

基于改进量子进化算法的作业车间调度研究

张建明

(浙江理工大学理学院, 310018 杭州)

摘要: 针对作业车间调度问题,以最大完工时间最小化为优化目标,提出了跳跃基因量子进化算法(JGQEA)。该算法在量子进化算法的基础上引入跳跃基因算子,同时采用动态调整量子旋转角策略以提高算法的搜索能力。通过仿真实验验证了算法的有效性,结果表明 JGQEA 优于 QEA 等几种进化算法。

关键词: 作业车间调度; 转换机制; 量子进化; 旋转角; 跳跃算子

中图分类号: TP18

文献标志码: A

0 引言

量子算法是以量子力学原理为基础,它最本质的特点是利用量子态的相干性和叠加性,以及量子比特之间的纠缠性,与经典算法最本质的区别在于量子并行性。因此,量子计算比经典计算在速度上会有指数级的提高。但基于量子力学原理的量子计算机还处于研制的初级阶段,因此,将量子计算与传统智能计算相结合的量子智能计算就运用而生。

1996 年 Narayanan 和 Moore 提出了量子遗传算法(quantum genetic algorithm, QGA),并应用于解决 TSP 问题^[1],开创了量子智能计算研究的新方向。2000 年,由 Han 等^[2]将量子位和量子门的概念引入进化算法,并用组合优化问题验证了算法的有效性。2002 年,Han 等^[3]在量子遗传算法的基础上,引入种群迁移机制,提出了量子衍生遗传算法(quantum inspired genetic algorithm, QIGA)。目前对 QGA 的改进方式主要包括算法结构、进化方式以及编码方法等,比如中国科技大学杨俊安等^[4]提出了一种多宇宙并行量子遗传算法,属于算法结构的改进;西南交通大学的张葛祥等^[5]提出的采用量子比特相位比较法更新量子门和自适应调整搜索网格策略,与西南交通大学的陈辉等^[6]提出的混沌

更新旋转门转角的量子遗传算法,都属于进化策略方面的改进;在编码方法上的改进有:清华大学王凌等^[7]给出的基于二进制编码的混合量子遗传算法和基于实数编码的混合量子遗传算法,李士勇等^[8-9]在求解连续优化问题时提出的基于实数编码和目标函数梯度信息的双链量子遗传算法,以及基于量子位 Bloch 球面的量子进化算法^[10]等。最近,在利用 QGA 对生产调度问题的研究中,Gu 等^[11-12]引入了量子交叉、变异等操作,获得了不错的优化效果。以上各种量子智能算法都是传统智能算法与量子算法相结合的产物。而跳跃基因算子在遗传算法中的成功实践,使我们有理由相信,提供了基因在种群内的水平传递的跳跃基因,有可能在量子算法中有利于种群多样性的保持,从而提高算法的优化性能。因此,本文针对作业车间调度问题,以最大完工时间最小为优化目标,在量子遗传算法中引入跳跃基因算子,提出跳跃基因量子遗传算法,以提高算法的优化性能。

1 作业车间调度问题的描述及数学模型

Job-shop 调度问题包括典型 Job-shop 调度和带并行机的 Job-shop 调度。Job-shop 调度问题是先进制造与自动化及其他调度技术相关研究领域中的

收稿日期: 2013-12-30

基金项目: 国家自然科学基金(10871181)

作者简介: 张建明(1972-),男,陕西延川人,博士、副教授,主要从事微分方程分支问题及智能计算方面的研究。

的经典热点问题。

生产系统有 n 个工件需要在 m 台机器上加工,调度的任务是把工序分配给机器上某个时间段,调度的目标就是确定每个机器上工件的加工顺序和每个工件的开始加工时间,使得完成所有工件所需的时间 Makespan 最小(加工周期或其他指标达到最优)。典型 Job-shop 调度问题的数学模型描述为^[13]:

$$\min\{\max\{\max c_{ik}\}\}.$$

满足约束条件:

$$c_{ik} - t_{ik} + \Omega(1 - a_{ijk}) \geq c_{ij}; i = 1, 2, \dots, n; j, k = 1, 2, \dots, m.$$

$$c_{jk} - c_{ik} + \Omega(1 - x_{ijk}) \geq t_{jk}; i, j = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

$$c_{ik} > 0; i = 1, 2, \dots, n; k = 1, 2, \dots, m.$$

其中 Ω 是一足够大的正数, c_{ik} 与 t_{ik} 分别表示第 i 个工件在第 k 台机器上的加工时间; a_{ijk} 与 x_{ijk} 为:

$$a_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 在机器 } j \text{ 上加工先于在机器 } k \text{ 上加工} \\ 0, & \text{工件 } i \text{ 在机器 } k \text{ 上加工先于在机器 } j \text{ 上加工} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 先于工件 } j \text{ 在机器 } k \text{ 上加工} \\ 0, & \text{工件 } j \text{ 先于工件 } i \text{ 在机器 } k \text{ 上加工} \end{cases}.$$

2 基于作业车间调度问题的量子进化算法

本节在量子遗传算法的基础上,引入跳跃基因算子,以提高优良个体性能在种群进化过程中的有效传递;在量子个体更新的过程中,引进新的量子角的选择模式,以保持种群在进化过程中的多样性,防止早熟收敛。

2.1 编码机制

在量子进化算法中,一个量子比特位可以表示为

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

其中 α, β 为复数,且满足归一化条件 $|\alpha|^2 + |\beta|^2 = 1$ 。

一个具有 l 位的量子个体就表示为:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_l \\ \beta_1 & \beta_2 & \cdots & \beta_l \end{bmatrix}.$$

其中 $|\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2, \dots, l$ 。在具有 n 个工件 m 台机器的作业车间调度问题中,我们取量子个体的长度为 $l = ([\log_2^n] + 1) \times n \times m$, 其中 $[x]$ 表示取整函数。由于 $[\log_2^n] + 1 \geq \log_2^n$, 因此,长度为 $[\log_2^n] + 1$ 的二进制串至少可以表示 n 个十进制数。所以,长度为 l 的二进制串足以表示作业车间调度的一个解。

2.2 解码机制

对于车间调度问题,我们的目的是在满足工艺

约束等条件下,合理安排各个工件在每台机器上的加工次序,使得最大完工时间等性能指标达到最优。而量子个体是通过概率幅对解的一种线性叠加态,因此,要通过解码机制,把线性叠加态的解解码为十进制的解。笔者在文献[10]中解码方法的基础上,提出以下解码方案。

第一步,设 $Q(t)$ 为 l 位的量子比特个 $Q(t) =$

$$\begin{bmatrix} \alpha_1^t & \alpha_2^t & \cdots & \alpha_l^t \\ \beta_1^t & \beta_2^t & \cdots & \beta_l^t \end{bmatrix}, \text{其中 } t \text{ 表示量子个体的代数。通常}$$

情况下为了使得每个解都有机会被搜索到,在初始种群中 α_i^0 与 $\beta_i^0 (i = 1, 2, \dots, l)$ 都取为 $\sqrt{2}/2$ 。

第二步,产生一个 0 与 1 之间的随机数 r , 若 $|\alpha_i^t|^2 > r^2$, 则令 $x_i(t) = 1$, 否则,令 $x_i(t) = 0 (i = 1, 2, \dots, l)$ 。因此,对每个 $Q(t)$ 可以得到一个长度为 l 的二进制串 $X(t) = (x_1^t, x_2^t, \dots, x_l^t)$ 。

第三步,在 $X(t)$ 中,将每 $[\log_2^n] + 1$ 个二进制串转化为十进制串,因此,可以得到长为 $m \times n$ 的十进制串 $D(t) = (d_1^t, d_2^t, \dots, d_{m \times n}^t)$ 。

第四步,将 $D(t)$ 中的数按从小到大排序,最小的 m 个数表示第一个工件,接下来的 m 个数表示第二个工件,依次类推。在此过程中保持每个数在 $D(t)$ 中的相对位置不变。因此,可以得到 n 个工件序号每个都重复 m 次的一个排列 $S(t)$ 。 $S(t)$ 中的第 k 次出现的数 i 将表示第 i 个工件的第 k 道工序。如果在 $D(t)$ 中出现相等的数字,则位置序号较小的数代表工序号较小的工件。

解码流程图如图 1 所示。

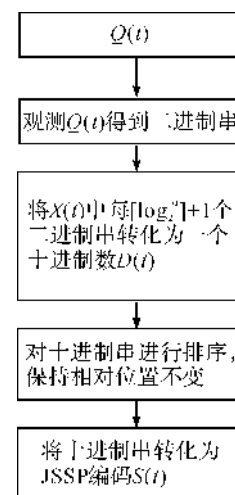


图1 编码及解码过程

以 3 个工件 2 台机器的 3×2 JSSP 调度问题为例:则 $Q(t)$ 为 12 位量子比特位的染色体,对 $Q(t)$ 进

行观测,若得到的12位二进制串 $X(t)=\{0,1,1,1,0,1,0,1,0,0,1,1\}$,则 $D(t)=(1,2,1,1,0,2)$,因此,位置1与位置5表示工件1;位置3与位置4表示工件2;位置2与位置6表示工件3。因此, $S(t)=(1,3,2,2,1,3)$,则位置1中的1表示第一个工件的第一道工序,位置5中的1表示第一个工件的第二道工序;位置3中的2表示第二个工件的第一道工序,依次类推。所以,任何一个量子个体都可以解码为一个可行调度,该方法的优点是解码方法简单,同时不会产生非法解。

2.3 量子旋转角

在QEA中,利用量子门进行量子个体的更新,而量子旋转门是被广泛采用的更新手段。在此过程中,量子旋转角的选择将起到至关重要的作用。对于旋转角,不仅要确定旋转角的大小还要确定旋转角的方向。旋转角 θ 和旋转方向可以通过表1得到,其中 r_i 为当前量子个体第 i 个量子比特位所观测到的二进制编码(0或1); b_i 为当前最优个体的第 i 量子比特位所观测到的二进制编码; $\Delta\theta_i$ 为旋转角的大小,控制算法收敛的速度; $s(\alpha_i, \beta_i)$ 为旋转角的方向,保证算法的收敛; $f(x)$ 为适应度函数。比如,当 $r_i=0, b_i=1, f(r)>f(b)$ 时,为使当前解收敛到一个具有更高适应度的量子个体,应增大当前解

取得0概率,即要使得 $|\alpha_i|^2$ 变大,此时,如果 (α_i, β_i) 在第一、三象限, θ 应顺时针方向旋转;如果 (α_i, β_i) 在第二、四象限, θ 应逆时针方向旋转,以增大 $|\alpha_i|^2$ 的值。因此,旋转角大小的选择对算法的收敛性能具有巨大的影响。在大部分量子进化算法或混合量子进化算法中,量子旋转角的大小总是通过查表给出,量子旋转角的大小往往是固定的。大小的选择通常也是根据具体问题由经验给出,与目标函数的本身没有太大的关系。尽管量子进化算法中个体的表示方式决定了量子种群比经典的进化算具有更强的多样性,但量子个体进化过程中 α_i 逐渐向0或1收敛而使种群失去多样性。为了保持种群的多样性,使得进化更具有目的性,本文采用动态调整旋转角策略。在进化初期由于 $|\alpha_i|^2$ 与 $|\beta_i|^2$ 都等于 $\frac{1}{2}$,为了使每个解都有被搜索到的可能,旋转角 θ 应小一些,以提高种群的探索能力;在进化中期种群会逐渐失去多样性,因此应增大旋转角 θ ;在进化后期为了提高种群的开发能力,增加种群的局部搜索能力,应再次减小旋转角 θ 。因此,笔者采用第一代到 $\frac{T}{2}$ (T 为最大进化代数)代内旋转角线性增加;在 $\frac{T}{2}$ 代之后旋转角线性减少。即:

表1 量子旋转角

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	$\Delta\theta_i$	-1	1	0	1 or -1
0	0	Ture	0	0	0	0	0
0	1	False	$\Delta\theta_i$	1	-1	0	0
0	1	Ture	0	0	0	0	0
1	0	False	$\Delta\theta_i$	-1	1	1 or -1	0
1	0	Ture	0	0	0	0	0
1	1	False	$\Delta\theta_i$	1	-1	0	1 or -1
1	1	Ture	0	0	0	0	0

$$\theta_t = \begin{cases} 2t(A-C)/T+C, & \text{当 } t < \frac{T}{2}, \\ -2(A-C)(t-T)/T+C, & \text{当 } t \geq \frac{T}{2}. \end{cases}$$

2.4 跳跃基因算子

目前,跳跃基因算子主要有两种形式:剪切粘贴置换与复制粘贴置换^[14]。这两种置换又分为在同一染色体内的置换与染色体之间的置换。同一染色体内的剪切粘贴置换就是在染色体内部随机产生一个剪切点和随机长度的跳跃基因,再随机产生一个新的粘贴点,将跳跃基因粘贴到新的粘贴点上,如图2

(a)。染色体间的剪切粘贴置换是在两个不同的染色体上都随机产生一个剪切点和随机长度的跳跃基因,同时每个染色体上随机产生一个新的粘贴点,然后将两个跳跃基因片段交叉粘贴到新的粘贴点处。染色体内的复制粘贴置换就是在染色体内部随机产生一个复制点和随机长度的跳跃基因,再随机产生一个新的粘贴点,将跳跃基因复制到新的粘贴点上,如图2(b)。在染色体间的复制粘贴置换是在两个不同的染色体上都随机产生一个复制点和随机长度的跳跃基因,同时每个染色体上随机产生一个新的粘贴点,然后将两个跳跃基因片段交叉复制到

新的粘贴点处。本文采用在同一染色体内的剪切粘贴置换,并把这种引入跳跃基因算子的量子遗传算法称为跳跃基因量子进化算法(jumping genes quantum evolutionary algorithm, JGQEA)。

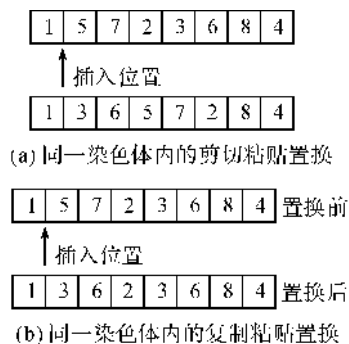


图2 同一染色体内的置换
针对JSSP的JGQEA伪代码为:

Begin

参数初始化

$t \leftarrow 0$

初始化种群 $Q(0)$

通过观测 $Q(0)$ 获得 $P(0)$

评价 $P(0)$

将 $P(0)$ 中的最优解保存至 Bp

While 终止条件不成立 Do

Begin

$t = t + 1$

通过观测 $Q(t)$ 获得 $P(t)$

评价 $P(t)$

将 $P(t)$ 与 Bp 中的最优解保存至 Bp

If 基因跳跃条件成立

实施跳跃操作

End

End

End

3 仿真结果与分析

为了验证 JGQEA 对 JSSP 的有效性,选取 Muth 与 Thompson 提出的经典的基准问题 mt06、mt10 及 mt20^[15-16] 和由 Applegate 与 Cook 提出的经典的基准问题 la01、la02、la03、la04、la05、la06、la21、la24、la33、la36 及 la38, 与 MA、GA、SGA、PSO 及 QEA 的优化效果进行比较。表 2 中 JGQEA 的测试参数为:种群规模为 300,跳跃概率为 0.1,最大代数 300。利用 Matlab 编程,测试环境为: Intel(R) Core 2 Duo 2.93GHz CPU, 2G 内存。表中 OPT 表示最优解, MA 表示基因算法(memetic algorithm)^[17], GA 表示遗传算法(genetic algorithm)^[18], SGA 表示 Nakan 与 Yamada 提出的遗传算法^[19], PSO 表示离子群算法(particle swarm optimization)^[20-21], QEA 表示普通量子进化算法。从表 2 可知,尽管 JGQEA 不是所有问题中都得到最优解,但所得到的最优解好于 GA、SGA、PSO、QEA 等方法获得的最优解,在 mt10、mt20、La38 中所获的最优解较 MA 算法获得的最优解差。由此可以说明, JGQEA 有较好的优化性能。在 QEA 与 JGQEA 的比较中可以发现,引入跳跃基因算子的 JGQEA 的优化性能比 QEA 都有所提高,这进一步说明跳跃基因算子可以改善算法的优化性能。为了比较不同的跳跃概率对算法收敛速度的影响,笔者以 La06 为例,选择不同的跳跃概率来比较它们的收敛速度,如图 3 所示。

表2 最优结果比较

基准问题	n	m	OPT	MA	GA	SGA	PSO	QEA	JGQEA
Mt06	6	6	55	55	55	55	55	55	55
Mt10	10	10	930	930	997	960	958	950	936
MT20	20	5	1 165	1 165	1 251	1 196	1 165	1 183	1 177
La01	10	5	666	666	666	666	666	666	666
La02	10	5	655	655	655	655	655	655	655
La03	10	5	957	957	957	957	957	957	957
La04	10	5	950	950	950	950	950	950	950
La05	10	5	593	593	593	593	593	593	593
La06	15	5	926	926	926	926	926	926	926
La21	15	10	1 040	1 055	1 180	1 156	1 047	1 132	1 056
La24	15	10	935	940	998	1 007	935	989	940
La33	30	10	1 719	1 719	1 719	1 719	1 719	1 719	1 719
La36	15	15	1 268	1 281	1 309	1 302	1 268	1 311	1 274
La38	30	10	1 196	1 216	1 231	1 228	1 196	1 227	1 231

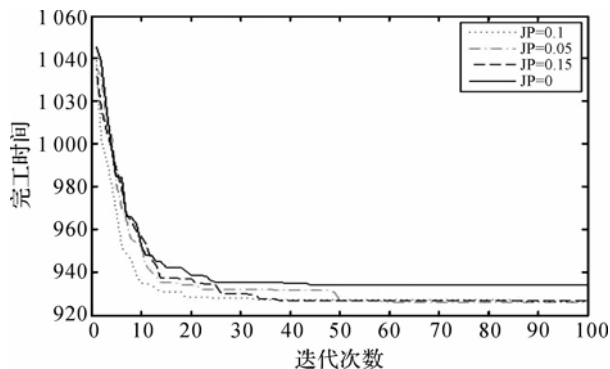


图3 不同跳跃概率下收敛速度比较(La06)

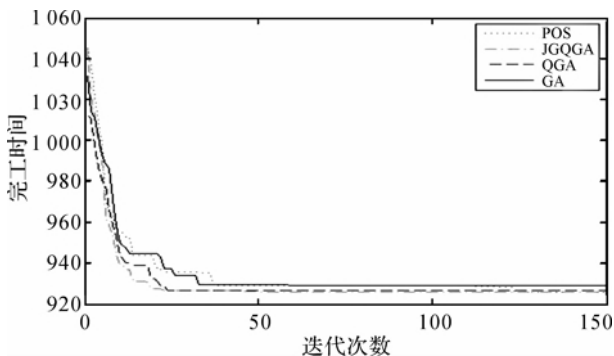


图4 对 La06 不同算法的比较

图3中,我们取 $A=0.03$, $C=0.01$;种群规模为300,最大代数100,运行次数为10次;跳跃概率分别取0、0.05、0.1、0.15;竖轴表示运行10次完工时间的平均值,横轴表示运行代数。从图中可以看出,当选择一个恰当的跳跃概率时,算法的收敛速度会得到有效的改善,特别是在跳跃概率分别取0.05、0.1、0.15时,其收敛速度都优于跳跃概率为零的情况。在图4中,以La06为模拟对象,将JGQGA与POS、QGA、GA的优化性能做了比较,结果显示JGQGA比这几种算法在优化质量上都有所提高。其中JGQGA中相关参数为:种群规模300,最大代数150, $A=0.03$, $C=0.01$,跳跃概率为0.1。

4 结论

本文提出了JGQEA用于解决JSSP问题。在JGQEA中采用动态调整量子旋转角的策略,同时引入跳跃基因,以改善量子种群的多样性,提高JGQEA的优化性能。对多个JSSP的基准问题以最大完工时间为优化目标进行优化调度,JGQEA所获地的最优值优于GA、SGA、PSO及QEA所获得的最优值,但是不及MA所获得的最优值。这一方面说明,JGQEA具有良好的优化性能,另一方面也说明JGQEA的优化性能还有进一步改进和提高

的要求。在优化目标方面,本文仅对最大完工时间作为优化目标对JGQEA的优化性能进行检验。实际的调度目标往往包括利润最大、库存最小、提前拖期惩罚最小等等的多目标调度,因此,如何将JGQEA用于JSSP的多目标调度具有非常重要的研究价值,这也是我们以后要重点研究的内容。

参考文献:

- [1] Narayanan A, Moore M. Quantum inspired genetic algorithm[C]//IEEE International Conference on Evolutionary Computation. Piscataway, 1996: 61-66.
- [2] Han K H, Kim J H. Genetic quantum algorithm and its application to combinatorial optimization problem[C]//IEEE International Conference on Evolutionary Computation. Lajolla, 2000: 1354-1360.
- [3] Han K H, Kim J H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(6): 580-593.
- [4] 杨俊安, 庄镇泉, 史亮. 多宇宙并行量子遗传算法[J]. 电子学报, 2004, 32(6): 923-928.
- [5] Zhang G, Gu Y, Hu L, et al. A novel quantum genetic algorithm and its application to digital filter design[J]. Intelligent Transportation Systems, 2003, 2: 1600-1605.
- [6] Chen H, Zhang J H, Zhang C. Chaos updating rotated gates quantum-inspired genetic algorithm proceedings of the international conference on communications [J]. Circuits and Systems, 2004, 2: 1108-1112.
- [7] Wang L, Tang F, Wu H. Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation[J]. Appl Math Comput, 2005, 171: 1141-1156.
- [8] Li P, Li S. Quantum-inspired evolutionary algorithm for continuous spaces optimization[J]. Chinese Journal of Electronics, 2008, 17(1): 80-84.
- [9] 李士勇, 李盼池. 基于实数编码和目标函数梯度的量子遗传算法[J]. 哈尔滨工业大学学报, 2006, 38(8): 1216-1218, 1223.
- [10] Li P, Li S. Quantum-inspired evolutionary algorithm for continuous spaces optimization based on Bloch coordinates of qubits[J]. Neurocomputing, 2008, 72: 581-591.
- [11] Gu J, Gu X, Gu M. A novel parallel quantum genetic algorithm for stochastic job shop scheduling[J]. J Math Anal Appl, 2009, 355(1): 63-81.
- [12] Gu J, Gu M, Cao C, et al. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job

- shop scheduling problem[J]. *Comput Oper Res*, 2010, 37(5): 927-937.
- [13] 张建明. 基于改进量子进化算法的生产调度问题研究[D]. 上海: 华东理工大学, 2013.
- [14] Ripon K S N, Sam K, Man K F. A real coding jumping gene genetic algorithm (RJGGA) for multi-objective optimization[J]. *Inf Sci*, 2007, 177(2): 632-654.
- [15] Muth J F, Thompson G L. *Industrial Scheduling*[M]. EngleWood Cliffs: Prentice-Hall, 1963.
- [16] Ripon K S N, Chi H T, Sam K. Multi-objective evolutionary job-shop scheduling using jumping genes genetic algorithm[C]//2006 International Joint Conference on Neural Networks. Vancouver, 2006: 3100-3107.
- [17] Gao L, Zhang G, Zhang L, et al. An efficient memetic algorithm for solving the job shop scheduling problem [J]. *Comput Indust Engin*, 2011, 60: 699-705.
- [18] Ombuki B M, Ventresca M. Local search genetic algorithms for job shop scheduling problem[J]. *Applied Intelligence*, 2004, 21: 99-109.
- [19] Nakano R, Yamada T. Conventional genetic algorithm for job-shop problems[C]//Proceedings of the 4th International Conference on Genetic Algorithms (ICGA'91). San Diego, 1991: 474-479.
- [20] Kennedy J, Eberhart R C. Particle swarm optimization [C]//IEEE International Conference on Neural Networks, Perth, 1995: 1942-1948.
- [21] Tasgetiren M F, Sevkli M, Liang Y C. A particle swarm optimization and differential evolution algorithms for job shop scheduling problem[J]. *Inter J Oper Res*, 2006, 3(2): 120-135.

A Novel Quantum Evolutionary Algorithm for Job-shop Scheduling

ZHANG Jian-ming

(School of Sciences, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: For job-shop scheduling, this paper proposes jumping gene quantum evolutionary algorithm (JGQEA) with the optimization objective of minimizing the makespan. This algorithm introduces jumping gene operator based on quantum evolutionary algorithm, and applies dynamic adjusting quantum rotation angle strategy to improve search capability of the algorithm. The effectiveness of the algorithm is verified by simulation experiment, and results show that JGQEA is superior to QEA and several other evolutionary algorithm.

Key words: job-shop scheduling; converting mechanism; quantum evolution; rotation angle; jumping operator

(责任编辑: 康 锋)