Simulation Model of Ant Colony Optimization for the FJSSP

Li-Ning Xing
National University of Defense Technology, China

Ying-Wu Chen
National University of Defense Technology, China

Ke-Wei Yang
National University of Defense Technology, China

INTRODUCTION

The job shop scheduling problem (JSSP) is generally defined as decision-making problems with the aim of optimizing one or more scheduling criteria. Many different approaches, such as simulated annealing (Wu et al., 2005), tabu search (Pezzella & Merelli, 2000), genetic algorithm (Watanabe, Ida, & Gen, 2005), ant colony optimization (Huang & Liao, 2007), neural networks (Wang, Qiao, &Wang, 2001), evolutionary algorithm (Tanev, Uozumi, & Morotome, 2004) and other heuristic approach (Chen & Luh, 2003; Huang & Yin, 2004; Jansen, Mastrolilli, & Solis-Oba, 2005; Tarantilis & Kiranoudis, 2002), have been successfully applied to JSSP.

Flexible job shop scheduling problem (FJSSP) is an extension of the classical JSSP which allows an operation to be processed by any machine from a given set. It is more complex than JSSP because of the addition need to determine the assignment of operations to machines. Bruker and Schlie (1990) were among the first to address this problem. The flexible job shop scheduling problem may be formulated as follows.

1. There is a set of n jobs that plan to process on m machines;
2. The set machine is noted M, $M = \{M_1, M_2, ..., M_m\}$;
3. Each job j consists of a sequence of n_j operations $O_{j1}, O_{j2}, ..., O_{jn_j}$;
4. The execution of each operation i of a job j (noted O_{ji}) requires one machine out of a set of given machines called $M_{ji} \subseteq M$.

The problem is thus to both determine an assignment and a sequence of the operations on all machines that minimize following criteria.

1. Maximal completion time of machines;
2. Total workload of the machines;
3. Critical machine workload.

The weighted sum of the above three objective values is taken as the objective function.

$$F(c) = 0.5*F_1(c) + 0.2*F_2(c) + 0.3*F_3(c) \quad (1)$$

Where $F(c)$ denotes the total evaluation value of the schedule c; $F_1(c)$ denotes the maximal completion time of machines (makespan) of the schedule c; $F_2(c)$ denotes the total workload of the machines of the schedule c; $F_3(c)$denotes the critical machine workload of the schedule c.

BACKGROUND

For solving the realistic case with more than two jobs, two types of approaches have been used: hierarchical approaches and integrated approaches (Xia & Wu, 2005).

In hierarchical approaches, assignment of operations to machines and the sequencing of operations on the machines are treated separately. Kacem, Hammadi, and Borne (2002a; 2002b) proposed a genetic algorithm controlled by the assigned model for the FJSSP. Xia and Wu (2005) present an effective hybrid optimization approach, which makes use of particle swarm optimization to assign operations on machines and simulated annealing algorithm to schedule operations, for the multi-objective FJSSP.

Integrated approaches were used by considering assignment and scheduling at the same time. The integrated approach which had been presented by Dauzere-Peres and Paulli (1997) was defined a neighborhood structure for the FJSSP where there is no distinction between reassigning and resequencing an operation, and the tabu search procedure is proposed based on the neighborhood structure. Mastrolilli and Gambardella (2002) improved Dauzere-Peres' tabu search techniques and presented two neighborhood functions. Most researchers were interested in applying tabu search techniques and genetic algorithms to FJSSP in the past (Xia & Wu, 2005).

SIMULATION MODEL OF ANT COLONY OPTIMIZATION

Framework of the Simulation Model

The framework of our proposed simulation model is displayed as Figure 1.

Input Subsystem

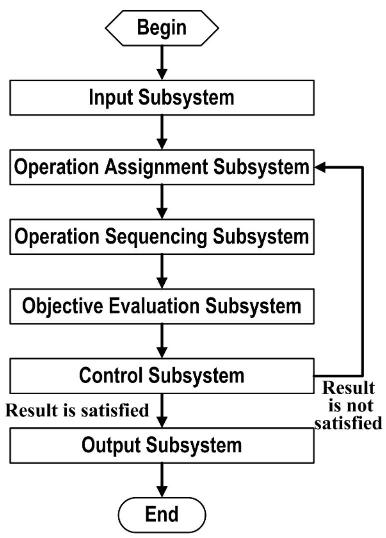
The leading function of input subsystem is inputting all necessary data for solving FJSSP. In our work, we apply file mode to implement the data inputting. Please note, it should have a verify function after data reading, for example, each input data should be a positive integer etc.

Operation Assignment Subsystem

The primary mission of operation assignment subsystem is achieving an excellent assignment of operations to machines. In this part, each assignment was evaluated by formula (2).

$$Fitness(\alpha) = 0.2*F_2(\alpha) + 0.8*F_3(\alpha) \quad (2)$$

Figure 1. The framework of our proposed simulation model



Also, the operation assignment machine knowledge (OAMK) is defined for operation assignment. OAMK is the accumulative knowledge of assigning the giving operation to a more appropriate machine. It was achieved from the near-optimal solution of FJSSP of each iterative. A knowledge matrix *OAMK* with size $|Oper| \times |Mach|$ is defined for the OAMK, where $|Mach|$ denotes the number of machines, and $|Oper|$ denotes the total account of all operations. For an arbitrary element $OAMK[i][j]$, it means the probability of assigning the giving operation i to the current machine j.

For enhancing the assignment performance, we try to assign each operation to the machines which process the giving operation with a minimal processing time or the second minimal processing time. The implementing flow of operation assignment is listed as follows.

Step 1. Select an operation (i.e., O_{ij}) among all operations which need to be assigned.

Step 2. Search all machines (e.g., M_{ij}) which process the giving operation with a minimal processing time or the second minimal processing time.

Step 3. Choice a machine (i.e., M_{ijk}, $M_{ijk} \in M_{ij}$) among the achieved machine set M_{ij} randomly with a probability distribution, which was indicated by the OAMK, and assign operationO_{ij} to the selected machine M_{ijk}.

Step 4. Repeat *Step 1* to *Step 3* until all operations were assigned to the appropriate machines.

When obtaining the global optimal solution (the most excellent solution from the beginning of the trial), the OAMK will be updated by applying the rule of (3) according to the global optimal solution.

$$OAMK(i,m) = OAMK(i,m) + Q_G \quad (3)$$

Where m denotes each giving machine, i denotes each operation processed in machine m, Q_G denotes the incremental level in the knowledge updating phase.

Operation Sequencing Subsystem

In order to enhance the sequencing performance, we try to arrange each operation to the giving machine using ant colony optimization (ACO) algorithm. The computational flow of ACO algorithm is displayed as Figure 2.

1. Schedule knowledge initialization. In this part, operation assignment position knowledge (OAPK) is defined according to the traditional pheromone definition. OAPK is the accumulative knowledge of the more appropriate operation processing sequence at a giving machine. It is achieved from the near-optimal solution of FJSSP of each iterative.