



## A multi-objective PSO for job-shop scheduling problems

D.Y. Sha<sup>a</sup>, Hsing-Hung Lin<sup>b,\*</sup>

<sup>a</sup> Department of Industrial Engineering and System Management, Chung Hua University, Hsin Chu, Taiwan, ROC

<sup>b</sup> Department of Industrial Engineering and Management, National Chiao Tung University, Hsin Chu, Taiwan, ROC

### ARTICLE INFO

#### Keywords:

Job-shop scheduling  
Particle swarm optimization  
Multiple objectives

### ABSTRACT

Most previous research into the job-shop scheduling problem has concentrated on finding a single optimal solution (e.g., makespan), even though the actual requirement of most production systems requires multi-objective optimization. The aim of this paper is to construct a particle swarm optimization (PSO) for an elaborate multi-objective job-shop scheduling problem. The original PSO was used to solve continuous optimization problems. Due to the discrete solution spaces of scheduling optimization problems, the authors modified the particle position representation, particle movement, and particle velocity in this study. The modified PSO was used to solve various benchmark problems. Test results demonstrated that the modified PSO performed better in search quality and efficiency than traditional evolutionary heuristics.

© 2009 Elsevier Ltd. All rights reserved.

### 1. Introduction

The job-shop scheduling problem (JSP) has been studied for more than 50 years in both academic and industrial environments. Jain and Meeran (1999) provided a concise overview of JSPs over the last few decades and highlighted the main techniques. The JSP is the most difficult class of combinatorial optimization. Garey, Johnson, and Sethi (1976) demonstrated that JSPs are non-deterministic polynomial-time hard (NP-hard); hence we cannot find an exact solution in a reasonable computation time. The single-objective JSP has attracted wide research attention. Most studies of single-objective JSPs result in a schedule to minimize the time required to complete all jobs, i.e., to minimize the makespan ( $C_{max}$ ). Many approximate methods have been developed to overcome the limitations of exact enumeration techniques. These approximate approaches include simulated annealing (SA) (Lour-enço, 1995), tabu search (Nowicki & Smutnicki, 1996; Pezzella & Merelli, 2000; Sun, Batta, & Lin, 1995) and genetic algorithms (GA) (Bean, 1994; Gonçalves, Mendes, & Resende, 2005; Kobayashi, Ono, & Yamamura, 1995; Wang & Zheng, 2001). However, real-world production systems require simultaneous achievement of multiple objective requirements. This means that the academic concentration of objectives in the JSP must be extended from single to multiple. Recent related JSP research with multiple objectives is summarized as below.

Ponnambalam, Ramkumar, and Jawahar (2001) has offered a multi-objective GA to derive optimal machine-wise priority dispatching rules for resolving job-shop problems with objective

functions that consider minimization of makespan, total tardiness, and total machine idle time. Ponnambalam's multi-objective genetic algorithm (MOGA) has been tested with various published benchmarks, and is capable of providing optimal or near-optimal solutions. A Pareto front provides a set of best solutions to determine the tradeoffs between the various objects, and good parameter settings and appropriate representations can enhance the behavior of an evolution algorithm. Esquivel, Ferrero, and Gallard (2002) studied the influence of distinct parameter combinations as well as different chromosome representations. Initial results showed that:

- (i) larger numbers of generations favor the building of a Pareto front because the search process does not stagnate, even though it may be rather slow,
- (ii) Multi-recombination helps to speed the search and to find a larger set size when seeking the Pareto optimal set, and
- (iii) operation-based representation is better than priority-list and job-based representation selected for contrast under recombination methods.

The Pareto archived simulated annealing (PASA) method, a meta-heuristic procedure based on the SA algorithm, was developed by Suresh and Mohanasndaram (2006) to find non-dominated solution sets for the JSP with the objectives of minimizing the makespan and the mean flow time of jobs. The superior performance of the PASA can be attributed to the mechanism it uses to accept the candidate solution. Candido, Khator, and Barcia (1998) addressed JSPs with numbers of more realistic constraints, such as jobs with several subassembly levels, alternative processing plans for parts and alternative resources of operations, and the

\* Corresponding author. Tel.: +886 937808216.

E-mail address: [hsinhung@gmail.com](mailto:hsinhung@gmail.com) (H.-H. Lin).

requirement for multiple resources to process an operation. The robust procedure worked well in all problem instances and proved to be a promising tool for solving more realistic JSPs. Lei and Wu (2006) first designed a crowding-measure-based multi-objective evolutionary algorithm (CMOEA) makes use of the crowding-measure to adjust the external population and assign different fitness for individuals. Compared to the strength Pareto evolutionary algorithm, CMOEA performs well in job-shop scheduling with two objectives including minimization of makespan and total tardiness.

One of the latest evolutionary techniques for unconstrained continuous optimization is particle swarm optimization (PSO) proposed by Kennedy and Eberhart (1995). PSO has been successfully used in different fields due to its ease of implementation and computational efficiency. Even so, application of PSO to the combination optimization problem is rare. Coello, Plido, and Lechga (2004) provided an approach in which Pareto dominance is incorporated into PSO to allow the heuristic to handle problems with several object functions. The algorithm uses a secondary repository of particles to guide particle flight. That approach was validated using several test functions and metrics drawn from the standard literature on evolutionary multi-objective optimization. The results show that the approach is highly competitive. Liang, Ge, Zho, and Guo (2005) invented a novel PSO-based algorithm for JSPs. That algorithm effectively exploits the capability of distributed and parallel computing systems, with simulation results showing the possibility of high-quality solutions for typical benchmark problems. Lei (2008) presented a PSO for the multi-objective JSP to minimize makespan and total job tardiness simultaneously. Job-shop scheduling can be converted into a continuous optimization problem by constructing the corresponding relationship between a real vector and a chromosome obtained using the priority rule-based representation method. The global best position selection is combined with crowding-measure-based archive maintenance to design a Pareto archive PSO. That algorithm is capable of producing a number of high-quality Pareto optimal scheduling plans.

Hybrid algorithms that combine different approaches to build on their strengths have led to another branch of research. Wang and Zheng (2001) combined GA with SA in a hybrid framework, in which the GA was introduced to present a parallel search architecture, and SA was used to increase the probability of escape from local optima at high temperatures. Computer simulation results showed that the hybrid strategy was very effective and robust, and could find optima for almost all benchmark instances. Xia and Wu (2005) developed an easily implemented approach for the multi-objective flexible JSP based on the combination of PSO and SA. They demonstrated that their proposed algorithm was a viable and effective approach to the multi-objective flexible JSP, especially for large-scale problems. Ripon (2007) extended the idea in the jumping genes genetic algorithm, a hybrid approach capable of searching for near-optimal and non-dominated solutions with better convergence by simultaneously optimizing criteria.

Previous literature indicates that there has been little study of the JSP with multiple objectives. In this study, we use a new evolutionary PSO technique to solve the JSP with multiple objectives.

## 2. Job-shop scheduling problem

A typical JSP can be formulated as follows. There are  $n$  jobs to be processed through  $m$  machines. Each job must pass through each machine once and only once. Each job should be processed through the machines in a particular order, and there are no precedence constraints among the different job operations. Each machine can perform only one job at a time, and it cannot be interrupted. In

addition, the operation time is fixed and known in advance. The objective of the JSP is to find a schedule to minimize the time required to complete all jobs, that is, to minimize the makespan  $C_{max}$ . In this study, we attempt to attain the three objectives (i.e., minimizing makespan, machine idle time, and total tardiness) simultaneously. We formulate the multi-objective JSP using the following notation:

- $n$  is the total number of jobs to be scheduled,
- $m$  is the total number of machines in the process,
- $t(i, j)$  is the processing time for job  $i$  on machine  $j$  ( $i = 1, 2, \dots, n$ ), ( $j = 1, 2, \dots, m$ ),
- $L_i$  is the lateness of job  $i$ ,
- $\{\pi_1, \pi_2, \dots, \pi_n\}$  is the permutation of jobs.

The objectives considered in this paper are formulated as follows:

Completion time (makespan)  $C(\pi, j)$

$$C(\pi_1, 1) = t(\pi_1, 1) \tag{1}$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + t(\pi_i, 1) \quad i = 2, \dots, n \tag{2}$$

$$C(\pi_1, j) = C(\pi_1, j - 1) + t(\pi, j) \quad j = 2, \dots, m \tag{3}$$

$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j - 1)\} + t(\pi_i, j) \quad i = 2, \dots, n; \tag{4}$$

$$j = 2, \dots, m$$

$$\text{Makespan, } f_{C_{max}} = C(\pi_n, m) \tag{5}$$

$$\text{Total tardiness, } f_{total\ tardiness} = \sum_{i=1}^n \max\{0, L_i\} \tag{6}$$

$$\text{Total idle time, } f_{total\ idle\ time} = \{C(\pi_1, j - 1) + \sum_{i=2}^n \{\max\{C(\pi_i, j - 1) - C(\pi_{i-1}, j), 0\}\} | j = 2 \dots m\} \tag{7}$$

## 3. PSO background

PSO is based on observations of the social behavior of animals, such as birds in flocks or fish in schools, as well as on swarm theory. The population consisting of individuals or particles is initialized randomly. Each particle is assigned with a randomized velocity according to its own movement experience and that of the rest of the population. The relationship between the swarm and particles in PSO is similar to the relationship between the population and chromosomes in a GA.

In PSO, the problem solution space is formulated as a search space. Each particle position in the search space is a correlated solution to the problem. Particles cooperate to determine the best position (solution) in the search space (solution space).

Suppose that the search space is  $D$ -dimensional and there are  $\rho$  particles in the swarm. Particle  $i$  is located at position  $X^i = \{x_1^i, x_2^i, \dots, x_D^i\}$  and has velocity  $V^i = \{v_1^i, v_2^i, \dots, v_D^i\}$ , where  $i = 1, 2, \dots, \rho$ . Based on the PSO algorithm, each particle move towards its own best position ( $pbest$ ), denoted as  $Pbest^i = \{pbest_1^i, pbest_2^i, \dots, pbest_n^i\}$ , and the best position of the whole swarm ( $gbest$ ) is denoted as  $Gbest = \{gbest_1, gbest_2, \dots, gbest_n\}$  with each iteration. Each particle changes its position according to its velocity, which is randomly generated toward the  $pbest$  and  $gbest$  positions. For each particle  $r$  and dimension  $s$ , the new velocity  $v_s^r$  and position  $x_s^r$  of particles can be calculated by the following equations:

$$v_s^r(\tau) = w \times v_s^r(\tau - 1) + c_1 \times rand_1 \times [pbest_s^r(\tau - 1) - x_s^r(\tau - 1)] + c_2 \times rand_2 \times [gbest_s^r(\tau - 1) - x_s^r(\tau - 1)] \tag{8}$$

$$x_s^r(\tau) = x_s^r(\tau - 1) + v_s^r(\tau - 1) \tag{9}$$

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات