Applied Soft Computing 26 (2015) 166–179

Contents lists available at ScienceDirect

Applied Soft Computing

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

A nature inspired intelligent water drops evolutionary algorithm for parallel processor scheduling with rejection

Hadi Mokhtari

Department of Industrial Engineering, Faculty of Engineering, University of Kashan, Kashan, Iran

ARTICLE INFO

Article history:
Received 20 July 2013
Received in revised form 10 July 2014
Accepted 19 September 2014
Available online 16 October 2014

Keywords:
Intelligent water drops
Parallel local search
Neighborhood structures
Order scheduling
Rejection

ABSTRACT

Scheduling has become a popular area for artificial intelligence and expert system researchers during last decade. In this paper, a new metaheuristic algorithm entitled intelligent water drops (IWD) is adapted for solving a generalized kind of order scheduling problem where rejection of received orders is allowed with a penalty cost. At the beginning of production period, a set of orders are received by manufacturer. Due to capacity limit, the manufacturer can only process a subset of orders and has to decide to reject some of undesirable orders. The accepted orders are proceed to be scheduled by a set of identical parallel processors in shop floor. The objective is to select the best set of orders with high contribution in manufacturer's benefit and then find the appropriate schedule of accepted orders minimizing the number of tardy orders. To effectively solve the suggested problem, the Lexicographic utility function is customized to address different objectives and then an IWD algorithm, which is based on the process of the natural rivers and the interactions among water drops in a river, is devised. To further enhance the performance of basic IWD, an Iterated Local Search (ILS) heuristic is also incorporated into the main algorithm. To demonstrate the applicability of suggested problem and also show the effectiveness of enhanced IWD with ILS, a real-world application in commercial printing industry is presented and the performance of algorithm is compared with traditional algorithms like GA, DE and ACO.

1. Introduction

Production scheduling is one of the most important fields in operations research and combinatorial optimization. Among scheduling problems addressed in literature, machine scheduling with parallel processors is one of the NP-hard optimization problems. The problem is generally defined as follows. In a parallel processor scheduling problem, there is a set of n orders $\{O_1, O_2, \ldots, O_n\}$; each of which can be processed on m processors $\{P_1, P_2, \ldots, P_n\}$. Each order should be processed without preemption on a given processor. Each processor can process only one order at a given time. In addition, all orders are available at time zero, they arrive to the shop floor at same time. One of the basic types of this problem is that all the processors are identical in terms of processing function and speed. Most studies on parallel processors scheduling assumed that the processors are identical. Guinet [1] studied the identical parallel processors problem with makespan minimization where setup times are sequence-dependent. In addition, a tabu search based metaheuristic was suggested by Franca et al. [2] so as to minimize the total completion time for a scheduling problem with parallel machines. In another study, Gendreau et al. [3] developed some lower bounds for a multiprocessor scheduling problem with sequence dependent setup times. Hurink and Knust [4] presented a heuristic for the same problem with the makespan objective. Eom et al. [5] discussed a parallel processor scheduling problem with the sequence-dependent family setup. Besides, a special case of problem with the makespan objective and a minimum amount of flow time was presented by Lin and Liao [6]. In another work, Dunstall and Wirth [7] investigated an identical parallel machine scheduling problem with family jobs and sequence-independent set-up times. Moreover, Tahar et al. [8] considered a scheduling problem of a set of jobs with sequence dependent setups and splitting condition. A hybrid algorithm integrating features from tabu search, simulated annealing and variable neighborhood search was presented by Anghinolfi and Paolucci [9] to cope with a parallel processor job scheduling problem. Rocha et al. [10] studied the performance of a VNS algorithm for parallel processor scheduling problem with sequence dependent setup times. Shim and Kim [11] discussed a scheduling problem with parallel processors and devised some dominance rules and lower bounds to optimize the tardiness of jobs. A branch and bound algorithm was also developed by using bounds and rules suggested. Biskup et al. [12]

E-mail address: mokhtari_ie@kashanu.ac.ir

http://dx.doi.org/10.1016/j.asoc.2014.09.040
1568-4946/

studied a scheduling model minimizing the total tardiness and then designed a general heuristic algorithm for achieving optimal or near-optimal schedules. Moreover, a branch-and-bound algorithm was devised by Tanaka and Araki [13] for a scheduling problem with multiple processors to minimize total tardiness. Pfund et al. [14] suggested an identical parallel machine problem with ready times and sequence dependent setup times, and then developed a heuristic method minimizing the total weighted tardiness. Yu and Gu [15] suggested a mixed integer model for parallel processor problem with makespan measure. Nessah et al. [16] developed some dominance rules and two performance lower bounds in order to minimize the total weighted completion time on identical parallel machines with release dates. In another research, Driessel and Mönch [17] studied a scheduling problem on identical parallel processors with ready times, precedence constraints, and sequence-dependent setup times. In a work presented by Hu et al. [18], a heuristic algorithm which is based on a combination of the largest amount of processing rule and the enhanced smallest machine load rule was presented for solving a parallel machine scheduling problem with precedence constraints and machine eligibility. In subsequent section, related literature of scheduling problems with the case of rejection will be reviewed.

2. Related works on rejection

In most traditional scheduling problems, all orders should be processed on the processors when they are received by the manufacturer and therefore rejection is not allowed for the orders. However in real world conditions, the manufacturer sometimes rejects some of received orders, for example, in cases where orders need special raw materials, skilled workers and also bring relatively small profits. In such cases, to reduce costs and obtain maximum profits, the manufacturers evaluate the overall desirability of received orders and then reject some of undesirable ones. The machine scheduling problem with rejection was first introduced by Bartal et al. [19] where the performance measure was to minimize the makespan and the penalty cost related to the rejected jobs. Afterwards, the production scheduling problem with the case of rejection has been treated with increasing research attention. For example, Seiden [20] considered the similar model where multi processors are available, and the preemption of orders was allowed. Hoogeveen et al. [21] discussed a job scheduling problem with identical, uniformly related, or unrelated parallel processors where rejection and preemption are allowed. In another study, Dosa and He [22] considered a scheduling problem with machine cost and rejection. To minimize the sum of the maximum completion time, purchasing cost of machines, and the total penalty of rejected jobs, an optimal online algorithm was also presented. Moreover, Cheng and Sun [23] studied a scheduling problem with single-machine, deterioration jobs and rejection. The received jobs can be rejected by paying penalties, and the objective is to minimize the makespan, total weighted completion time, the maximum lateness/tardiness, and the total penalty of the rejected jobs, simultaneously. In another work, Zhang et al. [24,25] discussed a single machine scheduling problem with release dates and rejection. The aim was to minimize the sum of the makespan of the accepted orders and the total rejection penalty of the rejected ones. Furthermore, Engels et al. [26] proposed a single machine scheduling with rejection case where the aim was to minimize the sum of the weighted completion times of the accepted jobs and the total penalty cost of the rejected jobs. Epstein et al. [27] studied an on-line scheduling problem with the unit time jobs and rejection on single a processor minimizing the total completion time. Lu et al. [28,29] discussed an unbounded parallel machine scheduling with release dates, batch processing machines and rejection. The job processing time is defined as the longest processing time of the jobs in batch. The problem was proved to be a binary NP-hard, and then a pseudo-polynomial-time algorithm was given. In another work presented by Cao et al. [30], some scheduling problems with the makespan objective, rejection case and with discretely compressible processing times were studied. Moreover, Cao and Zhang [31] analyzed the scheduling problem with rejection and non-identical job arrivals to optimize the sum of penalties of the rejected jobs and the maximum completion time of the accepted ones. The NP-hardness of off-line version of problem was proved and then a polynomial time approximation scheme was presented. A polynomial-time approximation scheme (PTAS) is an approximation algorithm used for solving optimization problems with a bounded approximation. It takes an instance of problem and, in polynomial time, produces a solution that is within an interval of being optimal. Cao and Yang [32] studies a parallel scheduling problem with batch processing machines and rejection case. The objective was to minimize the maximum completion time plus the penalty cost incurred by the rejected jobs. A polynomial time approximation scheme was described for the on-line version of presented scheduling model which is based on dynamic programming and rounding. Besides, Lu et al. [33] evaluated a bounded single machine scheduling problem with parallel-batch, release dates and job rejection. The accepted jobs are then processed on the machine while a certain penalty has to be paid for the rejected jobs. Sengupta [34] studied the scheduling problem with rejection so as to minimize the sum of the maximum lateness/tardiness of the accepted jobs and the total penalty cost of the rejected jobs. Furthermore, Khuller and Mestre [35] discussed the single machine scheduling problem with due dates and rejection so that minimize maximum lateness of jobs. Table 1 shows a brief view of above literature with the case of rejection.

As results presented in Table 1 shows, there is no paper in literature addressing the rejection with the number of tardy orders as scheduling measure. Hence in this paper, a bi-objective scheduling model is suggested in which the objective is to simultaneously minimize the penalty cost incurred by the rejected orders and number of accepted orders be late. Two different decisions should be made in the proposed model: (i) which orders should be accepted for processing? and (ii) what is the best schedule of accepted orders on processors? In order to achieve the best solution to the above questions, a new nature-inspired evolutionary method called as intelligent water drops (IWD) is customized for the problem in current paper. As it can be seen in Table 1, metaheuristic techniques have not been employed to address the rejection case, yet. Recently, artificial intelligence and evolutionary algorithms have been successfully used to address scheduling problems. IWD is also an intelligent metaheuristic algorithm which is based on the process of the natural rivers and the interactions among water drops in the river. We can find very few papers addressing the scheduling problems with IWD algorithm. For example, Niu et al. [36] devised an enhanced IWD for addressing a general scheduling problem in job shop environment. In order to achieve better solutions, five schemes are embedded into classic IWD to increase the diversity of the solution space. As another study, Ong et al. (2013) customized IWD algorithm to cope with single and multiple objective job shop scheduling problems. In this paper, we are going to adapt the IWD for solving a scheduling problem on parallel processors with rejection, number of tardy orders, and penalty cost.

The reminder of this paper is organized as follows. Section 3 describes a multi-objective approach based on Lexicographic utility function to address different objectives in the model. In Section 4, the problem characteristics and mathematical model are stated and the problem is illustrated via a numerical example. Section 5 introduces the IWD algorithm in general, and then the proposed IWD algorithm is elaborated in sequel. In Section 6, the ILS heuristic