41092 Seville, Spain

ARTICLE INFO

Article history:
Received 18 July 2013
Accepted 25 July 2014
Available online 12 August 2014

Keywords:
Flowshop
Makespan
Benchmark

ABSTRACT

In this work a new benchmark of hard instances for the permutation flowshop scheduling problem with the objective of minimising the makespan is proposed. The new benchmark consists of 240 large instances and 240 small instances with up to 800 jobs and 60 machines. One of the objectives of the work is to generate a benchmark which satisfies the desired characteristics of any benchmark: comprehensive, amenable for statistical analysis and discriminant when several algorithms are compared. An exhaustive experimental procedure is carried out in order to select the hard instances, generating thousands of instances and selecting the hardest ones from the point of view of a gap computed as the difference between very good upper and lower bounds for each instance. Extensive generation and computational experiments, which have taken almost six years of combined CPU time, demonstrate that the proposed benchmark is harder and with more discriminant power than the most common benchmark from the literature. Moreover, a website is developed for researchers in order to share sets of instances, best known solutions and lower bounds, etc. for any combinatorial optimisation problem.

1. Introduction

Advancements in algorithms in the field of operational research frequently require careful and comprehensive computational comparisons against well known and established benchmarks of instances. Once a standard set of problems is recognised as the de facto standard, different proposed techniques can be easily compared using this set. As per the recommendations of Beasley (1990), such benchmarks are nowadays shared easily through the Internet and the best known solutions (usually in the form of best known upper bounds in minimisation problems) are shared and used in order to compare presented algorithms against such bounds.

The importance of benchmarks cannot be overstated. A result is published only after showing better performance for a given problem in the standard benchmark accepted by peers most of the time. Therefore, the quality of the benchmark is of paramount importance. Poorly designed benchmarks might not be representative of real problems. Furthermore, other problems might arise. The set of instances might be of a limited size, too easy or specific for a given combination of input parameters. In such cases, if a given method outperforms another in the benchmark, it is not guaranteed that the performance can be generalised over the population of real instances.

One of the major fields in operational research is scheduling. This is recognised by Potts and Strusevich (2009) where it is stated that hundreds of papers are published per year in all relevant journals in the field. In scheduling, the pioneering work is the paper of Johnson (1954) where the famous two machine flowshop scheduling problem with makespan minimisation criterion was studied. Therefore, flowshop scheduling has been in the spotlight ever since. This prolific field is summarised in the reviews of Framinan, Gupta, and Leisten (2004); Ruiz and Maroto (2005); Hejazi and Saghafian (2005) or in Gupta and Stafford (2006). Reviews for other objectives apart from makespan are given in Vallada, Ruiz, and Minella (2008) for tardiness related criteria and in Pan and Ruiz (2013) for flowtime objectives. Literally hundreds of papers have been proposed just for the minimisation of the makespan in flowshop problems, even more if one considers all other studied objectives. This paper focuses specifically on the flowshop problem.

The most widely used benchmark for flowshop scheduling is that of Taillard (1993). There are other much less employed benchmarks, like the ones of Demirkol, Mehta, and Uzsoy (1998) or

* Corresponding author. Tel.: +34 96 387 70 07x74946; fax: +34 96 387 74 99.
E-mail addresses: evallada@eio.upv.es (E. Vallada), rruiz@eio.upv.es (R. Ruiz), framinan@us.es (J.M. Framinan).

http://dx.doi.org/10.1016/j.ejor.2014.07.033
0377-2217/

Reeves (1995) or older benchmarks that are not currently being used, like the ones of Carlier (1978) and Heller (1960). Taillard's benchmark comprises 120 instances for the flowshop problem that range from 20 jobs and 5 machines all the way up to 500 jobs and 20 machines. At the time of writing, only 28 instances[1] in the benchmark are "open" meaning that the optimum solution has not yet been found. As we will show, several authors have recently been unable to assess outperformance in Taillard's benchmark due to several factors that we will later point out. Notable examples are Dong and Ping Chen (2008) and Kalczynski and Kamburowski (2008). These authors could not find statistically better performance using Taillard's benchmark and showed that using other randomly generated instances of their own, better performance was observed. In a sense, Taillard's benchmark is reaching exhaustion.

The previous potential problems, along with other shortcomings motivate this research. In this paper we present a new, computationally challenging and comprehensive benchmark for the flowshop scheduling problem with makespan criterion. First, we define the flowshop problem and study the existing literature in an attempt at characterising the hardness of flowshop instances in Section 2. Then, following a large computational campaign, we present the new benchmark in Section 3. Contrary to existing research, where benchmarks are simply presented, we carry out a comprehensive computational and statistical testing of the presented benchmark in Section 4. We compare the statistical capability of the new benchmark against the benchmark of Taillard with successful results. Another contribution of this research is the new website of instances with many potentially useful features for other researchers to use. This web 2.0 portal contains different benchmarks along with historical data of best results, lower bounds and all types of information as the data is held in a database and in a content management system. All this is explained in Section 5. Finally, Section 6 concludes the paper and gives further research directions.

2. Flowshop scheduling problem and the hardness of the instances

The problem consists of determining a processing sequence of n jobs in a set of m machines that are disposed in series. All jobs must be processed sequentially in all machines. This processing sequence is, without loss of generality, $\{1,\ldots,m\}$. Each job j, $j=\{1,\ldots,n\}$ needs a processing time of p_{ij} units at each machine i, $i=\{1,\ldots,m\}$. This processing time is a non-negative, deterministic and known amount. A flowshop is a common production setting in factories where products start processing at machine or stage 1 and continue processing until they are finished in the last machine m. The determination of a production sequence for all machines needs the exploration of $(n!)^m$ sequences, as there are $n!$ possible job permutations at each machine and this permutation can be changed from machine to machine with what is known as job passing. However, a common simplification in the flowshop literature is to consider only $n!$ schedules and once the production sequence of jobs for the first machine is determined, is kept unaltered for all other machines. This simplified problem is known as the permutation flowshop scheduling problem or PFSP in short. The completion time of a job in the factory is denoted as C_j. The most common objective for the PFSP is the minimisation of the maximum C_j. This is referred to as makespan and denoted as $C_{\max}$.

Johnson (1954) represents the earliest known contribution in the literature, where the author studied the two machine flowshop problem with makespan minimisation. From this work, the well known Johnson's algorithm can be used to optimally solve the problem. In general, for m machines, the problem is denoted as $F/prmu/C_{\max}$ using the well known $\alpha/\beta/\gamma$ notation of Graham, Lawler, Lenstra, and Rinnooy Kan (1979). When $m \geqslant 3$ the flowshop problem is known to be $\mathcal{NP}$-hard for $C_{\max}$ minimisation as per the results of Garey, Johnson, and Sethi (1976).

According to the results of the computational comparison of Ruiz and Maroto (2005), the NEH heuristic of Nawaz, Enscore, and Ham (1983) is a clear performer. More recent methods, such as those of Dong and Ping Chen (2008); Kalczynski and Kamburowski (2008) Rad, Ruiz, and Boroojerdian (2009) have shown NEH outperforming algorithms. As regards metaheuristics, the list is also long. In this case, some of the best performing methods are the Hybrid Genetic Algorithm of Ruiz, Maroto, and Alcaraz (2006) and the Iterated Greedy of Ruiz and Stützle (2007). With Taillard's benchmark, the state-of-the-art as regards metaheuristics for the PFSP has reached a high level of maturity. For example, Vallada and Ruiz (2009) managed to obtain, with a parallel iterated greedy method, an average percentage deviation over the best known solutions of Taillard's benchmark of only 0.25%. However, this small deviation might just be another sign of Taillard's benchmark aging. From the 120 instances of Taillard, the optimum solution is known today for 92 instances. For the remaining 28, the average gap between the best known solution and the highest known lower bound is just 0.94%. Therefore, given the current state-of-the-art performance and how close to the best known solutions are for Taillard's instances, there is a big potential problem in the near future: New and better methods might end up being disregarded due to it not being possible to show better performance than existing algorithms in Taillard's benchmark. However, the fact that method A does not give better solutions than method B in a benchmark that has been practically solved does not mean that in another harder and/or bigger benchmark, method A would not give better solutions.

Let us note that other existing benchmarks for the PFPS and $C_{\max}$ criterion are not more difficult than Taillard's. For example, Demirkol et al. (1998) proposed a total of 600 instances for different flow and job shop problems, including objectives with due dates. As regards flowshop scheduling, the paper presented 120 instances for makespan minimisation. The problem with these instances is that they only reach 50 jobs and 20 machines, which is a much smaller size than the benchmark of Taillard.

In order to come up with a new benchmark one has to make sure that the instances are varied, numerous, representative of real-life situations and, above all, hard. The reason behind the needed hardness is that the benchmark needs to have discriminant power, i.e., given two methods A and B, we need to conclude if A is better than B. If both A and B are very good performers, they might be able to solve easy instances to almost optimality in most cases and thus, the benchmark will be of no use. In summary, the desired characteristics of a good benchmark are the following:

- Exhaustive: large number of instances, small and large sized instances, different combinations of instance size.
- Amenable for statistical analysis: equidistant, that is, the number of jobs and machines go up by a uniform quantity each time.
- Discriminant: statistically significant differences can be easily found when several algorithms are compared.

Benchmark instances have been constructed almost exclusively from uniform random distributions. It has been customary to draw the processing times from a $U[1,99]$ distribution. This is the case of Taillard's benchmark. It is also known that uniformly distributed processing times result in instances that are harder to solve by algorithms. This has created a number of debates. In real-life it is

[1] The list of best known solutions is found at http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/flowshop.dir/best_lb_up.txt.