

文章编号:1006-3080(2012)04-0482-06

## 量子进化算法在柔性作业车间调度问题中的应用

张建明<sup>1,2</sup>, 顾幸生<sup>1</sup>

(1. 华东理工大学自动化研究所, 上海 200237; 2. 浙江理工大学数学系, 杭州 310018)

**摘要:**针对柔性作业车间调度完工时间最小化问题,提出了一种基于量子计算的量子进化算法。根据柔性作业车间调度问题的特点,设计出基于工序编码和基于机器编码的量子编码及解码方法。引入动态旋转角策略和跳跃基因算子,并通过实例验证了算法的有效性。

**关键词:**柔性作业车间调度; 旋转角; 跳跃基因算子

**中图分类号:**TP301

**文献标志码:**A

## Quantum Evolutionary Algorithm for Flexible Job-Shop Scheduling Problems

ZHANG Jian-ming<sup>1,2</sup>, GU Xing-sheng<sup>1</sup>

(1. Institute of Automation, East China University of Science and Technology, Shanghai 200237, China; 2. Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** In this paper, a quantum evolutionary algorithm is proposed for flexible job-shop scheduling problems with the objective to minimize the makespan. Aiming at the features of the flexible job-shop scheduling problems, both the working-procedures-based encoding method and the machine-based decoding method are proposed. Moreover, dynamic rotation angle and jumping gens operator are utilized in the proposed algorithm. Finally, simulation results are provided to demonstrate the effectiveness and the applicability of the proposed algorithm.

**Key words:** flexible job-shop scheduling problem; rotation angle; jumping gens operator

柔性作业车间调度问题(Flexible Job-shop Scheduling Problem, FJSP)是经典作业车间调度问题(JSP)的扩展。在经典车间调度问题中,工件的每道工序只能在一台确定的机床上加工。而在柔性作业调度问题中,每道工序的加工有多台机器可以选择,并且选择不同的机器加工时间也不同。由于减少了对机器的约束,因此,柔性作业车间调度是比经典作业车间调度问题更为复杂的 NP-Hard 问题。同时,柔性作业车间调度问题比经典车间调度问题具有更为现实的应用背景,更符合现实的制造车间

的调度需求,所以,柔性作业车间调度问题的研究具有重要的理论和现实意义。

经典作业车间调度问题是要对已分配工序在各台机器上的加工顺序进行排序,使得预先确定的目标最优。由于柔性作业车间调度问题减少了对机器的约束,因此,柔性作业调度问题同时还要解决路径分配问题,即把所加工的工序分配给可选机器以确定加工路线。目前,常用的柔性作业调度问题求解方法有分枝定界法、整数规划法、基于规则的启发式算法、拉格朗日松弛法、禁忌搜索算法、遗传算法、

收稿日期:2011-12-29

基金项目:国家自然科学基金项目(61174040, 61104178),中央高校基本科研业务费资助

作者简介:张建明(1972-),男,陕西延川人,副教授,博士生,研究方向为微分方程与智能计算。E-mail:jmzhang48@hotmail.com

通讯联系人:顾幸生, E-mail:xsgu@ecust.edu.cn

模拟退火算法等。由于分枝定界法、整数规划法等精确算法只能求解小规模问题,因此,基于规则的启发式算法<sup>[1]</sup>、模拟退火算法<sup>[2]</sup>、遗传算法<sup>[3-5]</sup>、粒子群算法<sup>[6-7]</sup>、禁忌搜速算法<sup>[8]</sup>、蚁群算法<sup>[9]</sup>等近似方法能较快地得到问题的较优解,以满足实际问题的需要。目前求解 FJSP 的近似方法大致可以分为两类:分级法和集成法。分级法的原理是基于每道工序的加工机器确定后,FJSP 就转化为经典 JSP;集成法是将分配工序机器和确定工序方面的研究,同时考虑在机器上的加工顺序<sup>[10]</sup>。

量子力学是 20 世纪人类最伟大的科学成就之一,尽管目前人们对量子力学原理的理解和解释还存在不同的看法,但它的正确性是不容置疑的。诺贝尔物理学奖获得者 Feynman 曾经指出:量子力学的精妙之处在于引入了几率幅的概念。一旦用量子态来表示信息就实现了信息的量子化。量子进化算法是建立在量子态矢量表示基础之上,将量子比特的几率幅表示应用于染色体的编码,使一条染色体可以表示多个态的叠加,并利用量子门实现染色体的更新,从而实现对目标的优化求解。由于量子算法的天然并行性与量子态纠缠、叠加和相干等特性,从本质上改变了传统的计算理念,因此,近年来将量子计算与传统智能计算相结合,构造计算速度更快、优化性能更好的智能计算新方法成为当前智能优化的热点问题之一。1996 年,Narayanan 等<sup>[11]</sup>将量子理论引入遗传算法,它以量子理论和量子计算为基础,采用量子比特实现染色体编码,通过量子门对其进行更新。其主要优点是能用小规模的种群实现较大空间的搜索,同时由于量子并行性是量子算法的基本特征,即对于函数可以同时计算不同  $x$  处的值,因而量子遗传算法有更高的效率,被广泛应用于寻优问题中。后来,许多学者对量子遗传算法做了大量的改进工作,将量子算法与粒子群算法<sup>[12-14]</sup>、免疫算法<sup>[15]</sup>、模拟退火算法<sup>[16]</sup>、人工神经网络<sup>[17]</sup>等智能优化算法相结合提出了许多计算性能良好的量子混合进化算法。在流水车间调度及随机作业车间调度方面,Wang<sup>[18]</sup>及 Gu<sup>[19-20]</sup>利用量子遗传算法取得了良好的结果,进一步说明了量子智能算法在生产调度方面同样具有广阔的研究及应用价值。

目前,将量子遗传算法用来解决柔性作业车间调度问题鲜见报道。本文设计了适合于柔性作业调度问题特点的量子遗传算法编码和相应的种群个体的进化及跳跃基因算子,通过基准柔性作业车间调度问题的实例验证了该算法的有效性。

## 1 柔性作业车间调度问题的描述

柔性作业车间调度问题可描述为: $n$  个工件在  $m$  台机器上加工, $n$  个工件的集合表示为  $J = \{J_1, J_2, \dots, J_n\}$ ,  $m$  台机器的集合表示为  $M = \{M_1, M_2, \dots, M_m\}$ 。每个工件  $J_i (1 \leq i \leq n)$  有  $n_i$  道工序  $O_{ij} (1 \leq i \leq n, 1 \leq j \leq n_j)$  需要加工。每道工序可以在若干台机器上加工,选择不同的加工机器所需的加工时间也不同,并且每道工序必须按可行的工艺次序进行加工。工序  $O_{ij}$  可选择的机器集合用  $M_{ij} = \{M_{ij1}, M_{ij2}, \dots, M_{ijk_{ij}}\}$  表示。如果  $k = m$ ,则称为完全柔性作业车间调度问题;如果  $k < m$ ,则称为局部柔性作业车间调度问题。调度的目的是为每道工序选择最合适的机器、确定每台机器上各工件、工序的最佳加工顺序及开工时间,使系统的某些性能指标达到最优。同时,在加工过程中还需要满足如下条件:

- (1)同一时刻同一台机器只能加工一个工件;
- (2)每个工件在某一时刻只能在一台机器上加工,每道工序一旦开工将不能中断;
- (3)不同工件的工序没有先后约束,同一工件的工序之间必须按工艺次序进行加工;
- (4)所有工件和机器都从 0 时刻开始调度。

## 2 基于量子遗传算法的柔性作业车间调度问题

### 2.1 量子遗传算法的编码与解码

对于经典车间调度问题,目前应用较多的是基于工序的编码,它的优点是对于任意交叉置换染色体总能得到可行调度,能够避免死锁、对解空间表征具有完全性以及解码能得到主动调度。但对于柔性作业车间调度来说,不仅要确定工序的加工顺序,还要为每道工序分配加工机器,因此,仅采用基于机器的编码无法满足柔性作业车间调度问题的要求。本文根据柔性作业车间调度问题的特点提出基于量子遗传算法的染色体编码方法,通过不同的解码方法,把同一染色体解码为基于工序的编码,用来确定工序的加工顺序;同时解码为基于机器的编码,用来确定每道工序的加工机器。对于  $n$  个工件、 $m$  台机器的柔性作业车间调度问题,量子编码的染色体长度  $L = ([\log_2^n] + 1) \times n \times m$ ,  $[x]$  表示取不大于  $x$  的最大整数。由于长度为  $([\log_2^n] + 1)$  的二进制串所能表示的十进制数不会小于  $n$  个,因此,量子编码的染

色体表示为

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_L \\ \beta_1 & \beta_2 & \cdots & \beta_L \end{bmatrix}$$

其中 $|\alpha_i|^2+|\beta_i|^2=1,i=1,2,\cdots,L$ 。对量子位进行测量,使之对应的量子位塌陷到确定的“1”或“0”状态,这样就可以得到一个长度为 $L$ 的由“0”和“1”组成的二进制串。每 $(\lceil\log_2^m\rceil+1)$ 个二进制串转化为1个十进制数,这样可以获得1个长为 $m\times n$ 的十进制串。将十进制串按大小顺序排序,同时保持每个数在十进制串中的相对位置不变,最小的前 $m$ 个数表示工件1,次小的 $m$ 个数表示工件2,以此类推,最后最大的 $m$ 个数表示工件 $n$ ,于是就得到了基于工序编码十进制串。

以2个工件、2台机器为例:

|      |  |
|------|--|
| 量子编码 | $\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 & \alpha_7 & \alpha_8 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & \beta_7 & \beta_8 \end{bmatrix}$ |
| 二进制串 | 0 0 1 1 0 1 0 1  |
| 十进制串 | 0 3 1 1  |
| 工序编码 | 1 2 1 2  |

对于基于机器的编码,以上得到的十进制串记为 $d_{11},d_{12},\cdots,d_{1m},d_{21},d_{22},\cdots,d_{2m},\cdots,d_{nm}$ ,则基于机器的编码为 $c_{11},c_{12},\cdots,c_{1m},c_{21},c_{22},\cdots,c_{2m},\cdots,c_{nm}$ ,其中 $c_{ij}=\text{mod}(d_{ij},k_{ij})+1(1\leq i\leq n,1\leq j\leq m)$ , $\text{mod}(r,s)$ 表示取 $r$ 除以 $s$ 的余数, $k_{ij}$ 表示第 $i$ 个工件的第 $j$ 道工序可选择的加工机器的台数。

以3个工件、3台机器,每个工件的每道工序有2台机器可以选择为例,得到基于机器的编码如下:  
十进制串 2 0 3 2 1 0 1 2 3  
基于机器的编码 1 1 2 1 2 1 2 1 2

这个基于机器的编码表示第1个工件的第1道工序在 $M_{112}$ 机器上加工,第1个工件的第2道工序在 $M_{121}$ 机器上加工,第2个工件的第1道工序在 $M_{212}$ 机器上加工,以此类推。

2.2 量子更新操作

量子门是实现量子进化算法进化操作的执行机构,其构造是设计算法的关键,直接影响着算法的性能。根据薛定谔方程,量子门应满足 $U(t)U(t)'=I$ 。量子门的设计有多种形式,有量子非门、受控非门、Hadamard门等<sup>[3]</sup>,最常用的是量子旋门,即

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$

其中: $(\alpha_i,\beta_i)$ 为第 $i$ 个量子比特编码; $\theta_i$ 为旋转角。

在量子旋转门中,量子旋转角的选择对量子进

化算法是至关重要的。量子旋转角的幅度影响收敛速度,如果其幅度太大,会导致早熟,幅度太小,收敛速度又会变得很慢。在众多的文献中,量子个体采用量子旋转门进行更新,旋转角都是固定的,也就是说在每代进化过程中旋转角是不变的。文献[21]给出了依据自变量为进化代数的负指数的自适应调整策略,实际上是一种转角的迭代步长单调下降的调整策略,没有考虑种群中各染色体的差异,也没有利用目标函数的差异。为此,李士勇等<sup>[22-23]</sup>提出了一种用于连续空间优化的基于实数编码和目标函数的梯度信息的双链量子遗传算法,充分利用了目标函数的梯度信息。由于调度问题的解是离散变量,本文用一阶差分代替梯度,定义转角步长函数为

$$\Delta\theta_{it} = \Delta\theta_0 \times \exp\left(-\frac{\Delta f(i) - \min \Delta f}{\max \Delta f - \min \Delta f}\right)$$

其中: $\Delta\theta_{it}$ 表示 $t$ 代种群中第 $i$ 个个体的旋转角迭代步长; $\Delta\theta_0$ 为转角初值; $\Delta f(i)$ 为第 $i$ 个代与子代个体的适应度函数的一阶差分; $\min \Delta f$ 与 $\max \Delta f$ 分别为父代种群与子代种群适应度函数的一阶差分的最小值与最大值。这样的转角步长调整策略使得目标函数变化较大时,适当减小转角步长,反之适当加大转角步长。量子进化中,量子旋转角的大小及方向见表1。表1中, $r_i$ 表示对第 $i$ 个量子位进行测量所得的值; $b_i$ 表示对最优个体的第 $i$ 量子位测量所得的值; $f(r)$ 表示量子个体适应度值; $f(b)$ 表示最优个体的适应度值; $\Delta\theta_i$ 表示第 $i$ 个量子位的旋转角; $s(\alpha_i,\beta_i)$ 用来确定旋转角的方向,即取-1表示顺时针旋转,取1表示逆时针旋转。

2.3 跳跃基因操作

跳跃基因是生物界中同一个染色体内或者不同染色体之间基因的水平传递现象,而通常的基因交叉是基因在父代个体与子代个体之间的垂直传递现象。虽然传统的遗传算法在各种优化问题中表现出良好的优化性能,但同时也容易陷入局部最优而导致早熟收敛。为保证种群的多样性从而避免早熟收敛,在遗传算法中引入跳跃基因算子<sup>[24]</sup>,即首先给定跳跃概率 $r$ ,随机产生一个0与1之间的随机数 $\text{rand}$ ,若 $\text{rand}<r$ ,则对该染色体执行跳跃基因操作;否则,不执行跳跃基因操作。跳跃基因操作通常有两种形式,一种是在同代染色体之间的剪切粘贴方式;另一种是在同一个染色体内的复制粘贴方式,即在染色体中随机截取一段基因片段复制到随机产生的迁移位置,本文采用后一种方式。

表 1 量子旋转角  
Table 1 Lookup table of rotation angle

| $r_i$ | $b_i$ | $f(r_i)>f(b)$ | $\Delta\theta_i$    | $s(\alpha_i,\beta_i)$ |                     |              |             |
|-------|-------|---------------|---------------------|-----------------------|---------------------|--------------|-------------|
|       |       |               |                     | $\alpha_i\beta_i>0$   | $\alpha_i\beta_i<0$ | $\alpha_i=0$ | $\beta_i=0$ |
| 0     | 0     | False         | $\Delta\theta_{ii}$ | -1                    | 1                   | 0            | 1 or -1     |
| 0     | 0     | True          | 0                   | 0                     | 0                   | 0            | 0           |
| 0     | 1     | False         | $\Delta\theta_{in}$ | 1                     | -1                  | 0            | 0           |
| 0     | 1     | True          | 0                   | 0                     | 0                   | 0            | 0           |
| 1     | 0     | False         | $\Delta\theta_{in}$ | -1                    | 1                   | 1 or -1      | 0           |
| 1     | 0     | True          | 0                   | 0                     | 0                   | 0            | 0           |
| 1     | 1     | False         | $\Delta\theta_{ii}$ | 1                     | -1                  | 0            | 1 or -1     |
| 1     | 1     | True          | 0                   | 0                     | 0                   | 0            | 0           |

2.4 灾变操作

为了保持种群的多样性,当在一定的代数内最优解没有得到更新时,认为陷入局部最优解。在保留最优解的基础上,随机生成新的种群。

2.5 算法流程

基于量子遗传算法的柔性作业车间调度问题的算法流程如下:

Begin

(1)参数初始化及种群初始化。初始化参数包括种群规模  $N$ 、最大代数  $T$ 、变异概率  $r$ 、灾变操作的代数  $CT$  等。初始化种群产生  $N$  个量子比特编码长为  $L$  的个体种群

$$Q_{0i} = \begin{bmatrix} \alpha_{1i} & \alpha_{2i} & \cdots & \alpha_{Li} \\ \beta_{1i} & \beta_{2i} & \cdots & \beta_{Li} \end{bmatrix}, i = 1, 2, \dots, N$$

(2)对初始种群个体进行测量,并评估每个个体的适应度值,将最优个体保存于变量  $B$

(3)While 不满足结束条件 do

$$t=t+1$$

利用量子旋转门对种群进行更新

执行跳跃基因算子

对种群进行量子测量,并评估每个个体适应值

If 连续一定的代数最优个体没有得到更新

执行灾变操作

end

end

由于量子编码的特殊性,为了能使该算法具有更好的局部搜索能力,对于每一代种群进行测量并评估每个个体时,都进行多次测量与评估。解码时先根据基于机器分配编码的基因串选择每道工序的加工机器,然后按基于工序编码的基因串确定每台机器上的工序顺序。但是确定每台机器上的工序顺序时,按照工序在该序列上的顺序进行解码,序列上

第 1 道工序首先安排加工,然后取序列上第 2 道工序,将其插入到对应机器上最佳可行的加工时刻安排加工,以此方式直到序列上所有工序都安排在其最佳可行的地方,这样所得到的调度都将为主动调度。

3 仿真实验

为了比较跳跃基因算法及动态旋转角对算法性能的影响,采用文献[25-26] 中的  $8 \times 8$  部分柔性调度问题及  $10 \times 10$  完全柔性调度问题进行测试。采用 Matlab 进行编程,程序运行环境为 Intel(R) Core(TM)2 Duo E7500 2.93CPU,2 G 内存。仿真实验结果见图 1~图 4。

图 1 为不同跳跃概率算法性能比较结果,运行参数为:种群规模  $P=300$ ,终止代数  $T=120$ ,转角迭代初值  $\Delta\theta_0=0.12\pi$ ,每代的测试次数  $w=20$ ,运行次数  $S=10$ 。从图 1 可以看出,选择恰当的跳跃概率可以有效地提高算法性能。图 2 为固定旋转角与动态旋转角算法性能比较结果,跳跃概率  $r=0.2$ ,其余运行参数与图1完全相同,可以看出采用

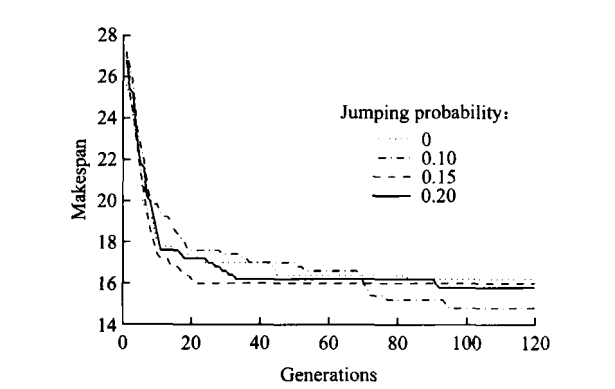


图 1 不同跳跃概率算法性能比较  
Fig. 1 Comparison for different jumping probability

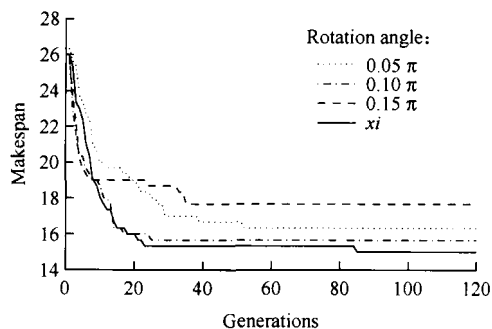


图2 固定旋转角与动态旋转角算法性能比较

Fig. 2 Comparison between invariant and dynamic rotation angle

动态旋转角能有效提高算法性能。图3为跳跃基因在固定旋转角情况下的算法性能比较结果,为了验证跳跃基因在固定旋转角情况下对算法性能的影响,选择种群规模  $N=300$ , 终止代数  $T=200$ , 固定旋转角  $\Delta\theta_0=0.05\pi$ , 从图3可以看出,在固定旋转角的情况下,跳跃基因依然可起到优化算法性能的作用。

图4为  $10\times 10$  完全柔性调度问题测试结果,其运行环境同上,采用固定旋转角  $\Delta\theta_0=0.16\pi$ , 对不同跳跃概率对算法性能进行比较。结果表明,不同的跳跃概率对算法的影响不同,当跳跃概率为0.05时,算法性能明显优于跳跃概率为零的情况,而跳跃概率为0.10,0.15及0.20时,算法性能不及跳跃概率为零的情况。以上的测试结果表明,动态调整旋转角的策略优于固定旋转角策略,恰当的跳跃概率可以提高算法的性能。跳跃基因算子和动态旋转角策略可以有效提高算法性能,主要原因是这两种策略能有效提高种群的多样性的同时增强了进化的目的性,从而提高算法的搜索能力。

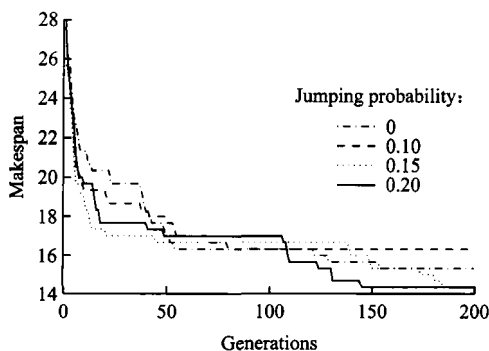


图3 跳跃基因在固定旋转角情况下的算法性能比较

Fig. 3 Comparison for different jumping probability

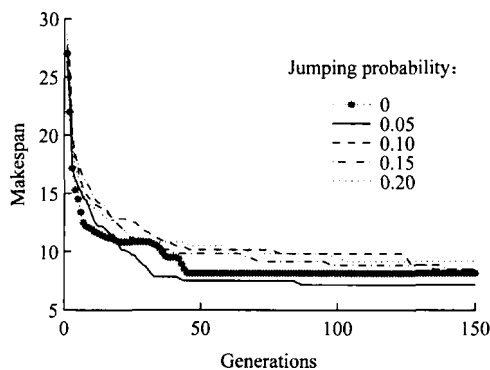


图4 固定旋转角下不同跳跃概率对算法性能影响比较

Fig. 4 Comparison for different jumping probability with invariant rotation angle

## 4 结 论

本文通过引入跳跃基因算子使得种群基因不但可以垂直传递同时也可以个体内水平传递,从而提高了种群的多样性;同时动态调整旋转角策略使得量子个体在进化时改变了绝大部分量子进化算法中旋转角步长没有考虑到个体适应值函数的状况,使得量子个体在进化过程中旋转角步长根据个体适应值实时调整,因此,提高了量子进化的目的性,从而可以有效提高算法性能。

## 参考文献:

- [1] Montazeri M, Van W. Analysis of scheduling rules for an FMS [J]. International Journal of Production Research, 1990, 28(4):785-802.
- [2] Najid N M, Dauzere-Pere S, Zaidat A. A modified simulated annealing method for flexible job-shop scheduling problem [C]// Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ: IEEE Press, 2002:6-12.
- [3] Chen H, Ihlow J, Lehmann C. A genetic algorithm for flexible job-shop scheduling [C]// IEEE International Conference on Robotics and Automation. Detroit: IEEE, 1999:1120-1125.
- [4] Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems [J]. IEEE Transaction on Systems, Man and Cybernetics: Part C, 2002, 32(1):1-13.
- [5] 张超勇, 饶运清, 李培根, 等. 柔性作业车间调度问题的两级遗传算法 [J]. 机械工程学报, 2007, 43(4):119-124.

- [6] Lian Zhigang. A united search particle swarm optimization algorithm for multiobjective scheduling problem[J]. Applied Mathematical Modelling, 2010, **34**(11):3518-3526.
- [8] Mastrolilli M, Gambardella L M. Effective neighborhood functions for the flexible job shop problem[J]. Journal of Scheduling, 2000, **3**(1):3-20.
- [9] 王万良,赵澄,熊婧,等.基于改进蚁群算法的柔性作业车间调度问题的求解方法[J].系统仿真学报,2008, **20**(16):4326-4329.
- [10] Xia Weijun, Wu Zhiming. An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problems[J]. Computers & Industrial Engineering, 2005, **48**(2):409-425.
- [11] Narayanan A, Moore M. Quantum inspired genetic algorithm [C]//IEEE International Conference on Evolutionary Computation. Nagoya, Japan: IEEE, 1996:61-66.
- [12] Sun Jun, Xu Wenbo, Feng Bin, et al. A global search strategy of quantum-behaved particle swarm optimization[C]//Cybernetics and Intelligent Systems, Proceedings of the 2004 IEEE Conference. USA: IEEE, 2004:111-116.
- [13] Sun Jun, Feng Bin, Xu Wenbo. Particle swarm optimization with particles having quantum behavior[C]//IEEE Proceedings of Congress on Evolutionary Computation. USA: IEEE, 2004:325-331.
- [14] 周殊,潘炜,罗斌,等.一种基于粒子群优化方法的改进量子遗传算法及应用[J].电子学报,2006, **34**(5):897-901.
- [15] 李映,张艳宁,张荣椿,等.免疫量子进化算法[J].西北工业大学学报,2005, **23**(4):537-543.
- [16] 张伟丰,王朝晖,郑建国.量子进化和模拟退火的混合优化算法[J].湖北汽车工业学院学报,2006, **20**(2):34-37.
- [17] Ahn J, Weinacht T C, Bucksbaum P H. Information storage and retrieval through quantum phase[J]. Science, 2000, **287**(5452):463-465.
- [18] Wang Ling, Wu Hao, Tang Fang, et al. A hybrid quantum-inspired genetic algorithm for flow shop scheduling[J]. Lecture Notes in Science, 2005, **3645**:636-644.
- [19] Gu Jinwei, Gu Xingsheng, GU Manzhao, et al. A novel parallel quantum genetic algorithm for stochastic job-shop scheduling [J]. Journal of Mathematical Analysis and Applications, 2009, **355**(1):63-81.
- [20] Gu Jinwei, Gu Manzhao, Cao Cuiwen, et al. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job-shop scheduling problem[J]. Computers and Operations Research, 2010, **37**(5):927-937.
- [21] Zhang Gexiang, Li Na, Jin Weidong, et al. A novel quantum genetic algorithm and its application[J]. ACTA Electronica Sinica, 2004, **32**(3):476-479.
- [22] Li Panchi, Li Shiyong. Quantum-inspired evolutionary algorithm for continuous spaces optimization[J]. Chinese Journal of Electronics, 2008, **17**(1):80-84.
- [23] 李士勇,李盼池.量子遗传算法及在模糊控制器参数优化中的应用[J].哈尔滨工业大学学报,2006, **38**(8):1216-1218.
- [24] Ripon K S N. Hybrid evolutionary approach for multi-objective job-shop scheduling problem[J]. Malaysian Journal of Computer Science, 2007, **20**(2):183-198.
- [25] Zhang Haipeng, Gen M. Multistage-based genetic algorithm for flexible job-shop scheduling problem[J]. Journal of Complexity International, 2005, **11**:223-232.
- [26] Kacem I, Hammadi S, Borne P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems[J]. IEEE Transactions on Systems, Man and Cybernetics, Part C, 2002, **32**(1):1-13.

# (上接第 481 页)

- [8] Sandoval-Basurto E A, Gosset G, Bolivar F, et al. Culture of *Escherichia coli* under dissolved oxygen gradients simulated in a two-compartment scale-down system: Metabolic response and production of recombinant protein [J]. Biotechnology and Bioengineering, 2005, **89**(4):453-463.
- [9] Oosterhuis N M G, Kossen N M F, Oliver A P C, et al. Scale-down and optimization studies of gluconic acid fermentation by *Gluconobacter oxydans* [J]. Biotechnology and Bioengineering, 1985, **27**(5):711-721.
- [10] De Leon A, Galindo E, Ramirez O T. Design and characterization of a one-compartment scale-down system for simulating dissolved oxygen tension gradients [J]. Journal of Chemical Technology and Biotechnology, 2010, **85**(7):950-956.
- [11] 宋翔,秦永峰,梁雪,等.谷氨酸发酵分批补料动力学模型的构建[J].发酵通讯科技,2009, **38**(3):10-11.
- [12] 云逢霖.氧的传递及供氧对谷氨酸发酵的影响(续)[J].发酵科技通讯,1981, **10**(2):56-63.
- [13] Enfors S O. Physiological responses to mixing in large scale bioreactors [J]. Journal of Biotechnology, 2001, **85**(2):175-185.
- [14] Bylund F. Substrate gradient formation in the large-scale bioreactor lowers cell yield and increases by-product formation [J]. Bioprocess Engineering, 1998, **18**(3):171-180.
- [15] Taymaz-Nikerel H, Gulik W M V, Heijnen J J. *Escherichia coli* responds with a rapid and large change in growth rate upon a shift from glucose-limited to glucose-excess conditions [J]. Metabolic Engineering, 2011, **13**(3):307-318.