Computers & Industrial Engineering 64 (2013) 937–958

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

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

A dual encoding-based meta-heuristic algorithm for solving a constrained hybrid flow shop scheduling problem

Antonio Costa [a,*], Fulvio Antonio Cappadonna [b], Sergio Fichera [a]

[a] Dipartimento di Ingegneria Industriale e Meccanica, University of Catania, Viale Andrea Doria 6, 95125 Catania, Italy
[b] Dipartimento di Ingegneria Elettrica Elettronica e Informatica, University of Catania, Viale Andrea Doria 6, 95125 Catania, Italy

ARTICLE INFO

Article history:
Received 16 February 2012
Received in revised form 7 January 2013
Accepted 8 January 2013
Available online 1 February 2013

Keywords:
Hybrid flowshop
Metaheuristics
Genetic algorithm
Encoding
Decoding
Optimization

ABSTRACT

Though scheduling problems have been largely investigated by literature over the last 50 years, this topic still influences the research activity of many experts and practitioners, especially due to a series of studies which recently emphasized the closeness between theory and industrial practice. In this paper the scheduling problem of a hybrid flow shop with *m* stages, inspired to a truly observed micro-electronics manufacturing environment, has been investigated. Overlap between jobs of the same type, waiting time limit of jobs within inter-stage buffers as well as machine unavailability time intervals represent just a part of the constraints which characterize the problem here investigated. A mixed integer linear programming model of the problem in hand has been developed with the aim to validate the performance concerning the proposed optimization technique, based on a two-phase metaheuristics (MEs). In the first phase the proposed ME algorithm evolves similarly to a genetic algorithm equipped with a regular permutation encoding. Subsequently, since the permutation encoding is not able to investigate the overall space of solutions, a random search algorithm equipped with an *m*-stage permutation encoding is launched for improving the algorithm strength in terms of both exploration and exploitation. Extensive numerical studies on a benchmark of problems, along with a properly arranged ANOVA analysis, demonstrate the statistical outperformance of the proposed approach with respect to the traditional optimization approach based on a single encoding. Finally, a comprehensive comparative analysis involving the proposed algorithm and several metaheuristics developed by literature demonstrated the effectiveness of the dual encoding based approach for solving HFS scheduling problems.

1. Introduction

In the last 50 years a huge amount of literature addressed the traditional scheduling problems as flow shop, flow lines, job shop, and parallel machines shops. This field of research has been investigated over the years by means of several optimization techniques entailing both exact approaches and heuristic techniques, also in relation to the degree of complexity of the problem in hand (Blazewicz, Ecker, Pesch, Schmidt, & Weglarz, 1992). However, the scheduling topic still represents a very active branch of research which recently got benefit from the study of several production environments strictly connected to the classical scheduling problems. As for example, scheduling optimization of hybrid or flexible configurations of the regular flow shop problem has been meeting with a great acceptance within the research community, basically due to the affinity between the way the theoretical problem is addressed and the real needs of the industrial practice. On the basis of what stated by Ruiz and Maroto (2006) there is a clear difference among Flexible Flow Line (FFL), Flow Shop with Multi-Processor (FSMP), Hybrid Flow Shop (HFS) and, finally, between Hybrid Flexible Flow Line (HFFL) and Hybrid Flexible Flow Shop (HFFS), even though all these production configurations may be considered as a variant of the traditional Flow Shop (FS) wherein one stage may hold more than one machine, at least. Both in the FFL and in the FSMP the machines available at each stage are identical. In addition, in the FSMP a given task of a job can be performed simultaneously by a set of parallel machines pertaining to a given stage. The HFS is significantly more complex than regular flow shop; each job must be processed by only one machine per stage, the production flow is unidirectional and machines within a given stage are unrelated. HFFS as well as HFFL are identical to the aforementioned HFS and FFL, the only difference being that jobs may skip some stages. An earlier research performed by Gupta (1988) demonstrated as the FFL problem with just two stages is $\mathcal{NP}$-Hard, though one of the two stages holds only one machine.

* Corresponding author. Tel.: +39 0957382457; fax: +39 095337994.
E-mail addresses: costa@diim.unict.it (A. Costa), fcappadonna@dieei.unict.it (F.A. Cappadonna), sfichera@diim.unict.it (S. Fichera).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.01.004

One year later Gourgand et al. (1999) stated as the HFS problem involves a total number of potential solutions equal to $n!(\prod_{i=1}^{m} m_i)^n$. Making full use of the criterion introduced by Linn and Zhang (1999) literary research on HFS can be roughly classified into three categories: (1) two-stage HFS; (2) three-stage HFS; (3) m-stage ($m > 3$) HFS. Since the level of complexity considerably increases in relation to the number of stages, the earliest studies have addressed the two-and three-stages problems. Arthanary and Ramaswamy (1971) developed a pioneer research aiming to cope with a two-stage HFS scheduling problem through a Branch and Bound algorithm. A couple of years later, SPS Name (SPS Year) addressed a more complicated no-wait flowshop with multi-processors and m-stages by means of a dynamic programming algorithm.

More recently, several studies focusing on the m-stage HFS problem have been elaborated. A comprehensive outline of the more relevant studies on the m-stage HFS has been provided by Linn and Zhang (1999) and Riane and Ariba (1999). Although optimal solutions of multi-stage HFS can be obtained via exact methods (Rajendran & Chaundhuri (1992); Vignier, Billaut, Proust, & T'Kindt (1996); Vignier, Commandeur, & Proust (1997)) when the problem size is small, the complexity connected to issues involving three or more stages justify the employment of heuristic approaches (Ying & Lin, 2009; Jungwattanakit, Reodecha, Chaovalitwongse, & Werner, 2008). Actually, heuristic methods able to address a HFS problem should be distinguished between constructive heuristics and meta-heuristics. The former, traditionally characterized by the drawback of the non-robustness, are able to yield a fast response but the provided solution may result drastically far from the global optimum. Brah and Loo (1999) selected five better performing flow shop heuristics and evaluated their performances for a flow shop with multiple processors problem both in terms of makespan and flow time. They found that two well-known heuristics were comparable in performance in a flow shop with multi processors and that one of those was more consistent and robust than the other one for both optimization criteria. Ruiz, Serifoglu, and Urilings (2008) proposed a mixed integer modelling and a set of constructive heuristics for the HFFS scheduling problem. Instances up to 15 jobs exhaustively have been solved by the MIP model while larger instances have been investigated by an adaptation of the NEH (Nawaz, Enscore, & Ham, 1983) algorithm as well as by means of a set of well-known despatching rules. Ying and Lin (2009) proposed an effective and efficient heuristic named Heuristic for the Multistage Hybrid Flowshop (HMHF) whose performance was properly compared with that of 10 heuristics and a tabu search based meta-heuristic from the relevant literature. On the other hand, meta-heuristic (ME) algorithms can be used for solving various NP-hard optimization problems, according to a proper adaptation to the structure of the problem to be tackled. MEs can reach near-optimal solutions of a large sized problem in a relatively narrow computational time than exact methods, but their implementation could result really sophisticated, also requiring arduous both coding and decoding tasks in relation to the kind of problem to be optimized. Tavakkoli-Moghddam, Sfaei, and Sassoni (2009) introduced an efficient Memetic Algorithm (MA), i.e. a meta-heuristic algorithm which mimics the cultural evolution, combined with a novel local search engine, named Nested Variable Neighbourhood Search (NVNS), for solving the flexible flow line scheduling problem with processor blocking and without intermediate buffers. It is worth pointing out that they adopted a structure of the chromosome for the problem representation composed by a matrix along with a vector, the former being used for job assignment to machines and the latter being used for the job permutation. The FSMP has been widely dealt with by the recent literature and several authors decided to use ME algorithms. Oguz, Zinder, Van Ha, Janiak, and Lichtenstein (2004) demonstrated as the introduction of precedence constraints in the FSMP problem makes even the simplest version of this problem NP-hard and, in addition, elaborated an approximation algorithm based on the idea of tabu search in order to address that kind of issue. Genetic Algorithms efficacy and efficiency for solving FSMP problems has been documented by Sivrikaya Serifoglu and Ulusoy (2004). Allaoui and Artiba (2004) deal with the hybrid flow shop scheduling problem under maintenance constraints to optimize several objectives based on flow time and due date. Setup, cleaning and transportation times have been take into account and a proper Simulated Annealing was implemented for optimizing the aforementioned problem. Ying and Lin (2006) developed a novel ant colony system for solving the hybrid flow shop with multiprocessor problem. To verify the proposed optimization technique, a thorough comparison with a genetic algorithm and a tabu search from the relevant literature has been carried out on the basis of two well-known benchmark problem sets. Tseng and Liao (2008) developed a Particle Swarm Optimization (PSO) algorithm, a novel meta-heuristic inspired by the flocking behavior of the birds, powered by a new encoding scheme, namely a new way to represent in terms of string the studied problem, i.e. a flow shop with multiprocessor tasks. To assess the effectiveness of the proposed encoding embedded within the PSO, several computational experiments have been carried out, also to make a comparison with a genetic algorithm and an Ant Colony System proposed by the relevant literature. The leading role of the problem encoding has been emphasized in the research work of Gholami, Zandieh, and Alem-Tabriz (2009) where a flow shop problem with multiprocessor, also including sequence-dependent setup times and machine breakdowns has been optimized. A parallel greedy algorithm approach to the hybrid flow shop with multiprocessor task scheduling problem recently has been carried out by Kaharaman Kaharaman, Engin, Kaya, and Elif Öztürk (2010). They considered a set of 240 numerical examples divided into 24 groups and compared the performance of their technique with a tabu search and a genetic algorithm based meta-heuristics, both arisen from the literature. A matching between simulation and optimization has been utilized by Rathinasamy and R (2010) for addressing the scheduling problem of an automotive vibration dampers manufacturing system arranged as a hybrid flow shop with multiprocessor. Recently a Particle Swarm Optimization PSO algorithm has been implemented for addressing a flexible flow shop with multi processors tasks (Singh & Mahapatra, 2012). Mutation, a commonly used operator in genetic algorithm, has been introduced in PSO so that trapping of solutions at local minima or premature convergence can be avoided.

With exception of the earliest researches, several papers recently dealt with the HFS scheduling problem through meta-heuristics and a lot of them highlighted the key role of the problem encoding for enhancing both the efficiency and the efficacy of such optimization algorithms. Ruiz and Maroto (2006) studied the makespan minimization of a HFS with sequence dependant setup times and machine eligibility scheduling problem through a genetic algorithm. They adopted a simple permutation problem encoding able to reach a good compromise between quality of solutions and computational time efficiency. A so-called rational problem encoding, basically a real number-based encoding different form the regular permutation one, has been embedded within an Immune Algorithm (IA) for solving a HFS scheduling problem with sequence dependent setup times (Zandieh, Fatemi Ghomi, & Moattar Husseini, 2006). The obtained results have been compared with a random key genetic algorithm. The same kind of approach inspired by theoretical immunology has been adopted by Engin and Döyen (2004) who showed as the Immune