APPLICATION OF IMPROVED PARTICLE SWARM OPTIMIZATION ALGORITHM IN JOB SHOP SCHEDULING PROBLEM

Qi ZHAO¹, Zhe YANG²

¹ College of Computer Science and Software, Tianjin Polytechnic University, Tianjin, China.
² College of Information and Computer Engineering, Taiyuan University of Technology, Taiyuan, Shanxi, China.

ABSTRACT

Job shop scheduling problem (JSSP) is a typical NP-hard combinatorial optimization problem in the field of production management. Many different algorithms have been applied to solve combinatorial optimization problems, such as genetic algorithm (GA), particle swarm optimization (PSO), heuristic algorithm and so on. However, it is difficult to guarantee the optimization of the JSSP even on a small scale. The original particle swarm optimization (PSO) algorithm is generally used to solve the continuous function problem and to help solve the discrete problem. This paper proposes an improved particle swarm optimization algorithm for JSSP. The idea of genetic algorithm is introduced into PSO to perform crossover mutation operation on the particles in solution space to ensure the diversity of particle population. In order to avoid the traditional particle swarm optimization algorithm easy to fall into local optimum when solving JSSP, a nonlinear dynamic weight method is adopted. The experiments show that the improved particle swarm optimization (IPSO) algorithm is more robust and efficient in dealing with the workshop scheduling problem.

KEYWORDS: Job-shop Scheduling Problem; PSO; IPSO; Inertia Weight.

1. INTRODUCTION:

JSSP is the core component of intelligent manufacturing execution system. Considering a large number of integrated entities and their interactions, the research of production system is complex. One of the reasons for this complexity is JSSP. Scheduling of operations involves allocating resources over a period of time to perform a set of tasks. This is one of the most critical issues in manufacturing process planning and management [1]. It is important for the industry to get the best solution to the scheduling problem, as the productivity and cost of any production plant depends on the dispatch used to control the plant's work [2]. In recent decades, many intelligence algorithms, such as GA, PSO and heuristic algorithm, have been applied to solve JSSP. Particle swarm algorithm (PSO) was proposed by Eberhart and Kennedy in 1995 [3,4]. Clerc and Kennedy studied the stability and convergence of particle swarm in multidimensional complex space [5]. Gohet et al. studied the competitive coevolution of multi-target particle swarm optimization algorithm [6]. Zhang, GH et al. proposed an effective hybrid particle swarm optimization algorithm for multi-objective flexible job shop scheduling [7]. Chen and Li [8,9], Shi and Eberhart [10] and Trelea[11] respectively studied the parameter selection of PSO. Yin and Zhang proposed a heuristic algorithm to solve JSSP [12]. The main content of this paper is as follows: firstly, the JSSP is described and the mathematical model is established in chapter 2. The third chapter to the PSO are described, the fourth chapter mainly puts forward the IPSO, in order to achieve the particle location value and feasible solution in the solution space mapping relationship, the coding method based on process introduces GA and combined with a nonlinear dynamic inertia weight method to optimize the existing algorithm. Finally, the feasibility and effectiveness of the IPSO are tested and concluded.

2. THE JOB-SHOP SCHEDULING PROBLEM:

JSSP can be described as follows. There are n jobs and m machines. Each job can only be completed after processing by m machines. The operation order of the machine is predetermined. Each procedure use one of the m machines to complete the work of a job within a fixed time interval. The preparation time of the job is ignored. Once the job procedure is carried out, it is not allowed to change. Each job corresponds to a set of processes on m machines. Scheduling is the time slots that assign a process to different machines. The goal of JSSP is to find an appropriate sequence of operations for all the artifacts to minimize the maximum total completion time ($C_{max}$). The following example illustrates the job shop scheduling problem. As shown in Table 1 and Fig1, there is a 3 x 2 JSSP. The machine startup sequence corresponding to each job and the time of each process is given. For the conceptual model, the notations are defined in Table 2.

According to the description listed above, the conceptual model of the JSSP can be defined as follows:

\[ \text{Minimize} \quad O_q = C_{max} \]

\[ O_q \leq O_{i} - \tau_i, \quad i = 0, 1, 2, ..., n \times m + 1; \quad q \in P \]

\[ \sum_{m \in M} \tau_m = 1; \quad m \in M; \quad t \geq 0 \]

\[ O_q = 0, \quad i = 0, 1, 2, ..., n \times m + 1 \]

Eq (1) is the objective fitness function, which minimizes the maximum total completion time. The precedence relationship constraint is determined by Eq. (2). Eq. (3) represents the constraint on a machine, which can only operate once at most. As shown in Eq (4), the completion time of the operation must be an affirmation of the constraint.

3. PARTICLE SWARM OPTIMIZATION:

Particle swarm optimization algorithm (PSO) is a kind of swarm intelligence optimization algorithm in the field of computational intelligence. This algorithm is derived from the research on bird feeding problem, which was first proposed by Kennedy and Eberhart in 1995. First, a group of particles is initialized in the feasible solution space, each of which represents a potential solution for extreme optimization. The particle moves in solution space and updates its position by

According to the description listed above, the conceptual model of the JSSP can be defined as follows:

\[ \text{Minimize} \quad O_q = C_{max} \]

\[ O_q \leq O_{i} - \tau_i, \quad i = 0, 1, 2, ..., n \times m + 1; \quad q \in P \]

\[ \sum_{m \in M} \tau_m = 1; \quad m \in M; \quad t \geq 0 \]

\[ O_q = 0, \quad i = 0, 1, 2, ..., n \times m + 1 \]

Eq (1) is the objective fitness function, which minimizes the maximum total completion time. The precedence relationship constraint is determined by Eq. (2). Eq. (3) represents the constraint on a machine, which can only operate once at most. As shown in Eq (4), the completion time of the operation must be an affirmation of the constraint.
tracking the optimal position of the individual \( (P_{\text{best}}) \) and the optimal position of the population \( (G_{\text{best}}) \). The optimal value calculated by fitness function in the positions experienced by particles refers to the optimal value of fitness searched by all particles in the population. The fitness value is updated every time the particle updates. The \( P_{\text{best}} \) and \( G_{\text{best}} \) is update by comparing the fitness value of the new particle with the fitness value of the individual extreme value and the population extreme value. The formula for updating speed and position is as follows:

\[
V_{id}^{t+1} = \omega V_{id}^{t} + c_1 r_1 (P_{\text{best},id}^{t} - X_{id}^{t}) + c_2 r_2 (G_{\text{best},id}^{t} - X_{id}^{t})
\]  
(5)

\[
X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1}
\]  
(6)

Eq. (5) is the d-dimensional velocity update equation of particle i, where \( c_1 \) and \( c_2 \) is the learning factor, \( \omega \) is the inertial factor, and \( r_1 \) and \( r_2 \) is the uniform random number in the range \([0,1]\) to increase the search randomness. Eq. (6) is the d-dimensional position updating formula of particle i, \( P_{\text{best}}^{t} \) represents the d-th dimension of velocity vector of \( K_{\text{best}} \), iteration particle i, and \( X_{id}^{t} \) represents the d-th dimension of position vector of \( K_{\text{best}} \), iteration particle i. The process of particle swarm algorithm is shown in Fig 2.

4. THE IMPROVED PARTICLE SWARM ALGORITHM FOR JSSP:

There are some problems in the application of PSO algorithm to solve the JSSP. PSO is mainly used to solve the continuous function problem and help to solve the discrete problem. JSSP is a combinatorial optimization problem with discrete solution space. So the first step is to find an efficient coding method to map the position value of the particle to the feasible solution in solution space. In this paper, the idea of GA is introduced into the algorithm by means of the process-based coding. At the same time, in order to enhance the local optimization ability of the original particle swarm optimization algorithm and prevent the premature convergence from falling into local optimization, the existing PSO algorithm was optimized by adopting a nonlinear dynamic adjustment method of inertial weight.

4.1 Encoding particle:

The searching space is created in an \( n \times m \) dimensions space form jobs on \( m \) machines JSSP. In order to simulate the operation permutation sequence of JSSP, the procedure based coding method is adopted. Each particle is composed of \( n \) genes representing processes, representing an operable sequence of JSSP, and each job appears \( m \) times.

For example, for a 33 JSSP problem, particle \( a = \{1, 3, 2, 2, 1, 3\} \) and the corresponding operation sequence is \( a = \{O_1, O_3, O_2, O_2, O_1, O_3\} \). The new particle is updated by crossing and swapping. For the two particles \( a = \{2, 1, 3, 1, 1, 2, 3\} \) and \( b = \{3, 1, 2, 2, 3, 1, 3, 2\} \), the crossover operator based on job as particle update way. First, the set of artifacts is randomly divided into groups J1 and J2, J1=\{1,2\}, J2=\{3\}. Particle a and b are respectively separated according to the workpiece contained in J1 and J2 to obtain two sets of sub-sequences a and b, and new particles a’ and b’ are finally obtained through crossing. In this paper, the method of inserting mutation is adopted for particle mutation operation to randomly select an element and randomly insert a position in the operation sequence. Crossover and mutation operations are shown in Fig 3 and Fig 4.

4.2 Nonlinear dynamic weight method:

The PSO algorithm contains some parameters that have a great impact on the performance of the algorithm. The selection of the parameters of the algorithm is still empirical and there is still no optimal selection scheme. There are many adjustment schemes for parameters, such as particle swarm algorithm with convergence factor [13] proposed by Clerc, linear decreasing inertial weight strategy proposed by Shi Y and RC Eberhart [14]. In this paper, the inertia weight of the original particle swarm algorithm is adjusted in a nonlinear dynamic way. The formula is as follows:

\[
\omega = f(iter) = \left( \frac{iter_{\text{max}} - iter}{iter_{\text{max}}} \right) \omega_{\text{max}} + \omega_{\text{min}}
\]  
(7)

\( \omega_{\text{max}} \) represents the initial weight at the start of a given run \( \omega_{\text{min}} \) represents the final weight at the end of a given run. \( f(iter) \) is fitness value of the current iteration number. \( iter_{\text{max}} \) is maximum number of iterations. \( iter \) is current iteration number.

4.3 Learning factor improvement:

Learning factors \( c_1 \) and \( c_2 \) determine the influence of the particle’s own experience and population experience on the particle’s motion trajectory. The improved particle swarm algorithm in this paper is \( c_1 \), first large and then small, and \( c_2 \), first small and then big. The basic idea mainly refers to the historical information of particles in the early stage and pays more attention to social information in the later stage. Improved formula of learning factor as follows:

\[
c_1 = 2 - \sin \left( \frac{\pi \cdot iter}{iter_{\text{max}}} \right)
\]  
(8)

\[
c_2 = 1 + \sin \left( \frac{\pi \cdot iter}{iter_{\text{max}}} \right)
\]  
(9)

Where \( k \) is the current number of iterations and \( iter_{\text{max}} \) is the maximum number of iterations.

5 EXPERIMENTAL RESULTS:

In order to illustrate the feasibility and effectiveness of the IPSO proposed in this paper in the application of JSSP, the standard test cases of JSSP were used for comparison test. The algorithm used the initial parameters \( c_1=2 \) and \( c_2=1 \) to conduct experiments on PSO and IPSO respectively. The output results are the maximum total completion time, average relative error percentage, optimization rate and standard deviation of the results. The PSO and IPSO are compared by four indexes. The experimental results are shown in Fig 5. The line chart of the four indexes including maximum completion time, average relative error percentage, optimization rate and standard deviation of the results shows that IPSO has higher stability and better optimization ability than PSO in dealing with the JSSP. The feasibility and effectiveness of the improved algorithm are verified. Comparison of PSO and IPSO fitness curves, average of 500 replicate experiments in Fig 6 shows that the PSO algorithm is more likely to fall into local optima, when dealing with job shop scheduling problems, while the improved particle swarm algorithm has a good ability to jump out of local optimal solutions.
6. CONCLUSIONS:
This paper presents an improved particle swarm optimization algorithm. Firstly, the mathematical model established for the job shop scheduling problem is applied to the coding problem of the particle swarm algorithm in solving the discrete problem of job shop scheduling. The process is coded to encode the particles to realize the particle position value and solution space. The mapping relationship of feasible solutions, and the idea of genetic algorithm is introduced to update the particles to increase the diversity of particle populations. In order to solve the problem that the original particle swarm optimization algorithm is easy to fall into local optimum in solving the job shop scheduling problem, a nonlinear dynamic adjustment method of inertia weight is adopted. At the same time, in order to balance the global search ability and local search ability of the particle swarm optimization algorithm, the particle swarm The learning factors in the algorithm were adjusted. The experimental results show that the proposed algorithm has good stability while improving the optimization ability and speeding up the convergence.

REFERENCES:
7. Zhang, GH;Shao, XY ;Li, PG; Gao, L ;(2008). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. COMPUTERS & INDUSTRIAL ENGINEERING 56(4) ,1309-1318