Knowledge-Based Systems 78 (2015) 59–74

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A hybrid biogeography-based optimization for the fuzzy flexible job-shop scheduling problem

Jian Lin

School of Information, Zhejiang University of Finance & Economics, Hangzhou, Zhejiang 310018, PR China

ARTICLE INFO

Article history:
Received 29 July 2014
Received in revised form 28 December 2014
Accepted 31 January 2015
Available online 7 February 2015

Keywords:
Fuzzy flexible job-shop scheduling
Biogeography-based optimization
Path relinking
Local search
Fuzzy processing time

ABSTRACT

Biogeography-based optimization is a novel evolutionary algorithm which mimics the immigration and emigration of species among habitats. In this paper, the biogeography-based optimization is combined with some heuristics to construct an effective hybrid algorithm for solving the fuzzy flexible job-shop scheduling problem. First, path relinking technique is employed as migration operation to generate a new solution. Then, an insertion-based local search heuristic is introduced and embedded in the biogeography-based optimization to modify the mutation operator. Moreover, an efficient machine assignment rule is also proposed to decode the representation based on the operation sequence. Consequently, the exploration and exploitation abilities of the hybrid algorithm are enhanced and well balanced. Computational results and the comparisons with some existing algorithms are presented to show the effectiveness of the proposed hybrid scheme.

1. Introduction

As an extension of the classical job-shop scheduling problem (JSP), the flexible job-shop scheduling problem (FJSP) plays a very important role in manufacturing systems and industrial process. In the FJSP, an operation is allowed to be processed on more than one machine, compared to the classical JSP, the FJSP is more difficult to solve since it needs to determine the assignment of machines to operations as well as the sequencing of the operations on the assigned machines.

After the pioneering work of Bruker and Schlie [1], where a polynomial algorithm was proposed to solve the FJSP with two jobs, many approaches have been proposed to solve the FJSP. Gomes et al. [2] developed an integer linear programming model to schedule flexible job-shop. A two-phase TS algorithm was proposed by Saidi-Mehrabad and Fattahi [3] for the FJSP with sequence dependent setups. In [4], Gao et al. presented a novel genetic algorithm (GA) hybridizing with the variable neighborhood search to solve the FJSP with makespan criterion. After that, many GAs based on various strategies were proposed to solve the FJSP. Typically, an improved GA with a variable generation and selection mechanism in [5], a simulation-based GA in [6] and a hybrid GA combined with local search heuristics in [7]. Besides, a variable neighborhood search algorithm with a knowledge module was designed by Karimi et al. in [8], a hybrid differential evolution (DE) with some heuristics was presented by Yuan [9]. Recently, Rossi [10] proposed a swarm intelligence approach based on a disjunctive graph model, Ziaee [11] developed an efficient heuristic based on a constructive procedure to obtain high-quality schedules quickly.

Among those researches, deterministic processing time and due-date are the common assumptions. However, the processing time of an operation cannot be known precisely and the due-date may be flexible in real-world, processing time and due-date with fuzzy value is quite usual nowadays in practice [12]. The fuzzy job-shop scheduling problem (fJSP) extends the JSP by considering the processing time or the due-date to be fuzzy value. The GA was also employed to deal with the fJSP by Sakawa and Kubota [12]. In [13], Lei developed a Pareto archive particle swarm optimization (PSO) for the fJSP with three objectives to obtain a set of Pareto optimal solutions. By using fuzzy decision-making theory, Gonzalez-Rodriguez et al. [14] proposed a new semantic for this type of problem, and GA was adopted to search possible schedules. Recently, Lei [14] presented a random key GA, Hu et al. [15] designed a modified DE algorithm, and Engin et al. [16] proposed a scatter search method.

By introducing more real-world constraints, the fuzzy flexible job-shop scheduling problem (FFJSP) is a combination of the FJSP and fJSP, which is more close to the production reality. Quite a few evolutionary algorithms (EAs) have been proposed during the past few years, namely, GA, PSO, DE, etc., but very few have been applied to address the FFJSP. Two efficient GAs, called

E-mail address: linjian1001@126.com

http://dx.doi.org/10.1016/j.knosys.2015.01.017
0950-7051/

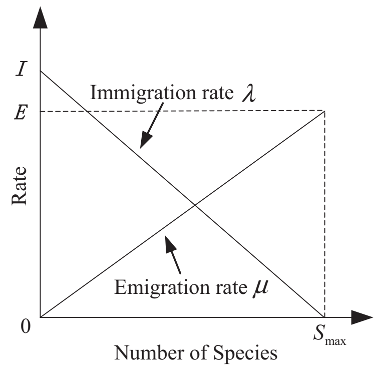

Fig. 1. The relationships between the number of species, emigration rate and immigration rate.

decomposition–integration genetic algorithm (DIGA) [17] and co-evolutionary genetic algorithm (CGA) [18], were developed by Lei [17]. More recently, an estimation of distribution algorithm (EDA) [19], a hybrid artificial bee colony (HABC) algorithm [20] and a teaching–learning-based optimization (TLBO) algorithm [21] were presented to solve the FJSPF.

The biogeography-based optimization (BBO) is a novel evolutionary algorithm, which mimics the immigration and emigration process of species among habitats [22]. The BBO has demonstrated good performance when compared with other EAs [23–29]. However, to the best of our knowledge, there is no research work about the BBO for solving the FFJSP. This paper aims at employing an effective hybrid biogeography-based optimization (HBBO) algorithm in solving the FFJSP with the objective to minimize the fuzzy makespan. Experiments and comparisons are conducted on the benchmark instances generated by Lei [17,18] to verify the effectiveness of the proposed scheme.

The remainder of the paper is organized as follows. In Sections 2 and 3, the FFJSP and the BBO are introduced, respectively. In Section 4, a HBBO scheme is proposed for the FFJSP. The computational results on benchmark instances together with comparison to some existing algorithms are presented in Section 5. Finally, a conclusion is given in Section 6.

2. Fuzzy flexible job-shop scheduling problem

2.1. Problem description

The FFJSP is commonly described as follows. There are a set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$ to be processed on m machines $M = \{M_1, M_2, \ldots, M_m\}$. Each job J_i is formed by a sequence of n_i operations $\{O_{i,1}, O_{i,2}, \ldots, O_{i,n_i}\}$. The execution of $O_{i,j}$ requires one machine out of a given set of machines out of M. The processing

Table 1
Fuzzy processing time.

Operations	Machines		
	M_1	M_2	M_3
$O_{1,1}$	(3,5,8)	(7,8,10)	(8,9,10)
$O_{1,2}$	(6,8,10)	(5,7,9)	(4,5,7)
$O_{1,3}$	(3,6,8)	(11,15,21)	(3,4,5)
$O_{2,1}$	(6,9,12)	(7,10,13)	(4,7,10)
$O_{2,2}$	(2,3,5)	(2,5,7)	(3,4,5)
$O_{3,1}$	(5,8,10)	(9,13,16)	(2,5,6)
$O_{3,2}$	(3,4,5)	(8,11,13)	(10,13,17)

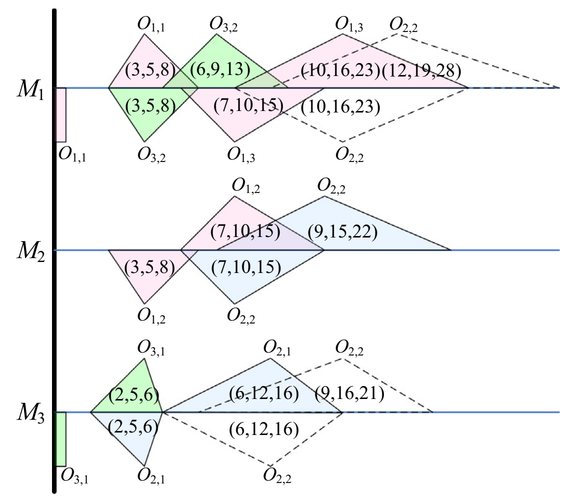

Fig. 3. The machine assignment for π by using the ECM rule.

time of the $O_{i,j}$ on machine M_k is represented as a triangular fuzzy number (TFN) $p_{i,j,k} = (p^1_{i,j,k}, p^2_{i,j,k}, p^3_{i,j,k})$, where $p^1_{i,j,k}$ is the best processing time, $p^2_{i,j,k}$ is the most probable processing time and $p^3_{i,j,k}$ is the worst processing time. Similarly, the fuzzy makespan of $O_{i,j}$ is represented as a TFN $C_{i,j} = (C^1_{i,j}, C^2_{i,j}, C^3_{i,j})$, where $C^1_{i,j}$ is the best makespan, $C^2_{i,j}$ is the most probable makespan and $C^3_{i,j}$ is the worst makespan. However, there are several assumptions that may not necessarily apply to this argument, such as:

- Each machine can process only one operation and no job can be processed more than one machine at a time.
- The operation cannot be interrupted during the processing.
- The transfer times between different machines are included in the processing time.

```
Procedure ECM rule (π)
  For each machine m
    γ_m = 0;
      Λ^m(γ_m) = Φ;
    Next for
    For t = 1 to n
      Find the machine m that can process the operation π(t) with the earliest completion time;
      γ_m = γ_m + 1;
      Λ^m(γ_m) = π(t)
    Next for
End procedure
```

Fig. 2. Pseudo code of the ECM rule.