Journal of Manufacturing Systems 36 (2015) 224–230

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical Paper

A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times

Marcelo Seido Nagano*, Hugo Hissashi Miyata, Daniella Castro Araújo

University of São Paulo, School of Engineering of São Carlos, Av. Trabalhador São-Carlense 400, 13566-590 São Carlos, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 26 May 2011
Received in revised form 18 June 2014
Accepted 18 June 2014
Available online 17 July 2014

Keywords:
Scheduling
Heuristic
No-wait flowshop
Sequence-dependent setup
Flowtime

ABSTRACT

In this paper, we addressed the problem of scheduling jobs in a no-wait flow shop with sequence-dependent setup times with the objective of minimizing the total flow time. As this problem is well-known for being NP-hard, we present a new constructive heuristic, named QUARTS, in order to obtain good approximate solutions in a short CPU time. QUARTS breaks the problem in quartets in order to minimize the total flow time. The method was tested with other literature methods: BAH and BIH by Bianco et al. (1999) [6], TRIPS, by Brown et al. (2004) [7] and the metaheuristic Iterated Greedy with Local Search proposed by Ruiz and Stützle (2007) [25]. The computational results showed that IG_{LS} obtained the best results and QUARTS presented the best performance regarding other constructive heuristics.

1. Introduction

In the flowshop literature, the majority of the papers assume that the setup time is negligible or part of the job processing time. Treating setup times separately from processing times allows operations to be performed simultaneously and hence improves resource utilization. This is, in particular, important in modern production management systems such as just-in-time (JIT), optimized production technology (OPT), group technology (GT), cellular manufacturing (CM), and time-based competition (see [1–4]). Another important area in scheduling arises in the no-wait flowshop problem (NWFSP), where jobs have to be processed without interruption between consecutive machines. In some processes, for example, the temperature or other characteristics (such as viscosity) of the material require that each operation follow the previous one immediately. There are several industries where the NWFSP applies including the metal, plastic, and chemical industries. This environment is also motivated by concepts such as JIT and zero inventory in modern manufacturing systems (see [5–7]).

The NWFSP has attracted the attention of many researchers. Hall and Sriskandarajah [5] give in their survey paper a detailed presentation of the applications and research on this problem. Fink and Voβ [8] proposed three constructive heuristics and several meta-heuristics considering the total flowtime as the criteria. Aldowaisan and Allahverdi [9] proposed six heuristics based on simulated annealing and genetic algorithms techniques for the F_m/no-wait/C_{max} problem. Aldowaisan and Allahverdi [10] proposed four constructive heuristics for the problem with total completion time as the criterion. Grabowski and Pempera [11] developed and compared different local search algorithms for the NWFSP with makespan criterion. Li et al. [12] considered the NWFSP with makespan minimization and proposed a composite heuristic for large-scale problems. Framinan and Nagano [13] proposed a new heuristic based on an analogy between the given problem and the well-known traveling salesman problem (TSP). Laha and Chakraborty [14] proposed a heuristic based on the principle of job insertion for minimizing makespan. Recently, Framinan et al. [15] proposed a constructive heuristic based on an analogy with the two-machine problem in order to select the candidate to be *appended* in the partial scheduling, to minimize the total completion time.

Considering both the characteristics of no-wait and separated setup times, Aldowaisan and Allahverdi [16] proposed a constructive heuristic for the two-machine problem with the objective of minimizing total flowtime. They considered sequence-independent setup times. Bianco et al. [6] was the first to study the NWFSP with sequence dependent setup times with the objective of minimizing makespan. They showed how to reduce this problem to the asymmetric traveling salesman problem (ATSP)

* Corresponding author at: Universidade de São Paulo, Escola de Engenharia de São Carlos, Departamento de Engenharia de Produção, Av. Trabalhador Sãocarlense 400, CEP: 13566-590 São Carlos, São Paulo, Brazil. Tel.: +55 16 3373 9428; fax: +55 16 3373 9425.
E-mail addresses: drnagano@usp.br, drnagano@sc.usp.br (M.S. Nagano).

http://dx.doi.org/10.1016/j.jmsy.2014.06.007
0278-6125/

and presented two lower bounds and two heuristics, named BAH and BIH. The computational results showed that BIH outperformed BAH in the quality solution. Allahverdi and Aldowaisan [17] found optimal solutions for the F_3/ST_{si}, no-wait$/\sum C_j$ problem, where the setup and processing times satisfy certain conditions, and presented five heuristics for the general problem. Later, Allahverdi and Aldowaisan [18] considered the F_2/ST_{sd}, no-wait$/\sum C_j$ problem and presented five heuristics that used a repeated insertion technique. Stafford and Tseng [19] proposed two mixed-integer linear programming (MILP) models to solve the m-machine NWFSP with sequence dependent setup times in order to minimize the makespan. Shyu et al. [20] presented an ant colony optimization algorithm for the F_2/ST_{si}, no-wait$/\sum C_j$ problem and showed that their algorithm outperformed earlier heuristics. Brown et al. [7] presented a non-polynomial time solution method and a heuristic named TRIPS for the NWFSP with sequence independent setup times, considering for the performance measures both the total flowtime and makespan. França et al. [21] considered the same problem as Bianco et al. [6] and solved it by an evolutionary approach. Their genetic algorithm outperformed BIH. Ruiz and Allahverdi [22] presented a domination relation for the F_4/ST_{si}, no-wait$/\sum C_j$ problem and proposed an iterated local search method and five heuristics for the same problem with m-machines. The results showed that three of their heuristics outperformed TRIPS and the ant colony algorithm of Shyu et al. [20]. Ruiz and Allahverdi [3] proposed seven heuristics and four genetic algorithms for the NWFSP with sequence independent setup times in order to minimize the maximum lateness. Their genetic algorithms outperformed the heuristics of Ruiz and Allahverdi [22].

Nagano et al. [28,29] presented the hybrid metaheuristic evolutionary cluster search (ECS) for the F_m/ST_{si}, no-wait$/\sum C_j$ and F_m/ST_{sd}, no-wait$/C_{\max}$ problems. Nagano and Araújo [27] addressed the problem of scheduling jobs in a no-wait flow-shop with sequence-dependent setup times with the objective of minimizing the makespan and the total flowtime. They presented two new constructive heuristics to obtain good approximate solutions for the problem in a short CPU time, named GAPH and QUARTS. Samarghandi and ElMekkawy [30] developed a mathematical model of the problem and the problem was reduced to a permutation problem. A straightforward algorithm for calculating the makespan of the permutation of jobs was developed. A particle swarm optimization (PSO) was applied on the encoded sequences for exploration of the solution space. Computational results on the available test problems revealed the efficiency of the PSO in finding good-quality solutions.

In this paper, we consider the problem of scheduling a no-wait flowshop with sequence dependent setup times (F_m/ST_{sd}, no-wait$/\sum C_j$). The NWFS problem consists of a set $J=\{j_1, j_2, j_3, \ldots, j_n\}$ of n jobs to be processed on a set $M=\{m_1, m_2, m_3, \ldots, m_m\}$ of m dedicated machines, each one being able to process only one job at a time. Job j_i consists of m operations $op_{li}, \ldots, op_{ki}, op_{k+li}, \ldots, op_{mi}$, to be executed in this order, where operation op_{ki} must be executed on machine k, with p_{ki} processing time, immediately before operation op_{k+1i}. There is a sequence dependent setup time s_{ij}^k between operations op_{ki} and op_{kj} in machine k. This paper is organized as follows. In Sections 2 and 3, we describe the set of heuristics available for the problem. In Section 4, we test the new heuristic effectiveness. Finally, conclusions and final considerations are given in Section 5.

2. Existing constructive heuristics for the problem

In this section, we review the main contributions to the problem regarding constructive methods. More specifically, we explain in detail the constructive heuristics BAH and BIH, from Bianco et al. [6], and TRIPS, from Brown et al. [7]. BAH and BIH heuristics were developed for no-wait flowshop with sequence-dependent setups for the makespan criteria, and TRIPS was developed for the no-wait flowshop with sequence-independent setups for both makespan and total flowtime criteria. In this paper, they will be adapted for the no-wait flowshop with sequence-dependent setups for the total flowtime criteria. The metaheuristic Iterated Greedy, proposed by Ruiz and Stützle [25] for permutation flow shop problems also will be adapted for the considered problem.

2.1. BAH

BAH algorithm finds a feasible sequence in n iterations. At each iteration, given a partial sequence of the scheduled jobs computed in the previous iteration, the algorithm examines a set of candidates of the unscheduled jobs, and appends a candidate job to a partial sequence minimizing the time when the shop is ready to process an unscheduled job.

The pseudo-code of the heuristic is as follows:

Given a set $J=\{j_1, j_2, j_3, \ldots, j_n\}$ of n jobs, let σ be the set of programmed jobs and U be the set of non-programmed jobs.

Step 1: $U \leftarrow J$; $\sigma \leftarrow \varnothing$;
Step 2: While $U \neq \varnothing$, do:
  Step 2.1: Choose the job $j_i \in U$ to be added at the end of the sequence σ, such that the flowtime is minimum;
  Step 2.2: Add job j_i to the end of the sequence σ;
  Step 2.3: $U \leftarrow U - j_i$.

2.2. BIH

The BIH algorithm also finds a sequence of n jobs on n iterations. However, in this algorithm, at each iteration it considers a sequence of a subset of jobs, and finds the best sequence obtained inserting an unscheduled job in any position of the given sequence.

A more detailed description of the heuristic is as follows:

Given a set $J=\{j_1, j_2, j_3, \ldots, j_n\}$ of n jobs, let σ be the set of programmed jobs, U be the set of non-programmed jobs and h the relative insertion position.

Step 1: $U \leftarrow J$; $\sigma \leftarrow \varnothing$;
Step 2: While $U \neq \varnothing$, do:
  Step 2.1: Choose the job $j_i \in U$ which can be inserted in the sequence σ, such that the flowtime is minimum. Let h be the relative insertion position;
  Step 2.2: Insert job j_i at position h in the sequence σ;
  Step 2.3: $U \leftarrow U - j_i$.

2.3. TRIPS

TRIPS heuristic was developed for the no-wait flowshop with sequence-independent setup times, for minimizing total flowtime ($F_m/ST_{si}/\sum C_j$) or makespan ($F_m/ST_{si}/C_{\max}$). In this paper, because there are only BIH and BAH constructive heuristics for the F_m/ST_{sd} problem, we will adapt it to this problem.

TRIPS examines all possible three-job combinations from the set of unscheduled jobs U and chooses the sequence $\{j_w j_x j_y\}$ that minimizes the three-job objective. Then, assigns job j_w to the last empty position in the sequence σ and removes j_w from U. The heuristic repeats the process, assigning one more job to σ for each set of triplets examined until only three jobs are left. Then, it selects the optimal sequence for these jobs and places them in the final positions of heuristic sequence σ.

In this paper, we propose an additional construction phase after the constructive TRIPS solution. We add the insertion mechanism of Nawaz et al. [26] (NEH) to improve the solution generated by TRIPS. Then, the pseudo code of the algorithm can be described as follows: