



Discrete Optimization

A scatter search algorithm for the distributed permutation flowshop scheduling problem

Bahman Naderi^a, Rubén Ruiz^{b,*}^a Department of Industrial Engineering, Faculty of Engineering, University of Kharazmi, Karaj, Iran^b Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Ciudad Politécnica de la Innovación, Universitat Politècnica de València, Edificio 8G, Acc. B, Camino de Vera s/n, 46021 València, Spain

ARTICLE INFO

Article history:

Received 11 January 2014

Accepted 15 May 2014

Available online 29 May 2014

Keywords:

Distributed scheduling

Permutation flowshop

Scatter search

ABSTRACT

The distributed permutation flowshop problem has been recently proposed as a generalization of the regular flowshop setting where more than one factory is available to process jobs. Distributed manufacturing is a common situation for large enterprises that compete in a globalized market. The problem has two dimensions: assigning jobs to factories and scheduling the jobs assigned to each factory. Despite being recently introduced, this interesting scheduling problem has attracted attention and several heuristic and metaheuristic methods have been proposed in the literature. In this paper we present a scatter search (SS) method for this problem to optimize makespan. SS has seldom been explored for flowshop settings. In the proposed algorithm we employ some advanced techniques like a reference set made up of complete and partial solutions along with other features like restarts and local search. A comprehensive computational campaign including 10 existing algorithms, together with statistical analyses, shows that the proposed scatter search algorithm produces better results than existing algorithms by a significant margin. Moreover all 720 known best solutions for this problem are improved.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Scheduling deals with the allocation of resources, typically machines, to tasks (commonly referred to as jobs) over time with the goal of optimizing a given objective (Pinedo, 2012). Scheduling is an important problem that appears mainly in manufacturing industries. It is well known that good schedules contribute greatly to the overall performance of a company (McKay, Pinedo, & Webster, 2002). The layout of the machines on the production floor, along with the flow of the jobs in the machines, together with a myriad of constraints and real life settings determine the type of scheduling problem to solve. The flowshop scheduling problem (FSP) is arguably the most common processing layout in practice as it is typical for manufacturing plants to manufacture a given family of products that have to visit machines in a known order. For example, in car manufacturing, the painting of the car body must go after the body as been welded and before any assembly operation, hence a flowshop structure. Reisman, Kumar, and Motwani (1997) reviewed practical cases and concluded that the flowshop problem has many real life applications. This applicability of the flowshop is also highlighted in the many exiting reviews

from the literature like Framinan, Gupta, and Leisten (2004), Ruiz and Maroto (2005), Hejazi and Saghafian (2005) and Gupta and Stafford (2006). As a matter of fact, once generalized to hybrid flowshops or flexible flowline problems, many production problems can be modeled after a flowshop (Linn & Zhang, 1999; Vignier, Billaut, & Proust, 1999; Wang, 2005; Quadt & Kuhn, 2007; Ruiz & Vázquez-Rodríguez, 2010; Ribas, Leisten, & Framinan, 2010). The FSP can be formally described as follows: A set N of n different and independent jobs has to be scheduled. Jobs usually model client orders or batches of products to be manufactured after a production planning process has been carried out (Pochet & Wolsey, 2006). Each job j , $j \in N$ has to visit, in order, all m machines in the set of machines M . Without loss of generality, each job visits first machine 1, then machine 2 and so on until machine m . A job cannot go to the next machine until it is finished in the current machine and a machine cannot process more than one job at the same time. As a result of the machines being disposed in series, each job is broken down into m tasks, one per machine. Each task from a job j , $j \in N$ needs a given processing time at each machine i , $i \in M$. This processing time is denoted as p_{ij} and it is deterministic, known in advance and usually non-negative, represented by an integer quantity.

The objective in the FSP is to find a schedule or processing sequence of all the jobs in the machines such that a given

* Corresponding author. Tel.: +34 96 387 70 07; fax: +34 96 387 74 99.

E-mail addresses: bahman.naderi@aut.ac.ir (B. Naderi), rruiz@eio.upv.es (R. Ruiz).

optimization criterion is optimized. According to the previously cited review papers, the most commonly studied objective is the minimization of the maximum completion time or makespan, denoted as C_{\max} . Given the completion time of a job in the last machine m , denoted as C_j , the makespan is then the minimization of the maximal C_j , $j = 1, \dots, n$. Since there are as many possible schedules as job sequences at each machine, the total number of solutions is $(n!)^m$, i.e., all possible job permutations at each machine, considering that these permutations can change from machine to machine. Given this huge search space, the FSP is usually simplified to what is called the Permutation Flowshop Scheduling Problem or PFSP by forbidding job-passing, i.e., once the production sequence is fixed for a machine, all machines follow the same production sequence. This brings down the total number of solutions to $n!$. Using the well known three field notation for scheduling problems (Graham, Lawler, Lenstra, & Rinnooy Kan, 1979; Pinedo, 2012), the PFSP with makespan criterion is denoted as $F/prmu/C_{\max}$.

This problem was first studied almost 60 years ago by Johnson (1954) where the well known Johnson's algorithm was proposed for solving the two machine version. For three or more machines, the problem is known to be \mathcal{NP} -Complete in the strong sense Garey, Johnson, and Sethi (1976). Nowadays, the literature on the PFSP is immense and the problem and many variants have been thoroughly studied. The topic is so widely studied that there are even some dedicated monographs such as Chakraborty (2009) and Emmons and Vairaktarakis (2012), or even for some variants, like lot streaming in Sarin and Jaiprakash (2007). However, there is one extension that was only recently presented. In Naderi and Ruiz (2010) and Naderi and Ruiz (2010) studied a variant that was referred to as the Distributed Permutation Flowshop Scheduling Problem or DPFSP. In essence, the regular PFSP considers one single factory where products are manufactured. However, multi-factory enterprises are much more competitive in a globalized economy. The literature on manufacturing systems abounds with examples where it is shown that distributed manufacturing is key for high product quality, low production costs and reduced management risks, among many other benefits (Wang, 1997; Moon, Kim, & Hur, 2002; Kahn, 2004, among many others). Distributed manufacturing is now a topic of interest as the recent editorial in a special issue of a reputable manufacturing journal shows (Chan & Chung, 2013). In that editorial and in many of the papers of the cited special issue the importance and benefits of distributed manufacturing are praised and highlighted. In the DPFSP there is an important added complexity with respect to the PFSP: Jobs need to be assigned to factories and then a schedule must be built for each factory. More formally, the DPFSP extends the regular permutation flowshop in the following way: The set N of n jobs must be processed by a set G of F identical factories. Each factory has the same set M of m machines. The processing times of all the tasks of a given job do not change from factory to factory. Once assigned to a factory, a job has to be completed in that factory. The objective is to minimize the maximum makespan among all factories. Naderi and Ruiz (2010) referred to this problem as $DF/prmu/C_{\max}$. The same authors demonstrated that no factory must be left empty with no jobs assigned (given $n > F$) as this does not improve the makespan value. They also concluded that the total number of solutions in the DPFSP is $\binom{n-1}{F-1} n!$. Additionally, since the DPFSP reduces to the regular PFSP if $F = 1$, it is easy to conclude that the DPFSP is also an \mathcal{NP} -Hard problem.

From the paper of Naderi and Ruiz (2010), several other authors built upon those results and several methodologies have been proposed to solve this new problem. Naderi and Ruiz (2010) proposed some mathematical models, simple heuristics and local search

methods. Therefore, more complex methodologies might reveal new interesting solutions to this hard combinatorial problem. Furthermore, given the existing recent methods proposed, it is also worthwhile comparing the effectiveness and efficiency of existing approaches to ascertain which are the state-of-the-art methods. These are some of the objectives of this paper. When deciding about which advanced techniques could be applied to the DPFSP we observed that simple local search based methods failed to escape strong local optima and therefore we chose a powerful methodology: Scatter Search (Glover, Laguna, & Martí, 2000; Laguna & Martí, 2003; Martí, Laguna, & Glover, 2006, among others). Contrary to many existing metaheuristic frameworks, which have been applied several times to flowshop problems, scatter search (SS) has seldom been used for these scheduling settings. References with applications of scatter search to regular flowshops are scarce. Nowicki and Smutnicki (2006) presented some methods, including ideas from path relinking and scatter search to the regular PFSP with makespan criterion but failed to significantly advance the state-of-the-art. In a short paper, Saravanan, Noorul Haq, Vivekraj, and Prasad (2008) proposed another scatter search method for the same problem and reported average percentage deviations over the best known solutions for the benchmark of Taillard (1993) of a little over 1%. This is clearly not better than the deviations below 0.5% given by the simpler Iterated Greedy (IG) method of Ruiz and Stützel (2007) or the deviations of just 0.22% given in Vallada and Ruiz (2009). As regards the PFSP, it seems that there are no other noteworthy scatter search applications. Therefore, it is plausible to think that scatter search methods for flowshop problems still have some headroom for improvement and therefore we choose them for this paper. Furthermore, the controlled diversification in scatter search shows, as we will empirically demonstrate, great strength in the DPFSP.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review on the DPFSP. Section 3 presents in detail the proposed scatter search approach. This method is calibrated in Section 4. In the same Section, almost all relevant algorithms from the literature on the DPFSP are reimplemented and carefully evaluated. Through comprehensive computational and statistical analyses we show that the presented scatter search algorithm can be considered as the new state-of-the-art method for the DPFSP and makespan minimization. Finally, Section 5 concludes this paper and proposes some avenues for future research.

2. Literature review

In Naderi and Ruiz (2010) the authors presented six different Mixed Integer Linear Programming models for the DPFSP together with 12 heuristics that resulted from applying two different job to factory assignment rules to six famous heuristics for the regular flowshop problem. The two rules are the following:

- Assign a given job j to the factory with the lowest current C_{\max} , not including job j .
- Assign job j to the factory which completes it at the earliest time, i.e., the factory resulting in the lowest C_{\max} after assigning job j .

The rules are applied each time a job is scheduled. From the six tested heuristics the NEH method of Nawaz, Ensore, and Ham (1983) with the second job to factory assignment rule (referred to as NEH2) resulted in the best heuristic performance. Apart from the heuristic methods, Naderi and Ruiz (2010) presented a simple Variable Neighborhood Descent (VND, Mladenović & Hansen (1997)) starting with the NEH2 solution and with two neighborhoods. One being the insertion local search for all factories (until

Download English Version:

<https://daneshyari.com/en/article/476635>

Download Persian Version:

<https://daneshyari.com/article/476635>

[Daneshyari.com](https://daneshyari.com)