Computers & Industrial Engineering 93 (2016) 36–44

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A quantum behaved particle swarm optimization for flexible job shop scheduling ☆

Manas Ranjan Singh, S.S. Mahapatra *

Department of Mechanical Engineering, National Institute of Technology, Rourkela 769008, India

A R T I C L E I N F O

Article history:
Received 4 February 2013
Accepted 3 December 2015
Available online 23 December 2015

Keywords:
Flexible job shop scheduling
QPSO
Mutation
Chaotic numbers
Makespan

A B S T R A C T

A flexible job shop scheduling problem (FJSP) is an extension of the classical job shop problem (JSP) where operations are allowed to be processed on any among a set of available machines at a facility. For such problems, it is not always possible to find optimal solution in a reasonable time. Hence, a large variety of heuristic procedures such as dispatching rules, local search, and meta-heuristic procedures are used to solve such problems and generate approximate solutions close to the optimum with considerably less computational time. PSO is an effective algorithm which gives quality solutions in a reasonable computational time and requires less number of parameters to be tuned in comparison to other evolutionary meta-heuristics. However, PSO has an inherent drawback of getting trapped at local optimum due to large reduction in velocity values as iteration proceeds and poses difficulty in reaching at best solution. This drawback can be effectively addressed using quantum-behaved particle swarm optimization (QPSO) due to its advanced global search ability. Mutation, a commonly used operator in genetic algorithm, has been introduced in QPSO so that premature convergence can be avoided. Logistic mapping is used to generate chaotic numbers in this paper. The performance of schedules is evaluated in terms of total completion time or makespan (C_{max}). The results are compared with different well-known algorithms used for the purpose from open literature. The results indicate that the proposed QPSO algorithm is quite effective in reducing makespan because small value of relative deviation is observed.

1. Introduction

Scheduling is a one of the important decision making processes in both manufacturing and service industries for improving organizational effectiveness and customer satisfaction. It deals with the allocation of operations on machines (i.e. a sequence of operations on machines) in such a manner that some performance goals such as flow time, tardiness, lateness, and makespan can be minimized. In the current competitive environment, effective scheduling has become a necessity for survival in the market place. Organizations must meet the deadline committed to customers because failure to do so may result in a significant loss of goodwill. The organizations need to schedule activities in such a manner that available resources should be used in an efficient manner. A classical job shop scheduling problem (JSP) deals with a set of n jobs to be processed by a set of machines. Each job is processed on machines in a given order with a given processing time and each machine can process only one job at a time. In contrast, the flexible job shop scheduling problem (FJSP) is an extension of the classical job shop problem (JSP) where operations are allowed to be processed on any among a set of available machines at a facility. FJSP is considered to be more difficult than the classical JSP because it contains an additional problem of assigning operations to machines at a facility. Scheduling in flexible job shop environment is considered as NP-hard problem (Garey, Johnson, & Sethi, 1976). For such problems, it is not always possible to find an optimal solution in a reasonable time. Hence, a large variety of heuristic procedures such as dispatching rules, local search, and meta-heuristic procedures like Tabu search (TS), simulated annealing (SA), genetic algorithm (GA), and particle swarm optimization (PSO) are used to solve such problems and generate approximate solutions close to the optimum with considerably less computational time. These methods can be classified into two main categories: hierarchical approach and integrated approach. In hierarchical approaches, the assignment of operations to machines and the sequencing of operations on the machines are treated separately. In effect, hierarchical approach is based on the idea of decomposing the original problem in order to reduce complexity. Brandimarte (1993) has applied hierarchical approach for FJSP based on decomposition and solved the routing sub-problem using dispatching rules and then concentrated on sequencing sub-problem which is solved by using a TS

☆ This manuscript was processed by Area Editor Subhash C. Sarin.

* Corresponding author.
E-mail addresses: manasranjan.singh@gmail.com (M.R. Singh), mahapatrass2003@yahoo.com (S.S. Mahapatra).

http://dx.doi.org/10.1016/j.cie.2015.12.004
0360-8352/

algorithm. Kacem, Hammadi, and Borne (2002) have proposed a localization approach to solve the resource assignment problem and an evolutionary approach controlled by the assignment model for the FJSP. Integrated approaches consider both assignment and sequencing sub-problems simultaneously. Usually, integrated approaches produce better solutions than hierarchical approaches but more difficult to solve. Chen, Ihlow, and Lehmann (1999) and Jia, Nee, Fuh, and Zhang (2003) have considered integrated approaches to solve FJSP by using genetic algorithm.

Particle swarm optimization, first proposed by Kennedy and Eberhart (1995), is one of the potential evolutionary meta-heuristics, which is inspired by adaptation of a natural system based on the metaphor of social communication and interaction. Originally, PSO was focused on solving nonlinear programming and nonlinear constrained optimization problems comprising of continuous variables. Later, the algorithm is applied to various scheduling problems with improved performance. PSO is an effective algorithm which gives high quality solutions in a reasonable computational time and consists of less number of parameters to be adjusted as compared to the other evolutionary meta-heuristics like GA (Xia & Wu, 2006). Xia and Wu (2005) have presented a practical hierarchical solution approach by making use of PSO to assign operations on machines and simulated annealing (SA) algorithm to schedule operations on each machine. Recently, a new variant of PSO, called quantum-behaved particle swarm optimization (QPSO) has been proposed in order to improve the global search ability of the original PSO (Sun, Feng, & Xu, 2004; Sun, Xu, & Feng, 2004a, 2004b). PSO has an inherent drawback of getting trapped at local optimum due to large reduction in velocity values as iteration proceeds and poses difficulty in reaching at best solution. However, this drawback can be effectively addressed using quantum-behaved particle swarm optimization (QPSO) due to its advanced global search ability of the original PSO. The iterative equations of QPSO is different from that of PSO in that it needs no velocity vectors for particles, needs fewer parameters to be adjusted and can be executed easily. It has been proved that such iterative equations leads QPSO to be global convergent (Clerc & Kennedy, 2002).

In this paper, the search mechanism of the QPSO is used to solve FJSP due its effective exploration and exploitation ability. The proposed approach uses QPSO to assign the operations of each job on available capable machines and sequence the operations on each machine. The objective considered in this paper is to minimize makespan. To improve the solution diversity, chaotic numbers are used to define the particles rather than random numbers (Prakash, Khilwani, Tiwari, & Cohen, 2008).

2. Brief literature review

Brucker and Schlie (1990) have developed a polynomial algorithm for solving the flexible job shop scheduling problem with two jobs. Pezzella, Morganti, and Ciaschetti (2008) have used genetic algorithm for FJSP in which a mix of different rules for generating the initial population, selection of individuals, and reproduction operators were used. Gao, Sun, and Gen (2008) employed a hybrid of GA and variable neighborhood descent (VND) for FJSP. VND involves two local search procedures: local search of moving one operation and local search of moving two operations. Fattahi, Saidi, and Jolai (2007) have proposed a mathematical model along with SA and TS algorithms for solving FJSP. Mastrolilli and Gambardella (2000) have proposed two neighborhood functions incorporated with TS algorithm to find better performances than other existing meta-heuristics in terms of computation time and solution quality. Xing, Chen, Wang, Zhao, and Xiong (2010) have proposed a knowledge-based ant colony optimization algorithm (KBACO) for solving the FJSP. Bagheri, Zandieh, Mahdavi, and Yazdani (2010) have employed an artificial immune algorithm to solve the flexible job shop problem. Ho, Tay, and Lai (2007) have developed an architecture called learnable genetic architecture (LEGA) for learning and evolving solutions for the FJSP. Zhang, Gao, and Shi (2011) have recommended an effective GA for solving the FJSP to minimize makespan. Yazdani, Amiri, and Zandieh (2010) have established a parallel variable neighborhood search (PVNS) algorithm based on six neighborhood structures. Defersha and Chen (2010) have developed a parallel GA to minimize the makespan in a complex flexible job shop, which includes sequence dependent setup times, machine release dates, and time lag requirements. Zhang and Gen (2005) have proposed a multistage operation based GA to deal with the flexible job shop scheduling problem from a point view of dynamic programming. The QPSO algorithm has motivated many researchers from different communities. It has been shown to successfully solve a wide range of continuous optimization problems. Among these applications, it has been used to tackle constraint optimization problems (Sun, Liu, & Xu, 2007), multi-objective optimization problems (Omkara, Khandelwala, Ananthb, Naika, & Gopalakrishnana, 2009), neural network training (Li, Wang, Hu, & Sun, 2007), electromagnetic design (Coelho & Alotto, 2008; Mikki & Kishk, 2006), semiconductor design (Sabat, Coelho, & Abraham, 2009), mechanical design (Mariani, Duck, Guerra, Coelho, & Rao, 2012), and image processing (Gao, Xu, Sun, & Tang, 2010; Lei & Fu, 2008). However, the algorithm is a suitable candidate for solving combinatorial optimization problem like FJSP.

3. Particle swarm optimization

Particle swarm optimization (PSO) algorithm, originally introduced by Kennedy and Eberhart (1995), is a population based evolutionary computation technique. It is motivated by the behavior of organisms such as bird flocking and fish schooling. In PSO, each member is called particle and each particle moves around in the multidimensional search space with a velocity which is constantly updated by the particle's own experience and the experience of the particle's neighbors or the experience of the whole swarm. The members of the entire population are maintained throughout the search procedure so that information is socially shared among individuals to direct the search towards the best position in the search space. Two variants of the PSO algorithm have been developed, namely PSO with a local neighborhood and PSO with a global neighborhood. According to the global neighborhood, each particle moves towards its best previous position and towards the best particle in the whole swarm, called the gbest model in the literature. On the other hand, based on the local variant so called the pbest model, each particle moves towards its best previous position and towards the best particle in its restricted neighborhood. Generally, PSO is characterized as a simple heuristic of well-balanced mechanism with flexibility to enhance and adapt to both global and local exploration abilities. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. PSO does not require that the optimization problem be differentiable as is required by classical optimization methods such as gradient descent and quasi-Newton methods. PSO can, therefore, also be used on optimization problems that are partially irregular and noisy. Due to the simple concept, easy implementation, and quick convergence, PSO has gained much attention and been successfully applied to a wide range of applications such as power and voltage control, neural network training, mass spring system, task assignment, supplier selection and ordering problem, automated drilling, and state estimation for electric power distribution systems (Abido, 2002; Brandstatter & Baumgartner, 2002; Onwubolu & Clerc, 2004; Salman, Ahmad, & Al-Madani, 2002; Van den Bergh & Engelbecht, 2000; Yoshida, Kawata, Fukuyama, & Nakanishi, 2001).