量子光学优化算法*

王金叶，马　良，刘　勇
（上海理工大学 管理学院，上海 200093）

摘　要：通过分析光学优化算法的特性，将光学优化算法中每个光源点都用量子空间中的一个粒子来描述，利用群体智慧的聚集性，建立了光学优化算法的量子势能场模型，并根据势能场模型的群体自组织性和协同性等特点提出了量子光学优化算法。通过对多个经典测试函仿真分析，得出量子光学优化算法在量子力学收敛理论下比光学优化算法控制参数少，设置简单，优化性能更好，收敛速度更快，优化了算法的收敛精度和速度。

关键词：量子力学；光学优化算法；量子势能场；仿真分析；优化性能

中图分类号：TP310.6　　**文献标志码**：A　　**文章编号**：1001-3695(2018)03-0654-04

doi:10.3969/j.issn.1001-3695.2018.03.003

Quantum-behaved optics inspired optimization

Wang Jinye, Ma Liang, Liu Yong
(*Business School, University of Shanghai for Science & Technology, Shanghai* 200093, *China*)

Abstract: By analyzing the character of optics optimization algorithm, this paper described every light point in the algorithm as a particle in the quantum space and found the quantum potential field model with the aggregation of swarm intelligence. Because of the points of self-organize and cooperation, this paper raised the quantum-behaved optics algorithm. Using several functions to analyze, the results show that the algorithm with the theory of quantum mechanics has fewer control parameters and simpler setup, and it has better performance and faster convergent speed than OIO algorithms.

Key words: quantum mechanics; optics inspired optimization (OIO); quantum potential energy field; simulation analysis; performance of optimization

0　引言

光学优化算法（optics inspired optimization，OIO）源于物理光学凸、凹面镜成像原理的优化算法[1]，由 Kashan 于 2015 年提出。其原理是把优化函数看做反射镜面，将函数的凸部分看做凸面镜，函数的凹部分看做凹面镜，每一个初始解则相当于一个初始光源点。每一个光源点的光线经函数镜面反射后，由反射面凸凹不平的性质，可得到正立或倒立的缩小的像，这一系列像点即可作为下一次寻优的初始光源点[2]。这种迭代模式在寻优的过程中可同时对任务进行探究，最终求得问题的最优解。与遗传算法[3]、粒子群算法[4]以及 NSGA Ⅱ算法相比，光学优化算法在处理优化问题上具有高度收敛精度与优势。

与其他智能算法一样，由于初始解的不确定性，OIO 算法存在易陷入局部最优、早熟收敛和后期收敛较慢等问题[5]。为克服这些缺陷，提高 OIO 算法中光源点的随机性和算法的全局搜索能力，受量子粒子群算法的启发，本文将 OIO 算法中每个光源点都用量子空间中的一个粒子来描述，以体现光源点的不确定性，并根据群体智慧的聚集性，建立量子势能场模型，然后根据群体自组织性和协同性等特点提出了 QOIO（quantum-behaved optics inspired optimization）算法。QOIO 算法控制参数少，设置简单，搜索能力强，具有较强的全局搜索能力。通过将 QOIO 算法对经典函数[6]的仿真分析，得出 QOIO 算法明显优于 OIO 算法。

1　QOIO 模型的建立

1.1　从 OIO 到 QOIO

文献[5,7,8]主要对光学算法的模型推导进行分析。在定义域内，随机选取初始光源点 $O_j^t=[o_{j1}^t o_{j2}^t \cdots o_{jn}^t]$，确定光源点的位置与高度，并在定义域内随机选取一变量 $F_{i_k}^t$，但 $i_k \neq j$ 时 $F_{i_k}^t \neq O_j^t$，假设 $F_{i_k}^t$ 是镜面顶点坐标。如果 $f(O_j^t) < f(F_{i_k}^t)$，则将对应的函数段看做凸面函数模型；如果 $f(O_j^t) > f(F_{i_k}^t)$，则将对应的镜面函数段看做是凹面函数模型。

一维空间中，假设第 j 光源点前面的镜子函数是凹面函数，而该光源点的函数位置 s_{j,i_k}^t 是集合 $U[f(O_j^t), f(O_j^t)+d_\infty]$ 中的随机数（d_∞ 是一无穷大的数）。运算初始时，设 $d_\infty = |\max\limits_{j=1,\cdots,NO}\{f(O_j^t)\}|$。光学算法中每矫正一次球面偏差，便更新一次 d_∞。由 p_{j,i_k}^t 的定义可得

$$p_{j,i_k}^t = s_{j,i_k}^t - f(F_{i_k}^t) \tag{1}$$

其中：$m_{i_k}^t$ 为曲面函数圆心到 X 轴的距离。凹面函数成像原理如图 1 所示。由图 1 可得，$m_{i_k}^t$ 也是 $U[f(O_j^t), f(O_j^t)+d_\infty]$ 中的随机数。由 r_{j,i_k}^t 的定义得

收稿日期：2016-12-03；**修回日期**：2017-01-17　**基金项目**：国家自然科学基金资助项目(71401106)；国家教育部人文社科规划基金项目(16YJA630037)；上海市高原学科建设项目；上海高校青年教师培养计划资助项目(ZZsl15018)；上海理工大学博士科研启动经费项目(1D-15-303-005)

作者简介：王金叶(1991-)，女，硕士，主要研究方向为系统工程、智能优化(Jinye_w@163.com)；马良(1964-)，男，教授，博导，博士，主要研究方向为系统工程、智能优化；刘勇(1982-)，男，讲师，博士(后)，主要研究方向为智能优化、系统工程.

$$r^t_{j,i_k} = m^t_{j,i_k} - f(F^t_{i_k}) \tag{2}$$

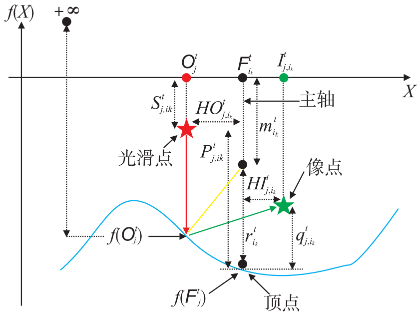

图1　凹面函数成像原理

当光源点前的函数为凸函数时,第 j 个光源点的位置 s^t_{j,i_k} 是 $U[f(F^t_{i_k}), f(F^t_{i_k})+d_\infty]$ 中的随机生成数。与凹面镜类似,由式(1)得出 p^t_{j,i_k},曲率圆心距 $m^t_{i_k}$ 是 $U[f(O^t_j)-d_\infty, f(O^t_j)]$ 内随机生成数,r^t_{j,i_k} 可由式(2)得出。与凹面函数不同的是,r^t_{j,i_k} 是一负值[5]。已知 p^t_{j,i_k} 和 r^t_{j,i_k},由镜面函数公式得出 q^t_{j,i_k}:

$$\frac{2}{r^t_{i_k}} = \frac{1}{p^t_{j,i_k}} + \frac{1}{q^t_{j,i_k}} \Rightarrow q^t_{j,i_k} = \frac{r^t_{i_k} p^t_{j,i_k}}{2p^t_{j,i_k} - r^t_{i_k}} \tag{3}$$

镜面函数的类型不同,则 q^t_{j,i_k} 的正负不同。

$$HO^t_{j,i_k} = \| O^t_j - F^t_{i_k} \| \tag{4}$$

由式(4)得,第 j 个光源点像长为

$$HI^t_{j,i_k} = -HO^t_{j,i_k} \frac{q^t_{j,i_k}}{p^t_{j,i_k}} \tag{5}$$

HI^t_{j,i_k} 可作为算法的搜索步长,第 j 个光源点在第 t 次迭代中像的位置即可以得出

$$I^t_{j,i_k} = F^t_{i_k} + HI^t_{j,i_k} \frac{(O^t_j - F^t_{i_k})}{\| O^t_j - F^t_{i_k} \|} = F^t_{i_k} - \frac{r^t_{i_k}}{2p^t_{j,i_k} - r^t_{i_k}} (O^t_j - F^t_{i_k}) \tag{6}$$

则 I^t_{j,i_k} 是定义域中一个新的搜索解。

光学启发式优化算法在镜面函数选择这一步中,利用轮盘赌适度分配的方法对解进行改进。轮盘赌适应度函数如下:

$$\mathrm{fit}(O^t_j) = \begin{cases} \dfrac{1}{1+f(O^t_j)} & f(O^t_j) \geqslant 0 \\ 1+\mathrm{abs}(f(O^t_j)) & f(O^t_j) < 0 \end{cases} \tag{7}$$

在函数搜索中第 j 个虚拟光源点,其位置坐标为 $[o^t_{j1}\, o^t_{j2} \cdots o^t_{jn} s^t_{j,i_k}]$。由式(6)可知,当 κ^t_{j,i_k} 大于给定阈值时,将会出现横向反射偏差。镜面的像差与第 j 个光源点在第 t 次迭代反射的镜面函数相关,表达式如下:

$$\kappa^t_{j,i_k} = \frac{r^t_{i_k}}{2\sqrt{(r^t_{i_k})^2 - (HO^t_{j,i_k})^2}} - \frac{|r^t_{i_k}|}{2} \tag{8}$$

当 $HO^t_{j,i_k} > |r^t_{j,i_k}|$ 时,表示第 j 个虚拟光源点离中心轴太远,像不能在虚拟光镜中呈现,则更正 $r^t_{i_k}$,增加曲面半径的长度,调节曲面镜面反射偏差。

从动力学的角度看,OIO 算法中粒子的收敛过程是以局部吸引点 I 为吸引子,随着寻优速度的减小,不断地接近 I 点,最后跌落到 I 点。因此在整个过程中,I 点实际上存在某种形式的吸引势能场吸引着粒子,这也正是整个粒子群能保持聚集性的原因。由于在 OIO 算法系统中光源点的收敛是以反射的形式实现的,并且反射的速度总是有限的,所以在搜索过程中光源点的搜索空间是一个有限的区域,不能覆盖整个可行空间,这是 OIO 算法本身所存在的局限性。

受量子粒子群算法的启发[9],本文提出将每个光源点都用量子空间中的一个粒子来描述,以体现光源点的不确定性,并根据群体智慧的聚集性建立量子势能场模型,然后根据人类群体自组织性和协同性等特点提出了 QOIO 算法。

1.2　QOIO 算法

假设 OIO 系统是一个量子系统,在量子空间中光源点的速度和位置不能同时确定,每个粒子的状态都由波函数 ψ 来确定。$|\psi|^2$ 是粒子位置的概率密度函数[10,11]。由上述对 OIO 算法系统中光源点的收敛分析可知,第 j 粒子在第 t 次迭代,该光源点的势阱为 $F^t_{i_k}$,则在第 $t+1$ 次迭代可以得到光源点 i 的函数为

$$\psi(x^{t+1}_{jd}) = \frac{1}{\sqrt{L_{jd}(t)}} \exp\left[-\frac{|x^t_{jd} - I_{jd}(t)|}{I_{jd}(t)}\right] \tag{9}$$

光源点的概率密度函数 Q 表示为

$$Q(x^{t+1}_{jd}) = [\psi(x^{t+1}_{jd})]^2 = \frac{1}{L_{jd}(t)} \exp\left[-\frac{2|x^t_{jd} - I_{jd}(t)|}{L_{jd}(t)}\right] \tag{10}$$

光源点的概率分布函数 T 表示为

$$T(x^{t+1}_{jd}) = \exp\left[-\frac{2|x^t_{jd} - I_{jd}(t)|}{L_{jd}(t)}\right] \tag{11}$$

在式(6)~(8)中,$L_{jd}(t)$ 是双指数分布的标准偏差[12]。应用 Monte Carlo 方法可以得到在第 $t+1$ 次迭代时,第 j 个粒子第 d 维的位置为

$$I^{t+1}_{jd} = I_{jd}(t) \pm \frac{L_{jd}(t)}{2} \ln[1/u_{id}(t)] \tag{12}$$

其中:$u_{id}(t)$ 是均匀分布在(0,1)上的随机数[13]。$L_{jd}(t)$ 的值由式(13)确定:

$$L_{jd}(t) = 2\alpha(t)\,|C_d - I^t_{jd}| \tag{13}$$

其中:参数 C 称为平均最优位置,它是所有粒子自身最优位置的中心点,可由式(14)计算得到。

$$C(t) = (C_1(t), C_2(t), \cdots, C_k(t)) = \frac{1}{N}\sum_{i=1}^{N} I_j(t) =$$
$$\left(\frac{1}{N}\sum_{j=1}^{N} I_{j1}(t), \frac{1}{N}\sum_{j=1}^{N} I_{j2}(t), \cdots, \frac{1}{N}\sum_{j=1}^{N} I_{jk}(t)\right) \tag{14}$$

综合式(9)(10)可以得到粒子的位置更新方程为

$$x^{t+1}_{jd} = F_{jd}(t) \pm \alpha(t)\,|C_d - x^t_{jd}| \times \ln[1/u_{id}(t)] \tag{15}$$

其中:参数 α 取值为算法迭代次数的线性函数,即

$$\alpha = (\alpha_{\max} - \alpha_{\min}) \times (\mathrm{iteration} - \mathrm{iter}) / \mathrm{iteration} + \alpha_{\min}$$

其中:iter 是当前迭代次数;iteration 是总的迭代次数;$\alpha_{\max}$、$\alpha_{\min}$ 是两个正的常数,通常分别取值为 1.0 与 0.5[14]。光源点的当前最优位置 I^t_j 和全局最优位置 I^t_g 的更新方式与 OIO 算法中相应参数的更新方式完全相同,即

$$I^{t+1}_j = \begin{cases} x^{t+1}_j & f(x^{t+1}_j) < f(I^t_j) \\ I^t_j & f(x^{t+1}_j) \geqslant f(I^t_j) \end{cases} \tag{16}$$

$$I^{t+1}_j = \arg\min_{1 \leqslant i \leqslant N} \{f(I^t_j)\} \tag{17}$$

其中:$f(\cdot)$ 为目标函数[15]。将式(16)作为粒子位置更新公式的 OIO 算法称为 QOIO 算法。

综上所述,QOIO 算法的步骤如下:

a)参数初始化。随机生成 NO 个初始光源点,求出初始最优点 G,以及初始最差解,令其等于 d_∞。

b)用轮盘赌选择方法从当前解中选择一个异于 O^t_j 的镜面函数顶点 F^t_j,确定镜面函数类型。

c)若 $\dfrac{(r^t_{i_k})^2}{2\sqrt{(r^t_{i_k})^2 - (HO^t_{j,i_k})^2}} - \dfrac{|r^t_{i_k}|}{2} > 0.01$ 或 $HO^t_{j,i_k} > |r^t_{i_k}|$ 时,有 $d_\infty = 2d_\infty$,修正镜面函数偏差。

d)根据式(11)计算粒子群的平均最优位置。

e) 计算粒子当前适应值，并与前一次迭代适应度值作比较，若小于，则根据光源点的位置更新为粒子当前最优位置，即若 $f(\boldsymbol{x}_j^{t+1}) < f(\boldsymbol{I}_j^t)$，$\boldsymbol{I}_j^{t+1} = \boldsymbol{x}_j^{t+1}$。

f) 将全局最优解 $\boldsymbol{I}_j^t$ 中的 c 个变量对应赋值于 $\boldsymbol{U}_j^t$ 中，若 $f(\boldsymbol{U}_j^t) < f(\boldsymbol{O}_j^t)$，则将第 j 个光源点用 $\boldsymbol{U}_j^t$ 代替；若 $f(\boldsymbol{U}_j^t) < f(\boldsymbol{G})$，则 $\boldsymbol{G} = \boldsymbol{U}_j^t$；若 $k = K$ 不成立，则重复 b) ~ f)。

g) 若停止条件满足，则 $\boldsymbol{G}$ 即为所求最优解。

2 仿真实验

本文选择了 10 个经典测试[16,17]函数来验证 QOIO 算法的有效性。表 1 给出了测试函数的定义、搜索空间和理论最优解。

表 1　经典测试函数

函数名称	公式	取值范围	最小值
Schwefel 2.22	$\sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]^n$	0
Rosenbrock	$\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2 + (x_i-1)^2)$	$[-30,30]^n$	0
Step	$\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	$[-100,100]^n$	0
Quartic	$\sum_{i=1}^{n} ix_i^4 + \mathrm{random}(0,1)$	$[-1.28,1.28]^n$	0
Rastrigin	$\sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12,5.12]^n$	0
Griewank	$\frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{i}) + 1$	$[-600,600]^n$	0
Penalized	$\frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_n-1)^2\} + \sum_{i}^{n}u(x_i,10,100,4)$, $y_i = 1+\frac{1}{4}(x_i+1)$; $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a\le x_i\le a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	$[-50,50]^n$	0
Six-hump camel-back	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^n$	−1.031 628 5
Branin	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6) + 10(1-\frac{1}{8\pi})\cos(x_1) + 10$	$[-5,10]\times[0,15]$	0.398
Goldstein-price	$[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	$[-2,2]^n$	3

算法求解相关的参数设置如下：光源点数为 40 个，最少镜面函数为 0，最多镜面数为 40 个。为了验证算法的收敛效果，对表 1 中的函数选取不同的维度运行 5 000 次，寻求各函数最小值及取值点。统计最优解、平均解及标准差，结果如表 2 所示。

表 2　仿真结果

函数名称	维数	OIO 算法			QOIO 算法		
		平均值	方差	最好值	平均值	方差	最好值
Schwefel 2.22	10	7.9455e−004	1.5478e−007	4.5287e−004	5.0343e−028	1.7666e−054	4.4231e−033
	30	3.801 6	0.739 7	2.660 8	1.6238e−024	1.3081e−047	1.3360e−026
	50	24.978 8	13.279 1	20.674 6	7.5807e−022	4.0518e−042	3.4589e−026
Rosenbrock	10	2.453 3	14.364 0	1.0431e−004	5.4246e−004	7.5137e−007	1.7666e−005
	30	5.758 0	1.4627e+002	0.001 0	0.008 9	1.9787e−004	4.2715e−005
	50	4.874 2	2.3499e+002	0.005 1	0.016 8	3.6203e−004	3.5277e−005
Step	10	3.3560e−005	2.0038e−009	2.9577e−006	5.0890e−010	2.9394e−019	4.8348e−011
	30	1.2329e+002	1.7544e+003	80.004 3	1.7233e−005	9.2673e−010	1.0076e−007
	50	3.1975e+003	5.1202e+005	2.4755e+003	1.5376e−004	4.6594e−008	1.7144e−007
Quartic	10	9.8962e−004	3.5008e−006	4.6802e−007	0	0	0
	30	15.446 2	3.016 4	12.736 9	0	0	0
	50	63.235 3	18.788 4	55.990 1	0	0	0
Rastrigin	10	0.793 9	0.651 7	0.001 1	0	0	0
	30	44.215 0	18.704 1	37.938 7	0	0	0
	50	1.7211e+002	5.4291e+002	1.3285e+002	0	0	0
Griewank	10	0.042 8	6.0622e−004	2.5254e−004	0	0	0
	30	1.990 7	0.524 1	1.193 8	0	0	0
	50	28.123 1	1.1467e+002	16.895 2	0	0	0
Penalized	10	3.4420e−006	7.0355e−011	6.1711e−008	2.0111e−010	4.5282e−020	2.7953e−011
	30	7.917 5	8.947 0	4.389 1	9.5940e−006	1.0423e−010	1.0177e−008
	50	4.0636e+005	2.8095e+011	6.3912e+003	4.1617e−006	1.7526e−011	2.2626e−007
Six-hump camel-back	2	−1.031 6	2.1912e−032	−1.031 6	−1.031 6	0	−1.031 6
Branin	2	0.397 8	0	0.397 8	0.397 8	0	0.397 8
Goldstein-price	2	3	2.1036e−030	3	3	4.3825e−032	3

实验结果可以看出,QOIO 算法对于经典函数的优化结果精度明显优于 OIO 算法。对于 Quartic、Rastrigin、Griewank、Six-hump camel-back、Branin、Goldstein_price 函数,QOIO 算法都可以收敛到全局最优解。

为进一步验证 QOIO 算法的收敛速度和精度的优越性,用表 1 中的基准函数,在操作环境为 Intel i3 处理器,Windows 7 操作系统,MATLAB 2010 软件下,设置相同随机数以产生相同初始种群及相同的随机数序列,将 QOIO 与 OIO 算法进行对比测试,设定种群规模为 40,迭代次数为 10 000,结果如图 2 ~ 11 所示。其中,纵轴是寻找到的最小值,取以 10 为底的对数;横轴是迭代次数。

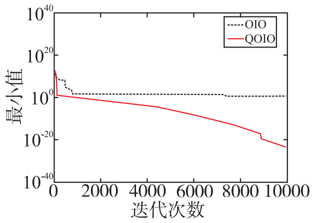

图2　Schwefel 2.22函数(D=30)迭代最优值变化曲线

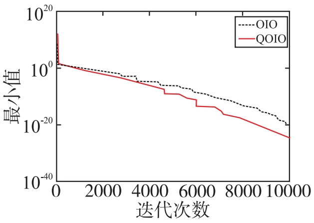

图3　Rosenbrock函数(D=30)迭代最优值变化曲线

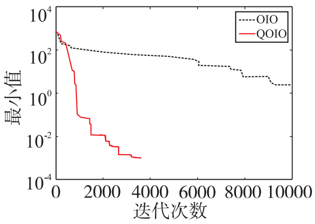

图4　Step函数(D=30)迭代最优值变化曲线

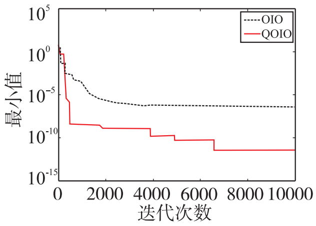

图5　Quartic函数(D=30)迭代最优值变化曲线

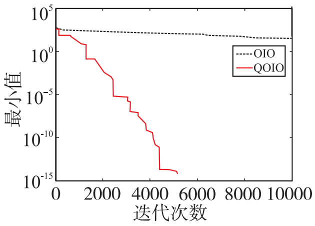

图6　Rastrigin函数(D=30)迭代最优值变化曲线

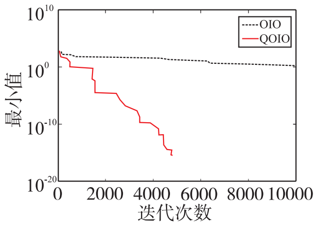

图7　Griewank函数(D=30)迭代最优值变化曲线

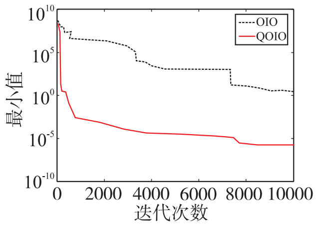

图8　Penalized函数(D=30)迭代最优值变化曲线

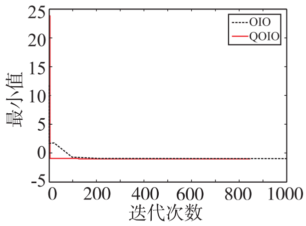

图9　Six-hump camel-back函数(D=2)迭代最优值变化曲线

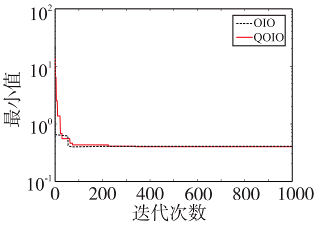

图10　Branin函数(D=2)迭代最优值变化曲线

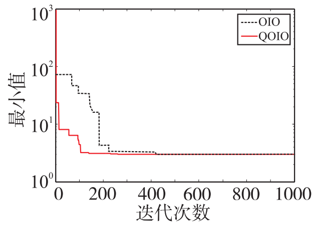

图11　Goldstein-price函数(D=2)迭代最优值变化曲线

从图 2 ~ 11 中每次迭代最优值变化的曲线可以看出,QOIO 算法的收敛速度明显快于 OIO 算法;图 4、6、7 中可以看出,OIO 算法很快即可收敛到最优解。相比于 OIO 算法的精度不高、易发生早熟和停滞现象,QOIO 算法则克服了这些缺点,以新的解的更新方式建立了全局搜索与局部搜索之间更为有效的平衡机制,加快了算法的收敛速度,使算法更易收敛于全局最优解,优化了算法的收敛精度和速度。

3　结束语

针对 OIO 算法在求解函数优化问题时收敛速度慢、收敛精度低等问题,QOIO 算法借鉴了量子力学原理,改进了解的更新方式;QOIO 算法控制参数少,设置简单,搜索能力强,具有较强的全局搜索能力,提高了算法的优化性能。经典测试函数通过设置不同的值得到多组非劣解表明,QOIO 算法增强了 OIO 算法的稳定性,提高了收敛速度和收敛精度,均表明了 QOIO 算法的有效性和实用性。

今后研究中,将 QOIO 算法进一步应用到多目标模型的求解中,并结合实际应用对算法进行深入的讨论。

参考文献:

[1] Formato R A. Central force optimization: a new deterministic gradient-like optimization metaheuristic[J]. Opsearch, 2009, 46(1): 25-51.

[2] Kashan A H. A new metaheuristic for optimization: optics inspired optimization (OIO)[J]. Computers & Operations Research, 2015, 55(3): 99-125.

[3] 边霞, 米良. 遗传算法理论及其应用研究进展[J]. 计算机应用研究, 2010, 27(7): 2425-2429, 2434.

[4] 周驰, 高海兵, 高亮, 等. 粒子群优化算法[J]. 计算机应用研究, 2003, 20(12): 7-11.

[5] Kashan A H. An effective algorithm for constrained optimization based on optics inspired optimization (OIO)[J]. Computer-Aided Design, 2015, 63(6): 52-71.

[6] 王凌. 量子进化算法研究进展[J]. 控制与决策, 2008, 23(12): 1321-1326.

[7] Konkar R. Analysing spherical aberration in concave mirrors[J]. Resonance Journal of Science Education, 2012, 17(8): 779-790.

[8] 郭秀芝, 高丽丽. 凹面镜成像规律的讨论[J]. 大学物理实验, 2003, 16(3): 36-37.

[9] 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): 3686-3694.

[10] Tang Wenling, Tian Guihua. Solving ground eigenvalue and eigenfunction of spheroidal wave equation at low frequency by supersymmetric quantum mechanics method[J]. Chinese Physics B, 2011, 20(1): 121-127.

[11] Lalwani P, Banka H, Kumar C. CRWO: clustering and routing in wireless sensor networks using optics inspired optimization[J]. Peer-to-Peer Networking and Applications, 2017, 10(3): 453-471.

[12] Xu Suhui, Mu Xiaodong, Chai Dong, *et al*. Multi-objective quantum-behaved particle swarm optimization algorithm with double-potential well and share-learning[J]. Optik-International Journal for Light and Electron Optics, 2016, 127(12): 4921-4927.

[13] 杜卫林, 李斌, 田宇. 量子退火算法研究进展[J]. 计算机研究与发展, 2008, 45(9): 1501-1508.

[14] 张毅, 卢凯, 高颖慧. 量子算法与量子衍生算法[J]. 计算机学报, 2013, 36(9): 1835-1842.

[15] 吕强, 陈如清, 俞金寿. 量子连续粒子群优化算法及其应用[J]. 系统工程理论与实践, 2008, 28(5): 122-130.

[16] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3): 459-171.

[17] Kashan A H. League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships[J]. Applied Soft Computing, 2014, 16(3): 171-200.