

7-30-2020

## Improved Quantum-behaved Particle Swarm Optimization for Solving Multi-objective Flexible Job-Shop Scheduling Problems

Tian Na

*1. Institute of Educational Informatization, Jiangnan University, Wuxi 214122, China; ;2. Institute of Electrical Automation, Jiangnan University, Wuxi 214122, China;*

Zhicheng Ji

*2. Institute of Electrical Automation, Jiangnan University, Wuxi 214122, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Improved Quantum-behaved Particle Swarm Optimization for Solving Multi-objective Flexible Job-Shop Scheduling Problems

## Abstract

**Abstract:** Due to the complexity of flexible job-shop scheduling problem (FJSP), it is still the hot topic for research. FJSP was given deep insight into with three objectives to be minimized simultaneously: makespan, maximal machine workload and total workload. *Quantum-behaved particle swarm optimization (QPSO) with different coefficient selection methods was compared.* The benchmark function tests show that QPSO with adaptive coefficient outperforms other selection methods in unimodal functions, while QPSO with cosine coefficient performs better in multi-modal functions. Therefore, QPSO with cosine decreasing coefficient is adopted to solve the multi-objective FJSP, which is a complex multi-modal optimization problem. Simulation results of four representative FJSP examples indicate the effectiveness and efficiency of the proposed method.

## Keywords

quantum-behaved particle swarm optimization, adaptive coefficient, cosine coefficient, multi-objective problem flexible job-shop scheduling problems, critical path

## Recommended Citation

Tian Na, Ji Zhicheng. Improved Quantum-behaved Particle Swarm Optimization for Solving Multi-objective Flexible Job-Shop Scheduling Problems[J]. *Journal of System Simulation*, 2015, 27(12): 2948-2957.

## 改进量子粒子群求解多目标柔性作业车间调度

田娜<sup>1,2</sup>, 纪志成<sup>2</sup>

(1.江南大学教育信息化研究中心, 江苏 无锡 214122; 2.江南大学电气自动化研究所, 江苏 无锡 214122)

**摘要:** 柔性作业车间调度问题(FJSP), 由于其求解的复杂性, 仍然是研究者们研究热点。对基于不同的缩放系数选择策略的量子粒子群算法(QPSO)进行了比较研究, 标准测试函数的仿真结果表明, 自适应的缩放系数在单峰问题上优于其他选择策略; 而余弦递减系数由于帮助粒子避免了陷入早熟而在多峰问题上表现比较好, 故将其应用于求解多目标柔性作业车间调度问题(最大完工时间, 最大机器工作时间, 全部机器工作时间)。4 个经典的仿真实例测试结果表明了算法的有效性和相较于其他算法的优越性。

**关键词:** 量子粒子群算法; 自适应系数; 余弦系数; 多目标柔性作业车间调度; 关键路径  
中图分类号: TP183 文献标识码: A 文章编号: 1004-731X (2015) 12-2948-10

Improved Quantum-behaved Particle Swarm Optimization  
for Solving Multi-objective Flexible Job-Shop Scheduling ProblemsTian Na<sup>1,2</sup>, Ji Zhicheng<sup>2</sup>(1. Institute of Educational Informatization, Jiangnan University, Wuxi 214122, China;  
2. Institute of Electrical Automation, Jiangnan University, Wuxi 214122, China)

**Abstract:** Due to the complexity of flexible job-shop scheduling problem (FJSP), it is still the hot topic for research. FJSP was given deep insight into with three objectives to be minimized simultaneously: makespan, maximal machine workload and total workload. *Quantum-behaved particle swarm optimization (QPSO) with different coefficient selection methods was compared.* The benchmark function tests show that QPSO with adaptive coefficient outperforms other selection methods in unimodal functions, while QPSO with cosine coefficient performs better in multi-modal functions. Therefore, QPSO with cosine decreasing coefficient is adopted to solve the multi-objective FJSP, which is a complex multi-modal optimization problem. Simulation results of four representative FJSP examples indicate the effectiveness and efficiency of the proposed method.

**Keywords:** quantum-behaved particle swarm optimization; adaptive coefficient; cosine coefficient; multi-objective problem flexible job-shop scheduling problems; critical path

## 引言

作业车间调度问题(JSP), 是生产调度的一个

重要分支, 属于 NP-hard 组合优化问题。经典的 JSP 问题是将一组工件安排到一组机器上处理, 在满足一定约束条件的前提下, 实现某种优化标准。柔性作业车间调度问题(FJSP)是 JSP 的扩展, 它允许一道工序在一组机器的任何一台机器上处理, 更接近真实的生产情况。由于需要额外的工作给工序分配机器, 所以 FJSP 的求解过程比 JSP 更复杂。



收稿日期: 2015-03-30 修回日期: 2015-09-06;  
基金项目: 江苏省博士后基金(1401004B); 国家高技术  
研究发展计划项目(2013AA040405);  
作者简介: 田娜(1983-), 女, 河北, 博士, 副教授,  
研究方向为智能控制, 系统辨识; 纪志成(1959-), 男,  
浙江, 博士, 教授, 研究方向为智能控制, 系统辨识。

<http://www.china-simulation.com>

• 2948 •

在过去的 10 年间, 越来越多的学者开始研究 FJSP 问题, 尤其是多目标问题, 由于目标之间存在冲突, 所以使得优化过程更加复杂。Brandimarte<sup>[1]</sup> 首先将问题分解成多个子问题, 然后用禁忌搜索算法求解, 但是局部搜索也存在非常大的局限性。在文献[2]中, ZHANG 基于粒子群优化算法(PSO)和禁忌搜索算法提出了一种混合算法, 求解多目标 FJSP, 比单独用禁忌搜索有了很大的改进。与禁忌搜索类似, 李俊等<sup>[3]</sup>提出一种改进的模拟退火算法求解 FJSP。Davarzani<sup>[4]</sup> 采用人工免疫算法求解多目标 FJSP, 并与 AL+CGA<sup>[5]</sup>, PSO+SA<sup>[6]</sup>进行了比较。此外各种改进版本的 PSO 也被应用于求解 FJSP<sup>[7-9]</sup>, 基于教与同伴学的粒子群算法用于求解柔性作业车间调度问题<sup>[10]</sup>, 以改善 PSO 的全局搜索能力。但是, PSO 算法中的粒子有两个状态变量描述(位置和速度), 迭代过程还需要额外的工作去设计速度的改变形式, 比较复杂。另外, 还极少有研究工作对序列的交叉和变异的概率控制进行研究, 例如, 在前期, 需要较大的交叉和变异概率以提高种群的多样性, 而在后期, 交叉和变异概率需要相应进行调整, 以保证种群收敛。

在本文中, 基于改进缩放系数的量子粒子群优化算法(QPSO)用于求解多目标柔性作业车间调度问题。相比 PSO 而言, QPSO 中粒子只有一个位置状态, 也只有一个参数需要调整, 所以更加简单, 也节省了很多计算工作。而且, 由于 FJSP 是一个复杂的多峰组合优化问题, 传统的 QPSO 很容易陷入局部最优解, 所以, 我们提出两个选择缩放系数(QPSO 算法的唯一参数)的策略(自适应和余弦)来控制算法的收敛性。另外, 我们采用基于 pareto 排序的多目标优化方法用来评估粒子的适应值, 利用外部非支配集引导粒子的搜索过程。

## 1 多目标 FJSP 问题的数学描述

一般的多目标最小化问题可以表示为:  $f(x)$  在满足一些约束条件的前提下, 有  $P (P > 1)$  个决策变量和  $Q (Q > 1)$  个目标<sup>[11]</sup>。

$$\text{Min } f(x) = (f_1(x), \dots, f_q(x), \dots, f_Q(x)) \quad (1)$$

其中  $\Omega$  是搜索空间,  $x = (x_1, x_2, \dots, x_p, \dots, x_P)$  是一组  $P$  维的决策变量(连续的或者离散的)( $1 \leq p \leq P$ ), 代表问题的一个可能解;  $f_q(x)$  是第  $q$  个目标函数( $1 \leq q \leq Q$ )。显然, 此问题的唯一精确解是不存在的, 因为多个目标函数可能是冲突的, 只能找到所有目标函数的一个平衡。

FJSP 问题可以描述如下: 有一组  $n$  个工件和一组  $m$  台机器,  $M$  表示所有机器的集合。每个工件  $i$  包含一系列 ( $n_i$ ) 的工序。每道工序  $O_{i,j}$  ( $i=1, 2, \dots, n; j=1, 2, \dots, n_i$ ) 可以在机器集  $M_{i,j}$  上处理 ( $M_{i,j} \subseteq M$ ), 工序需要从其中选择一台机器进行处理。FJSP 问题需要确定 2 个问题: 机器的选取和工序的排序。如果至少有一道工序存在  $M_{i,j} \subset M$ , 那么问题定义为部分柔性(P-FJSP); 而如果每道工序都满足  $M_{i,j} = M$ , 那么问题定义为全柔性(T-FJSP)<sup>[12]</sup>。在本文中, 目标函数定义如下:

(1) 机器的最大完工时间

$$\min f_1 = \max_{1 \leq k \leq m} (C_k) \quad (2)$$

(2) 机器的最大工作时间

$$\min f_2 = \max_{1 \leq k \leq m} (W_k) \quad (3)$$

(3) 所有机器的工作时间

$$\min f_3 = \sum_{k=1}^m W_k \quad (4)$$

在本文中用到的符号总结如下:

$n$ : 工件总数;

$m$ : 机器总数;

$n_i$ : 工件  $i$  的工序数;

$O_{i,j}$ : 工件  $i$  的第  $j$  道工序;

$M_{i,j}$ : 工序  $O_{i,j}$  的可用机器集合;

$P_{i,j,k}$ : 工序  $O_{i,j}$  在机器  $k$  上的处理时间;

$t_{i,j,k}$ : 工序  $O_{i,j}$  在机器  $k$  上的开始时间;

$C_{i,j}$ : 工序  $O_{i,j}$  的完成时间;

$C_k$ : 机器  $M_k$  的完工时间;

$W_k$ : 机器  $M_k$  的工作时间;

$$x_{i,j,k} = \begin{cases} 1, & \text{if } O_{ij} \text{ operates on machine } k \\ 0, & \text{else} \end{cases}$$

在问题的求解过程中，要满足一下假设：

(1) 每道工序在处理过程中不能被打断；

$$\sum_{k \in M_{i,j}} x_{i,j,k} = 1, \quad \forall i, j \quad (5)$$

(2) 每台机器在某一个时间只能处理一道工序；

$$\begin{cases} [(C_{h,g} - t_{h,g,k} - C_{i,j})x_{h,g,k}x_{i,j,k} \geq 0] \vee \\ [(C_{i,j} - t_{i,j,k} - C_{h,g})x_{h,g,k}x_{i,j,k} \geq 0] \\ \forall (i, j), (h, g), k \end{cases} \quad (6)$$

(3) 工件的工序顺序约束要满足；

$$C_{i,j} - C_{i,(j-1)} \geq P_{i,j,k}x_{i,j,k} \quad (7)$$

(4) 所有的机器都是在时刻 0 可用的。

基于 Pareto 方法，采用了支配的概念，是求解多目标问题最常用的方法，已经被证明在很多工作中非常有效<sup>[12-13]</sup>。

## 2 量子行为粒子群优化算法

### 2.1 粒子群优化算法

粒子群优化算法(PSO)是基于群体智能的优化技术，最初由 Kennedy 和 Eberhart 在 1995 提出的<sup>[14]</sup>。该算法的概念是来源于鸟群或鱼群的社会行为。该系统有一个粒子群体，其中每个粒子代表一个优化问题可能的解。已经证明 PSO 算法与遗传算法(GA)有相当的性能<sup>[15-16]</sup>。

在有  $M$  个粒子， $D$  维空间的 PSO 中，第  $i$  个粒子第  $k$  次迭代的位置向量和速度向量表示为：

$$X_i(k) = (X_{i1}(k), X_{i2}(k), \dots, X_{iD}(k)),$$

$$V_i(k) = (V_{i1}(k), V_{i2}(k), \dots, V_{iD}(k))。$$

粒子根据以下迭代公式更新速度和位置：

$$V_i(k+1) = \omega V_i(k) + c_1 r_1 (P_i(k) - X_i(k)) + c_2 r_2 (P_g(k) - X_i(k)) \quad (8)$$

$$X_i(k+1) = X_i(k) + V_i(k+1) \quad (9)$$

其中  $i=1,2,\dots,M$ ， $j=1,2,\dots,D$ ， $\omega$  是惯性权重，是  $c_1$  和  $c_2$  是加速系数。 $r_1$  和  $r_2$  是均匀分布在(0,1)中的随机数。向量  $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$  是第  $i$  个粒子的历史最佳位置。向量  $P_g = (P_{g1}, P_{g2}, \dots, P_{gD})$  是整个

种群的历史最佳位置。

### 2.2 量子行为粒子群优化算法

PSO 算法的缺点是不能确保收敛到全局最优解<sup>[17]</sup>，为了克服这个缺点，Sun 在 2004 年提出了量子行为粒子群优化算法的概念<sup>[18]</sup>，文献[17]中的轨迹分析说明，当每个粒子收敛到  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  时，PSO 算法实现收敛：

$$p_i(k) = \varphi P_i(k) + (1 - \varphi) P_g(k) \quad (10)$$

其中  $\varphi \in (0,1)$ 。

在量子世界中，粒子的速度是没有意义的。所以在 QPSO 中，位置是唯一描述粒子状态的变量，更新公式如下：

$$X_i(k+1) = p_i(k) \pm \alpha |mbest(k) - X_i(k)| \ln(1/u) \quad (11)$$

其中  $mbest(k)$  称为平均最优位置，定义为所有粒子历史最优位置的平均值：

$$mbest(k) = \left( \frac{1}{M} \sum_{i=1}^M P_{i1}(k), \dots, \frac{1}{M} \sum_{i=1}^M P_{iD}(k) \right) \quad (12)$$

公式(11)中的参数  $\alpha$  是缩放系数，通过调整可以控制收敛速度。

由于 QPSO 算法不需要速度向量，并且只有一个参数需要控制，使得算法更容易实现。标准测试函数证明，QPSO 算法的性能要优于 PSO 算法<sup>[18-21]</sup>。

### 2.3 基于缩放系数控制的 QPSO

在过去的文献中，为了改善粒子群和量子粒子群算法的全局搜索能力，很多工作都针对缩放系数做了改进。其中，Sugeno 函数用来改进惯性权重：

$$\omega = \frac{(1 - \beta)}{(1 - s\beta)} \quad (13)$$

其中  $\beta = k / k_{max}$ ， $s$  是一个  $>-1$  的常数。

Nickabadi<sup>[22]</sup>总结了各种版本的惯性权重修改版本(常数，线性递减，非线性递减和自适应)。文献[23]提出了一个基于粒子排序的自适应方法：

$$\omega_i = \omega_{min} + (\omega_{max} - \omega_{min}) Rank_i / M \quad (14)$$

其中  $\omega_{\max}$  和  $\omega_{\min}$  是惯性权重的最大和最小值。  
 $Rank_i$  是根据粒子适应值排序得到的粒子  $i$  的名次。

在 QPSO 中, 缩放系数的值越大, 粒子搜索范围越广; 相反, 缩放系数的值越小, 粒子会在局部仔细搜索。选取合适的缩放系数可以使得算法在全局搜索和局部搜索取得一个平衡, 进而找到全局最优解。最常用的缩放系数选取方法是随着迭代步数线性递减:

$$\alpha = (\alpha_{\max} - \alpha_{\min})(k_{\max} - k) / k_{\max} + \alpha_{\min} \quad (15)$$

Sun 在文献[21]中提出, 常数  $\alpha = 0.75$  能在简单的单峰问题上能取得很好的效果。但是对于复杂的多峰问题, 就很容易陷入局部最优。文献[24]提出 2 个非线性递减方法:

$$\alpha = (\alpha_{\max} - \alpha_{\min})(k/k_{\max})^2 + (\alpha_{\min} - \alpha_{\max})(2k/k_{\max}) + \alpha_{\max} \quad (16)$$

$$\alpha = (\alpha_{\max} - \alpha_{\min})\cos(\pi k/2k_{\max}) + \alpha_{\min} \quad (17)$$

从图 1 可以看出, 抛物线下降的速度比直线要快, 而余弦曲线下降的速度要比直线慢。

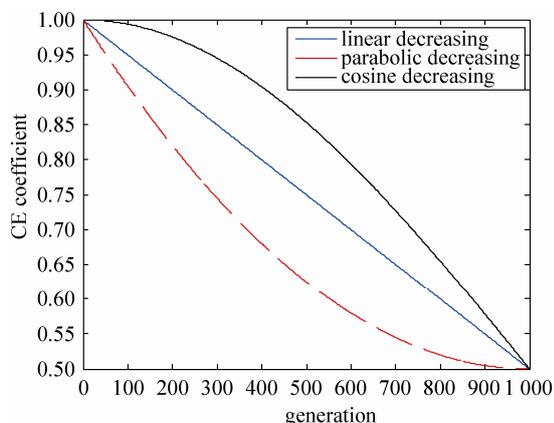


图 1 缩放系数的 3 种递减策略

本文提出一种自适应的缩放系数选取方法, 并且与其他方法做对比。首先, 定义成功比:

$$S(i, k) = \begin{cases} 1, & \text{if } f(pbest_i^k) < f(pbest_i^{k-1}) \\ 0, & \text{if } f(pbest_i^k) = f(pbest_i^{k-1}) \end{cases} \quad (18)$$

$$P_s(k) = \sum_{i=1}^M S(i, k) / M \quad (19)$$

其中  $pbest_i^k$  粒子  $i$  的个体最优位置,  $P_s \in [0, 1]$  是粒子的个体最优位置得到改善的次数比。成功比越

大, 说明粒子们已经收敛到一个远离最优解的点, 并且整个群体正在很缓慢的向最优解移动。相似的, 成功比越小, 说明粒子们在最优解附近徘徊, 迭代过程中, 粒子的适应值并没有得到很大的改善。因此, 选择一个与  $P_s$  成正比的缩放系数  $\alpha$  是比较合理的:

$$\alpha(k) = g(P_s(k)) \quad (20)$$

其中  $g(\cdot)$  可以是任何随着  $P_s$  递增的函数。

我们采用一个线性函数, 将  $P_s$  映射到缩放系数  $\alpha$ :

$$\alpha = (\alpha_{\max} - \alpha_{\min})P_s(k) + \alpha_{\min} \quad (21)$$

其中  $([\alpha_{\min}, \alpha_{\max}])$  的值选取为  $[0, 1.0]$ 。

## 2.4 标准测试函数仿真分析

为了测试并且比较不同缩放系数选择策略的性能, 采用一组常用的标准测试函数(表 1)来做仿真, 其中,  $f_1, f_2$  和  $f_3$  是单峰问题,  $f_4, f_5, f_6, f_7$  是复杂多峰问题。仿真中用到的搜索空间的维数  $D$  设置为 30。

表 1 标准测试函数

公式	范围
$f_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]$
$f_2 = \sum_{i=1}^D (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	$[-30, 30]$
$f_3 = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$[-100, 100]$
$f_4 = \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$
$f_5 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$
$f_6 = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	$[-100, 100]$
$f_7 = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$

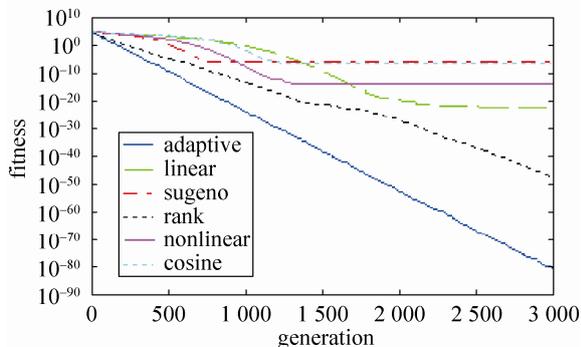
我们将自适应的缩放系数选择策略(adaptive)与其余 6 个选择策略从收敛速度和收敛精度两方面进行了比较: 线性递减(linear)、Sugeno 函数(sugeno)、基于名次的(rank)、非线性递减(nonlinear)和余弦递减(cosine)。

在以下仿真实验中，粒子的个数设为  $M = 20$ ，最大迭代步数为  $K_{\max} = 3000$ ， $\alpha$  的范围设为  $[0.5, 1.0]$ 。表 2 是运行 50 次的平均最优结果和标准偏差值。图 2 给出了 7 种算法在 6 个标准函数上的收敛曲线，从中可以看出，对于简单的单峰问题，

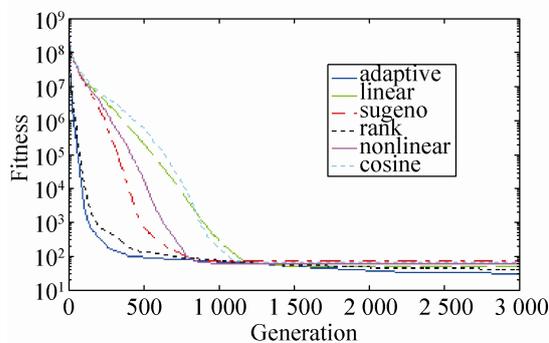
自适应缩放系数无论从平均最优解还是收敛速度上，均优于其他方法。然而对于复杂多峰问题  $f_4 \sim f_7$ ，余弦递减的缩放系数，由于下降的速度比较慢，粒子有更多的时间进行较大范围的搜索，虽然收敛速度比较慢，但是收敛的结果优于其他方法。

表 2 标准测试函数结果

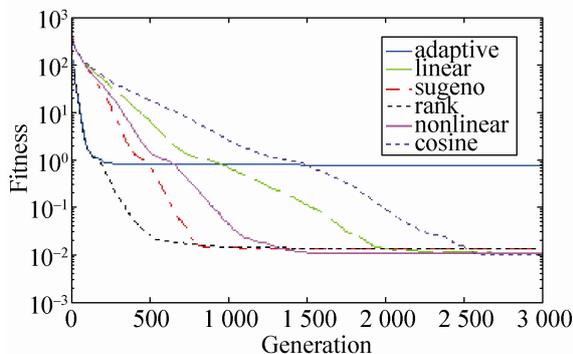
function	Mean optimum (Standard deviation)					
	Adaptive	Linear	Sugeno	Rank	Parabolic	Cosine
$f_1$	6.39E-82 (6.5E-162)	2.71E-23 (6.33E-45)	2.23E-07 1.59E-12	2.21E-48 (2.40E-94)	9.66E-15 (1.99E-27)	1.56E-07 (8.26E-13)
$f_2$	28.01 (724.48)	45.13 (1.46E03)	67.81 (3.55E03)	38.57 (1.20E03)	55.50 (2.80E03)	59.95 (2.37E03)
$f_3$	0.18 0.021	253.18 2.44E04	2.38E03 (1.00E06)	28.35 5.62E03	721.99 1.44E05	2.34E03 (8.55E05)
$f_4$	72.33 (488.44)	28.48 (99.59)	34.13 (95.58)	33.26 (94.60)	29.40 (75.94)	27.07 (94.74)
$f_5$	2.90E-02 (7.25E-04)	1.1E-02 (1.82E-04)	1.27E-02 1.80E-04	1.32E-02 (6.53E-04)	1.02E-02 (2.97E-04)	9.5E-03 (1.4E-04)
$f_6$	9.6E-03 (6.19E-05)	7.2E-03 (6.22E-05)	7.8E-03 (1.5E-05)	7.5E-03 (3.65E-05)	7.6E-03 (1.60E-05)	6.8E-03 (1.9E-05)
$f_7$	0.14 (0.20)	1.31E-05 (3.1E-10)	2.19E-05 (9.6E-10)	0.12 (0.18)	7.24E-09 (2.12E-16)	2.01E-12 (5.8E-24)



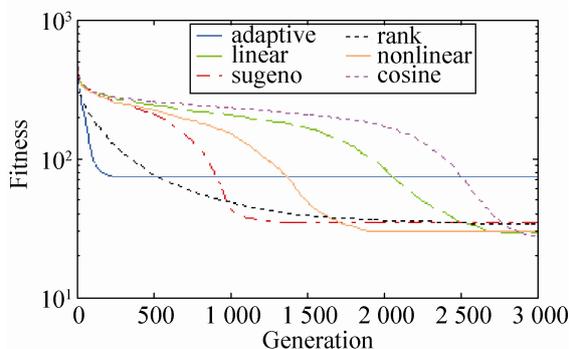
(a) f1



(b) f2



(c) f3



(d) f4

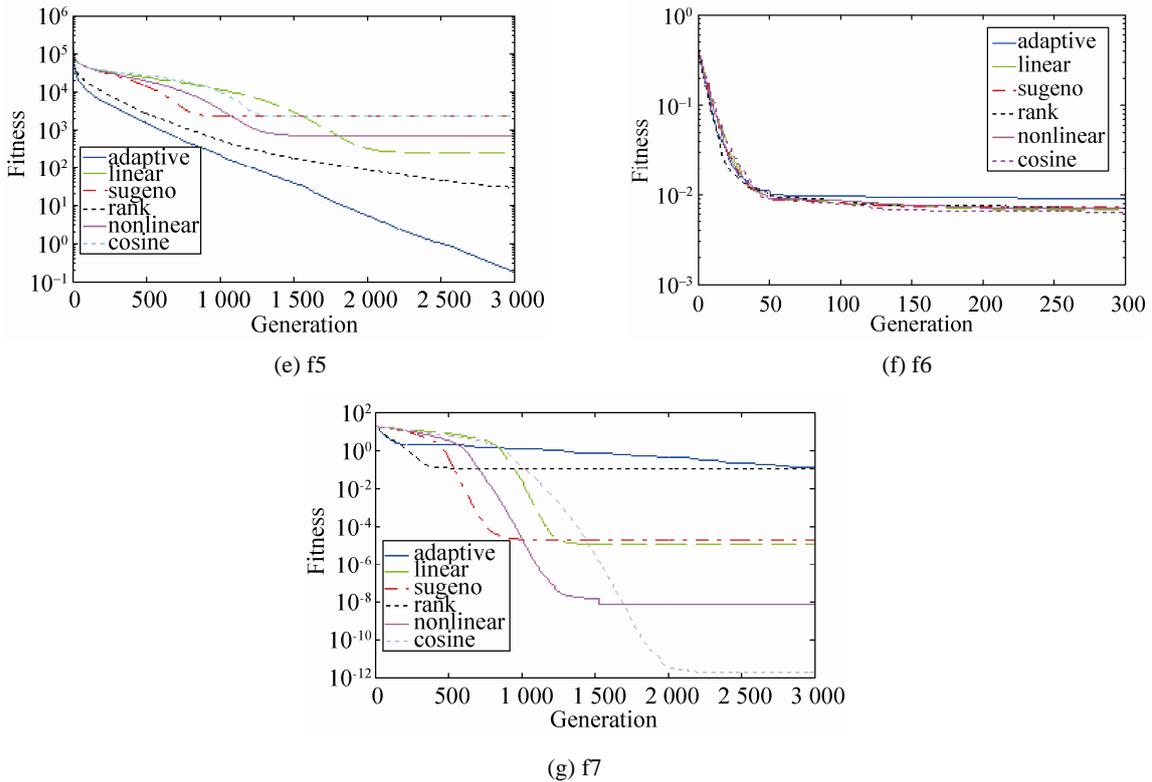


图 2 6 个算法在 7 个标准测试函数上的收敛曲线

### 3 余弦递减缩放系数 QPSO 求解柔性作业车间调度问题

#### 3.1 编码

用 PSO 或者 QPSO 解决 FJSP 问题, 对粒子的位置编码是最关键的一步。在本文中, 粒子的位置向量包含两部分(以表 3 为例), 第 1 部分表示工序序列  $(O_{1,1}, O_{2,1}, O_{3,1}, O_{3,2}, O_{1,2}, O_{1,3}, O_{2,2}, O_{2,3})$ , 第 2 部分表示工序选取的机器, 如图 3 所示, 1000 表示  $O_{1,1}$  在机器  $M_1$  上处理, 0001 表示  $O_{1,2}$  在  $M_4$  上处理。位置向量的长度等于  $\sum_{i=1}^n n_i(1+m)$ 。这个编码方

法的优点是, 可以比较容易的对粒子的工序部分和机器部分分别进行交叉和变异操作, 缺点是对存储空间复杂度比较高。

表 3 3 个工件 4 台机器实例

工件	工序	$M_1$	$M_2$	$M_3$	$M_4$
$J_1$	$Q_{11}$	1	3	4	1
	$Q_{12}$	3	8	2	1
	$Q_{13}$	3	5	4	7
$J_2$	$Q_{21}$	4	1	1	4
	$Q_{22}$	2	3	9	3
	$Q_{23}$	9	1	2	2
$J_3$	$Q_{31}$	8	6	3	5
	$Q_{32}$	4	5	8	1

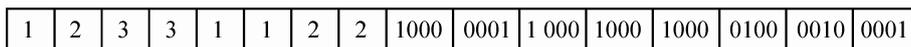


图 3 粒子的位置向量编码

#### 3.2 粒子位置初始化

粒子的初始化对算法的性能有很大的影响, 由于 QPSO 算法的粒子只有一个位置向量, 本文采用一个有指导性的初始化方法。但是为了确保群体的

多样性, 50%的粒子随机初始化, 另外 50%的粒子采用以下方法进行初始化。

第 1 步: 初始化

- (a) 随机生成工件的一个排列  $J$  (例如 2, 3, 1);
- (b) 将每个工件的第一道工序  $O_{2,1}, O_{3,1}, O_{1,1}$  根据

规则(2b)分配到合适的机器处理, 如果此时机器上没有其他工序正在处理, 则  $O_{x,l}$  从时刻 0 开始处理; 否则,  $O_{x,l}$  是从机器的最后一道工序的结束时间开始处理。

第 2 步: 给工序找一个合适的机器:

(a) 找出结束时间最早的机器  $M_s$  以及此台机器的最后一道工序  $O_{i,j}$ 。如果有多个机器满足条件, 随机选取一台, 令  $t$  为机器  $M_s$  的最后完成时间;

(b) 根据以下步骤为工序  $O_{i,j+1}$  选合适的机器:

\*找出所有能处理工序  $O_{i,j+1}$  的机器:  $P(O_{i,j+1})$ ;

\*计算每台机器  $M_k \in P(O_{i,j+1})$  的等待时间: 如果  $M_k$  在时刻  $t$  没有处理任何工序, 则等待时间等于工序  $O_{i,j+1}$  在机器  $M_k$  上的处理时间; 否则, 如果机器  $M_k$  正在处理工序  $O_{z,y}$ , 则等待时间等于机器  $M_k$  上等待队伍里所有工序的处理时间+ (工序  $O_{z,y}$  的剩余处理时间)+工序  $O_{i,j+1}$  在机器  $M_k$  上的处理时间。

\*工序  $O_{i,j+1}$  分配到等待时间最短的机器上处理。

第 3 步: 跳转到第 2 步。

### 3.3 交叉和变异

对于组合优化问题, 有效的信息交换可以帮助找到更好的解。而交叉和变异是实现信息交换的最常用的方法。本文分别对工序序列部分和机器部分进行交叉和变异。图 4 给出工序部分的交叉过程。其中, 子粒子 C1 和 C2 直接从父粒子 P1 和 P2 中继承一部分基因({1},{2,3}), 剩余部分基因则从 P2 和 P1 中按顺序拷贝。

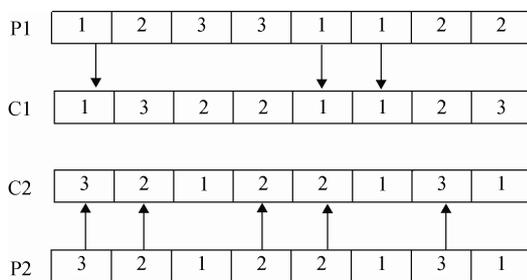


图 4 工序部分的交叉操作

机器部分的交叉操作如图 5 所示, 其中, 随机选取 2 个位置, 然后进行 2 点交叉。

P1	1000	0001	1000	1000	1000	0100	0010	0001
P2	0100	0010	0100	0010	0100	0010	0010	0010
C1	0100	0010	1000	1000	0100	0010	0010	0010
C2	1000	0001	0100	0100	0100	0010	0010	0001

图 5 机器部分的交叉操作

粒子的变异也分为工序部分变异和机器部分变异, 工序部分变异如图 6 所以, 任意选取两个位置, 然后颠倒两个位置之间的工序。对于机器部分的变异, 不能将工序任意选取一台机器进行处理, 因为最大完工时间和工作时间是由一个序列的关键路径决定的。因而, 只有移动关键路径上的工序才有可能使得序列有所改善。以图 6 为例, 工序集合  $\{O_{4,1}, O_{2,1}, O_{2,2}, O_{2,3}\}$  构成一条关键工序(图 7a), 然后将工序  $O_{4,1}$  从机器  $M_1$  移动到  $M_4$ , 显而易见, 最大完工时间从 12 减到了 11(图 7b)。

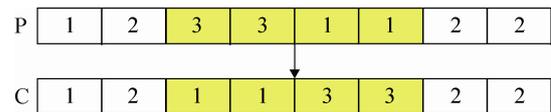
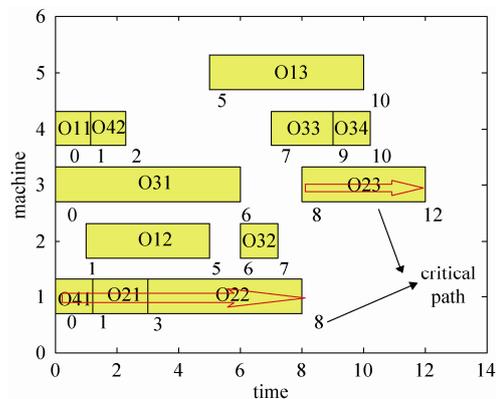
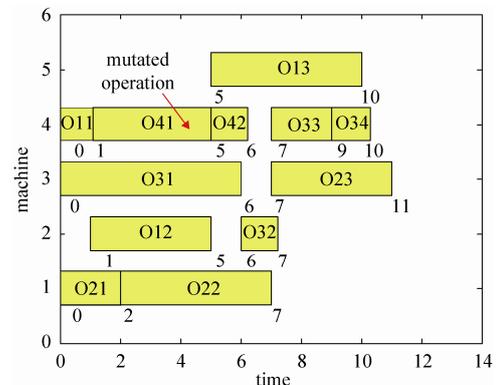


图 6 工序部分的变异操作



(a) 关键路径



(b) 移动工序

图 7 找出并移动关键路径上的工序

### 3.4 算法流程

步骤 1: 初始化

粒子的个数  $M=100$ ; 缩放系数  $\alpha$  的范围  $[0.5, 1.0]$ ; 粒子的位置向量的维度  $D$  根据具体的问题来定义; 50% 的粒子位置根据 3.2 节的规则初始化, 50% 的粒子位置随机初始化, 构成  $X(0)$ ; 初始化个体最优位置  $P(0)$ ; 初始化非支配集  $nds$ ; 根据 Sigma 规则从  $nds$  中选取一个全局最优位置  $P_g$ 。迭代步数  $k=0$ 。

步骤 2: 迭代

(1) 对种群中的每个粒子, 进行如下步骤:

(a) 交叉

以缩放系数  $\alpha$  为交叉概率, 将  $X_i(k)$  中的工序部分和机器部分, 分别与  $P_g$  和  $P_i(k)$  进行交叉, 得到  $X_1$  和  $X_2$ 。

若  $X_1$  支配  $X_2$ , 并且  $X_1$  支配  $X_i(k)$ , 则  $P_i(k)=X_1$ 。

若  $X_2$  支配  $X_1$ , 并且  $X_2$  支配  $X_i(k)$ , 则  $P_i(k)=X_2$ 。

同时将  $nds$  中被  $P_i(k)$  支配的粒子移出  $nds$ , 若  $nds$  中不存在支配  $P_i(k)$  的粒子, 则  $P_i(k)$  加入  $nds$ 。

(b) 变异

以  $0.1 \times \alpha$  为变异概率, 对  $X_i(k)$  的工序部分进行编译。同时计算  $X_i(k)$  对应的关键工序以及关键路径, 对关键路径中的工序所选择的机器进行调整, 得到  $X_m$ 。

若  $X_m$  支配  $X_i(k)$ , 则  $P_i(k)=X_m$ 。同时将  $nds$  中被  $P_i(k)$  支配的粒子移出  $nds$ , 若  $nds$  中不存在支配  $P_i(k)$  的粒子, 则  $P_i(k)$  加入  $nds$ 。

(2) 根据公式(17)更新缩放系数。

$k=k+1$ ;

步骤 3: 根据 Sigma 规则选择非支配集中最好的粒子, 作为最优调度序列输出。

### 3.5 仿真分析

经过第 2 节的 benchmark 测试函数比较分析可知, 对于复杂多峰问题, 余弦递减的缩放系数有比

较好的性能。所以本节将余弦递减缩放系数的 QPSO 应用于解决多目标 FJSP 问题。为了验证算法的有效性, 选取了 4 个代表性的实例<sup>[12]</sup>, 分别是  $4 \times 5$ ,  $8 \times 8$ ,  $10 \times 10$  和  $15 \times 10$ 。

参数设置如下: 粒子个数  $M=100$ ; 非支配集的大小  $ND=100$ ; 粒子维数  $\sum_{i=1}^n n_i(1+m)$ ; 最大迭代次数  $k_{\max}=2000$ 。

从表 4 可以看出, 对于问题  $4 \times 5$ ,  $8 \times 8$  和  $15 \times 10$ , 余弦递减缩放系数 QPSO 得到的 3 个目标函数值明显支配其他算法得到的解。对于问题  $10 \times 10$ , 余弦递减缩放系数 QPSO 得到的解与其他算法得到的解互不支配, 由此可见余弦递减缩放系数 QPSO 在解决多目标柔性作业车间调度问题的优势。

表 4 4 个算法在 4 个实例上的比较测试结果

problem	Obj	AL+CGA	PSO+SA	GA	QPSO+cosine
$4 \times 5$	f1	16	11	12	11
	f2	10	10	11	10
	f3	34	32	33	32
$8 \times 8$	f1	16	16	15	14
	f2	15	13	14	13
	f3	75	73	73	73
$10 \times 10$	f1	7	7	7	7
	f2	5	6	6	5
	f3	45	44	43	44
$15 \times 10$	f1	24	12	12	11
	f2	11	11	11	11
	f3	91	91	92	91

由于本文求解的是多目标问题(最大完工时间, 单台机器最大工作时间, 所有机器的最大工作时间), 故这里不再给出算法的收敛曲线。

图 8~11 给出 4 个由余弦递减缩放系数 QPSO 得到的最优解的甘特图。其中  $15 \times 10$  问题难度最大, 原因之一是工件的数量比机器的数量少很多, 从而一个调度序列所包含的关键路径有若干条, 想调整某道工序的机器, 相当困难, 所以在此时使用局部搜索是很重要的。

另外, 多目标问题不同于单目标, 有的调度序列, 虽然最大完工时间相同, 但是甘特图对应的机

器最大工作时间和所有机器的全部工作时间却不  
相同，所以要兼顾考虑。

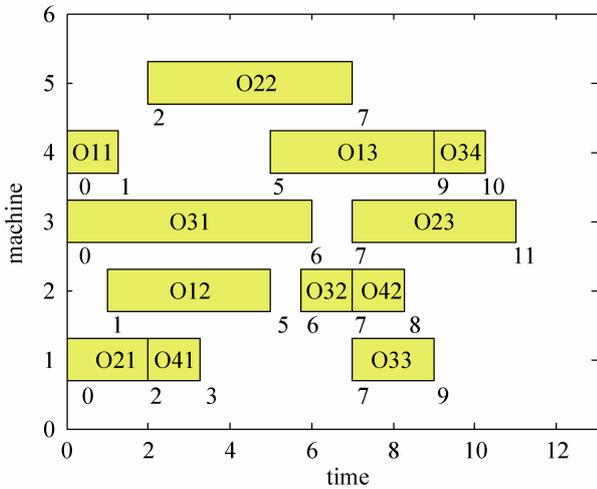


图 8 问题 4×5 的甘特图

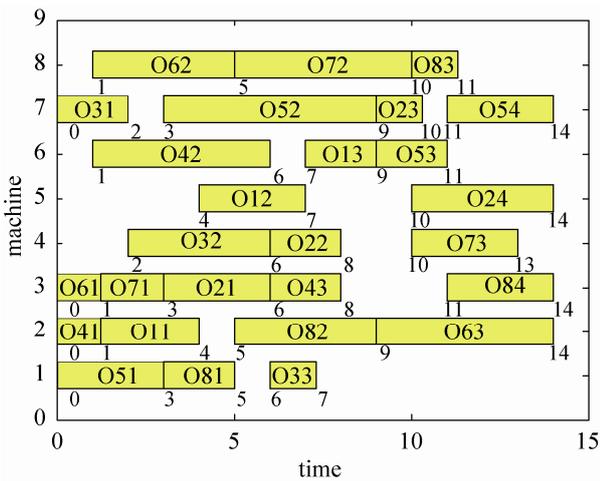


图 9 问题 8×8 的甘特图

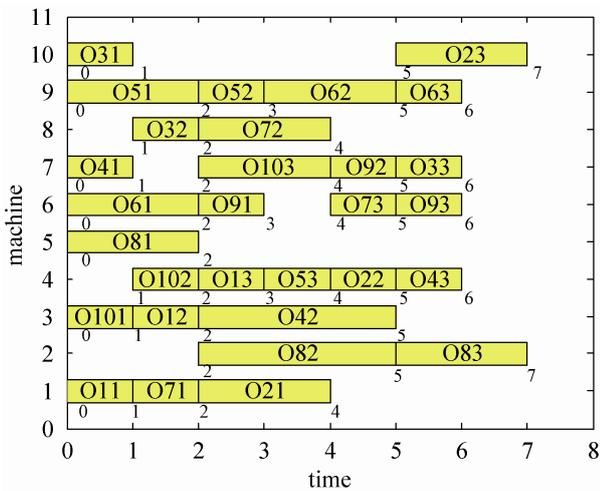


图 10 问题 10×10 的甘特图

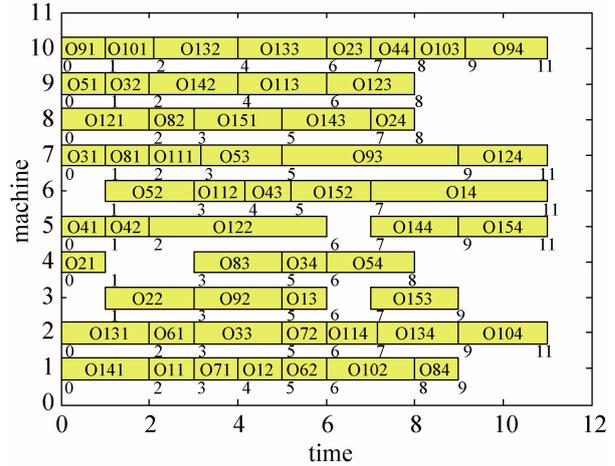


图 11 问题 15×10 的甘特图

## 4 结论

本文针对量子粒子群算法的缩放系数，采用六种不同的选择策略，以帮助算法避免陷入局部最优，标准测试函数仿真结果显示，自适应选择策略在求解简单单峰函数时，性能优于其他算法，而对于复杂多峰问题，余弦递减策略虽然收敛速度比较慢，但是收敛的全局最优解最好。考虑到多目标柔性作业车间调度问题的非线性和复杂性，我们采用余弦递减策略缩放系数的量子粒子群算法应用于此问题的求解。经过四个标准模型的仿真，结果表明，此方法能得到比较有竞争力的结果。

## 参考文献:

- [1] P Brandimarte. Routing and scheduling in a flexible job shop by tabu search [J]. Annals of Operations Research (S1572-9338), 1993, 41(3): 157-183.
- [2] G Zhang, X Shao. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem [J]. Computers & Industrial Engineering (S0360-8352), 2009, 56(4): 1309-1318.
- [3] 李俊, 刘志雄, 张煜, 等. 柔性作业车间调度优化的改进模拟退火算法 [J]. 武汉科技大学学报, 2015, 38(2): 111-116.
- [4] Z Davarzani, M Akbarzadeh, N Khairdoost. Multiobjective artificial immune algorithm for flexible job shop scheduling problem [J]. International Journal of Hybrid Information Technology (S1738-9968), 2012, 5(3): 75-88.
- [5] I Kacem, S Hammadi, P Borne, Approach by localization

- and multi-objective evolutionary optimization for flexible job-shop scheduling problems [J]. *IEEE Transactions on Systems, Man and Cybernetics (S0018-9472)*, 2002, 32(1): 1-13.
- [6] G Moslehi, M Mahnam. A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search [J]. *International Journal of Production Economics (S0925-5273)*, 2011, 129(1): 14-22.
- [7] T Lin, S Horng. An efficient job-shop scheduling algorithm based on particle swarm optimization [J]. *Expert Systems with Applications (S0957-4174)*, 2009, 37(3): 2629-2636.
- [8] X Shao, W Liu, Q Liu, *et al.* Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem [J]. *International Journal of Advanced Manufacturing Technology (S1433-3015)*, 2013, 67(9/12): 2885-2901.
- [9] B Panigrahi, V Pandi, S Das, Adaptive particle swarm optimization approach for static and dynamic economic load dispatch [J]. *Energy Conversion and Management (S0196-8904)*, 2008, 49(6): 1407-1415.
- [10] 吴定会, 孔飞, 田娜, 等. 教与同伴学习粒子群算法求解柔性作业车间调度问题 [J]. *计算机应用 (S1001-9081)*, 2015, 35(6): 1617-1622.
- [11] K Deb. *Multi-objective optimization using evolutionary algorithms*[M], Chichester, UK: Wiley, 2001.
- [12] I Kacem, S Hammadi, P Borne. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic [J]. *Mathematics and Computers in Simulation (S0378-4754)*, 2002, 60(3-5): 245-276.
- [13] T Hsu, R Dupas, D Jolly, *et al.* Evaluation of mutation heuristics for the solving of multi-objective flexible job shop by an evolutionary algorithm [C]// *Proc. IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia. USA: IEEE Press, 2002, 5: 655-660.*
- [14] J Kennedy, R Eberhart. Particle swarm optimization [C]// *IEEE International Conference on Neural Networks, Perth, Australia. USA: IEEE Press, 1995: 1942-1948.*
- [15] R Eberhart, Y Shi. Comparison between genetic algorithm and particle swarm optimization [C]// *Evolutionary Programming VII, Lecture Notes in Computer Science 1447. Germany: Springer Berlin, Heidelberg, 1998: 611-616.*
- [16] F Van Den Bergh. An analysis of particle swarm optimizers [D]. South Africa: University of Pretoria, 2001.
- [17] M Clerc, J Kennedy. The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space [J]. *IEEE Transaction on Evolutionary Computation (S1089-778X)*, 2002, 6(1): 58-73.
- [18] J Sun, B Feng, W Xu. Particle swarm optimization with particles having quantum behaviour [C]// *Proc. IEEE Congress Evolutionary Computation, Portland. USA: IEEE Press, 2004: 325-331.*
- [19] J Sun, W Xu, J Liu, Parameter Selection of Quantum-behaved particle Swarm optimization [J]. *Lecture Notes in Computer Science (S0302-9743)*, 2005, 3612: 543-552.
- [20] J Sun, W Xu, B Feng. A global search strategy of quantum-behaved particle swarm optimization [C]// *IEEE Conference on Cybernetics and Intelligent Systems, Singapore. USA: IEEE Press, 2004: 111-116.*
- [21] J Sun, W Fang, X Wu. Quantum-behaved particle swarm optimization: analysis of the individual particle's behavior and parameter selection [J]. *Evolutionary Computation (S1530-9304)*, 2012, 20 (3): 349-393.
- [22] A Nickabadi, M Ebadzadeh, R Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight [J]. *Applied Soft Computing (S1568-4946)*, 2011, 11(4): 3658-3670.
- [23] K Lei, Y Qiu, Y He. A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization [C]// *Proc. Symposium Systems and Control in Aerospace and Astronautics, Harbin, China. USA: IEEE Press, 2006: 19-21.*
- [24] N Tian, C Lai, K Pericleous. Contraction-expansion coefficient learning in quantum-behaved particle swarm optimization [C]// *10th International Symposium on Distributed and Applications to Business, Engineering and Science, Wuxi, China. USA: IEEE Press, 2011: 304-308.*