Expert Systems with Applications 38 (2011) 9248–9255

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# An adaptive annealing genetic algorithm for the job-shop planning and scheduling problem

Min Liu *, Zhi-jiang Sun, Jun-wei Yan, Jing-song Kang *

*School of Electronic and Information Engineering, Tongji University, Shanghai 201804, China*

ARTICLE INFO

*Keywords:*
Adaptive annealing genetic algorithm
Genetic algorithm
Job-shop planning
Scheduling

ABSTRACT

The genetic algorithm, the simulated annealing algorithm and the optimum individual protecting algorithm are based on the order of nature, and there exist some application limitations on global astringency, population precocity and convergence rapidity. An adaptive annealing genetic algorithm is proposed to deal with the job-shop planning and scheduling problem for the single-piece, small-batch, custom production mode. In the AAGA, the adaptive mutation probability is included to improve upon the convergence rapidity of the genetic algorithm, and to avoid local optimization, the Boltzmann probability selection mechanism from the simulated annealing algorithm, which solves the population precocity and the local convergence problems, is applied to select the crossover parents. Finally, the AAGA-based job-shop planning and scheduling problem is discussed, and the computing results of AAGA and GA are depicted and compared.



## 1. Introduction

With increasing customer demand and the globalization of the market, more and more enterprises make the products using the single-piece, small-batch, custom production mode (SPSBCP) (Sohlenius, 1992), which requires that the product design, process planning, job-shop planning and scheduling be re-designed for every order.

The integrated system model of Computer Aided Process Planning and Production Planning & Scheduling based on distributed and dynamic process planning solves the integration and concurrence problems of CAPP and PPS in SPSBCP mode (Liu, Bai, & Zhang, 2004). In job-shop planning, the replacing scenario, such as changing operation process and adjusting equipments, should be taken into account to finish the daily job smoothly, and the manufacturing jobs that require special equipment should be scheduled properly to assure that the key resources have no latency time when job-shop planning is completed.

A wide literature base has been published on production scheduling, focusing mostly on scheduling for various types of production systems at the shop floor or assembly-line level, such as job-shop scheduling (Adibi, Zandieh, & Amiri, 2010; Guo, Wong, Leung, Fan, & Chan, 2006; Sha & Lin, 2010; Zhang, Gao, & Shi, 2011), flow-shop scheduling (Chiang, Cheng, & Fu, 2011; Khademi Zare & Fakhrzad, 2011; Yagmahan & Yenisey, 2010), machine scheduling (Baek & Yoon, 2002; Balin, 2011; Behnamian, Zandieh, & Fatemi Ghomi, 2009), assembly-line scheduling (Guo, Wong, Leung, Fan, & Chan, 2008; Zhang, Kano, & Kyoya, 2000) and order scheduling (Ashby & Uzsoy, 1995; Axsater, 2005; Chen & Pundoor, 2006; Guo, Wong, & Leung, 2008).

The genetic algorithm, the simulated annealing algorithm and the optimum individual protecting algorithm are based on the order of nature and have been applied to solve complex combination optimization problems (Fogel, 1994) and intelligent control problems (Maniezzo, 1994) because of their higher steadiness and global optimization (Holland, 1975). However, there are some application limitations, such as global astringency, population precocity and convergence rapidity (Qi & Palmieri, 1994; Rudolph, 1994; Zhang, Xu, & Liang, 1997). Sirag put forth the unified thermodynamic operator (Sirag & Weisser, 1987), Bosesniuk applied the Boltzmann-, Darwin- and Heackel- strategies in the optimization problems (Bosesniuk & Ebeling, 1990), and Golberg provided the parallel simulated annealing algorithm (Golberg & Mahfoud, 1992) to improve those algorithms.

The parallel variable neighborhood search (PVNS) algorithm uses various neighborhood structures that carry the responsibility of making changes in the assignment and sequencing of operations for generating neighboring solutions (Yazdani, Amiri, & Zandieh, 2010). Pezzella, Morganti, and Ciaschetti (2008) propounded a GA for flexible flow-shop scheduling based on the integrated approach, in which a mix of different strategies for generating the initial population, selecting individuals for reproduction, and reproducing new individuals is presented. Gao, Sun, and Gen (2008) studied flow-shop scheduling with three objectives: min (minimum) makespan, min maximal machine workload, and min

* Corresponding authors.
*E-mail address:* Lmin@tongji.edu.cn (M. Liu).

0957-4174/$ - see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.01.136

total workload. They developed a hybrid genetic algorithm based on the integrated approach for this problem. Pan proposed a hybrid genetic algorithm to solve the no-wait job-shop scheduling problem (Pan & Huang, 2009). The variable neighborhood search (VNS) (Amiri, Zandieh, Yazdani, & Bagheri, 2010) is a modern meta-heuristic based on systematic changes of the neighborhood structure within a search to solve combinatorial optimization problems.

In this paper, the adaptive annealing genetic algorithm (AAGA) is proposed to solve the job-shop planning and scheduling problems for the single-piece, small-batch, custom production mode. AAGA adopts the population diversity, the adaptive mutation probability and the Boltzmann probability selection mechanism to improve the convergence rapidity of the genetic algorithm. The remainder of this paper is organized as follows. Section 2 describes the AAGA. The adaptive mutation probability and the Boltzmann probability selection mechanism are included in the AAGA, and then the global astringency of AAGA is proven. Section 3 describes the job-shop planning and scheduling model of the SPSBCP production mode. Section 4 designs the adaptive annealing genetic algorithm for the job-shop planning and scheduling problems. Section 5 compares the results of the GA and AAGA. Section 6 summarizes the contributions of this work and outlines some future considerations.

## 2. Adaptive annealing genetic algorithm

In the genetic algorithm, the mutation probability is fixed for the entire optimum process of the domain in question, which results in local optimization and increases the searching time. The adaptive mutation probability is provided to improve the convergence rapidity of the genetic algorithm, and the Boltzmann probability selection mechanism is included to select the crossover parents and to assure the population variety. The adaptive mutation probability-based and the Boltzmann probability-based AAGA meet the global astringency requirement.

### 2.1. Some concepts related to the AAGA

#### 2.1.1. Crossover, mutation and selection

$|X^{n0}|$ and $|X^{n1}|$ denote the element numbers of the space $X^{n0}$ and the space $X^{n1}$, respectively, as below:

$$X^{n0} = \{S^{(1)}, \ldots, S^{(i)}, S^{(i+1)}, \ldots, S^{(X^{n0})}\},$$
$$X^{n1} = \{P^{(1)}, \ldots, P^{(i)}, P^{(i+1)}, \ldots, P^{(X^{n1})}\},$$

where $S^{(i)}(i = 1,2,\ldots,|X^{n0}|)$ is the element of the parent population space $X^{n0}$,$P^{(j)}(j = 1,2,\ldots,|X^{n1}|)$ is the element of the crossover population space $X^{n1}$, $P^{(m)}(m = 1,2,\ldots,|X^{n1}|)$ is the element of the mid population space $X^{n1}$, $C_{ij}(k)$ is the crossover probability from $S^{(i)}$ into $P^{(j)}$ in step $k$, $M_{jm}(k)$ is the mutation probability from $P^{(j)}$ into $P^{(m)}$, and $S^{(i)}_{mq}(k)$is the transfer probability from the midst population $P^{(m)}$ into a new parent population $S^{(q)}$ when the parent population is $S^{(i)}$.

According to the definition of crossover, mutation and selection in the genetic algorithm, $C_{ij}(k)$, $M_{jm}(k)$, and $S^{(i)}_{mq}(k)$ are subject to:

$$\sum_{j=1}^{|X^{n1}|} C_{ij}(k) = 1 \quad i = 1,2,\ldots,|X^{n0}|, \quad k = 0,1,2,\ldots \tag{1}$$

$$M_{jm}(k) = \prod_{i=1}^{n1} (P_M(k))^{h_1}(1 - P_M(k))^{1-h_1} \quad m,j = 1,2,\ldots,|X^{n1}|, \quad k = 0,1,2,\ldots \tag{2}$$

$$S^{(i)}_{mq}(k) = \begin{cases} \prod_{x_i \in S^{(q)}} J_k(f_i) \Big/ \left( \sum_{x_j \in P^m} J_k(f_j) \right)^{n0} & S^q \subset P^m, \\ 0 & \text{other}, \end{cases} \tag{3}$$

where $h_1,h_2,\ldots,h_{n1}$ are the Hamming distances between the individual $P^{(j)}$ and $P^{(m)}$, and $P_m(k)$ is the mutation probability in step $k$.

#### 2.1.2. Adaptive mutation probability

In the genetic algorithm, the mutation probability is fixed for entire optimum process of the domain in question, and the adaptive mutation probability is given based on the mutation probability method advised by Doctor Jin-tao MA (Ma, 1995).

At the beginning of the mutation process, the mutation probability $P_m(k)$ is supposed to be a larger value to spur the individual mutation, whose fitness value is small. The $P_m(k)$ value is decreased to restrain the individual mutation to improve the computing speed and widen the searching scale when the result is near the optimum. The adaptive mutation probability is computed as:

$$P^i_m(k) = \begin{cases} P_M & f_i \geqslant f_a, \\ P_M\left(1 + EXP\left(\eta \frac{f_a - f_i}{f_a}\right) EXP(-k)\right) & \text{other}, \end{cases} \tag{4}$$

where $P^i_m(k)$ is the mutation probability of the $i$th individual at the $k$ iteration, $f_a$ is the average fitness value, $f_i$ is the fitness value of the $i$th individual $x_i$, $k$ is the iteration number, $P_M$ is the initialization of mutation probability, and $\eta$ is a constant.

#### 2.1.3. Whole annealing selection

The new parent population $F_k$ is generated randomly and independently into $N$ individuals from the population $P_k$ according to proportion in the genetic algorithm. If the repeat selection is permitted, the selection probability of $x_i \in P_k$ is calculated as

$$P(x_i) = J_k(f_i) \Big/ \sum_{x_j \in P_k} J_k(f_j),$$

where $J(f(x))$ is a fitness function, $J : R \to R$ is a rigorous increasing function, and $J(f(x)) > 0$.

The Boltzmann probability selection mechanism in the annealing algorithm is imported into GA, and the filial generation producing method under the whole annealing selection is calculated as:

$$P(x_i) = e^{f_i/T_k} \Big/ \sum_{x_j \in P_k} e^{f_i/T_k}, \tag{5}$$

where $T_k$ is the annealing temperature close to 0, and formula (5) is the selection probability under the whole annealing selection. The corresponding fitness function is calculated as:

$$J_k(f(x)) = e^{f(x)/T_k}. \tag{6}$$

### 2.2. Global astringency of the AAGA

In the AAGA, the Boltzmann probability selection mechanism is imported into the GA from the annealing algorithm. That is to say, the fitness function changes to formula (6) when a set $F_k$ is generated from another set $P_k$ according to probability formula (5). Thus, formula (3) in GA will change to:

$$S^{(i)}_{mq}(k) = \begin{cases} \prod_{x_i \in S^{(q)}} e^{f_i/T_k} \Big/ \left( \sum_{x_j \in P^m} e^{f_i/T_k} \right)^{n0} & S^{(q)} \subset P^m, \\ 0 & \text{other}. \end{cases} \tag{7}$$

To simplify this problem, given that the mutation probability $P_M(k)$ and the crossover mode are not changed along with $k$ and that $G(T_k) = (G_{iq}(k))$, where: