Computers and Operations Research 80 (2017) 50–60

Contents lists available at ScienceDirect

Computers and Operations Research

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

Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows

Quan-Ke Pan[a], Rubén Ruiz[b,*], Pedro Alfaro-Fernández[b]

[a] State Key Laboratory of Digital Manufacturing Equipment & Technology in Huazhong University of Science & Technology, Wuhan, 430074, PR China
[b] 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, 46021, València, Spain

ARTICLE INFO

Article history:
Received 30 September 2015
Revised 22 June 2016
Accepted 21 November 2016
Available online 22 November 2016

Keywords:
Hybrid flowshop
Due date window
Heuristics
Iterated local search
Iterated greedy

ABSTRACT

In practice due dates usually behave more like intervals rather than specific points in time. This paper studies hybrid flowshops where jobs, if completed inside a due window, are considered on time. The objective is therefore the minimization of the weighted earliness and tardiness from the due window. This objective has seldom been studied and there are almost no previous works for hybrid flowshops. We present methods based on the simple concepts of iterated greedy and iterated local search. We introduce some novel operators and characteristics, like an optimal idle time insertion procedure and a two stage local search where, in the second stage, a limited local search on a exact representation is carried out. We also present a comprehensive computational campaign, including the reimplementation and comparison of 9 competing procedures. A thorough evaluation of all methods with more than 3000 instances shows that our presented approaches yield superior results which are also demonstrated to be statistically significant. Experiments also show the contribution of the new operators in the presented methods.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Production scheduling is a very important step for manufacturing industries. Production schedules obtained by manual methods or without optimization techniques are known to have ample room for improvement as commented in [12]. Therefore, using optimization methods results in large gains ([29,37], among others). As a result, scheduling is a very active field inside operations research. Since the mid-fifties, thousands of papers have been published. However, the application of such research developments in real industry remains rare, or at least the proportion of papers tackling with realistic production settings is much lower that those papers dealing with simplified problem settings [28,29,43].

Among the different types of scheduling problems, we are interested in what is probably the most commonly found situation in many industries, which is the combination of the parallel machines and flowshop problems. The addition of the two is usually referred to as hybrid flowshop and can be defined as follows: There is a set M of $m \geq 2$ production stages where $M = \{1, \ldots, m\}$. The stages are disposed in series. At each stage i, $i \in M$ we have a set M_i of m_i identical parallel machines where $M_i = \{1, \ldots, m_i\}$ and $m_i \geq 1$, $\forall i \in M$ and $\exists i \in M$, $m_i > 1$. A set N of n jobs has to be processed on the stages. More precisely, each job j, $j \in N$ visits first stage 1, then stage 2 and so on until stage m. At each stage i, each job is to be processed by exactly one of the available m_i parallel machines. As a result, a job is made up of m different tasks, one per stage. The processing time of any job j at any stage i is a positive integer denoted by p_{ij}. Jobs are processed without interruptions.

The hybrid flowshop or HFS in short has attracted a lot of interest in the literature, given its potential practical applications. As such, there are several reviews published. Some of them are [25,39,56,57] and more recently [41,45]. According to these last two most recent hybrid flowshop reviews, by far the most commonly studied optimization criterion is the minimization of the makespan or ($C_{\max}$), defined as $C_{\max} = \max_{j=1}^{n} C_j$ where C_j is the time at which job j is finished at stage m. C_j is commonly referred to as the completion time of job j. The HFS with makespan criterion can be represented as $((PM^{(i)})_{i=1}^{m})//C_{\max}$, following the well known and accepted three field notation of Graham et al. [13] and the extension for hybrid flowshops proposed by Vignier et al. [56]. Note that there are further assumptions stemming from the flowshop nature of the problem, these are well known and are detailed elsewhere, for example in [12,31]. In the HFS for each job and stage we have to decide to which machine the job is assigned.

* Corresponding author. Fax: +34 96 387 74 99.
E-mail addresses: 2281393146@qq.com (Q.-K. Pan), rruiz@eio.upv.es (R. Ruiz), pedro.al.fe@gmail.com (P. Alfaro-Fernández).

http://dx.doi.org/10.1016/j.cor.2016.11.022
0305-0548/© 2016 Elsevier Ltd. All rights reserved.

Then, for each machine at each stage, a sequence for the processing of the assigned jobs has to be determined. The simplest possible HFS with only two stages where one stage has one single machine and the other has just two parallel machines was shown to be $\mathcal{NP}$-Hard by Gupta [14]. As a result, the best state-of-the-art exact techniques can only cope with small problems with few stages, machines and jobs. For real-sized problems metaheuristics are currently the most promising solution approach.

While makespan minimization is the most studied objective in the scheduling literature, accounting for more than 60% of the published papers in the HFS according to the study of Ruiz and Vázquez-Rodríguez [45], it is not the most appropriate objective to study nowadays. As explained in [12], makespan minimization increases machine utilization, which made sense in the early days of manufacturing where costs had to be minimized and production used to be the bottleneck. Presently, service level objectives are much more important. We define by d_j the due date of job j, $j \in J$. If $C_j > d_j$ then job j is tardy by $C_j - d_j$ units of time. The tardiness of job j is $T_j = \max\{C_j - d_j, 0\}$ and the total tardiness objective is $TT = \sum_{j=1}^{n} T_j$. About 6% of the published research in HFS considers this objective. Additionally, not all jobs are equally important. Therefore, we can use a weighted tardiness version by defining w_j as the weight of job j, $j \in J$ and the total weighted tardiness objective as $TWT = \sum_{j=1}^{n} w_j T_j$. About 2% of the reviewed literature in [45] consider this objective. Note that when $C_j < d_j$ the tardiness is always 0 regardless of how small C_j is. In real problems, finishing products early results in tied up inventory and financial burden. A possibility is to account for the earliness as well, defined as $E_j = \max\{d_j - C_j, 0\}$ or $TWE = \sum_{j=1}^{n} w'_j E_j$ with weights. Note the different weight for earliness w'_j and tardiness w_j. Minimizing earliness only potentially negates the tardiness and the solution is to minimize the total earliness and tardiness or $TWET = \sum_{j=1}^{n} \left(w'_j E_j + w_j T_j \right)$. Only 1% of the reviewed literature reviewed in [45] deal with this objective and most of the time without weights. Furthermore, the introduction of earliness into the objective results in a non-regular function with enormous implications as the objective function value can be improved by arbitrarily delaying the start or completion time of jobs at machines and semi-active schedules no longer contain the optimal solution.

Due dates that represent a specific point in time are not realistic [46]. Real due dates are actually due windows. Within the window the job is considered to be delivered on time. Therefore, we define job's j due window as $\left[d_j^-, d_j^+ \right]$. Earliness and tardiness are redefined to consider this window as $T_j^{dw} = \max\{C_j - d_j^+, 0\}$ and $E_j^{dw} = \max\{d_j^- - C_j, 0\}$, respectively. The objective function that we consider is therefore the minimization of the weighted earliness and tardiness from a distinct due date window for each job, with different weights for earliness and tardiness or $TWET^{dw} = \sum_{j=1}^{n} \left(w'_j E_j^{dw} + w_j T_j^{dw} \right)$. The full problem studied is then denoted as $((PM^{(i)})_{i=1}^{m})//TWET^{dw}$ and we will refer to it as HFSDW in short. The result is a more realistic problem that might capture more easily industrial practice with the consideration of a more real objective function where not only the tardiness is considered but also earliness to avoid stocking product way before due dates. Distinct due dates for each job and different weights for earliness and tardiness allow practitioners to transfer to the objective priorities, rush orders and many other situations. Additionally, the important consideration of due windows also captures the reality where some jobs might have a tight due window of only one morning during the week whereas other jobs might be finished without penalties in longer time frames of several days. From the complexity hierarchies of objective functions given in [37] we conclude that the studied problem in this paper is $\mathcal{NP}$-Hard since the makespan version already belongs to this complexity class. Note that we could reach the same conclusion by considering that single machine problems with due-windows are already $\mathcal{NP}$-Hard. To the best of our knowledge, the problem studied in this paper with the $TWET^{dw}$ objective function has not been considered in the literature to this date. In this paper we propose simple methods based on Iterated Local Search and on Iterated Greedy for solving the problem. The choice of simple methods is mainly due to reasons: First, simple methods are easier to understand, implement and extend to other problem variants. They have fewer parameters and the results are easy to replicate. Second, recent studies in many different scheduling problems [12,31,54] indicate that simple methods give state-of-the-art results when compared to more complex approaches. As a result, these simple methods are easily transferable to industries.

The remainder of this paper is organized as follows. In the next section we briefly review the related literature. Sections 3 and 4 describe in detail the proposed methods, which are later comprehensively tested in Section 5. Finally, Section 6 gives some concluding remarks and further research.

2. Literature review

Only 1% of the papers reviewed by Ruiz and Vázquez-Rodríguez [45] dealt with hybrid flowshop problems and earliness tardiness objectives (not considering due windows). This includes Chang and Liao [7] and Liu and Chang [26] where a *TWET* objective is considered in a linear combination with other objectives. Janiak et al. [19] also combine earliness and tardiness (*ET*) with other functions. Finke et al. [11] presented Tabu Search methods to solve the HFS problem with *ET* objectives in a simplified setting where job to machine assignments are given. Khalouli et al. [24] presented an Ant Colony Optimization (ACO) method for the HFS with *TWET* objective. Behnamian et al. [3] presented a complex hybrid of population based methods, ACO, Simulated Annealing and Variable Neighborhood Search for the same problem but with the addition of sequence dependent setup times. A similar work is that of Behnamian et al. [6] where in this case Particle Swarm Optimization is used for a related problem of group scheduling. More recently, Behnamian and Zandieh [5] have included learning effects into the problem. To the best of our knowledge, no other studies deal with *ET* in HFS. Other less closely related works are those of [20] where a no-wait hybrid flowshop with a common due date window is studied with maximization of the profit for processing jobs. Sheikh [48] adds the earliness-tardiness criterion to the previous problem. Lastly, Pan et al. [35] studies a real problem in the steelmaking industry considering, among other things, earliness an tardiness criterion. Most existing research in the hybrid flowshop literature considers, as mentioned, makespan or tardiness objectives, as in [21] but not earliness-tardiness.

As regards due windows, most existing research has been carried out on single machine problems. Starting with the work of Cheng [8] where the single machine with a common due window for all jobs is studied. Two machine flowshop problems with common due windows are studied in [58]. The combination of hybrid flowshops and due windows, as stated previously is almost non-existent in the literature. We only find recent related papers, the first one by Huang and Yu [17] which study a two stage hybrid flowshop with distinct due windows and an objective function which is a linear combination of makespan, *ET* without weights. In the second paper, the same authors Huang et al. [18] introduce reentry of jobs into the problem. These two papers disregard different weights for the jobs and for the earliness and tardiness and only consider two stages in the hybrid flowshop. Therefore and as we can see, the general hybrid flowshop with m stages and $TWET^{dw}$ objective has not been yet considered in the literature.

Download English Version:

https://daneshyari.com/en/article/4959089

Download Persian Version:

https://daneshyari.com/article/4959089

Daneshyari.com