Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 61 (2015) 521 – 526

Procedia Computer Science

Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

A Bayesian Optimization-based Evolutionary Algorithm for Flexible Job Shop Scheduling

Lu Sun[a], Lin Lin[a,b,*], Yan Wang[a] ,Mitsuo Gen[b,c] and Hiroshi Kawakami[d]

[a]*Dalian University of Technology, 116620, China*
[b]*Fuzzy Logic Systems Institute, 820-0067, Japan*
[c]*Tokyo University of Science, 113-8656, Japan*
[d]*Kyoto University, 606-8306, Japan*

Abstract

Flexible Job-shop Scheduling Problem (fJSP) is a typical and important scheduling problem in Flexible Manufacturing System (FMS). The fJSP is an extended version of Job-shop Scheduling (JSP) that is NP hard problem. Due to it according with the real production system, we adopt a hybrid evolutionary computation algorithm to solve the fJSP problems. Among them, the Bayesian Optimization Algorithm (BOA) is introduced to the characteristics of scheduling and uncertainty characteristics of the time in the fJSP. On this basis, we propose a distributed evolutionary algorithm and parameter adaptive mechanism. Finally, through experiments, we conclude that the proposed hybrid evolutionary algorithm based on BOA with grouping mechanism get better solution than original algorithm and improve robustness of algorithm. Meanwhile, the paper also have objective perspective, that is we can group the data different from each other, make the whole population into sub-populations, and then make the experiment separately on different and parallel machines in distributed environment, so that not only optimizes the best solution, but also enhance the efficiency and shortened the time.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology

Keywords: Bayesian Optimization Algorithm; Flexible Job-shop Scheduling Problem; Evolutionary Algorithms

1. Introduction

In today's society, the scheduling models often need to adapt to different application requirements, and designing the scheduling model one by one for each customer's demands is clearly unrealistic, so the highly flexible scheduling is particularly important. In this paper, we introduce a fJSP (flexible Job-shop Scheduling Problem) model to simulate the automatic scheduling for a flexible manufacturing model. The fJSP as an extension of Job-shop Scheduling Problem (JSP) is a typical combinatorial optimization problem and it is a NP-hard problem under

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2015.09.207

the constraint of priorities and resources (Lawler, 1993; Gen & Cheng, 2000; Gen, Gao & Lin, 2009). Meanwhile, it has the characteristics of resources non-uniqueness: for an operation, it can choose the machine in available set to complete. For a traditional JSP (tJSP), researchers often propose the following assumption: for an operation, machine and processing time are fixed. With the development, multi-purpose equipment is replacing traditional equipment in order to increase efficiency and decrease cost in manufacturing. Therefore, considering the fJSP research is closer to the actual production mode.

Recently, Xing provided an effective integration between Ant Colony Optimization (ACO) models and applies the existing knowledge to guide the current heuristic searching, and indicates that the proposed algorithm is effective (Xing, *et al,* 2010). Zhang proposed algorithm with Global Selection (GS) and Local Selection (LS) routines designed to generate high-quality initial population in the initialization stage for minimizing makespan (Zhang, 2011). Nouri, *et al* proposed a combined genetic algorithm and tabu search with a scheduler agent applying a Neighborhood-based Genetic Algorithm (NGA) to guide the research in promising regions (Nouri, *et al*, 2015). Chang and Liu proposed a Hybrid Genetic Algorithm (HGA) for solving the distributed and flexible job-shop scheduling problem (DfJSP) and demonstrated the effectiveness of the algorithm through the experiment (Chang & Liu, 2015). All best solutions by above algorithms are generated by repeated iteration, but not learning from a learning criterion and existing research only considered the processing time and processing sequences, however, does not take process machine resources of multiple factors into account. So it is lack of related research for flexibility and the effectiveness analysis of the scheduling; design of optimization method, especially for the craft flexibility route and machine optional of flexible scheduling problem.

In this paper, we propose a hybrid evolutionary algorithm with Bayesian Network (BN) to solve S-fJSP that aims to minimize the makespan of processing time. We choose the Particle Swarm Optimization (PSO) proposed by (Kennedy & Eberhart, 1995) as the basic algorithm and group the chromosomes randomly at first. Generate the candidate network by PSO and choose the best BN that is closest to the real structure with the maximum likelihood function. Then, group the chromosomes again according to the BN structure. Do same operations mentioned above every m generations. Among the generations, adapt the parameters of PSO in the process adaptively to ensure to get better solutions. Section 2 introduces formulating process of fJSP model; Section 3 proposes the hybrid evolutionary algorithm with BN; Section 4 provides the detailed numerical experiment and result; finally, Section 5 gives the conclusion of the whole paper.

2. Modeling of fJSP

For the fJSP model, different operations of different tasks are processed on different machines, the processing time of each operation is fixed, and the machine for one operation is only one at one time. In fact, the fJSP model is an extension of JSP, so we can describe the formulation process as follows:

(1) Machine allocation: choose a machine for each operation from the available machine set.

(2) Operation sequence: all necessary operations for completing the jobs which satisfying the precedence constraints.

(3) Sequencing robustness: a best solution is not sensitive to data.

In the fJSP, each job i consists of n_i operations ($O_{i1}, O_{i2}, \ldots, O_{in_i}$). For each operation O_{ik}, processing machine must be from the machine set A_{ik}. The operations must be completed in sequence for one job.

The symbols used in the S-fJSP are defined as follows:

Indices:

i, h: job index, $i, h = 1,2.,\ldots,n$

j: machine index, $j = 1,2,\ldots,m$

k, g: operation index, k, $g = 1,2,\ldots, n_i$

Parameters:

n: total number of jobs

m: total number of machines

n_i: total number of operations of job i

t_{ikj}: processing time of kth operation of job i

Decision variables: