Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 131 – 136

Procedia Engineering

www.elsevier.com/locate/procedia

An Artificial Bee Colony Algorithm Based on Problem Data Properties for Scheduling Job Shops

Rui Zhang*

School of Economics and Management, Nanchang University, Nanchang 330031, PR China

Abstract

To solve the job shop scheduling problem with the objective of minimizing total weighted tardiness, an artificial bee colony algorithm based on problem data analysis is proposed. First, characteristic values are defined to describe the criticality of each job in the process of scheduling and optimization. Then, a fuzzy inference system is employed to evaluate the characteristic values according to practical scheduling knowledge. Finally, a local search mechanism is designed based on the idea that critical jobs should be processed with higher priority. Numerical computations are conducted with an artificial bee colony algorithm which integrates the local search module. The computational results for problems of different sizes show that the proposed algorithm is both effective and efficient.

Keywords: Artificial bee colony algorithm, Job shop scheduling problem, Local search, Problem data

1. Introduction

The job shop scheduling problem (JSSP) has been known as a notoriously hard combinatorial optimization problem since the 1950s. In terms of computational complexity, JSSP is $\mathcal{NP}$-hard in the strong sense. Therefore, even for very small JSSP instances, it is by no means easy to guarantee the optimal solution. In recent years, the meta-heuristics — such as genetic algorithm (GA) [1], tabu search (TS) [2], particle swarm optimization (PSO) [3], and ant colony optimization (ACO) [4] — have clearly become the research focus in practical optimization methods for solving JSSPs.

However, when the problem size grows, meta-heuristic algorithms usually take excessive time to converge. To enhance the efficiency of these algorithms, two types of approaches may be roughly identified in the literature:

(1) Focusing on the optimization algorithm: the conventional operations (or parameters) in the standard version of these algorithms have been modified or redesigned to promote their performance in the neighborhood search.

(2) Focusing on the features of the pending problem: problem-specific or instance-specific information is extracted and utilized in the searching process to accelerate the convergence speed of these algorithms.

The former approach is independent of problem classes. But according to the no free lunch theorem [5], such improvements on the optimization algorithm alone cannot guarantee good performance for all problems. In terms of the latter approach, embedded local search can utilize the characteristic information to improve the solutions in

*Corresponding author
Email address: zhangrui05@gmail.com (Rui Zhang)

doi:10.1016/j.proeng.2011.11.2478

the optimization process. However, how to effectively extract and describe the characteristic information remains a challenging but rewarding research topic.

In this paper, we devise a fuzzy inference system based on intuitive knowledge to evaluate the criticality value of each job. Then, this information is used in a local search mechanism to promote the optimization efficiency of the artificial bee colony (ABC) algorithm. The paper is organized as follows. The discussed job shop scheduling problem is formulated in Section 2. Sections 3 and 4 presents the detailed algorithms. The computational results and a brief analysis are provided in Section 5. Finally, the conclusions are given in Section 6.

2. Problem formulation

Job shop is one of the most frequently adopted models when dealing with scheduling problems. In the job shop scheduling problem (JSSP), a set of n jobs $\{J_i\}_{i=1}^n$ are to be processed on a set of m machines $\{M_k\}_{k=1}^m$. Each job has a fixed processing route which traverses all the machines in a predetermined order. Besides, a preset due date and a weight are given for each job. JSSP can also be described by a disjunctive graph $G(N, A, E)$, in which $N = \{0, 1, \ldots, *\}$ represents the set of nodes (including two dummy nodes, 0 and $*$); A is the set of conjunctive arcs and $E = \bigcup_{k\in M} E_k$ is the set of disjunctive arcs (E_k represents the disjunctive arcs that correspond to machine M_k). Then the discussed JSSP can be formulated as follows:

$$\begin{cases} \min & TWT = \sum_{j\in C} w_j \left(t_j + p_j - d_j\right)^+, \\ \text{s.t.} & \\ & t_i + p_i \leq t_j, \ \ (i, j) \in A, \\ & t_i + p_i \leq t_j \vee t_j + p_j \leq t_i, \ \ (i, j) \in E_k, k = 1, \ldots, m, \\ & t_i \geq 0, \ \ i \in N. \end{cases}$$

In this formulation, C is the set of the last operations of each job; w_j and d_j are respectively the weight and the due date of the job which operation j belongs to; p_j and t_j (the decision variable) are the processing time and the starting time of operation j, respectively; $(x)^+ = \max\{x, 0\}$. The scheduling objective considered in this paper is to determine the processing sequence of operations on each machine such that the total weighted tardiness is minimized.

3. Evaluation of the criticality values

In this section, we provide a detailed description of the fuzzy inference system which is designed to calculate the criticality value (CV) of each job.

Based on a given schedule (including the completion time of each job), we can define:

- The relative distance between job i's completion time and its due date: $g_i = (\hat{F}_i - d_i)/\sum_{j\in J_i} p_j$, where $\hat{F}_i$ denotes job i's completion time under the current schedule; $\sum_{j\in J_i} p_j$ equals the total processing time of job i.
- The relative slack time of job i: $h_i = (d_i - c_i - \sum_{j\in J_i'} p_j)/\sum_{j\in J_i} p_j$, where c_i refers to the completion time of the currently considered operation of job i and also the release time of J_i' which is the set of its succeeding operations in job i. Note that h_i corresponds to specific operations of job i and thus reflects the features of different processing stages of the job.
- The normalized weight of each job: $v_i = \overline{w}_i = (w_i - w_{\min})/(w_{\max} - w_{\min})$, where $w_{\max} = \max_{i=1}^n w_i$, $w_{\min} = \min_{i=1}^n w_i$.

Based on these variables, a fuzzy controller is designed to calculate the CV. The fuzzy controller takes g_i, h_i and v_i as input variables, and outputs the CV for each job. In the fuzzy inference system, the four input/output linguistic variables are respectively denoted by G, H, V and B, and are divided into three fuzzy sets as follows.

- $G, H = \{\text{NL}, \text{Z}, \text{PL}\}$, i.e. {Negative, Around Zero, Positive}.
- $V = \{\text{S}, \text{M}, \text{L}\}$, i.e. {Small, Medium, Large}.
- $B = \{\text{N}, \text{PC}, \text{C}\}$, i.e. {Not a critical job, Possibly a critical job, A critical job}.

In this fuzzy inference system, all the relevant membership functions are chosen to have symmetrical triangular shapes. For example, the membership functions of the three fuzzy sets related to G are illustrated in Figure 1. We could