Applied Soft Computing 19 (2014) 93–101

Contents lists available at ScienceDirect

Applied Soft Computing

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

Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan

M.K. Marichelvam [a,*], T. Prabaharan [b], X.S. Yang [c]

[a] Department of Mechanical Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, Tamilnadu 626001, India
[b] Department of Mechanical Engineering, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu 626005, India
[c] School of Science and Technology, Middlesex University, London NW4 4BT, UK

ARTICLE INFO

Article history:
Received 30 May 2012
Received in revised form 27 January 2014
Accepted 3 February 2014
Available online 13 February 2014

Keywords:
Hybrid flow shop (HFS)
Scheduling
NP-hard
Improved cuckoo search (ICS)
Metaheuristics

ABSTRACT

The multistage hybrid flow shop (HFS) scheduling problems are considered in this paper. Hybrid flowshop scheduling problems were proved to be NP-hard. A recently developed cuckoo search (CS) metaheuristic algorithm is presented in this paper to minimize the makespan for the HFS scheduling problems. A constructive heuristic called NEH heuristic is incorporated with the initial solutions to obtain the optimal or near optimal solutions rapidly in the improved cuckoo search (ICS) algorithm. The proposed algorithm is validated with the data from a leading furniture manufacturing company. Computational results show that the ICS algorithm outperforms many other metaheuristics.

1. Introduction

Scheduling is a process of allocating resources over time to perform a collection of required tasks [1]. It is a decision making process. Effective scheduling plays a vital role for the growth of any kind of industries. Different types of scheduling problems were addressed in the literature with different performance measures. Single machine scheduling, flow shop scheduling, parallel machine scheduling, job shop scheduling, HFS scheduling and project scheduling are some of the important types of scheduling environments. Among them HFS scheduling problems are considered in this paper. Minimization of makespan, total flow time, mean flow time, total earliness, total tardiness, weighted earliness and tardiness and number of tardy jobs are some of the important objective functions of scheduling problems. The objective of this paper is to minimize makespan. Many researchers concentrated on the HFS scheduling problems for the past several decades since it was proposed by Arthanari and Ramamurthy [2] due to their practical importance. Many industries like steel, ceramic, electronics, chemical and textile industries resemble the HFS environment. The HFS environment can be assumed to be the combination of flow shop and parallel machines. In the HFS environment, machines are arranged into several stages in series. Each stage has one or more identical machines in parallel. A job must pass through all stages. Moreover, the job must be processed by exactly one machine at every stage. The layout of a k stage hybrid flowshop scheduling is shown in Fig. 1 where M_{mk} represents the machine in the mth parallel and kth stage [3].

Researchers have proposed many different approaches to solve the HFS problems. Exact solution methods, heuristics and metaheuristics are the three important approaches addressed in the literature [4]. One of most important, exact solution methods used for the HFS problem is the branch & bound algorithm [5–8]. Portmann and Vignier [9] improved the performance of the branch & bound algorithm by hybridizing the genetic algorithm (GA). Neron et al. [10] solved the HFS scheduling problems using branch and bound algorithm. They applied energetic reasoning and global operations for enhancing the efficiency of branch and bound algorithm to minimize the makespan. The HFS scheduling problems were proved to be NP-hard by Gupta [11] and Hoogeveen et al. [12]. Hence, many heuristics and metaheuristics have been developed to search for optimal or near-optimal solutions for solving the HFS scheduling problems. Gupta and Tunc [13] proposed four heuristics to minimize makespan for two stage HFS scheduling problems with separable setup and removal times. Lee and Vairaktarakis [14] addressed heuristics to minimize makespan for solving the HFS problems. Hunsucker and Shah [15] evaluated the performance of different dispatching rules for a constrained HFS scheduling environment. Brah and Loo [16] developed a heuristic procedure to solve flow shop scheduling problems with multiple processors.

* Corresponding author. Tel.: +91 9751043410; fax: +91 04549 378171.
E-mail address: mkmarichelvamme@gmail.com (M.K. Marichelvam).

http://dx.doi.org/10.1016/j.asoc.2014.02.005
1568-4946/

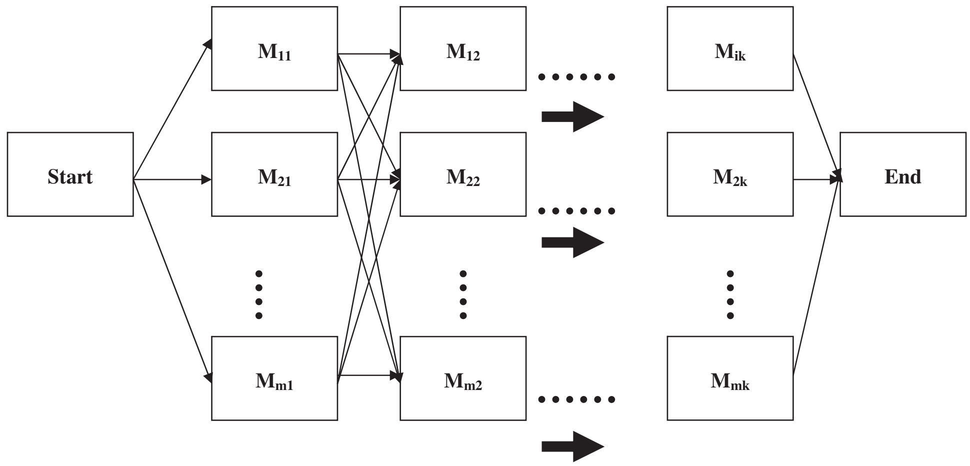

Fig. 1. Layout of the hybrid flow shop environment.

Negenman [17] presented a local search method to solve HFS problems. Soewnadi and Elmaghraby [18] suggested several heuristics for the three-stage HFS with the multiple processors to minimize makespan. Oğuz et al. [19] also proposed some heuristics to solve multiprocessor task scheduling in a two-stage flow shop scheduling problems. He and Babayan [20] solved three stage flexible flow shop scheduling problems using an agent based approach. Kyparisis and Koulamas [21] developed some heuristics to minimize the makespan for the multistage flexible flow shop scheduling problem with uniform parallel machines in each stage.

Recently, researchers have applied many metaheuristics to solve the HFS problems. Engin and Döyen [22] proposed an artificial immune system algorithm for solving the HFS problems to minimize the makespan. Zandieh et al. [23] also presented an immune algorithm approach for solving the HFS problems. They considered the sequence-dependent setup times. Ying [24] proposed a new hybrid immune algorithm and iterated greedy algorithm to minimize the makespan in the multistage HFS scheduling problems. Yang et al. [25] solved the flow shop scheduling with multiple processor problems by using a tabu search simulation optimization approach. They have considered a case study problem of a multilayer ceramic capacitor manufacturing company to minimize tardiness. Bożejko et al. [26] addressed a parallel tabu search algorithm to solve the HFS scheduling problems. Many researchers have applied different versions of genetic algorithm (GA) to solve the HFS problems [27–32]. Ying and Lin [33] developed an ant colony system heuristic for the multistage HFS problem with multiprocessor tasks to minimize the makespan. They tested the algorithm over the benchmark problems from the literature. Alaykýran et al. [34] also suggested an ant colony optimization algorithm for solving the HFS problems. Wang et al. [35] proposed a simulated annealing (SA) algorithm for the HFS scheduling problems to minimize the makespan. Tseng and Liao [36] presented particle swarm optimization (PSO) algorithm to solve the HFS problems. Recently, Liao et al. [4] developed a PSO hybridized with bottleneck heuristic to solve the HFS problems to minimize makespan. The different HFS environments, constraints and solution methodologies can be found in [37] and [38].

In this paper, a new metaheuristic algorithm called cuckoo search (CS) algorithm is used to solve the HFS problems. Cuckoo search algorithm is one of the population based metaheuristic algorithms developed by Yang and Deb in 2009 [39]. Yang and Deb [40] solved various optimization problems using the cuckoo search algorithm. Yang and Deb [41] also proposed a multiobjective cuckoo search algorithm for design optimization problems. Gandomi et al. [42] solved the structural optimization problems using cuckoo search algorithm. Walton et al. [43] proposed a modified cuckoo search algorithm to solve the optimization problems. Layeb [44] addressed a novel quantum inspired cuckoo search for knapsack problems. Durgun and Yildiz [45] used the cuckoo search algorithm to solve design optimization problems for vehicle components. Yildiz [46] determined the optimal machining parameters for milling operations using the cuckoo search algorithm.

From the current literature, one can easily understand that the cuckoo search algorithm is applied to solve different types of optimization problems. However, the application of cuckoo search algorithm to solve the scheduling problems is very limited. Marichelvam [47] proposed an improved hybrid cuckoo search algorithm for solving the permutation flow shop scheduling problems. Chandrasekaran and Simon [48] proposed a hybrid cuckoo search algorithm integrated with fuzzy system for solving multi-objective scheduling problems. Burnwal and Deb [49] solved the flexible manufacturing scheduling problem using the cuckoo search algorithm. They compared the results with GA and PSO. Hence in this paper, we apply the cuckoo search algorithm to solve the HFS problems. Similar to many metaheuristic algorithms, the initial solutions in the cuckoo search algorithm are also generated randomly. But in this paper a constructive heuristic called NEH heuristic developed by Nawaz et al. [50] is applied to generate one of the initial solutions to improve the solution quality. In addition, the parameters of the cuckoo search algorithm are tuned in this paper to increase the efficiency of the algorithm. Hence, in this paper, an improved cuckoo search (ICS) algorithm is presented to minimize makespan for the HFS scheduling problems.

The paper is organized as follows: the problem definition is presented in Section 2. The CS and ICS algorithms are explained in Section 3. Computational results are presented in Section 4. Conclusions and future research opportunities will be discussed in Section 5.

2. Problem definition

The HFS scheduling problems considered in this work consist of k stages in series. Each stage i ($i = 1,2,\ldots,k$) consists of m identical machines in parallel. A set of N jobs j ($j = 1,2,\ldots,N$) has to be processed on anyone of the machines available at each stage. The processing times of job j at various stages are $P_{1j}, P_{2j}, \ldots, P_{kj}$, and it may be zero for some jobs as the jobs are not processed in some stages. Processing time is the time required to process a job in a stage. The objective is to minimize the makespan (C_{max}). The makespan for a scheduling problem is defined as the completion