Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 1350–1373
computers & operations research
www.elsevier.com/locate/cor

Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics

Eva Vallada*, Rubén Ruiz, Gerardo Minella

Dpto. de Estadística e Investigación Operativa Aplicadas y Calidad, Universidad Politécnica de Valencia, Edificio I-3, Camino de Vera S/N, 46021 Valencia, Spain

Available online 10 October 2006

Abstract

In this work, a review and comprehensive evaluation of heuristics and metaheuristics for the m-machine flowshop scheduling problem with the objective of minimising total tardiness is presented. Published reviews about this objective usually deal with a single machine or parallel machines and no recent methods are compared. Moreover, the existing reviews do not use the same benchmark of instances and the results are difficult to reproduce and generalise. We have implemented a total of 40 different heuristics and metaheuristics and we have analysed their performance under the same benchmark of instances in order to make a global and fair comparison. In this comparison, we study from the classical priority rules to the most recent tabu search, simulated annealing and genetic algorithms. In the evaluations we use the experimental design approach and careful statistical analyses to validate the effectiveness of the different methods tested. The results allow us to clearly identify the state-of-the-art methods.

Keywords: Flowshop; Scheduling; Tardiness; Heuristics; Metaheuristics

1. Introduction

The flowshop scheduling problem is one of the most thoroughly studied problems in combinatorial optimisation. Its simple definition conceals a very difficult problem which has attracted the attention of numerous researchers in the past decades. In a flowshop there is a set $N = \{1, \dots, n\}$ of n jobs that have to be processed on a set $M = \{1, \dots, m\}$ of m machines. All jobs follow the same route in the machines and $p_{ij} \geqslant 0$ denotes the fixed processing time of job j, $j \in N$, on machine i, $i \in M$. Among the existing regular criteria, the minimisation of the maximum completion time or makespan ($C_{\max}$) is widely used. The simplification of considering only permutation schedules (the processing order of the jobs is the same for all machines) is very common and the resulting problem is called the permutation flowshop problem (PFSP) and denoted as $F/prmu/C_{\max}$ [1]. This problem was shown to be $\mathcal{NP}$-complete by [2] for $m \geqslant 3$. We can find in [3] an extensive comparison and evaluation of different methods for this objective. Recently, research has also focused on the development of algorithms with the objective of minimising total tardiness of jobs owing its importance to real-life considerations. In industrial plants, criteria based on due dates for delivery have become more important than those based on makespan. Let d_j, $j \in N$, be the *due date* for the job j, the job's tardiness is defined as $T_j = \max\{C_j - d_j, 0\}$, where C_j is the completion time of job j. The objective is to minimise the tardiness over all jobs or

* Corresponding author. Tel.: +34 96 387 70 07x74911; fax: +34 96 387 74 99.
E-mail addresses: evallada@eio.upv.es (E. Vallada), rruiz@eio.upv.es (R. Ruiz), mgerar@eio.upv.es (G. Minella).

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.08.016

total tardiness: $\sum_{j=1}^{n} T_j$. This problem is denoted as $F/prmu/\sum T_j$ [1] and it is known to be $\mathcal{NP}$-hard in the ordinary sense even when there is only one machine and $\mathcal{NP}$-hard in the strong sense for $m \geqslant 2$. For more details see [4].

In the literature of scheduling to minimise total tardiness of jobs we can find a few review papers, most of them dealing only with one machine. In [5] the authors presented a review focused on one machine and branch and bound methods where the scheduling problems are classified according to different objectives involving due dates. In [6], a review and evaluation of heuristics for several problems is presented. The authors survey different heuristics in the single machine, parallel machine, flowshop and jobshop settings. Several methods are proposed and evaluated for both the single machine and the parallel machine tardiness problems. In [7], a comparison of heuristics for the single machine total weighted tardiness problem is carried out. Finally, [8] reviews research on the total tardiness and total weighted tardiness problems in the single machine environment, where some extensions of these problems for multi-machine environments are also presented.

As we can see, all these reviews and evaluations are mainly focused on the single machine case and no comparison deals with the most recent heuristics and metaheuristics available for the multi-machine flowshop scheduling problem. Moreover, it is very difficult to compare the methods proposed in the literature because the evaluations are partial and the benchmarks used are not standard, therefore the results are many times difficult to reproduce.

The objective of this paper is to give an up-to-date review and evaluation of many existing heuristics and metaheuristics for the m-machine PFSP to minimise the total tardiness of the jobs. We evaluate 40 methods, from the classical despatching rules to the most recent heuristic and metaheuristic methods such as tabu search, genetic algorithms and differential evolution algorithms. Furthermore, we propose a benchmark to evaluate all the methods under a common data set, so that the results can be generalised.

The remainder of this paper is organised as follows: In Section 2 we review some exact methods for the PFSP with the objective to minimise the total tardiness of the jobs. In Section 3, most well-known heuristics for the same problem are surveyed. Section 4 deals with the metaheuristics and a comparison of all methods can be seen in Section 5. Finally, some conclusions are given in Section 6.

2. Exact methods

Due to the complexity of flowshop scheduling problems, using exact methods to solve them is impracticable for instances of more than a few jobs and/or machines. In [9], a branch and bound algorithm is proposed for the two-machine flowshop problem. The authors proposed a theorem based on the interchange of jobs which should appear consecutively if they satisfy some specific rules. This theorem is used to prune some branches in the search. The authors presented a comparison against the earliest due date (EDD), shortest processing time (SPT) and minimum slack (SLACK) despatching rules. The experiments were carried out using a set of 640 problems where the processing times for both machines are randomly generated from a uniform distribution over the values 1 and 10. The due dates are also randomly generated from a uniform distribution between $P(1 - T - R/2)$ and $P(1 - T + R/2)$ following the procedure presented in [10] where T and R are two parameters called *tardiness factor* and *due date range*. The P is commonly a lower bound on the makespan but in this case is the sum of the processing times in machine two plus the smallest processing time in machine one. Several problems were proposed where $T = \{0.25, 0.5, 0.75, 1\}$, $R = \{0.25, 0.5, 0.75, 1\}$ and $n = \{6, 8, 10, 12\}$. Therefore, 64 combinations are generated each of which is repeated 10 times. The results show that the SPT rule performs very well for large T values and the number of nodes processed by the branch and bound to find an optimal solution tends to increase with increasing values of T and R.

Another branch and bound algorithm for the two-machine case is that of [11]. The authors proposed a lower bound based on the sum of two lower bounds computed from the set of jobs in the partial sequence and the set of jobs not included in it, respectively. Dominance rules to prune sequences are also presented. A total of 240 problems were randomly generated to test the performance of the method. Processing times were uniformly distributed between 1 and 30 and due dates of the jobs were computed using the method previously commented where P in this case is the sum of the processing times of all operations divided by two. Several combinations of the parameters T, R and the number of jobs were considered where $T = \{0.1, 0.2, 0.3, 0.4, 0.5\}$, $R = \{0.8, 1, 1.2, 1.4, 1.6, 1.8\}$ and $n = \{10, 11, 12, 13, 14, 15\}$. The author compares the proposed branch and bound using the lower bounds and dominance rules presented in the paper against the branch and bound proposed in [9] and a similar branch and bound but adding the dominance rule presented in [9]. The results show that the branch and bound of [9] was outperformed by the other two proposed algorithms which showed a very similar performance.