Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-12 (2016) 408–413

The migrating birds optimization metaheuristic for the permutation flow shop with sequence dependent setup times

Imène Benkalai * **Djamal Rebaine** * **Caroline Gagné** * **Pierre Baptiste** **

* *Université du Québec à Chicoutimi, Saguenay (QC), Canada G7H-2B1(e-mails: imene.benkalai1@uqac.ca ; djamal.rebaine@uqac.ca ; caroline.gagne@uqac.ca).*

** *École Polytechnique de Montréal, Montréal (QC), Canada H3T-1J4 (e-mail: pbaptiste@polymtl.ca)*

Abstract: This paper addresses the problem of scheduling a set of independent jobs with setup times on a set of machines in a permutation flow shop environment. A metaheuristic known as the Migrating Birds Optimization (MBO for short) is designed for the minimization of the overall completion time. A computational study is conducted to analyze the efficiency of this approach on instances that can be found in *Sistemas de Optimizacion Aplicada* (`http://soa.iti.es/problem-instances`).

Keywords: Scheduling, flow shop, setup times, metaheuristic, migrating birds optimization.

1. INTRODUCTION

We address in this paper the permutation flow shop problem with sequence dependent setup times. This problem can be briefly described as follows. We are given a set of n independent jobs to be processed on a set of m machines. Each job comprises m operations that have to be processed without preemption, in this order, on machine M_1, machine M_2, and so on until it completes on machine M_m. The operating sequence of the jobs is the same on every machine, *i.e.* if one job is at the j-th position on machine M_1, then this job will be at the j-th position on all the machines. Furthermore, if operations j and k are processed in that order on machine M_i, then a setup time s_{ijk} must elapse at the completion of operation j and before operation k takes place on machine M_i. This time is needed to adjust machines before they execute the next job in the processing sequence. We seek a valid schedule of the jobs in order to minimize the overall completion time, known as the makespan. Using the classical scheduling notation (see e.g. Pinedo [2002]), this problem is represented as $F_m|prmu, s_{jk}|C_{max}$.

In cases involving sequence dependent setup times, the algorithms for minimizing the makespan tend to be complicated ; indeed, the consideration of setup times adds to the complexity of scheduling problems, as it is the case in the one-machine environment for example [Pinedo [2002]].

Flow shop scheduling problems with setup times exist naturally in many real world situations. Various applications are given in the literature where setup times may be needed for instance to change temperature or a manufacturing mold. We may cite the paper cutting where the machines need to be adjusted when changing from one cutting batch to another [Ruiz and Stutzle [2008]]. Another example is the problem that arises when manufacturing different types of tires requiring to change the manufacturing mold of the multi-purpose machines. A third and last example is the bakery problem where the baked products need different temperatures to cook, so adjustments would be required between two products.

The permutation flow shop scheduling problem with setup times is known to be $\mathcal{NP}$-hard in the strong sense [Gupta [1986]], even for the case of $m = 1$ machine [Ruiz et al. [2005]]. Therefore, the approximation approach is well justified as a solving method. In recent years, intelligence-oriented search algorithms such as Genetic Algorithms, Tabu Search, Simulated Annealing, and so on, that mimic nature phenomena, have been employed to solve scheduling problems. In this paper, we focus on a metaheuristic known as the Migrating Birds Optimization method (MBO for short), recently introduced by Duman *et al.* [Duman et al. [2012]]. This metaheuristic has only been applied to a very few problems. In addition, the promising results of this method on the Quadratic Assignment Problem [Duman et al. [2012]] make it a good candidate to solve the problem under study.

This paper is organized as follows. Section 2 describes the problem under study. Section 3 presents a brief review of the previous works related to this problem. Section 4 discusses the Migrating Birds Optimization method. Section 5 presents the experimental study we conducted on the efficiency of this method. Finally, in Section 6, we present our concluding remarks.

2. PROBLEM DESCRIPTION

The flow shop scheduling problem with setup times can be described as follows. Let $J = \{1, 2, \ldots, n\}$ be a set

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.640

of n jobs to be processed without preemption by $M = \{M_1, M_2, \ldots, M_m\}$ a set of m machines. A processing time p_{ij} is given to denote the time spent by job j on machine M_i, and a setup time s_{ijk} denotes the time that must elapse between the completion of job j and the start of job k whenever the former precedes the latter on machine M_i in a given sequence. We seek a valid schedule of the jobs in order to minimize the makespan. More formally, we seek a permutation $\pi = (\pi(1), \ldots, \pi(n))$ on set J such that

$$C_{max}(\pi^*) = \max_{\pi \in \Pi} C_{max}(\pi),$$

where Π denotes the set of the $n!$ permutations over set J, π and π^* are in Π, and $C_{max}(\pi)$ is the overall completion time associated with permutation π.

The value of the makespan for a given permutation π is computed as follows. Let $C(\pi(j), i)$ be the completion time of job $\pi(j)$ on machine M_i. It then follows $C_{max}(\pi) = C(\pi(n), m)$, where $C(\pi(j), k)$ of job $\pi(j)$ on machine M_k is given by the following recursive formula.

$$C(\pi(1), k) = \sum_{i=1}^{k} p_{i,\pi(1)}; k = 1, \ell, m$$

$$C(\pi(j), 1) = \sum_{q=1}^{j-1} (p_{1,\pi(q)} + s_{1,\pi(q),\pi(q+1)}) + p_{1,\pi(j)}; j = 2, \ell, n$$

$$C(\pi(j), k) = \max\{C_{k-1,\pi(j)}, C_{k,\pi(j-1)} + s_{k,\pi(j-1),\pi(j)}\}$$
$$+p_{k,\pi(j)}; k = 2, \ell, m \ ; \ j = 2, \ell, n.$$

3. RELATED WORKS

Since the problem is $\mathcal{N}P$-hard in the strong sense, the metaheuristic approach has been extensively utilized as a solving approach for the $F_m|prmu, s_{jk}|C_{max}$ problem. In this section, we review the state of the art of the important works that have been done in this area. A general survey is given in Allahverdi *et al.* [Allahverdi et al. [2008]].

Let us first start with the Genetic Algorithm designed by Ruiz *et al.* [Ruiz et al. [2005]]. The authors compare the efficiency of Genetic Algorithms with other approaches that have been shown to produce good results on the standard flow shop (without setup times). They extended Taillard's standard flow shop instances. Their Genetic Algorithm modifies a part of the population if the search happens to "stagnate" [1], and has a selection strategy based on The Roulette Wheel and Tournament Selection [Talbi [2009]]. The offspring are taken to build the next population along with the best parents. The authors also introduce a new crossover operator that they consider more efficient. It allows to preserve sequences of the solutions by copying "blocs" of two jobs that are present in both parents directly in the offspring and then applies the two-point crossover. It is claimed in that paper this algorithm outperforms by far the other algorithms listed in their study.

Kumar and Singhal [Kumar and Singhal [2013]] also proposed a Genetic Algorithm for the resolution of the same problem as above with splittings. This algorithm has the same evolution strategy as the previous one, a tournament selection strategy, and uses a two-point crossover followed by a repair phase [2]. Its mutation operator consists in repositioning a job. The authors also tested the effects of the variation of crossover and mutation probabilities on the quality of the results. Their algorithm often manages to find the optimal solutions for the instances in their testbed more or less rapidly depending on the previously cited probabilities.

Another algorithm, the Iterated Greedy Algorithm, was also applied to the same problem as above by Ruiz and Stützle [Ruiz and Stutzle [2008]]. This algorithm consists of two phases: a phase of destruction that produces a partial solution, and a phase of construction that completes the latter and replaces the current solution according to some acceptance criterion. Their experimental study shows that the efficiency of this algorithm is quite promising.

Let us also cite the Tabu Search algorithm designed by Santos *et al.* [Santos et al. [2014]], in which the total weighted tardiness is now the criterion to minimize. The idea here is to explore a small neighborhood based on job insertion with a dynamic tabu list, being supported by a speedup due to a bookkeeping of solution information. Their algorithm is quite efficient as it improves the best known solutions within short running times.

To solve a flow shop with the so-called family setup times, Lin *et al.* [Lin et al. [2011]] suggest using a multi-start version of the Simulated Annealing approach. The job set can be split into "families", and the setup times occur only between two jobs that belong to different families. Their approach takes advantage of the main properties of the Simulated Annealing (*e.g.* effective convergence, efficient use of memory, and easy implementation) and those of multi-start hill climbing strategies (*e.g.* sufficient diversification, and efficient sampling of the neighborhood solution space). It performs a given number of restarts from different solutions, thus, offering more chances to escape from local optima. The explored neighborhood is based on family jobs swaps and insertions. Their approach was shown to perform extremely well compared to other existing algorithms such as the Mimetic Algorithm of França *et al.* [França et al. [2005]] and the Tabu Search algorithms of Hendizadeh *et al.* [Hendizadeh et al. [2008]].

Finally, we cite the work of Dong *et al.* [Dong et al. [2009]] in which different priority rules are compared such as the NEHT-RB heuristic [Rìos-Mercado and Bard [1998]], and the PB heuristic by Tseng *et al.* [Tseng et al. [2006]]. This study resulted in a classification of which heuristic is best suitable depending on the setup times.

On the other hand, the Migrating Birds Optimization is a rather recent method that has only been applied to a very few problems to date. First designed by Duman *et al.*, its first application was on the Quadratic Assignment Problem [Duman et al. [2012]]. The authors used a 2-interchange neighborhood that consisted in switching two cells in a solution. The promising results it provided made researchers consider it as a solving method for scheduling problems. It was adapted to solve the permutation flow shop scheduling problem by Tongur and Ulker [Tongur and

[1] The algorithm does not manage to improve the current best solution for a certain number of iterations.

[2] To fix the unfeasible solutions.