文章编号: 1001-0920 (2016) 09-1678-07 **DOI:** 10.13195/j.kzyjc.2015.0606

量子行为引力搜索算法

曹茂俊[1,2], 李盼池[2], 尚福华[2]

(1. 中国石油勘探开发研究院，北京 100083；2. 东北石油大学 计算机与信息技术学院，黑龙江 大庆 163318)

摘 要: 为提高引力搜索算法的优化能力, 通过在原始算法中融合量子计算, 提出一种量子行为引力搜索算法. 该算法采用类似量子行为粒子群优化的寻优机制, 在每步迭代中, 计算个体适应度, 根据适应度计算个体质量, 取前 K 个质量最大的个体作为候选集. 采用轮盘赌方法在候选集中选择一个作为Delta势阱的中心, 调整其他个体向该中心移动完成一步优化, 在优化过程中使 K 值单调下降, 以期达到探索和开发的平衡. 标准函数极值优化的实验表明, 所提出的算法比原算法在优化能力和优化效率两方面都有明显提高.

关键词: 引力搜索；量子势阱；智能优化；算法设计

中图分类号: TP18 **文献标志码:** A

Quantum-behaved gravitational search algorithm

CAO Mao-jun[1,2], *LI Pan-chi*[2], *SHANG Fu-hua*[2]

(1. Research Institute of Petroleum Exploration & Development，Beijing 100083，China；2. School of Computer and Information Technology，Northeast Petroleum University，Daqing 163318，China．Correspondent：SHANG Fu-hua, E-mail：lipanchi69@126.com)

Abstract: To enhance the optimization performance of the gravitational search algorithm, by introducing quantum computing to the original algorithm, a quantum-behaved gravitational search algorithm is proposed. The optimization strategies of the proposed algorithm are similar to the quantum-behaved particle swarm. In each of iteration, the fitness of each individual is evaluated, and then the quality of each individual is calculated based on its fitness. The first K greatest individuals are taken as a candidate set, in which an individual is randomly selected with roulette and is taken as the center of the Delta potential well. By adjusting other individuals move to the potential well centre, a single-step optimization is completed. In the process of optimization, the value of K is made decreasing monotonically to achieve a balance of exploration and exploitation. The experimental results on benchmark functions extreme optimization show that the proposed algorithm is obviously superior to the original one in performance and efficiency.

Keywords: qravitational search；potential well；intelligent optimization；algorithm design

0 引 言

万有引力搜索算法(GSA)是由Esmat等[1]于2009年提出的一种源于对物理学中的万有引力进行模拟的新的优化搜索技术, 它通过群体中各粒子之间万有引力相互作用产生的群体智能指导优化搜索. 在该算法中, 个体是受牛顿引力作用的质量集合, 其移动方式受牛顿运动定律支配. 文献[1]给出的结果表明, 算法在收敛速度和收敛精度上明显优于粒子群算法(PSO)[2]、遗传算法(GA)[3]等其他智能优化算法. 因其概念简单、易于实现, 目前已引起国内外众多学者的广泛关注, 并已在生产调度[4]、光学工程[5]和电磁系统[6]等领域获得了较为成功的应用. 在标准的GSA中, 一个粒子(个体)由质量和位置描述, 它们决定了个体移动的轨迹. 在量子力学中, 粒子沿着确定的轨迹移动是没有意义的, 因为根据量子力学的测不准原理, 粒子的位置和速度不能同时确定. 这表明研究启发式搜索算法和量子计算这两种新型计算的融合是有意义的.

自引入量子计算以来, 提出了很多融合量子机制的智能搜索算法, 这些算法可分为两类: 一类是

收稿日期: 2015-05-13；**修回日期:** 2015-12-09.

基金项目: 国家自然科学基金项目(61170132)；中国石油天然气集团公司重大专项项目(2013E-3809)；国家科技重大专项项目(2016ZX05019).

作者简介: 曹茂俊(1978−), 男, 副教授, 博士生, 从事智能计算在测井曲线识别中应用的研究；李盼池(1969−), 男, 教授, 博士生导师, 从事量子衍生智能优化算法等研究.

只能在量子计算机上运行的量子算法, 开始于20世纪90年代, 典型的有量子搜索算法、因子分解算法, 这些算法在解决某些问题时比经典算法更有效; 另一类是面向经典计算机的量子启发式搜索算法, 典型的有量子衍生搜索算法和量子行为搜索算法. 量子衍生搜索算法的特点是利用某些量子力学或量子计算的原理, 如干涉、相干、叠加、概率幅、量子门、测量等, 用与概率相关的各种态描述系统的行为. 量子行为搜索算法[7-10]假定每个粒子存在于一个带有势阱的量子空间, 根据量子力学原理, 每个粒子都将按概率向着代表最优解的势阱中心移动, 依此实现进化搜索.

本文提出一种量子行为引力搜索算法 (QBGSA), 采用与量子行为粒子群相似的寻优策略, 通过使个体向着质量最大的粒子移动实现进化搜索, 所提出的算法只有一个控制参数, 容易调整. 标准函数极值优化的实验结果表明, 在优化能力和优化效率两方面, 所提出的算法比原算法有明显提高.

1　引力搜索算法

标准引力搜索算法受牛顿引力和运动定律启发于2009年提出, 其基本思想是: 在宇宙中的所有粒子之间均存在万有引力, 两个粒子的质量越大且距离越近, 引力越大; 两个粒子之间的距离越远, 引力越小. 粒子将在受其他粒子引力的合力方向上产生加速度, 如图1所示.

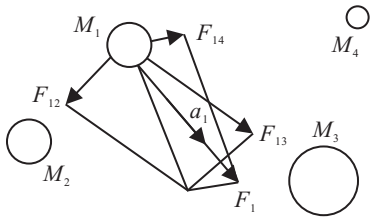

图 1　4个粒子之间的引力

图1中, 4个粒子之间均存在万有引力, 其中 M_1 受其他3个粒子的引力分别为 F_{12}、F_{13}、F_{14}, 合力为 F_1, 加速度为 a_1.

在GSA中, 个体的质量通过当前位置的适应度计算, 如下式所示:

$$m_i(t)=\frac{\mathrm{fit}_i(t)-\mathrm{worst}(t)}{\mathrm{best}(t)-\mathrm{worst}(t)}, \tag{1}$$

$$M_i(t)=\frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}. \tag{2}$$

其中: N 为种群规模, $\mathrm{fit}_i(t)$ 为第 i 个体在第 t 次迭代的适应度. 以最小值优化为例, $\mathrm{best}(t)$ 和 $\mathrm{worst}(t)$ 分别定义为

$$\begin{aligned}&\mathrm{best}(t)=\min\{\mathrm{fit}_j(t)\},\ j=1,2,\cdots,N;\\&\mathrm{worst}(t)=\max\{\mathrm{fit}_j(t)\},\ j=1,2,\cdots,N.\end{aligned} \tag{3}$$

根据牛顿运动定律, 计算个体 i 在第 d 维的加速度为

$$a_i^d=G(t)\sum_{j\in K_{\mathrm{best}},\ j\neq i}\mathrm{rand}_j\frac{M_j(t)(x_j^d(t)-x_i^d(t))}{R_{ij}(t)+\varepsilon}. \tag{4}$$

其中: $d=1,2,\cdots,D$, $i=1,2,\cdots,N$; $x_i^d(t)$ 为第 i 个体在第 d 维的位置; ε 为小正数; $R_{ij}(t)$ 为个体 i 与个体 j 之间的欧氏距离; rand_j 为 $(0,1)$ 区间均匀分布的随机数; t 时刻的引力常数为

$$G(t)=G_0\exp(-\alpha t/\max\_N), \tag{5}$$

G_0 为初值, t 为迭代步数, $\max\_N$ 为最大迭代步数; K_{best} 为前 K 个质量最大粒子的集合, 该集合也为代数 t 的函数, 最大值为 K_0, 随代数 t 线性下降, 终值为 K_1, 有

$$K_{\mathrm{best}}=\lfloor K_0+(K_1-K_0)t/\max\_N\rfloor. \tag{6}$$

粒子在相邻两代 (t 与 $t+1$) 之间的位置和速度更新为

$$v_i^d(t+1)=\mathrm{rand}_i v_i^d(t)+a_i^d(t), \tag{7}$$

$$x_i^d(t+1)=x_i^d(t)+v_i^d(t+1). \tag{8}$$

个体的位置即为优化问题的候选解, 上述过程反复迭代直到满足终止条件 (通常为一个足够好的适应度值或一个预先设定的最大迭代步数). 引力搜索算法的算法流程如下.

Step 1: 种群初始化

$$\boldsymbol{X}_i=(x_i^1,x_i^2,\cdots,x_i^D),\ i=1,2,\cdots,N;$$

Step 2: 计算个体适应度;

Step 3: 按式 (2) 更新 $M(t)$, 按式 (3) 更新 $\mathrm{best}(t)$ 和 $\mathrm{worst}(t)$, 按式 (5) 和 (6) 更新 $G(t)$ 和 K_{best};

Step 4: 按式 (4) 和 (7) 计算加速度和速度;

Step 5: 按式 (8) 产生 $\boldsymbol{X}_i(t+1)$, $i=1,2,\cdots,N$;

Step 6: 重复 Step 2 ~ Step 5, 直到满足终止条件.

2　量子行为引力搜索算法

2.1　量子势阱建模

在量子力学中, 粒子动态行为一般用如下薛定谔方程描述:

$$j\hbar\frac{\partial}{\partial t}\Psi(r,t)=\Big(-\frac{\hbar^2}{2m}\nabla^2+V(r)\Big)\Psi(r,t). \tag{9}$$

其中: $\hbar$ 为普朗克常数, m 为粒子质量, $V(r)$ 为势场能量分布函数. 在薛定谔方程中, 未知量是波函数 $\Psi(r,t)$, 根据波函数的统计诠释, 该函数幅度的平方为粒子在 r 处出现的概率密度.

量子势阱的建模思想为, 首先选择某种不显含时间 t 的势阱 $V(r)$; 然后通过求解薛定谔方程得到变量分离形式的波函数 $\Psi(r)$, 进而得到粒子在势场中出现

的概率密度函数 $|\Psi(r)|^2$; 最后通过将势阱中心设置为质量最大粒子的位置, 并合理设计势阱参数, 使其他粒子以大概率逼近处于势阱中心的最优粒子. 记势阱中心为 P, 第 k 次迭代粒子的位置为 x_k, 以 Delta 势阱为例, 粒子位置的迭代方程可用下式[7]描述:

$$x_{k+1} = P \pm \alpha|x_k - P|\ln(1/u)|. \tag{10}$$

2.2 量子行为引力搜索算法

量子行为引力搜索算法的设计思路是: 首先按普通 GSA 计算粒子质量, 建立由前 K 个质量最大粒子组成的候选解集 $K_{\rm best}$; 然后用轮盘赌策略在候选集中随机选取一个粒子作为势阱中心; 最后采用 Delta 势阱的迭代方程更新粒子位置.

关于 $K_{\rm best}$的更新, 采用指数下降的方法. 记种群规模为 N, 更新为

$$K_{\rm best} = \lfloor N \times (0.1)^{t/\max\_N} \rfloor. \tag{11}$$

记第 t 代用轮盘赌选择的粒子为 $X_{\rm best}(t)$, 种群平均粒子为

$$X_{\rm mean}(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t).$$

本文采用随机选择如下两式之一更新粒子位置:

$$x_i^d(t+1) = x_{\rm best}^d(t) \pm \alpha|x_i^d(t) - x_{\rm best}^d(t)|\ln\left(\frac{1}{u}\right), \tag{12}$$

$$x_i^d(t+1) = x_i^d(t) \pm \alpha|x_i^d(t) - x_{\rm mean}^d(t)|\ln\left(\frac{1}{u}\right). \tag{13}$$

式 (12) 和 (13) 的正负号均等概率地随机选择. 由式 (11) ∼ (13) 可知, 在 QBGSA 中, 除了种群规模、变量维数、迭代步数等所有智能优化算法都需要事先设定的共性参数外, 体现 QBGSA 自身特性的控制参数只有一个 α. 另外, 值得指出, QBGSA 与现有的量子行为粒子群算法 (QPSO)[8,10]有两点不同:

1) 在 QBGSA 中, 每步迭代用作势阱中心的粒子来自于由前 K 个质量最大粒子组成的候选集 (其中 K 随迭代步数单调下降), 具体采用哪个粒子由轮盘赌策略决定. 这样做的目的是使算法在初期有较强的全局探索能力, 在后期有较强的局部开发能力, 而 QPSO 自始至终分别采用自身最优粒子和自身最优粒子的算术平均作为势阱中心.

2) 关于粒子位置的更新. QBGSA 除采用由量子势阱建模得到的更新式外, 还采用另一个类似的更新式. 该式与文献 [8,10] 的不同之处在于, 直接使用当代种群粒子 (而不是自身最优粒子) 的算术平均作为势阱中心, 每步迭代两个公式等概率地随机选择使用.

对于迭代式 (12) 和 (13) 的合理性给出如下解释:

1) 在根据势阱理论得到的迭代式 (10) 中, 只需将势阱中心 P 替换为全局最优粒子 $x_{\rm best}^d$ 即可得到式 (12), 在势阱中粒子向着势阱中心移动, 式 (12) 使粒子向着全局最优粒子移动, 因此式 (12) 是合理的.

2) 式 (13) 的作用是增强种群多样性, 避免早熟收敛, 式中的正负号等概率地随机选取, 可使粒子等概率地向中心粒子或偏离中心粒子移动, 具体作用如下所示:

$$x_i^d \begin{cases} \text{远离 } x_{\rm mean}^d, \text{ 取 } +, \ x_i^d > x_{\rm mean}^d; \\ \text{靠近 } x_{\rm mean}^d, \text{ 取 } -, \ x_i^d > x_{\rm mean}^d; \\ \text{靠近 } x_{\rm mean}^d, \text{ 取 } +, \ x_i^d < x_{\rm mean}^d; \\ \text{远离 } x_{\rm mean}^d, \text{ 取 } -, \ x_i^d < x_{\rm mean}^d. \end{cases} \tag{14}$$

在迭代过程中, 若单纯使用式 (12) 则容易导致早熟收敛. 因为式 (12) 和 (13) 等概率随机选取, 两式中的正负号也等概率随机选取, 式 (13) 的独特之处在于允许粒子以 1/4 的概率偏离平均粒子, 从而能够在一定程度上抑制早熟收敛, 所以式 (13) 是合理的.

QBGSA 的算法流程如下.

Step 1: 种群初始化

$$\boldsymbol{X}_i = (x_i^1, x_{i2}^2, \cdots, x_i^D), \ i = 1, 2, \cdots, N;$$

Step 2: 计算个体适应度;

Step 3: 按式 (2) 更新 $M(t)$, 按式 (3) 更新 $\rm best(t)$ 和 $\rm worst(t)$, 按式 (11) 更新 $K_{\rm best}$;

Step 4: 采用轮盘赌策略, 在候选集 $K_{\rm best}$ 中选择一个粒子作为势阱中心;

Step 5: 在式 (12) 和 (13) 中, 等概率选择其一, 产生下一代个体 $\boldsymbol{X}_i(t+1)$, $i = 1, 2, \cdots, N$;

Step 6: 重复 Step 2 ∼ Step 5, 直到满足终止条件.

3 对比实验

3.1 测试函数

采用 CEC2013 定义的 28 个标准函数[9]作为优化对象, 如表 1 所示. 其中: (S) 表示该函数的变量是可分离的, (N) 表示变量是不可分离的, n 表示该函数由其他 n 个基本函数复合而成, 所有函数均为极小值优化.

3.2 参数设置

为体现所提出算法的优良性能, 所有函数的优化结果将与普通引力搜索算法 (GSA)、文献 [10] 中的量子行为粒子群算法 (QPSO) 进行对比. 为保证对比客观公正, QBGSA、GSA、QPSO 采用相同的种群规模和迭代步数 (即函数评估次数). 为尽量消除对比结果的随机性并增强对比结果的可信度, 所有函数均取 $D = 50$ 维和 $D = 100$ 维两种情况, 且每个函数均用 3 种算法各自独立优化 30 次, 取 30 次优化的平均

表 1　28 个 CEC'13 测试函数简介

	序号	名 称	最小值
单峰函数	1	Sphere (S)	−1 400
	2	Rotated High Conditioned Elliptic (N)	−1 300
	3	Rotated Bent Cigar (N)	−1 200
	4	Rotated Discus (N)	−1 100
	5	Different Powers (S)	−1 000
基本多峰函数	6	Rotated Rosenbrock's (N)	−900
	7	Rotated Schaffers F7 (N)	−800
	8	Rotated Ackley's (N)	−700
	9	Rotated Weierstrass (N)	−600
	10	Rotated Griewank's (N)	−500
	11	Rastrigin's (S)	−400
	12	Rotated Rastrigin's (N)	−300
	13	Non-Continuous Rotated Rastrigin's (N)	−200
	14	Schwefel's (N)	−100
	15	Rastrigin's Schwefel's (N)	100
	16	Rotated Katsuura (N)	200
	17	Lunacek Bi_Rastrigin (S)	300
	18	Rotated Lunacek Bi_Rastrigin (N)	400
	19	Expanded Griewank's plus Rosenbrock's (N)	500
	20	Expanded Scaffer's F6 (N)	600
复合函数	21	Composition Function 1 (n=5,Rotated) (N)	700
	22	Composition Function 2 (n=3,Unrotated)(S)	800
	23	Composition Function 3 (n=3,Rotated) (N)	900
	24	Composition Function 4 (n=3,Rotated) (N)	1 000
	25	Composition Function 5 (n=3,Rotated) (N)	1 100
	26	Composition Function 6 (n=5,Rotated) (N)	1 200
	27	Composition Function 7 (n=5,Rotated) (N)	1 300
	28	Composition Function 8 (n=5,Rotated) (N)	1 400
搜索范围: $[-100, 100]^D$			

结果作为对比指标.

3 种算法的共性参数设置如下: 当维数 $D = 50$ 时, 种群规模取 50; 当维数 $D = 100$ 时, 种群规模取 100; 终止条件为限定迭代步数, 两种维数下的迭代步数均取 10 000.

对于个性参数, QBGSA 只有一个 α, 经过多次实验, 在 $D = 50$ 和 $D = 100$ 两种情况下均取 $\alpha = 0.3$; GSA 的个性参数有两个, 经过多次实验, 万有引力常数 G 的初值取 $G_0 = 100$, G 单调下降的指数因子取 $\alpha = 10$; QPSO 的控制参数取 $\alpha = 0.5$.

3.3　优化结果对比

所有实验均在配置 Intel(R) Core(TM) i5-3470 CPU 3.20 GHz 处理器, 4.00 GB 内存, window 7 操作系统的计算机上, 采用 Matlab 7.0 编程实现. 为了凸显 QBGSA 的优化能力和优化效率, 对于 QBGSA、普通 GSA、文献 [10] 中的 QPSO, 同时给出平均优化结果对比和平均运行时间对比, 如表 2 和表 3 所示, 表中粗体表示对比结果获胜.

表 2　QBGSA、GSA、QPSO 的平均优化结果对比

序号	$D = 50$			$D = 100$		
	QBGSA	GSA	QPSO	QBGSA	GSA	QPSO
1	**−1.40e+3**	−1.39e+3	−1.39e+3	**−1.40e+3**	−1.39e+3	−1.39e+3
2	4.535e+6	**5.408e+5**	3.800e+7	1.426e+7	**4.815e+6**	1.250e+8
3	**3.945e+6**	1.294e+8	6.377e+9	**8.195e+6**	9.799e+9	3.19e+10
4	2.742e+5	8.965e+4	**3.345e+4**	5.726e+5	1.822e+5	**9.457e+4**
5	−9.99e+2	−9.99e+2	**−1.00e+3**	**−9.99e+2**	−9.98e+2	−9.98e+2
6	**−8.54e+2**	−8.18e+2	−8.27e+2	−6.65e+2	**−7.00e+2**	−6.19e+2
7	**−7.98e+2**	−7.54e+2	−7.11e+2	**−7.97e+2**	−6.81e+2	−6.77e+2
8	**−6.79e+2**	−6.78e+2	−6.78e+2	**−6.79e+2**	−6.78e+2	−6.78e+2
9	**−5.86e+2**	−5.60e+2	−5.38e+2	**−5.76e+2**	−4.81e+2	−4.51e+2
10	**−4.99e+2**	−4.98e+2	−4.94e+2	**−4.99e+2**	−4.98e+2	−3.50e+2
11	**−3.60e+2**	−7.68e+1	−1.61e+2	**−3.09e+2**	6.280e+2	3.040e+2
12	**−2.28e+2**	2.458e+2	1.034e+2	7.339e+2	1.608e+3	**7.330e+2**
13	**1.205e+1**	4.666e+2	2.226e+2	**8.396e+2**	2.054e+3	8.405e+2
14	**6.141e+2**	5.435e+3	1.157e+4	**1.618e+3**	1.334e+4	2.757e+4
15	**5.560e+3**	7.836e+3	1.415e+4	3.435e+4	**1.403e+4**	3.029e+4
16	2.073e+2	**2.000e+2**	2.034e+2	2.067e+2	**2.000e+2**	2.040e+2
17	3.941e+2	**3.870e+2**	6.637e+2	**5.105e+2**	8.627e+2	1.243e+3
18	9.307e+2	**4.845e+2**	8.394e+2	1.587e+3	1.474e+3	**9.121e+2**
19	5.057e+2	**5.045e+2**	5.262e+2	**5.110e+2**	5.332e+2	5.873e+2
20	6.236e+2	6.249e+2	**6.228e+2**	**6.499e+2**	6.500e+2	6.500e+2
21	**9.636e+2**	1.669e+3	1.725e+3	**1.070e+3**	1.100e+3	1.130e+3
22	**1.665e+3**	1.161e+4	1.336e+4	**2.474e+3**	2.463e+4	3.082e+4
23	**6.215e+3**	1.141e+4	1.577e+4	3.492e+4	3.343e+4	**2.226e+4**
24	**1.218e+3**	1.296e+3	1.289e+3	**1.224e+3**	3.183e+3	1.399e+3
25	**1.378e+3**	1.620e+3	1.508e+3	**1.447e+3**	2.369e+3	1.751e+3
26	1.499e+3	1.568e+3	**1.412e+3**	**1.562e+3**	1.889e+3	1.760e+3
27	**1.878e+3**	2.821e+3	2.854e+3	**1.970e+3**	4.754e+3	4.080e+3
28	**1.800e+3**	7.852e+3	1.807e+3	**4.974e+3**	1.608e+4	5.299e+3

由表 2 可见, 就平均优化结果而言, 28 个测试函数中, 当 $D = 50$ 时, QBGSA 优于普通 GSA 和 QPSO 的有 19 个, 劣于普通 GSA 的仅有 5 个, 劣于 QPSO 的仅有 4 个; 当 $D = 100$ 时, QBGSA 优于普通 GSA 和 QPSO 的有 20 个, 劣于普通 GSA 的仅有 4 个, 劣于 QPSO 的仅有 4 个. 这表明, QBGSA 不仅明显优于普通 GSA 和 QPSO, 而且随着维数增加, QBGSA 的优势略趋于明显.

下面考察 3 种算法的优化效率. 对于第 i 个函数, 设 3 种算法 30 次优化的平均时间分别为 T^i_{QBGSA}、T^i_{GSA}、T^i_{QPSO}, 则 QBGSA 与 GSA 的平均时间比值和 QBGSA 与 QSPO 的平均时间比值分别为

$$\frac{T_{\mathrm{QBGSA}}}{T_{\mathrm{GSA}}} = \frac{1}{28}\sum_{i=1}^{28}\frac{T^i_{\mathrm{QBGSA}}}{T^i_{\mathrm{GSA}}}, \tag{15}$$

$$\frac{T_{\mathrm{QBGSA}}}{T_{\mathrm{QPSO}}} = \frac{1}{28}\sum_{i=1}^{28}\frac{T^i_{\mathrm{QBGSA}}}{T^i_{\mathrm{QPSO}}}. \tag{16}$$

利用表 3 数据, 当 $D = 50$ 时, $T_{\mathrm{QBGSA}}/T_{\mathrm{GSA}} = 0.1344$, $T_{\mathrm{QBGSA}}/T_{\mathrm{QPSO}} = 1.1984$; 当 $D = 100$ 时,

表 3 QBGSA、GSA、QPSO 的平均运行时间对比 s

序号	$D=50$			$D=100$		
	QBGSA	GSA	QPSO	QBGSA	GSA	QPSO
1	36.74	379.6	**21.56**	114.4	609.1	**78.65**
2	39.54	382.2	**26.76**	136.7	626.2	**127.9**
3	40.37	384.7	**34.18**	147.1	632.6	**106.5**
4	38.45	381.6	**32.15**	131.4	622.8	**116.1**
5	36.85	380.7	**27.28**	116.1	615.8	**105.2**
6	38.17	382.1	**31.46**	128.6	625.4	**103.9**
7	48.33	388.0	**35.97**	177.2	643.4	**138.7**
8	44.71	385.0	**30.97**	167.1	640.1	**121.1**
9	**95.33**	472.7	134.6	**371.2**	731.7	570.5
10	40.62	383.5	**25.44**	138.9	627.9	**98.86**
11	41.28	382.7	**32.94**	133.9	625.9	**105.9**
12	45.88	387.6	**36.14**	175.4	648.3	**157.1**
13	45.56	389.0	**37.27**	174.8	647.6	**158.8**
14	41.53	396.5	**33.11**	135.5	626.1	**109.7**
15	43.06	401.6	**38.27**	149.5	633.3	**125.9**
16	**52.08**	455.3	119.6	**192.2**	718.4	495.7
17	39.21	397.2	**30.58**	125.9	620.5	**99.68**
18	42.13	400.8	**29.84**	152.4	625.0	**112.1**
19	38.67	398.2	**23.47**	131.8	610.1	**97.98**
20	41.14	399.7	**26.49**	157.4	614.6	**112.2**
21	55.63	411.4	**45.66**	230.7	649.3	**250.5**
22	57.04	406.3	**40.66**	196.2	619.8	**153.4**
23	61.51	414.1	**44.00**	237.4	642.0	**214.5**
24	**114.7**	510.6	168.0	**477.5**	753.3	748.9
25	**117.4**	510.8	169.0	**487.9**	753.9	749.9
26	**128.2**	517.7	187.5	**541.7**	773.7	850.8
27	**122.5**	512.5	184.3	**517.2**	765.6	811.1
28	**73.07**	424.0	75.15	**310.5**	668.9	344.0

$T_{\mathrm{QBGSA}}/T_{\mathrm{GSA}}=0.3238$, $T_{\mathrm{QBGSA}}/T_{\mathrm{QPSO}}=1.0902$. 这表明, QBGSA 的优化效率明显高于普通 GSA, 但略低于 QPSO. 为充分考察 QBGSA 的优化能力, 以 $D=50$ 为例, 将 QPSO 的限定步数增加到 QBGSA 的 1.5 倍 (即 15 000), 对每个测试函数独立优化 30 次, 然后利用每个函数的平均优化结果与 QBSGA 进行对比. 对比结果为: QBGSA 优于 QPSO 的函数有 20 个, 劣于 QPSO 的只有 7 个, 等于 QPSO 的有 1 个, 图 2 给出了对比结果.

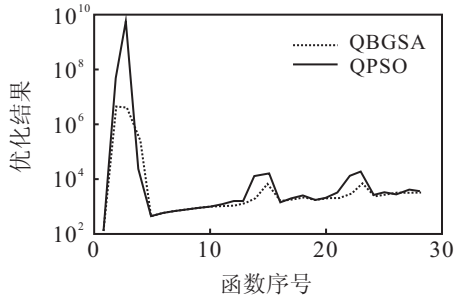

图 2 QBGSA (10^4步) 与 QPSO (1.5×10^4步) 优化结果对比

为使对比直观, 纵轴采用对数刻度, 考虑到负数不能取对数, 在绘图时, 两种算法的所有结果都增加一个常数 1 500. 对比结果表明, QBGSA 不仅在相同迭代步数下优于 QPSO, 而且在相近的优化时间下, QBGSA 也同样优于 QPSO.

虽然表 2 和表 3 的统计结果对算法性能提供了较为充分的对比, 但对于一个好的算法, 通常还需要进行算法优越性的显著性检验. 为了确定 BQGSA 是否获得了在统计意义上比其他算法更优的解, 对于 28 个测试函数, 根据 QBGSA、GSA、QPSO 分别优化 30 次的结果, 利用统计阈值 $\alpha=0.05$ 进行了威尔逊秩和检验. 该检验的原假设是: 对于同一函数, QBGSA 的优化结果和其他算法的优化结果没有差异. QBGSA 与两种对比算法基于威尔逊秩和检验的实验结果分别如表 4 和表 5 所示. 表 4 和表 5 中: P 表示原假设 (两种算法没有差异) 为真的概率; $T+$ 和 $T-$ 分别为 QBGSA 和对比算法的秩和; W 表示显著性结果, + 表示原假设被拒绝, QBGSA 以 95% 的显著性水平优

表 4 $D=50$ 时算法成对测试的显著性结果对比

序号	QBGSA VS. GSA				QBGSA VS. QPSO			
	P	T+	T−	W	P	T+	T−	W
1	7.57e-12	465	1 365	+	2.49e-07	486	1 344	+
2	3.01e-11	1 365	465	−	3.01e-11	465	1 365	+
3	1.28e-09	504	1 326	+	3.01e-11	465	1 365	+
4	3.01e-11	465	1 365	+	3.01e-11	1 365	465	−
5	2.89e-11	465	1 365	+	0.000 333	1 065	765	−
6	0.029 205	767	1 063	+	0.043 584	778	1 052	+
7	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
8	0.053 685	1 046	784	=	3.01e-11	465	1 365	+
9	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
10	3.01e-11	465	1 365	+	6.69e-11	473	1 357	+
11	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
12	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
13	3.01e-11	465	1 365	+	5.57e-10	495	1 335	+
14	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
15	5.49e-11	471	1 359	+	3.01e-11	465	1 365	+
16	3.01e-11	1 365	465	−	3.01e-11	465	1 365	+
17	0.122 350	1 020	810	=	3.01e-11	465	1 365	+
18	3.01e-11	1 365	465	−	3.01e-11	465	1 365	+
19	0.000 140	1 173	657	−	3.01e-11	465	1 365	+
20	2.95e-11	465	1 365	+	1.69e-08	1 297	533	−
21	9.64e-10	501	1 329	+	3.44e-10	489	1 341	+
22	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
23	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
24	3.33e-11	466	1 364	+	3.01e-11	465	1 365	+
25	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
26	0.001 952	705	1 125	+	0.000 398	1 155	675	−
27	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
28	2.93e-11	465	1 365	+	0.655 660	907	923	=
	+/=/−		22/2/4		+/=/−		23/1/4	

表 5　$D = 100$ 时算法成对测试的显著性结果对比

序号	QBGSA VS. GSA				QBGSA VS. QPSO			
	P	T+	T−	W	P	T+	T−	W
1	2.07e-11	465	1 365	+	2.07e-11	465	1 365	+
2	3.68e-11	1 363	467	−	3.01e-11	465	1 365	+
3	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
4	3.01e-11	465	1 365	+	3.01e-11	1 365	465	−
5	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
6	0.008 916	1 168	662	−	7.73e-06	612	1 218	+
7	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
8	0.684 320	887	943	=	3.01e-11	465	1 365	+
9	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
10	0.005084	725	1 105	+	3.01e-11	465	1 365	+
11	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
12	3.01e-11	465	1 365	+	0.589450	952	878	=
13	3.01e-11	465	1 365	+	0.739400	892	938	=
14	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
15	3.01e-11	1 365	465	−	3.01e-11	465	1 365	+
16	3.01e-11	1 365	465	−	3.01e-11	465	1 365	+
17	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
18	3.01e-11	465	1 365	+	2.87e-10	1 342	488	−
19	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
20	0.160 800	465	1 365	=	0.544030	493	1 337	=
21	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
22	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
23	3.01e-11	465	1 365	+	9.06e-08	1 277	553	−
24	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
25	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
26	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
27	3.01e-11	465	1 365	+	3.01e-11	465	1 365	+
28	3.01e-11	465	1 365	+	0.093341	801	1 029	=
	+/=/−		22/2/4		+/=/−		21/4/3	

于另一算法, “−”表示原假设被拒绝, QBGSA 以 95% 的显著性水平劣于另一算法, “=”表示原假设被接受, 两种算法没有统计差别; 最后一行给出每组算法中 QBGSA 优于、等于、劣于对比算法的函数个数. 显著性测试结果进一步验证了所提出算法的优势.

3.4　对优化结果的分析

对于上述实验结果, 给出如下理论分析:

1) QBGSA 的优化能力明显优于普通 GSA, 这是由于采用了量子势阱机制, 将经典 GSA 中粒子趋于向质量最大的粒子移动, 改进为量子势阱中的粒子向具有最低能量的势阱中心 (即最大质量的粒子) 移动. 粒子在量子势阱中的移动是概率性的, 除以大概率向势阱中心移动外, 还可能以小概率逃离势阱中心, 即模拟量子力学中的隧穿效应, 这实际上扮演了变异的角色, 有助于增强种群的多样性, 从而导致 QBGSA 的优化能力明显优于普通 GSA.

2) QBGSA 的优化效率明显高于普通 GSA, 这是由于 QBGSA 的计算量明显小于普通 GSA. 从算法结构看, QBGSA 仅比普通 GSA 多了一步轮盘赌选择操作, 但比普通 GSA 减少了计算引力常数、计算加速度、计算任意两个粒子之间的距离、计算速度这 4 种操作, 其中任意两个粒子之间的距离计算最为耗时, 直接导致普通 GSA 的低效率.

3) QBGSA 的优化能力明显高于 QPSO, 这是由于在 QBGSA 中, 采用了与 QPSO 不同的势阱中心设置方法, 同时增加一个新的位置更新式. 在 QBGSA 中, 每次迭代均从最优解候选集中用轮盘赌策略选择势阱中心, 同时提出一个基于粒子当前位置和平均位置的新的位置更新式, 以等概率从两个位置更新式中随机选择其一, 这些措施有效增强了种群多样性. 另外, 最优解候选集中粒子的个数随迭代步数单调下降, 也较好地保持了寻优过程中探索与开发之间的平衡.

4　实际应用

利用所提出算法解决洛阳拖拉机厂普通工人庄益敏提出的挂论问题, 问题表述为: 从齿数为 20, 21, $\cdots$, 100 的齿轮中, 选取两对齿轮, 使输出速度 (或传动比) 尽可能接近于 π. 该问题相当于在 20 与 100 之间, 找出 4 个整数 a、b、c、d, 使 $|\pi - (a \times b)/(c \times d)|$ 达到最小. 《工程手册》给出

$$\frac{ab}{cd} = \frac{52 \times 29}{20 \times 24} = \frac{377}{120}, \tag{17}$$

庄益敏发现

$$\frac{ab}{cd} = \frac{68 \times 62}{22 \times 61} = \frac{2\,108}{671},$$

比《工程手册》上的结果更好些, 他想知道是否还有更好的结果[11]. 事实上, 这是一个丢番图逼近问题的特例, 应用丢番图逼近理论可以得出全局最优解为

$$\frac{ab}{cd} = \frac{51 \times 77}{25 \times 50} = \frac{3\,927}{1\,250}.$$

本节采用 QBGSA 解决这一问题, 并与 GSA 进行对比.

该问题为 4 维整数优化, 在获得实数的最优解后必须按四舍五入转换为整数. 两种算法的参数设置为: 种群规模取 50, 限定迭代步数取 1 000, QBGSA 的控制参数 $\alpha = 0.8$; GSA 的万有引力常数 G 的初值取 $G_0 = 50$, G 单调下降的指数因子取 $\alpha = 10$. 两种算法分别独立优化 30 次, 优化结果如表 6 所示. 由优化结果可见, 对于 QBGSA, 30 次独立优化中, 11 次得到了全局最优解 (表 6 中黑体部分); 20 次优于庄益敏的结果; 所有 30 次优化均好于《工程手册》给出的结果. 对于 GSA, 30 次独立优化均未得到全局最优解, 且均劣于庄益敏的结果, 但有 13 次 (1、2、6、10、

11、16、20、21、22、23、24、28、29)优于《工程手册》的结果. 应用结果表明, 本文提出的算法不仅明显优于传统引力搜索算法, 而且对于优化求解实际问题也具有一定潜力.

表 6 挂轮问题QBGSA和GSA的优化结果对比

序号	QBGSA					GSA				
	目标值	a	b	c	d	目标值	a	b	c	d
1	**7.34e-06**	**51**	**77**	**25**	**50**	7.21e-05	68	79	38	45
2	1.05e-05	88	88	85	29	1.86e-05	89	85	43	56
3	1.29e-05	68	93	61	33	9.82e-05	76	78	51	37
4	**7.34e-06**	**51**	**77**	**50**	**25**	7.40e-05	78	87	45	48
5	1.77e-05	100	77	43	57	7.40e-05	87	78	48	45
6	**7.34e-06**	**77**	**51**	**25**	**50**	2.31e-05	75	79	46	41
7	1.05e-05	88	88	85	29	0.000 217	76	79	39	49
8	**7.34e-06**	**51**	**77**	**25**	**50**	7.40e-05	78	87	45	48
9	1.05e-05	88	88	85	29	7.40e-05	87	78	48	45
10	1.05e-05	88	88	29	85	2.31e-05	79	75	46	41
11	1.29e-05	93	68	61	33	2.31e-05	79	75	41	46
12	**7.34e-06**	**51**	**77**	**25**	**50**	0.000 117	91	86	47	53
13	1.05e-05	88	88	85	29	9.82e-05	78	76	37	51
14	1.77e-05	77	100	43	57	7.40e-05	78	87	45	48
15	1.86e-05	85	89	43	56	0.000117	86	91	47	53
16	1.29e-05	93	68	33	61	2.31e-05	75	79	41	46
17	**7.34e-06**	**51**	**77**	**25**	**50**	0.000 190	75	78	38	49
18	1.29e-05	62	68	61	22	7.40e-05	87	78	48	45
19	1.05e-05	88	88	29	85	0.000 356	71	78	43	41
20	1.05e-05	88	88	85	29	2.96e-05	63	75	32	47
21	**7.34e-06**	**51**	**77**	**25**	**50**	7.21e-05	68	79	38	45
22	**7.34e-06**	**77**	**51**	**50**	**25**	2.31e-05	79	75	46	41
23	1.05e-05	88	88	85	29	2.31e-05	75	79	41	46
24	**7.34e-06**	**77**	**51**	**50**	**25**	1.86e-05	85	89	56	43
25	**7.34e-06**	**51**	**77**	**25**	**50**	7.40e-05	87	78	40	54
26	1.05e-05	88	88	29	85	0.000 233	83	83	51	43
27	1.86e-05	89	85	43	56	0.000 117	91	86	47	53
28	1.77e-05	77	100	57	43	2.31e-05	75	79	41	46
29	**7.34e-06**	**77**	**51**	**50**	**25**	1.86e-05	85	89	56	43
30	1.29e-05	62	68	22	61	7.40e-05	87	78	45	48

5 结 论

经典引力搜索算法的优化能力, 特别是优化效率有待于进一步提高. 本文尝试将量子计算中的量子势阱机制引入经典引力搜索算法, 利用粒子在势阱中的移动, 代替其在万有引力场中的移动, 以期提高优化能力和优化效率. 实验结果表明, 这种尝试是成功的, 新算法呈现出明显优于原算法的性能, 从而揭示出, 在原算法中引入量子计算机制是提高其优化性能的有效途径.

参考文献(References)

[1] Esmat R, Hossein N, Saeid S. GSA: A gravitational search algorithm[J]. Information Sciences, 2009, 179(6): 2232-2248.

[2] Van B F, Engelbrecht A P. A study of particle swarm optimization particle trajectories[J]. Information Sciences, 2006, 176(8): 937-971.

[3] Tang K S, Man K F, Kwong S, et al. Genetic algorithms and their applications[J]. Signal Processing Magazine, 1996, 13(6): 22-37.

[4] Roy P K. Solution of unit commitment problem using gravitational search algorithm[J]. Int J of Electrical Power & Energy Systems, 2013, 53(1): 85-94.

[5] Saucer T W, Sih V. Optimizing nanophotonic cavity designs with the gravitational search algorithm[J]. Optics Express, 2013, 21(18): 20831-20836.

[6] 李超顺, 周建中, 肖剑. 基于改进引力搜索算法的励磁控制PID参数优化[J]. 华中科技大学学报, 2012, 40(10): 119-122.
(Li C S, Zhou J Z, Xiao J. PID parameter optimization of excitation control systems by using improved gravitational search algorithm[J]. J of Huazhong University of Science and Technology, 2012, 40(10): 119-122.)

[7] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]. Proc of Congress on Evolutionary Computation. New York: IEEE, 2004, 8: 325-331.

[8] Xi M L, Sun J, Xu W B. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position[J]. Applied Mathematics and Computation, 2008, 205(2): 751-759.

[9] Liang J J, Qu B Y, Sugamthan P N, et al. Problems definition and evaluation criteria for the CEC 2013 special session on real-parameter optimization[EB/OL]. Http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm.

[10] 方伟, 孙俊, 谢振平, 等. 量子粒子群优化算法的收敛性分析及控制参数研究[J]. 物理学报, 2010, 59(6): 3683-3694.
(Fang W, Sun J, Xie Z P, et al. Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter[J]. Acta Physica Sinica, 2010, 59(6): 3683-3694.)

[11] 华罗庚, 王元. 数学模型选谈[M]. 大连: 大连理工大学出版社, 2011: 49-65.
(Hua L G, Wang Y. Some topics on mathematical modeling[M]. Dalian: Dalian University of Technology Press, 2011: 49-65.)

(责任编辑: 郑晓蕾)