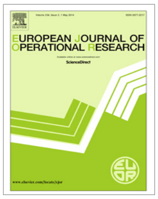
European Journal of Operational Research 237 (2014) 713–720

Contents lists available at ScienceDirect

European Journal of Operational Research

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



Innovative Application of O.R.

Flow shop scheduling with heterogeneous workers

Alexander J. Benavides [a], Marcus Ritt [a,*], Cristóbal Miralles [b]

[a] Instituto de Informática, Universidade Federal do Rio Grande do Sul, Brazil
[b] ROGLE, Departamento de Organización de Empresas, Universitat Politècnica de València, Spain

ARTICLE INFO

Article history:
Received 12 March 2013
Accepted 4 February 2014
Available online 12 February 2014

Keywords:
Scheduling
Metaheuristics
Flow shop
Integer programming
Heterogeneous workers

ABSTRACT

We propose an extension to the flow shop scheduling problem named Heterogeneous Flow Shop Scheduling Problem (Het-FSSP), where two simultaneous issues have to be resolved: finding the best worker assignment to the workstations, and solving the corresponding scheduling problem. This problem is motivated by Sheltered Work centers for Disabled, whose main objective is the labor integration of persons with disabilities, an important aim not only for these centers but for any company desiring to overcome the traditional standardized vision of the workforce. In such a scenario the goal is to maintain high productivity levels by minimizing the maximum completion time, while respecting the diverse capabilities and paces of the heterogeneous workers, which increases the complexity of finding an optimal schedule. We present a mathematical model that extends a flow shop model to admit a heterogeneous worker assignment, and propose a heuristic based on scatter search and path relinking to solve the problem. Computational results show that this approach finds good solutions within a short time, providing the production managers with practical approaches for this combined assignment and scheduling problem.

1. Introduction

According to the International Labour Organization, persons with disabilities represent an estimated ten percent of the worlds population. This amounts to about 700 million people worldwide, out of whom almost 500 million are in working age and usually suffering much higher unemployment rates than other people. Governments have implemented different policies for the integration of persons with disabilities such as reserving a percentage of jobs in companies, or creating Sheltered Work centers for Disabled (SWDs). SWDs have been adopted successfully in countries such as Spain, facilitating jobs for disabled workers, both as a transition formula towards real integration or as stable workplaces, by overcoming certain prejudices and by considering the workforce as heterogeneous as it actually is. Although SWDs receive some institutional support, they compete in real labor markets. Thus, they need to be efficient and competitive, not only for their survival but also to be able to grow (to promote new jobs), and they must ensure socialemployment integration for their workers, taking into account their limitations and aiming to evolve positively their capabilities and capacities (Miralles, Marin-Garcia, Ferrus, & Costa, 2010).

Most Operations Research/Management Science (OR/MS) approaches and tools standardize the processing time of every operation independently of the worker that performs it. This assumption is not realistic and may cause serious planning and control problems. Moreover, the decision making process of defining the master schedule or the workplace assignments becomes harder when the production manager has to cope with rigid information systems that disregard heterogeneity of skills. In these cases the next best solution is often to compensate in advance the existent deviations in the workforce (sometimes with rules of thumb), or by using a not very efficient “worst case” scenario. These measures lead to implement suboptimal solutions and many output results will be difficult to check, because of aggregated effects from the planning compensations, and because of rush adaptations made by the workers themselves, with clear differences between the defined scenarios and the reality, and without any control in the corresponding indicators. Instead, it would be more reliable to assume the workers heterogeneity, to estimate a priori the deviations, and to feed the planning/scheduling process with real, approximate data including this heterogeneity.

The OR/MS area is progressively proposing approaches that address the human diversity in the procedures for Design, Planning & Control of productive systems, thus contributing to narrow the gap between research and practice. Some authors do this by using a fuzzy or stochastic model of task execution times (e.g. Erel, Sabuncuoglu, & Sekerci, 2005), and others by directly categorizing

---

* Corresponding author. Tel.: +55 5133086818; fax: +55 5133087308.
E-mail addresses: ajbenavides@inf.ufrgs.br (A.J. Benavides), marcus.ritt@inf.ufrgs.br (M. Ritt), cmiralles@omp.upv.es (C. Miralles).

http://dx.doi.org/10.1016/j.ejor.2014.02.012
0377-2217/© 2014 Elsevier B.V. All rights reserved.

different kinds of workers (e.g. Corominas, Pastor, & Plans, 2008). The Assembly Line Worker Assignment and Balancing Problem (ALWABP) focuses in the heterogeneity of task times and the presence of incompatibilities, defining a new set of realistic hypotheses originally inspired by assembly lines in SWDs. Thus, from the initial paper of Miralles, Garcia-Sabater, Andrés, and Cardos (2007), many other authors have contributed to give this problem visibility proposing different extensions and methods to solve it (see for example Miralles et al. (2010), Blum & Miralles (2011), Moreira, Ritt, Costa, & Chaves (2012) and Corominas, Ferrer, & Pastor (2012)).

1.1. Contribution and outline of the paper

To the best of our knowledge, the problem of worker diversity when scheduling jobs in flow shop systems has not been addressed in the literature. If the workers are heterogeneous, then the processing time of an operation performed on a machine will depend on the assigned worker. The term "machine" is the usual term in the flow shop literature, but it refers to a work center in which the process performed may be automated or manual. Furthermore, in such a scenario, the complete solution of the problem consists of two elements: as usual, a schedule of jobs that optimizes a given objective function, and, additionally, the optimal allocation of workers to work centers (machines) that helps to get the best possible solution. The additional worker assignment increases the number of possible solutions of the normal non-permutation flow shop by a factor of $m!$. Therefore, sophisticated resolution methods are necessary in order to obtain the optimal worker allocation and the corresponding optimal schedule in reasonable computation time. In fact computation time is important due to rescheduling caused by workers rotation, turnover and absenteeism, mainly in SWDs where periodical health and psychological support are mandatory (Miralles et al., 2007).

Although the initial scenario that inspired this research was a SWD (as a perfect test bed where more than 70% of workers are disabled with different skills), this paper aims to present new models and approaches to cope with issues that arise in ordinary companies when the natural human diversity of workers needs to be considered, with or without regard to disabilities.

The next section reviews the main references on flow shop scheduling in the literature. Section 3 gives an example of the problem of finding a flow shop schedule for a heterogeneous workforce. In Section 4 we define the problem formally by a mathematical model, and in Section 5 we propose a scatter search with path relinking for solving it. We define a new set of instances which models the situation of heterogeneous workers in Section 6, and present computational experiments with standard solvers and the proposed heuristic. Finally, in Section 7, we analyze and discuss the results and conclude.

2. The flow shop scheduling problem

In a flow shop scheduling problem (FSSP) we have to schedule a set of jobs $J_1, \ldots, J_n$ on machines $M_1, \ldots, M_m$. Job J_j consists of m operations $o_{1j}, \ldots, o_{mj}$. Operation o_{ij} must be processed on machine M_i in time p_{ij} without preemption. The jobs cannot be processed in parallel, and the machines can execute only one operation at any given instant. Each job must be processed on the machines in the given order. If, in addition, each machine has to process all jobs in the same order, we obtain the particular case of a *permutation flow shop scheduling problem* (PFSSP). The most common objective function in flow shop scheduling is to minimize the makespan $C_{\max}$, i.e. the maximum completion time of any operation, which is also the focus of this paper. The two problems are also denoted by $F||C_{\max}$ and $F|prmu|C_{\max}$. For up to three machines both have the same optimal makespan. $F||C_{\max}$ can be solved in time $O(n \log n)$ for two machines, but is strongly NP-hard for three or more machines.

Fig. 1 shows an example of the FSSP. The table on the left gives the processing times of four jobs on four machines. The optimal schedule shown on the right has makespan 11. Note that this is a non-permutation schedule, since jobs two and three exchange their processing order on machine three. In this paper we permit non-permutation schedules, since our motivating application in SWDs is not restricted to permutation schedules.

The makespan of a schedule is defined by a *critical path*, which is a sequence of consecutive operations on the same machine or of the same job that leads to the longest overall duration. A critical path can be decomposed in maximal blocks of operations that are executed on the same machine. Critical paths are important in exact and heuristic algorithms, since we must reduce their total time in order to improve a schedule. In the example of the schedule of Fig. 1, the critical path is given by the striped operations.

2.1. Related work

Since most of the extensive literature on flow shop scheduling focuses on the PFSSP, and these results are often also useful in the general case, we give first a brief overview on methods restricted to permutation schedules, and then discuss previous work on non-permutation schedules. For a more detailed account we refer to the excellent surveys of Ruiz and Maroto (2005) and Potts and Strusevich (2009).

Due to the difficulty of the PFSSP, most authors propose heuristics for obtaining approximate solutions. Among the constructive heuristics proposed for the PFSSP, variants of the algorithm NEH of Nawaz, Enscore, and Ham (1983) have been consistently found to perform best (Kalczynski & Kamburowski, 2011; Ruiz & Maroto, 2005).

The currently best performing improvement and recombination heuristics are the ant colony algorithms of Rajendran and Ziegler (2004), the hybrid genetic algorithm of Ruiz, Marato, and Alcaraz (2006), the hybrid particle swarm optimization of Tasgetiren, Liang, Sevkli, and Gencyilmaz (2007), the iterated greedy algorithm of Ruiz and Stützle (2007), and the hybridization of a genetic algorithm with variable neighborhood search of Zobolas, Tarantilis, and Ioannou (2009). On standard instances these heuristics produce in time $nm/10$ seconds schedules whose makespan deviates less than 1% from the best known values.

Algorithms with proven guarantees are much more limited: the best (randomized) polynomial-time approximation algorithm comes only within a factor $O(\sqrt{\min\{m,n\}})$ of the optimal makespan (Nagarajan & Sviridenko, 2009). State-of-the-art exact algorithms based on branch-and-bound solve most of the standard instances up to 10 machines within a few hours, but are unable to solve instances with more machines (Companys & Mateo, 2007; Ladhari & Haouari, 2005).

Since the worst case gap between non-permutation and permutation schedules is a factor of $\Theta(\sqrt{m})$ (Nagarajan & Sviridenko, 2009), non-permutation schedules may be much shorter. The gain in practical instances, however, seems limited to a few percent (Liao, Liao, & Tseng, 2006; Tandon, Cummings, & Levan, 1991).

Several constructive heuristics that produce non-permutation schedules have been proposed. Gonzalez and Sahni (1978) solve two-machine flow shop problems on subsequent machine pairs optimally and join the obtained schedules, in time $O(mn \log n)$. Their algorithm is a $\lceil m/2 \rceil$-approximation. The guarantee has been slightly improved by Chen, Glass, Potts, and Strusevich (1996) to $m/2 + 1/6$. The heuristic of Koulamas (1998) solves all $\binom{m}{2}$