Expert Systems with Applications 42 (2015) 7132–7147

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A Monte Carlo simulation based chaotic differential evolution algorithm for scheduling a stochastic parallel processor system

Hadi Mokhtari [a,*], Ali Salmasnia [b]

[a] Department of Industrial Engineering, Faculty of Engineering, University of Kashan, Kashan, Iran
[b] Department of Industrial Engineering, Faculty of Engineering and Technology, University of Qom, Qom, Iran

ARTICLE INFO

Article history:
Available online 18 May 2015

Keywords:
Chaos theory
Chaotic maps
Simulation-based optimization
Differential evolution
Parallel processor system scheduling

ABSTRACT

One of the main limitation of the application of evolutionary algorithms (EA) is the tendency to converge prematurely to a local optimum. The EAs suffer with the disadvantage of premature convergence and hence the study on convergence of EAs is always one of the most important research fields. Due to outstanding capability of chaos to avoid being trapped in local optimum, it can be considered as an efficient search tool. Therefore, in current paper, in order to taking properties of chaos, eight chaotic maps are employed within a differential evolution (DE) algorithm for solving a stochastic job scheduling problem. To speedup searching and avoid local optimum traps, the random sequences produced from chaotic maps are utilized instead of random variables in DE. Furthermore, to address the uncertainties arising in scheduling environments, Monte Carlo simulation is used. However, simulation is not an optimization approach. Therefore, we design the simulation-based optimization approach where a simulator is combined with chaotic DE. The simulation experiments are used to evaluate the quality of candidate solutions and the chaotic DE is utilized to find best-compromised solutions and then guide the search direction. The performance of simulation-based chaotic DE algorithm is investigated in a computational study, and the results show the outperformance of suggested method with respect to the traditional methods.

1. Introduction and related works

Among scheduling problems addressed in literature, production scheduling with parallel processors has been a subject of interest by researchers and practitioners for a long time (Chang, Chen, & Lin, 2005). In fact, any situations with a series of processing units and a set of jobs to be carried out on units are real-life examples of such problem (Ying & Cheng, 2010). Scheduling of jobs on parallel processors leads to a major operational problem for running a time-sharing system. Moreover, it is a generalization of the single processor problem and of a particular case of problems arising in flexible manufacturing systems (Balin, 2011). This problem can be classified into three main groups (Cevikcan, Durmusoglu, & Baskak, 2011): (i) identical parallel processor scheduling problem, where processing times are same for each processor, (ii) uniform parallel processor scheduling problem, where processors have a parametric relationship in terms of processing time differences, and (iii) unrelated parallel processor scheduling problem, where processing time differences among processors cannot be expressed in a parametric relationship. Generally, parallel processor scheduling problem has many real-world applications, particularly, in semiconductor industry, wafer probing and integrated circuit (IC) manufacturing (Pearn, Chung, & Yang, 2002).

1.1. Parallel processor scheduling problem

The first paper on parallel processor scheduling is presented by McNaughton (1959), where three performance criteria were introduced for parallel-identical-processors. The measures are: makespan, total weighted tardiness and total weighted flow time. Hu (1961) presented a label scheduling algorithm entitled critical path method, to solve a set of non-preemptive jobs on parallel environment. Afterwards, Chen and Liu (1975) generalized Hu's method to address the non-preemptive partial order unit jobs on an arbitrary number of processors. Moreover, Fujii, Kasami, and Ninomiya (1969), Fujii, Kasami, and Ninomiya (1971) investigated such problem with a set of unit jobs and arbitrary precedence relations. In order to minimize either the makespan or flow time, Weber (1982) studied identical parallel machines with the exponential probability distribution of processing times.

Afterwards, many researchers have investigated the problem under different assumptions. Arnaout, Rabadi, and Mun (2006) considered a stochastic parallel machine scheduling problem to

* Corresponding author.
E-mail address: mokhtari_ie@kashanu.ac.ir (H. Mokhtari).

http://dx.doi.org/10.1016/j.eswa.2015.05.015
0957-4174/

minimize total weighted mean completion time and developed a heuristic solution method. Ying and Cheng (2010) developed an iterated greedy heuristic for dynamic parallel processor scheduling problem and extensive computational experiments were implemented to indicate its effectiveness as opposed to state-of-the-art algorithms. In addition, an integration of parts design characteristics and scheduling on parallel processors was presented by Cevikcan et al. (2011) in order to reduce the total set up times. To make the proposed methodology compatible with cable sequencing, an expert system is developed by the authors so as to reduce the burden of supervisors and improve effectiveness. Moreover, Behnamian, Zandieh, and Fatemi Ghomi (2009) developed a hybrid algorithm based on ant colony optimization (ACO), simulated annealing (SA), and variable neighborhood search (VNS) metaheuristics for a kind of parallel processor scheduling with sequence-dependent set up times. In order to minimize the makespan, an initial population generation method based on ACO, a SA for solution evolution, and a VNS which involves three local search procedures to improve the population. A two-phase sub population genetic algorithm is devised in another study by Chang et al. (2005) for solving traditional problem of scheduling with parallel processors. Balin (2011) describes a special case of parallel processors with non-identical parallel processors and applies the genetic algorithm to solve it.

Moreover, Yang (2013) presented an unrelated parallel processor scheduling problem with deterioration effects and maintenance to minimize the sum of completion times. An unrelated parallel batch processing machine scheduling problem was studied by Li, Huang, Tan, and Chen (2013). In order to minimize makespan, several heuristics classified into two groups were proposed based on different solving mechanisms. An another work, Rodriguez, Lozano, Blum, and Garcia-Martinez (2013) analyzed the problem of scheduling a set of jobs on a set of unrelated parallel machines with the objective of total weighted completion times. An iterated greedy heuristic was also designed to generate a sequence of schedules by iterating over a constructive heuristic using destruction and construction mechanisms. Zhang (2013) devised a three-stage algorithm for solving a parallel machine scheduling problem with random processing and setup times and adjustable production minimizing the tardiness of jobs and set the machine speeds. Cheng, Wang, Yang, and Hu (2013) proposed a mixed integer programming model and an ant colony optimization method to address the problem of scheduling parallel batching machines with jobs of arbitrary sizes with identical capacity and processing velocity. In order to overcome the immature convergence, they used a metropolis criterion to select the paths of ants. Lin, Fowler, and Pfund (2013) studied a multiple-objective unrelated parallel machine scheduling problem and proposed two heuristics and a genetic algorithm to find non-dominated solutions. The aim of their model is to simultaneously minimize makespan, total weighted completion time, and total weighted tardiness. As another multi-objective approach, Bandyopadhyay and Bhattacharya (2013) suggested another multi-objective approach to parallel machine scheduling problem minimizing total cost due tardiness, the deterioration cost and makespan, and then employed a Nondominated sorting Genetic Algorithm-II (NSGA-II) as solving approach. Hsu, Ji, Guo, and Yang (2013) solved parallel-machine scheduling problems with unrelated machines, aging effects, deteriorating maintenance, non-resumable jobs, and linear function of maintenance length in terms of starting time.

In year 2014, Lin and Ying (2014) considered an unrelated parallel machine scheduling problem with machine-dependent and job sequence-dependent setup times and devised an artificial bee colony algorithm address the makespan minimization. In another research, Eroglu, Ozmutlu and Ozmutlu (2014) designed an improved genetic algorithm with local search to cope with an unrelated parallel machine scheduling with sequence-dependent set-up times minimizing the makespan. In order to minimize the workload imbalance among the machines, Ouazene, Yalaoui, Chehade, and Yalaoui (2014) studied an identical parallel machines scheduling. A linear mixed integer program formulated to minimize the difference between the greatest and smallest workload assigned to each machine. Moreover, in another work presented by Bilyk, Monch, and Almeder (2014), a variable neighborhood search and a greedy randomized adaptive search procedure were presented to address a scheduling problem faced with in a semiconductor wafer fabrication facility with parallel batch machines and jobs ready times. Liao, Chang, Kuo, and Liao (2014) analyzed performance of five hybrid metaheuristic algorithms for solving unrelated parallel-machine scheduling problem.

In addition, Shen, Monch, and Buscher (2014) suggested a variable neighborhood search algorithm and an iterative scheme for a parallel machine scheduling problem to minimize the total weighted completion time with product families. Bitar, Dauzère-Pérès, Yugma, and Roussel (in press) proposed a memetic metaheuristic to solve an unrelated parallel machine scheduling problem with auxiliary resources and two separately criteria: the weighted flow time and the number of processed products. Mokhtari (2015) suggested an intelligent water drops (IWD) algorithm, as a new metaheuristic, for coping with a parallel processor scheduling problem where rejection of jobs is allowed with a penalty cost. Neamatian Monemi et al. (2015) considered a special parallel machine scheduling problem founded in maintenance planning of heterogeneous wells and also in the context of workover rig scheduling and solved it by a learning mechanism guided hyper-heuristic algorithm. Sels, Coelho, Manuel Dias, and Vanhoucke (2015) designed a genetic algorithm, a tabu search algorithm and a hybrid branch-and-bound procedure to minimize makespan on unrelated parallel machines. Hsieh, Yang, and Yang (2015) proposed controllable processing times for such problem where processing time of a job can be controlled by the allocation of extra resources. To optimize three scheduling measures, i.e. the completion time, the total machine load, and the total earliness and tardiness penalties, a decision support toll was also designed. More recently, Torabi, Sahebjamnia, Mansouri, and Aramon Bajestani (2015) discussed a parallel machine scheduling problem with non-zero ready times, sequence and machine-dependent setup times, and secondary resource constraints. They also considered an uncertainty in processing times and due dates with fuzzy natures. To concurrently minimize total weighted flow time, total weighted tardiness, and total machine load variation, a fuzzy multi-objective particle swarm optimization algorithm was suggested.

1.2. Scheduling under stochastic environment

As can be seen, most of papers presented in literature considered the parallel processor scheduling problem with deterministic parameters. We found few research papers (Arnaout et al., 2006; Torabi et al., 2015; Weber, 1982; Zhang, 2013) considering uncertain processing times within parallel processor scheduling problem. As can be seen, these existing papers considered a diffused range of assumptions and/or solved the problem just via a specific distribution function or by fuzzy assumptions. However, there are factors involved in real-world scheduling problems that are often uncertain in nature with different patterns of uncertainty.

Considering a system in a stochastic context is more realistic than in a deterministic one and works remain to be done concerning the stochastic version of this problem. To address uncertain nature of such problems, Monte Carlo simulation is a simple but