

用带变异因子的 QPSO 算法解决 Job-Shop 调度问题

石锦风,冯 斌,孙 俊

SHI Jin-feng, FENG Bin, SUN Jun

江南大学 信息工程学院,江苏 无锡 214122

School of Information Technology, Southern Yangtze University, Wuxi, Jiangsu 214122, China

E-mail: mystone2001@163.com

SHI Jin-feng, FENG Bin, SUN Jun. Solving job-shop scheduling problem with QPSO algorithm with mutation operator. *Computer Engineering and Applications*, 2008, 44(8): 49-52.

Abstract: Because Quantum-behaved Particle Swarm Optimization (QPSO) algorithm possibly run into prematurity, the mutation mechanism is introduced into QPSO algorithm to escape from local optima and strengthen its global search ability, and the improved QPSO algorithm is applied to solve Job-Shop Scheduling Problem. The simulation results show that this algorithm has better global convergence ability and more rapid convergence, and it is superior to Genetic Algorithm, Particle Swarm Optimization algorithm and QPSO algorithm.

Key words: mutation mechanism; Job-Shop Scheduling; Genetic Algorithm (GA); Particle Swarm Optimization algorithm; Quantum-behaved Particle Swarm Optimization algorithm

摘 要: 由于量子粒子群优化算法仍有可能出现早熟现象,因此将变异机制引入量子粒子群优化算法以使算法跳出局部最优并增强其全局搜索能力,并将改进后的量子粒子群优化算法用于求解作业车间调度问题。仿真实例表明,该算法具有良好的全局收敛性能和快捷的收敛速度,调度效果优于遗传算法、粒子群优化算法和量子粒子群优化算法。

关键词: 变异机制; 作业车间调度; 遗传算法; 粒子群优化算法; 量子粒子群优化算法

文章编号: 1002-8331(2008)08-0049-04 **文献标识码:** A **中图分类号:** TP391

1 引言

作业车间调度问题(Job-Shop Scheduling Problem, JSSP)是实现先进制造和提高生产效益的基础和关键,受到机械工程、计算科学和运筹学的广泛关注。然而,JSSP具有建模复杂、计算复杂和多约束等特点,属于组合优化问题,被证明是典型的NP-hard问题^[1]。

JSSP是关于计算机集成制造系统中的设计和制造控制中至关重要的一个环节。然而在很多情况下,即使是解决一个很普通的JSSP也会很艰难;并且仅仅针对某些调度问题,才有最佳的调度方案。所以,本文希望找到一个可行的解决方案解决这个问题。

目前,将粒子群优化(Particle Swarm Optimization, PSO)算法用于解决JSSP是国内外正在研究的热点。PSO算法用于求解JSSP要比其它经典的调度算法具有更大的优势和可行性,和以往经典的调度理论相比,也取得了更好的调度效果。

但在实际应用中,PSO算法还是会经常陷入局部极值。相比于PSO算法,虽然量子粒子群优化(Quantum-behaved Particle Swarm Optimization, QPSO)算法具有更好的全局搜索能

力,但是QPSO算法仍有可能陷入局部最优。

针对PSO算法和QPSO算法的上述不足,考虑将变异机制引入QPSO算法以使算法跳出局部最优并增强其全局搜索能力。这种引入了进化思想的QPSO算法,在搜索最优解的后期阶段可以增加解的多样性,进而可以使算法跳出局部最优解。

2 JSSP描述

JSSP研究 n 个工件在 m 台机器上加工,已知各操作步骤的加工时间和各工件在各机器上的加工次序约束,要求确定与工艺约束条件相容的各机器上所有工件的加工开始时间或完成时间或加工次序,使加工性能指标达到最优。

用 m 表示机器的数量, n 表示需要加工的工件的数量。假如用 i 表示第 i 个工件, k 表示第 k 个工序,则 $m(i,k)$ 表示加工第 i 个工件的第 k 个工序的机器的编号。 $t_{i,k}$ 表示工序 (i,k) 的加工时间, $m_{i,k}$ 是加工工序 (i,k) 的机器编号, ki 表示第 i 个工件的最后一道工序。

则JSSP可描述为:

(1)每个工件由 m 道工序组成,每道工序在不同的机器上

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60474030)。

作者简介:石锦风(1978-),男,硕士研究生,主要研究领域为进化计算及其应用;冯斌(1956-),女,博士,副教授,主要研究领域为智能控制技术;孙俊(1971-),男,博士研究生,主要研究领域为人工智能、进化计算。

收稿日期:2007-08-06 修回日期:2007-10-24

加工;

(2)每道工序必须在指定的机器上加工;

(3)按照加工工艺的规定,每道工序必须在它前面的工序加工完毕后再加工;

(4)每道工序从开始到结束,不会被另外的工序所中断;

(5)一个机器在某一时间只能加工一个工件,且每个工件在某一时间只能由一台机器加工。

可以看出 JSSP 是一个典型的线性规划问题。若用 $S_{i,k}$ 表示第 i 个工件的第 k 个工序的加工开始时间,可以定义如下三个约束条件:

$$S_{i,k}-S_{i,k-1} \geq t_{i,k-1}, 1 \leq i \leq n; 2 \leq k \leq ki \quad (a)$$

$$S_{i,1} \geq 0, 1 \leq i \leq n \quad (b)$$

$$S_{i,k}-S_{j,p}+t_{i,k} \leq 0 \text{ or } S_{j,p}-S_{i,k}+t_{j,p} \leq 0, 1 \leq i, j \leq n; 1 \leq k, p \leq ki(c)$$

在这些约束中,约束(a)表示工序(i,k)必须在工序($i,k-1$)之后被加工,约束(b)表示工序的开始加工时间必须是不小于零的,约束(c)表示一个特定的机器在某一时间只能加工一个工件,这样可以避免两个工件产生冲突。

调度的目标是使全部工件的加工结束时间最小,也就是说让全部工件尽可能早的被加工完成。因此,调度的目标函数是 $\max(S_{i,ki}+t_{i,ki})$,那么最小的总完工时间是 $\min(\max(S_{i,ki}+t_{i,ki}))$ 。

3 算法简介

3.1 PSO 算法

PSO 算法是一种基于群体智能方法的演化计算技术,是由 Kennedy 和 Eberhart 于 1995 年提出的,源于对鸟群捕食行为的研究。近年来,因 PSO 算法有个体数目少、计算简单和鲁棒性好等优点,已被成功应用于函数优化、系统辨识、神经网络训练、模糊系统控制等领域。

设在 D 维空间中有一个粒子数为 M 的粒子群(种群),记第 i 个粒子的位置矢量和速度矢量分别为 $X_i=(X_{i1}, X_{i2}, \dots, X_{iD})$ 和 $V_i=(V_{i1}, V_{i2}, \dots, V_{iD}), i=1, 2, \dots, M$ 。 $P_i=(P_{i1}, P_{i2}, \dots, P_{iD})$ 为粒子 i 迄今为止所搜索到的最优位置,称为个体最好位置或局部最好位置($pbest$)。 $P_g=(P_{g1}, P_{g2}, \dots, P_{gD})$ 为整个粒子群中的所有粒子迄今为止搜索到的最优位置,称为全局最好位置($gbest$)。

设 $f(X)$ 为求最小值的目标函数,则第 i 个粒子的个体最好位置 P_i 由下式确定:

$$P_i(t+1)=\begin{cases} X_i(t+1), & f(X_i(t+1)) < f(P_i(t)) \\ P_i(t), & f(X_i(t+1)) \geq f(P_i(t)) \end{cases} \quad (1)$$

粒子群中所有粒子均按如下进化方程来更新粒子的状态^[2]:

$$V_{ij}(t+1)=\omega V_{ij}(t)+c_1 r_{1j}(t)(P_{ij}(t)-X_{ij}(t))+c_2 r_{2j}(t)(P_{gj}(t)-X_{ij}(t)) \quad (2)$$

$$X_{ij}(t+1)=X_{ij}(t)+V_{ij}(t+1) \quad (3)$$

其中, $i=1, 2, \dots, M, j=1, 2, \dots, D, t$ 表示进化的当前代数, c_1 和 c_2 为学习因子, ω 为惯性权重, $r_{1j}(t)$ 和 $r_{2j}(t)$ 均为介于 $[0, 1]$ 之间的随机数。 ω 的具体取值视情况而定, c_1 和 c_2 为待定常数,通常在 $(0, 2]$ 区间内取值。

根据对粒子收敛行为的分析^[3],要保证算法的收敛性,每个粒子必须收敛于各自的 p 点,这是由粒子的追随性和粒子群的聚集性决定的。

3.2 QPSO 算法

受量子物理基本理论的启发,根据粒子群的基本收敛性质, Sun 等人于 2004 年提出的 QPSO 算法是对整个 PSO 算法

进化搜索策略的改变,并且进化方程中不需要速度向量,而且进化方程的形式更简单,参数更少且更容易控制^[4]。在量子空间中,粒子在整个可行解空间中进行搜索,因而 QPSO 算法的全局搜索性能远远优于标准 PSO 算法。

QPSO 算法利用波函数 $\psi(x, t)$ 来描述粒子的状态,并通过求解薛定谔方程得到粒子在空间某一点出现的概率密度函数,再通过 Monte Carlo 随机模拟得到粒子的位置方程为:

$$X(t)=p \pm \frac{L}{2} \ln\left(\frac{1}{u}\right) \quad (4)$$

其中, u 为在 $[0, 1]$ 上服从均匀分布的随机数。

在粒子群中引入了一个全局点 $mbest$ 来计算粒子的下一迭代步的变量 L , 它定义为所有粒子的个体最好位置的平均值。其计算公式如下:

$$mbest = \frac{1}{M} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}, \frac{1}{M} \sum_{i=1}^M P_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M P_{iD} \right) \quad (5)$$

其中 M 为种群中粒子的数目, P_i 为粒子 i 的个体最好位置。则 L 的值可由下式确定:

$$L(t+1)=2*\beta*mbest-X(t) \quad (6)$$

为了保证算法的收敛性,每一个粒子必须收敛于各自的 p 点, $p=(p_1, p_2, \dots, p_D)$, 第 i 个粒子的 p 点的第 d 维坐标为:

$$p_{id} = \varphi * P_{id} + (1-\varphi) * P_{gd} \quad (7)$$

其中 φ 是在 $[0, 1]$ 上均匀分布的随机数, D 为粒子的维数, P_i 和 P_g 分别表示粒子 i 的个体最好位置和种群的全局最好位置。

最后得到 QPSO 算法的进化方程为:

$$X_{id}(t+1)=p_{id} \pm \beta * mbest_d - X_{id}(t) * \ln\left(\frac{1}{u}\right) \quad (8)$$

其中, β 称为收缩扩张系数,是 QPSO 算法唯一的参数,一般取 $\beta=(1.0-1.5)*(MAXITER-t)/MAXITER+0.5$ 。

3.3 带变异因子的 QPSO 算法

在 QPSO 算法的每次循环中,虽然每个粒子的搜索空间是问题的整个可行解空间,但是与 PSO 算法及其他基于种群的进化算法一样,由于算法的聚集性,种群中多样性的丢失是不可避免的。相关实验证明,在后期的搜索阶段,粒子有聚集在一起的趋势,并且其搜索空间受到限制,因此整个种群易陷入局部极值。所以,本文提出变异机制以使 QPSO 算法跳出局部最优并增加种群的多样性。

在进化规划中,通过调整随机函数的参数或者步长,进化规划可以很理想地微调最优解附近的搜索空间。而在 QPSO 算法中,因为全局最好粒子 $gbest$ 吸引种群中的所有粒子向其飞行,同时也吸引着中间变量 $mbest$ 向其飞行。 $mbest$ 表示每个粒子的最好位置之间具有一定的相关性。因此,通过对某个粒子进行变异来引导种群离开当前的位置是有可能的,假如这个变异过的个体成为了新的全局最好位置或平均最好位置。基于这个目标,本文提出了一个以概率 P_m 来变异 $gbest$ 或者 $mbest$ 的变异算子。

通过添加一个高斯或柯西分布的随机值对每个向量元素独立地执行变异操作,变异过程可用如下公式表示:

$$x' = x + \phi D(\cdot) \quad (9)$$

其中, x' 是 x 变异后的值, $D(\cdot)$ 是一个概率分布的随机变量。当高斯分布和柯西分布有相同的步长时,柯西分布可以得到更大的扰动,这意味着柯西分布比高斯分布有更高的概率跳出局部最优,因此本文选择了柯西分布^[5]。柯西分布的概率方程定义为:

$$g(x) = \frac{a}{\pi * (x^2 + a^2)} \quad (10)$$

这里 a 的取值 0.2。为表达方便,将带变异因子的 QPSO 算法表示为 QPSO-M 算法。

QPSO-M 算法可以如下描述:

初始化种群中粒子的位置向量

do

$$\beta = (1.0 - 1.5) * (\text{MAXITER} - t) / \text{MAXITER} + 0.5$$

for $i=1$ to 种群中粒子数目 M

if $f(X_i) < f(P_i)$ then $P_i = X_i$

$$P_g = \min(P_i)$$

计算 $mbest$

if $P_m < \text{rand}(0,1)$

$$mbest = \text{Cauchy}(mbest)$$

for $d=1$ to 维数 D

$$\varphi = \text{rand}(0,1)$$

$$p_{id} = \varphi * P_{id} + (1 - \varphi) * P_{gd}$$

$$u = \text{rand}(0,1)$$

if $\text{rand}(0,1) < 0.5$

$$X_{id} = p_{id} - \beta * mbest_{id} - X_{id} * \ln(1/u)$$

else

$$X_{id} = p_{id} + \beta * mbest_{id} - X_{id} * \ln(1/u)$$

直到满足算法终止条件

对 $gbest$ 参数进行变异的 QPSO-M 算法和上述算法类似,唯一的不同是柯西分布函数是对 $gbest$ 参数变异。 P_m 表示随优化函数变化的变异概率。

4 基于 QPSO-M 算法的 JSSP

4.1 基于 GA、PSO 和 QPSO 算法的 JSSP 分析

Cai L W 等在 2000 年证明了 GA 特别适用于 JSSP,GA 使用了简单的编码和新的交叉、变异操作。GA 在调度领域中已经得到了比较广泛的应用和发展,几乎每一种调度问题都有人用 GA 求解,因此可以认为 GA 的适用性是比较好的。

基于 GA 的调度理论虽然很成熟,但大都是在简化实际调度问题基础上进行研究的,因此与解决实际调度问题还有相当大的差距。并且 GA 有自身的缺陷,它在解决大规模的 JSSP 时存在着进化速度过慢和过早收敛的局限,且随着问题规模的增大,算法有效性呈下降趋势。

据 PSO 算法的基本原理,将 PSO 算法应用于 JSSP 具有理论上的可行性,实例仿真也证明了 PSO 算法的有效性,PSO 算法在求解 JSSP 时取得了比 GA 更好的调度效果。

但在实际的调度中,PSO 算法还是会经常出现陷入局部极值的早熟现象,PSO 算法并不能从根本上解决早熟收敛问题。

QPSO 算法应用于 JSSP 时虽然比 PSO 算法具有更好的全局收敛能力,也更易于得到调度的最优解。但是相关实验证明,QPSO 算法仍有可能会出现早熟现象而无法得到全局最优解。

4.2 粒子的编码设计

针对 GA、PSO 算法和 QPSO 算法的上述缺陷,本文提出将 QPSO-M 算法应用于 JSSP。QPSO-M 算法中的粒子对应于 GA 中的染色体,采用基于操作的编码方式。因此,对于一个 n 个工件在 m 台机器上加工的 JSSP,问题的每一个调度均被编码成一个由 $n \times m$ 个代表工序的基因所组成的染色体串,该染色体

串对应所有工件的所有工序在机器上的一个排列,其中各工件号均出现 m 次。解码时先将染色体串转化为一个有序的操作表,然后基于此操作表以及工艺约束条件对各操作以最早允许的加工时间逐一进行加工,从而产生调度方案。

4.3 QPSO-M 算法的流程设计

对于一个 n 个工件在 m 台机器上加工的调度问题,假设种群的粒子数为 s ,采用 QPSO-M 算法对问题进行求解时,将该问题中的每个调度编码为一个 $m \times n$ 的矩阵,以此矩阵作为算法的粒子进行进化,由此在一个 $s \times m \times n$ 的三维空间内搜索最优调度。算法根据种群的规模,按照上述粒子的结构随机产生一定数目的粒子组成种群,不同的粒子代表一个不同的调度,同时初始化 $pbest$ 、 $gbest$ 。

如前所述,调度的目标函数是 $\max(S_{i,ki} + t_{i,ki})$ 。为了使得到的调度满足第一个约束条件,在目标函数中引入一个惩罚因子 $PNSH(S_{i,k} - S_{i,k-1} - t_{i,k-1})$ 。最后,得到的目标函数是:

$$A \times \max(S_{i,ki} + t_{i,ki}) + B \times \sum_i \sum_k PNSH(S_{i,k} - S_{i,k-1} - t_{i,k-1})$$

其中, A 和 B 是权重。以此作为目标函数,评价种群中的所有粒子,从中找到最佳粒子,并判断是否需要更新种群的 $pbest$ 与 $gbest$ 。然后计算种群的 $mbest$,并根据变异概率 P_m 对 $mbest$ 进行变异操作。按照 QPSO-M 算法的进化方程更新每一个粒子的位置向量,以此产生新的粒子。如此反复进行直到满足算法终止条件。算法的终止条件有两种,一种是当目标函数的值小于一个给定值时算法终止,另外可以限定算法的迭代次数作为终止条件。

最后,基于 QPSO-M 算法的 JSSP 的流程描述如下:

步骤 1 初始化种群,根据调度规则设定各粒子的随机位置,并初始化 P_i 和 P_g ;

步骤 2 根据当前迭代代数动态调整收缩扩张系数,从而实现算法搜索空间由全局逐步过渡到局部;

步骤 3 计算每一粒子的适应度值,并确定是否更新 P_i 和 P_g ;

步骤 4 计算 $mbest$ 的值;

步骤 5 根据 P_m 的值判断是否对 $mbest$ 进行变异操作;

步骤 6 按种群的进化方程生成新的粒子;

步骤 7 判断是否满足算法的终止条件,若满足则转到步骤 8,不满足则转到步骤 2;

步骤 8 输出最佳调度,算法结束。

5 实例仿真

以某加工车间的工件加工为例,要在 10 台机器上加工 10 个工件,机器加工的工序集矩阵 $m(i,k)$ 及对应于工序集的机器加工工件的时间矩阵 $t_{i,k}$ 如下所示。

$$m(i,k) = \begin{pmatrix} 1 & 2 & 5 & 3 & 7 & 6 & 4 & 9 & 8 & 10 \\ 3 & 6 & 4 & 10 & 8 & 2 & 7 & 1 & 9 & 5 \\ 2 & 1 & 9 & 3 & 10 & 7 & 8 & 6 & 5 & 4 \\ 4 & 6 & 8 & 7 & 1 & 9 & 2 & 3 & 5 & 10 \\ 7 & 6 & 10 & 9 & 5 & 1 & 2 & 3 & 4 & 8 \\ 3 & 8 & 9 & 4 & 6 & 7 & 1 & 2 & 5 & 10 \\ 3 & 7 & 4 & 9 & 10 & 1 & 2 & 5 & 6 & 8 \\ 3 & 4 & 10 & 8 & 6 & 5 & 2 & 7 & 1 & 9 \\ 10 & 7 & 4 & 9 & 3 & 8 & 2 & 6 & 5 & 1 \\ 3 & 10 & 4 & 8 & 6 & 5 & 9 & 7 & 1 & 2 \end{pmatrix}$$

$$t_{i,k} = \begin{pmatrix} 6 & 3 & 2 & 8 & 3 & 3 & 4 & 8 & 2 & 6 \\ 1 & 5 & 3 & 8 & 6 & 9 & 2 & 5 & 5 & 9 \\ 9 & 5 & 7 & 2 & 9 & 2 & 3 & 4 & 4 & 8 \\ 5 & 6 & 9 & 5 & 4 & 5 & 8 & 4 & 9 & 9 \\ 2 & 6 & 10 & 6 & 1 & 1 & 2 & 8 & 3 & 7 \\ 3 & 9 & 4 & 6 & 4 & 2 & 12 & 3 & 2 & 9 \\ 3 & 1 & 12 & 4 & 2 & 8 & 3 & 2 & 8 & 7 \\ 3 & 5 & 4 & 3 & 4 & 10 & 5 & 2 & 3 & 11 \\ 8 & 1 & 12 & 4 & 2 & 8 & 3 & 8 & 8 & 7 \\ 6 & 5 & 4 & 3 & 4 & 10 & 5 & 6 & 3 & 11 \end{pmatrix}$$

分别用 GA、PSO 算法、QPSO 算法和 QPSO-M 算法等四种算法来求解此调度问题,进而比较这些算法在求解大规模 JSSP 时的性能。结合算法和调度的效率综合考虑,并经多次仿真测试,设 GA 的交叉概率 $P_c=0.9$,变异概率 $P_m=0.4$;PSO 算法的加速常数 $c_1=c_2=1.85$,惯性权重 ω 从 0.8 到 0.4 线性减小;QPSO 算法和 QPSO-M 算法的收缩扩张系数 β 都是从 1.0 到 0.5 线性减小。四种算法的种群规模均为 30,迭代次数均为 600,均分别运行 50 次。这些算法得到的最小完工时间、平均最小完工时间和平均运行时间等如表 1 所示;分别运行 50 次后得到的平均收敛曲线如图 1 所示。

表 1 四种算法的调度结果

算法	最小完工时间	平均最小完工时间	平均运行时间
GA	113	127.46	1 785
PSO	101	109.39	1 196
QPSO	98	101.63	817
QPSO-M	96	98.52	829

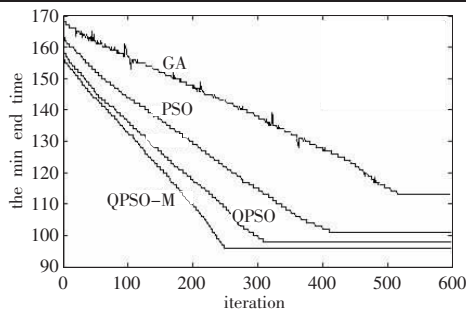


图 1 四种算法的收敛曲线图

(上接 36 页)

熵的信息测度。作为相似性测度,对多模态图像进行刚体配准实验,在多方面对这些测度进行了对比分析。实验表明,在不损失配准精度的情况下,当 α 或 t 取接近 1 的数值 ($SRI_{0.9}$ 或 $GMI_{0.9}$) 时,基于广义熵的信息测度的收敛性能优于归一化互信息测度,有效地克服了互信息对噪声的敏感性。不同的插值方案和搜索策略对测度的影响是下一步的研究重点。

参考文献:

- [1] Collignon A, Maes F, Delaere D, et al. Automated multimodality image registration using information theory[J]. Information Processing in Medical Imaging, 1995: 263-274.
- [2] Pluim J P W, Maintz J B A, Viergever M A. Mutual information based registration of medical images: a survey[J]. IEEE Trans Med Imag, 2003, 22(8): 986-1004.
- [3] Liu Chang-chun, Li Ke, Liu Zhong-guo. Medical image registration by maximization of combined mutual information and edge correlative deviation[C]//IEEE Conference on EMBS, 2005: 6379-6382.

比较这四种算法的调度结果,可以看出 QPSO-M 算法在最小完工时间和平均最小完工时间两个指标均要优于其他三种算法,而 50 次运行的平均运行时间指标也优于 GA 和 PSO 算法, QPSO-M 算法和 QPSO 算法具有相近的运行速度。由此证明了 QPSO-M 算法在求解 JSSP 时的有效性,算法也取得了比其他三种算法更好的调度效果。通过比较四种算法的收敛曲线图,可以看出 QPSO-M 算法的收敛速度更快,全局收敛能力更强,由此证明了该算法在收敛性能及解决 JSSP 上的优越性。

6 结束语

本文在讨论典型 JSSP 的基础上,提出了将带变异因子的 QPSO 算法用于求解 JSSP。QPSO-M 算法相比于 GA、PSO 算法等计算简单、收敛速度快,相比 QPSO 算法可以更有效地防止算法早熟,有利于发现更优解,从而具有更好的全局收敛性能。该算法搜索成功率远远高于其它算法,这对于求解大规模的 JSSP 是十分有利的。仿真实例将 QPSO-M 算法应用于求解复杂的 JSSP,调度结果显示了该算法具有良好的调度效果,证明了算法的有效性。

参考文献:

- [1] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003.
- [2] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]//Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, 1998. NJ: IEEE Press, 1998: 69-73.
- [3] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization[C]//Proceedings of CEC 1999, Piscataway, NJ: IEEE Press, 1999: 1951-1957.
- [4] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]//Proceedings of 2004 Congress on Evolutionary Computation, 2004: 325-331.
- [5] Yao X, Liu Y. Fast evolutionary strategies[C]//Proc 6th Conf Evolutionary Programming, 1997: 151-161.
- [6] Sun J, Xu W B, Feng B A. A global search strategy of quantum-behaved particle swarm optimization[C]//Proceedings of IEEE conference on Cybernetics and Intelligent Systems, 2004: 111-116.
- [7] Pluim J P W, Maintz J B A, Viergever M A. F-Information measures in medical image registration[J]. IEEE Trans Med Imag, 2004, 23(12): 1508-1516.
- [8] Wachowiak M P, Smolnikova R, Tourassis G D, et al. Similarity metrics based on nonadditive entropies for 2D-3D multimodal biomedical image registration[C]//SPIE 2003: Medical Imaging, 2003, 5032: 1090-1100.
- [9] Bardera A, Feixas M, Boada I. Normalized similarity measures for medical image registration[C]//SPIE 2004: Medical Imaging, 2004, 5370: 108-118.
- [10] Studholme C, Hill D L G, Hawkes D J. An overlap invariant entropy measure of 3D medical image alignment[J]. Pattern Recognit, 1999, 32(1): 71-86.
- [11] Rényi A. On measures of entropy and information[C]//Proc Fourth Berkeley Symp Math Stat and Probability, 1961, 1: 547-561.
- [12] Arimoto S. Information-theoretic considerations on estimation problems[J]. Information and Control, 1971, 19: 181-190.
- [13] Evans A C. BrainWeb: online simulated brain database[DB/OL]. [2007-03-01]. <http://www.bic.mni.mcgill.ca/brainweb>.