文章编号: 1001-0920(2010)01-0099-06

一种二进制编码的量子粒子群优化算法

奚茂龙[1], 孙 俊[2], 吴 勇[1]
(1. 无锡职业技术学院 机电技术学院, 江苏 无锡 214121; 2. 江南大学 信息学院, 江苏 无锡 214122)

摘 要: 针对离散空间优化问题,给出二进制编码的量子粒子群优化(BQPSO)算法的设计思路,重新定义粒子的位置矢量和粒子之间的距离,提出了 BQPSO 算法的进化方程. 通过泛函分析的方法分析了 BQPSO 算法的收敛性,得出全局收敛的结论,并通过多个测试函数测试了 BQPSO 算法的性能. 求解结果验证了算法的优越性.

关键词: 量子粒子群算法; 二进制编码; 全局收敛
中图分类号: TP301.6 **文献标识码**: A

Quantum-behaved particle swarm optimization with binary encoding

XI Mao-long[1], *SUN Jun*[2], *WU Yong*[1]
(1. School of Mechatronics Technology, Wuxi Institute of Technology, Wuxi 214121, China; 2. School of Information Technology, Southern Yangtze University, Wuxi 214122, China. Correspondent: XI Mao-long, E-mail: wx_xml@hotmail.com)

Abstract: The thought of quantum-behaved particle swarm optimization with binary encoding (BQPSO) is discussed, and evolution equations are given which are completely different from the QPSO algorithm. Position vector and distance between two positions are redefined, and QPSO algorithm with binary encoding is proposed. The convergence of BQPSO algorithm is analyzed by using functional analysis method, and conclusion of global convergence is derived. The test result for BQPSO algorithm shows its better performance in solving test functions.

Key words: QPSO; Binary encoding; Global convergence

1 引 言

粒子群优化算法自 1995 年被 Kennedy 提出以来,作为一种优化技术已广泛应用于实数解空间的优化问题,取得了较好的应用效果[1]. 此后他针对离散空间优化问题又提出了二进制粒子群优化(BPSO)算法[2]. 在此基础上他又提出了一种改进的二进制粒子群优化算法[3],扩展了粒子群优化算法的应用范围. Sun 等[4]在分析粒子群优化算法的基础上,深入研究了智能群体进化过程,提出了具有量子行为的粒子群优化算法(QPSO). QPSO 算法相对于 PSO 算法具有进化方程简单、控制参数少、收敛速度快、运算简单等优点,在测试函数和很多实际应用中都取得了优于 PSO 算法的效果[5-7]. 随后许多学者针对 QPSO 算法本身提出了一些改进算法[8-10]. 但这些应用都是针对实数解空间的优化问题,对于离散空间的优化问题,QPSO 算法并没有一个有效的解决方法.

为了解决该问题,本文针对离散空间的优化问题,提出了二进制编码的量子粒子群优化算法(BQPSO),给出了完全不同于 QPSO 算法的迭代方程,分析了方程的内在含义,进行了收敛性分析;同时通过测试函数验证了 BQPSO 算法的有效性,给出了一些指导性的结论.

2 BQPSO 算法思想

为了使量子粒子群算法适应实际问题中离散搜索空间的问题,受 BPSO 算法的启发,在量子粒子群算法中引入了二进制编码的概念,提出了二进制编码的量子行为粒子群算法. 为分析方便,首先写出量子粒子群算法的进化方程

$$p(t) = \phi \cdot P_i(t) + (1 - \phi) \cdot P_g(t),$$
$$\phi \sim U(0,1), \quad (1)$$
$$m_{\text{best}} = \left(\frac{1}{M}\sum_{i}^{M} P_{i,1}(t), \frac{1}{M}\sum_{i}^{M} P_{i,2}(t), \cdots,\right.$$

收稿日期: 2009-02-19; **修回日期**: 2009-04-06.
基金项目: 国家自然科学基金项目(60474030).
作者简介: 奚茂龙(1977—),男,江苏盐城人,讲师,博士,从事智能进化算法的研究;孙俊(1972—),男,江苏无锡人,副教授,博士,从事智能计算的研究.

$$\frac{1}{M}\sum_{i}^{M}P_{i,n}(t)\Big), \tag{2}$$

$$x(t+1) = p(t) \pm \beta\,|\, m_{\text{best}} - x(t)\,|\ln(1/u). \tag{3}$$

在量子粒子群算法中,没有速度和轨迹的概念,只有粒子位置点和粒子之间距离的概念. BPSO 算法的生成策略并不适用于 BQPSO 算法. 在 BQPSO 中,为了表示两个粒子之间的距离,引入了 Hamming 距离. 假设定义了两个粒子 (X_1, X_2),它们分别有两个决策变量 (X_{11}, X_{12}),(X_{21}, X_{22}),每一个决策变量由 5 位二进制编码,如图 1 所示.

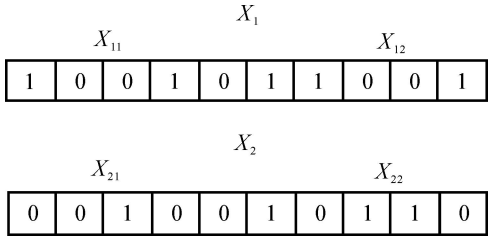

图 1 粒子位置的二进制编码

在 BQPSO 算法中,定义粒子含有决策变量的个数即为粒子的维数,如 X_{id},它的下标 i 表示 X_{id} 群体中的第 i 粒子,d 表示 X_{id} 的第 d 维,即第 d 个决策变量. X_i 和 X_{id} 的长度分别用 l 和 l_d 表示,则

$$l = \sum_{i}^{d} l_d,\ d = 1, 2, \cdots, D. \tag{4}$$

粒子 X_1, X_2 的距离可由 Hamming 距离表示,即

$$|\, X_1 - X_2\,| = d_H(X_1, X_2), \tag{5}$$

其中 $d_H(\cdot)$ 是计算 Hamming 距离的函数,它的值为两个位串对应位的不同值的数目.

根据量子粒子群算法,修改算法中平均最优位置 m_{best} 的值,生成 BQPSO 中的 m_{best}. 在 BQPSO 算法中,表示 m_{best} 的二进制位串中每一位的值由群体中所有表示粒子最优个体信息的二进制对应位串信息表示,通过统计群体中统计粒子二进制编码的每一位出现 0,1 的概率的大小,出现 0 的次数多,则 m_{best} 对应的为 0;反之,则为 1. 如图 2 所示.

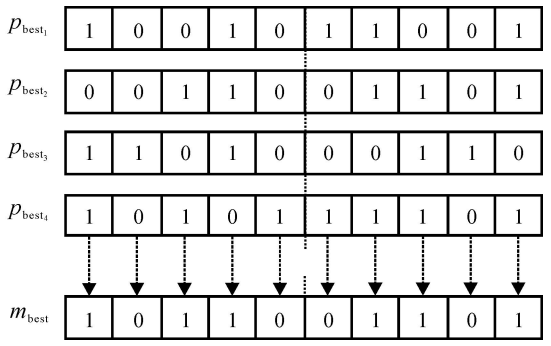

图 2 平均最优位置值

图 2 中,群体中有 4 个粒子,每个粒子当前的最优个体信息分别为 p_{best_1}, p_{best_2}, p_{best_3}, p_{best_4},它们对应的第 1 个二进制位的值分别为 1,0,1,1,则粒子群的 m_{best} 的值对应的第 1 位二进制的值则为 1. 依此类推,得到 m_{best} 的值为 1011001101. 如果对应位中出现的 0 和 1 的次数相同,则 m_{best} 随机选择为 0 或 1,如图 2 中对应粒子的第 6 位,分别出现了 2 次 0 和 2 次 1,则 m_{best} 值的第 6 位随机产生,图中随机生成为 0. 这样,便可写出 BQPSO 算法中获得 m_{best} 值的 Get_m_{best}( ) 函数:

```
Get_m_best ( p_best )
for j = 1 to l (the length of binary string)
   sum = 0;
   for each particle i
      sum = sum + p_best[i][j];
   endfor
   avg = sum/M;
   if avg > 0.5 m_best[j] = 1; endif
   if avg < 0.5 m_best[j] = 0; endif
   if avg = 0.5
      if rand( ) < 0.5 m_best[j] = 0;
      else m_best[j] = 1;
      endif
   endif
endfor
Return m_best
```

在 QPSO 算法中,式 (1) 是计算群体的局部吸引子 P_i, p_{id} 的值在 $p_{\text{best}_{id}}$ 和 g_{best_d} 之间,$P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ 则位于以 p_{best_i} 和 g_{best} 为对角线两端的超矩形中. P_i 到 p_{best_i} 或者 g_{best} 的距离都必须小于对角线的长度,即

$$|\, P_i - p_{\text{best}_i}\,| \leqslant |\, p_{\text{best}_i} - g_{\text{best}}\,|, \tag{6}$$

$$|\, P_i - g_{\text{best}}\,| \leqslant |\, p_{\text{best}_i} - g_{\text{best}}\,|. \tag{7}$$

通过 P_i 值的计算,可使群体产生多样性,跳出粒子的局部搜索区域. 在 BQPSO 算法中,P_i 的产生方式通过父代 p_{best_i} 和 g_{best} 中的每一位随机交叉,生成新的子代,即 P_i,显然 P_i 满足式 (6) 和 (7) 定义的 Hamming 距离,如图 3 所示.

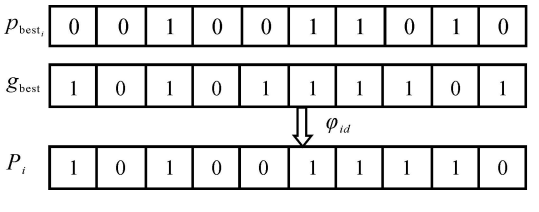

图 3 二进制编码粒子的多点交叉

P_i 的每一位由下式:

表 1　QPSO 算法和 BQPSO 算法进化方程的表达式

QPSO 算法	BQPSO 算法
$m_{\text{best}} = \left(\frac{1}{N}\sum_{i=1}^{N}P_{i1}(t), \frac{1}{N}\sum_{i=1}^{N}P_{i2}(t), \cdots, \frac{1}{N}\sum_{i=1}^{N}P_{iD}(t)\right)$	$\text{Get\_}m_{\text{best}}(p_{\text{best}})$
$p_{id}(t) = \varphi_d(t)\,P_{id}(t) + [1-\varphi_d(t)]\,P_{gd}(t), \varphi_d(t) \sim U(0,1)$	$\text{Get\_}P(p_{\text{best}s_i}, g_{\text{best}})$
$x_{id}(t+1) = p_{id}(t) \pm \beta \mid M_d(t) - x_{id}(t) \mid \ln[1/u_{id}(t)],\ u_{id}(t) \sim U(0,1)$	$\text{Transf}(P_{id}, p_r)$

$$P_{id} = g_{\text{best}_d} \cdot \varphi_d + p_{\text{best}_{id}} \cdot (1-\varphi_d) \tag{8}$$

计算得到,其中 φ 是服从在 $[0.0,1.0]$ 之间的随机数. 这样便可由式(8)写出计算 P_i 值的 Get_$P(\,)$ 函数.

QPSO 算法中的方程(3)可改写为

$$|x_{id} - p_{id}| = \beta |m_{\text{best}_d} - x_{id}| \ln[1/u_{id}],\ u_{id} \sim U(0,1). \tag{9}$$

其中 $|x_{id} - p_{id}|$ 在BQPSO算法中,可看作 $d_H(X_{id}, m_{\text{best}_d})$,则式(9)可写为

$$d_H(X_{id}, P_{id}) = [b]. \tag{10}$$

其中

$$b = \beta \times d_H(X_{id}, m_{\text{best}_d}) \ln(1/u),\ u = r_{\text{rand}}(\,). \tag{11}$$

式中:β 为BQPSO 算法的系数,$d_H(X_{id}, P_{id})$ 是粒子 i 第 d 维的 X_{id} 和局部吸引子第 d 维的 P_{id} 的Hamming距离,因为 Hamming距离是整数,需要对 b 值使用 $[\cdot]$ 取整. 通过式(10),可根据 b 的值和 P_{id} 的值反求得 X_{id} 的位串,时间复杂度为 $O(b \cdot l_d)$. 为了减少计算开销,重新改写了 X_{id} 的计算过程,通过变异 P_{id} 每一位的值生成新的 X_{id},变异的概率表达式为

$$p_r = \begin{cases} b/l_d; \\ 1, \text{ if } b/l_d > 1. \end{cases} \tag{12}$$

其中 l_d 为粒子第 d 维的长度. 这样便可写出计算 X_{id} 的函数 Transf( ):

```
Transf(P_id, P_r)
for each bit in the substring p_id;
  if rand() < P_r
    if the state of the bit is 1
      Set its state to 0;
    else set its sate to 1;
    endif
  endif
endfor
X_id = P_id;
Return X_id
```

Transf( ) 时间复杂度为 $O(l_d)$. 由以上 3 种操作可得BQPSO 算法的时间复杂度为 $O(tML)$. 其中 t 为算法迭代次数,M 为群体规模,l 为粒子长度.

QPSO算法和BQPSO算法进化方程的表达式如表1所示.

通过以上分析,可以得到二进制量子粒子群算法的步骤如下:

Step1: 用二进制位串的形式初始化群体中的每个粒子 X,并使得 $p_{\text{best}} = X$;

Step2: 根据方程 $m_{\text{best}} = \text{Get\_}m_{\text{best}}(p_{\text{best}})$ 计算 m_{best} 的值;

Step3: 根据适应度函数值(最小化问题)计算群体中每一个粒子的值,并与前次的局部最优值比较,如果 $f(X_i) < f(p_{\text{best}_i})$,则 $p_{\text{best}_i} = X_i$;反之,则不更新;

Step4: 计算群体中全局最优粒子 p_{best_g},并与前次的全局最优值 g_{best} 比较,如果 $f(p_{\text{best}_g}) < f(g_{\text{best}})$,则 $g_{\text{best}} = p_{\text{best}}$;反之,则不更新;

Step5: 根据方程 $P_i = \text{Get\_}P(p_{\text{best}_i}, g_{\text{best}})$,计算 P_i 的值;

Step6: 根据式

$$b = \beta \times d_H(X_{id}, m_{\text{best}_d}) \ln(1/u),\ u = \text{rand}(\,),$$

$$p_r = \begin{cases} b/l_d; \\ 1, \text{ if } b/l_d > 1 \end{cases}$$

计算 p_r 的值;

Step7: 根据方程 $X_{id} = \text{Transf}(P_{id}, p_r)$ 计算 X_{id} 的值,并连接生成 X;

Step8: 重复 Step2 ～ Step7,直到满足算法结束条件.

3　BQPSO 算法的收敛性分析

在这一节将用泛函分析法分析BQPSO 算法的收敛性.

令二进制字符集 $K = \{0,1\}$,粒子的位置由长为 D 的二进制位串表示,编码空间 $S = \{0,1\}^D$,$|S| = 2^D$. 设 X 表示粒子群当前位置的集合,P 表示粒子群个体最好位置的集合,其规模为 M,粒子群当前位置空间

$$S^X = \{(X_1, X_2, \cdots, X_M),\ X_j \in S,\ j = 1,2,\cdots,M\}.$$

粒子群个体最好位置空间

$$S^P = \{(P_1, P_2, \cdots, P_M),\ P_j \in S,\ j = 1,2,\cdots,M\},$$

全局最好位置空间

$$S^g = \{P_g, P_g \in S\} = S.$$

由 BQPSO 算法的工作流程可以看出,BQPSO 算法的求解过程是一个循环迭代过程. 在每一次迭代中,具体操作包括:目标函数值评价, p_{best} 位置的更新, g_{best} 粒子的选择,以及粒子的位置更新,从而使群体从一种状态进化到更为优越的另一种状态. 因此,可以将上述工作流程进一步抽象为随机映射运算.

设 (Ω, A, P) 是概率空间,可以给出 BQPSO 算子的定义.

定义 1 BQPSO 算子 T 是根据算法的迭代方式,通过当前位置的迭代,得到新的个体最好位置的过程. 它是一种从当前位置状态空间、个体最好位置状态空间和全局最好位置状态空间到个体最好位置状态空间的随机映射,即

$$T: \Omega \times S^X \times S^P \times S^g \to S^P.$$

BQPSO 算法在求解过程中,每一次迭代相当于产生一次映射 T_{BQPSO}. 设 $X(t)$ 和 $P(t)$ 为第 t 次迭代产生的当前位置群体和个体最好位置群体, $X(t+1)$, $P(t+1)$, $P_g(t+1)$ 为 $t+1$ 次迭代产生的群体以及全局最好位置,则

$$P(t+1) = T(\omega, X(t), P(t), P_g(t)), \quad (13)$$

其中 $t = 0, 1, \cdots, T-1$, T 为最大迭代次数.

仿真表明,BQPSO 算法每次迭代产生的全局最好位置的适应值优于上次迭代的适应值,即各代全局最好位置的适应值序列形成一个非减序列(最大化问题)

$$f(P_{g,t-1}) \leq f(P_{g,t}) \leq f(P_{g,t+1}). \quad (14)$$

其中: $P_{g,t-1} \in P(t-1)$, $P_{g,t} \in P(t)$, $p_{g,t+1} \in P(t+1)$ 分别表示 $t-1, t, t+1$ 代的全局最好位置.

在用 BQPSO 算法求解优化问题时,一般只关心搜索过程中满意解的变化情况. 为分析方便,不妨把迭代群体用其全局最好位置来代替. 因此上述 BQPSO 算法所产生的映射可重新定义为

$$P_{g,t+1} = T(\omega, P_{g,t}),$$
$$P_{g,t} \in P(t), \ P_{g,t+1} \in P(t+1). \quad (15)$$

BQPSO 算法的求解过程可看作由解空间到解空间的映射. 下面从映射角度对 BQPSO 算法的收敛性进行分析.

定义 2 定义度量 $d: S \times S \to R$,其中 d 的表达式为

$$d(P_{g,i}, P_{g,j}) = |(c - f(P_{g,j})) - (c - f(P_{g,j}))| = |f(P_{g,i}) - f(P_{g,j})|, \ P_{g,i}, P_{g,j} \in S. \quad (16)$$

其中: c 是一个很大的正数,则 (S, d) 是完备可分的度量空间.

证明 首先证明 (S, d) 是度量空间. 一般要求 S 为非空集合, d 为 $S \times S$ 上的实值函数. 对于 $P_{g,i}$, $P_{g,j}$, $P_{g,k} \in S$,对应一个实数 $d(P_{g,i}, P_{g,j}) \in S$ 满足:

1) $d(P_{g,i}, P_{g,j}) \geq 0$,当且仅当 $P_{g,i} = P_{g,j}$ 时, $d(P_{g,i}, P_{g,j}) = 0$,满足非负性;

2) $d(P_{g,i}, P_{g,j}) = |(c - f(P_{g,i})) - (c - f(P_{g,j}))| = |(c - f(P_{g,j})) - (c - f(P_{g,i}))|$,

满足对称性;

3) $d(P_{g,i}, P_{g,j}) = |(c - f(P_{g,i})) - (c - f(P_{g,j}))| = |(c - f(P_{g,i})) - (c - f(P_{g,k})) + (c - f(P_{g,k})) - (c - f(P_{g,j}))| \leq |(c - f(P_{g,i})) - (c - f(P_{g,k}))| + |(c - f(P_{g,k})) - (c - f(P_{g,j}))|$,

满足三角不等式,所以 (S, d) 为度量空间.

其次证明 (S, d) 为完备度量空间. S 是一有限状态空间,即 S 中位串个体的数目是有限的. 对当前粒子群最好个体位置的柯西序列 $\{P_{g,i}\}$ $(P_{g,i} \in S)$,以及任意 $\varepsilon > 0$,存在自然数 N. 当自然数 $n > N$ 时, $d(P_{g,n}, P_{g,m}) < \varepsilon$;当 $n \to \infty$ 时, $P_{g,n} \to P_g$. 因此, (S, d) 是完备的度量空间.

最后证明 (S, d) 是可分的. 设 $G \subseteq S$,由于 S 为有限集合,因此 G 为可数子集. 又 G 的闭包包含 S 中所有元素,所以 G 在 S 中稠密,这就证明了 (S, d) 是可分的. 故 (S, d) 是完备可分的度量空间. □

定义 3 随机算子 $T: \Omega \times S \to S$ 称为随机压缩算子,如果存在非负实值随机变量 $K(\omega) < 1$, a. s. ,使得

$$p(\{\omega, d(T(\omega, P_{g,i}), T(\omega, P_{g,j+1})) \leq K(\omega) d(P_{g,i}, P_{g,i+1})\}) = 1, \ P_{g,i}, P_{g,i+1} \in S. \quad (17)$$

定理 1 BQPSO 算法的操作形成的映射 T_{BQPSO} 是随机压缩算子.

证明 根据 BQSPO 算法的运行机理,每迭代一次均可能产生比上一代更好的个体,所以存在一个非负实值随机变量, $0 \leq K(\omega) < 1$, a. s. ,使得

$$d(T_{\text{BQPSO}}(\omega, P_{g,i-1}), T_{\text{BQPSO}}(\omega, P_{g,i})) = d(P_{g,i}, P_{g,i+1}) = |(c - f(P_{g,i})) - (c - f(P_{g,i+1}))| \leq K(\omega) |(c - f(P_{g,i-1})) - (c - f(P_{g,i}))| = K(\omega) d(P_{g,i-1}, P_{g,i}),$$
$$\Omega_0 = \{\omega: d(T_{\text{BQPSO}}(\omega, P_{g,i-1}), T_{\text{BQPSO}}(\omega, P_{g,i-1})) \leq$$

$K(\omega)\, d(P_{g,i-1}, P_{g,i})\} \subseteq \Omega_0,\ p(\Omega_0) = 1.$

所以 BQPSO 算法迭代所形成的映射

$$T_{\mathrm{BQPSO}}: \Omega \times S^X \times S^P \times S^g \to S^P$$

是一个随机压缩算子. □

定理 2 (随机压缩映射定理)　设随机算子 $T: \Omega \times S \to S$ 满足对几乎所有的 $\omega \in S$, $T(\omega)$ 均为压缩算子,即存在 $\Omega_0 \in \Omega$, $p(\Omega_0) = 1$,使任一 $\omega \in \Omega_0$,有

$$d(T(\omega, P_{g,i}), T(\omega, P_{g,i-1})) \leq K(\omega)\, d(P_{g,i}, P_{g,i+1}),$$

$P_{g,i}, P_{g,i-1}, P_{g,i+1} \in S$, $0 \leq K(\omega) < 1$,

则 $T(\omega)$ 有唯一随机不动点 $r(\omega)$,即

$$T(r(\omega), \omega) = r(\omega).$$

证明　利用巴拿赫压缩映射定理,对 $\omega \in \Omega_0$,存在 $h(\omega) \in S$ 为 $T(\omega)$ 的唯一不动点,对 $P_{g,i} \in S$,令

$$r(\omega) = \begin{cases} h(\omega), & \omega \in \Omega_0; \\ P_{g,i}, & \omega \in \Omega - \Omega_0 \end{cases}$$

为 $T(\omega)$ 的唯一不动点.

下面证明 $r(\omega)$ 的可测性:对任一 $P_{g,0} \in S$,令

$$P_{g,1}(\omega) = T(\omega, P_{g,0}),$$

$$P_{g,i+1}(\omega) = T(\omega, P_{g,i}),\ i = 1, 2, \cdots \quad (18)$$

由于 $P_{g,1}(\omega) \to P_{g,0}$, $T(\omega, P_{g,i}(\omega)) \to T(\omega, P_{g,0})$, $P_{g,i} \in S$ 即 $T(\omega)$ 连续. 根据符合定理知 $\{P_{g,i}\}$ 为一随机变量序列,又根据巴拿赫压缩映射定理, $P_{g,i}(\omega) \to r(\omega)$, a. s. ,由随机变量的极限定理可知, $r(\omega)$ 为一随机变量,从而 $r(\omega)$ 为 $T(\omega)$ 的随机不动点. □

由以上分析可知,BQPSO 算法的求解迭代过程是一种随机压缩映射,根据定理 2 可知该迭代过程是收敛的.

4　仿真与讨论

本文使用与文献[2]中相同的测试函数,见表 2. 对每个测试函数都进行了 3 组仿真,3 组仿真的群体数目分别为 20,40 和 80,每组仿真都应用 BPSO 算法和 BQPSO 算法独立运行 100 次. BPSO 算法的参数设置与文献[2]一致, $V_{\max}$ 设置为 6, c_1 和 c_2 设置为 1;BQPSO 算法的控制参数 β 设置为 0. 6. 算法进化 200 次,表 3 中记录了两种算法的平均最优值,100 轮仿真中的算法最大值和最小值以及方差.

表 2　测试函数

数测试函数	维数 ×每维长度 = 编码长度	函数最优值
f_1	3 ×10 = 30	78. 6
f_2	2 ×12 = 24	3905. 93
f_3	5 ×10 = 50	55
f_4	30 ×16 = 480	1248. 2
f_5	2 ×17 = 34	500

表 3 给出了两种算法对每个测试函数在不同粒子数的情况下得到的仿真结果,包括 100 次运行结果的目标函数的平均值、标准方差、最大值和最小值. 由表 3 中的数据可知,在简单的低维测试函数中,两种算法的测试结果接近,但对于绝大多数仿真结果,BQPSO 算法取得了优于 BPSO 算法的结果;在复杂的高维测试函数 f_5 的条件下,BQPSO算法

表 3　BQPSO 和 BPSO 算法在 5 个测试函数中的仿真结果

函数	粒子数	BPSO				BQPSO			
		平均最优值	方差	最大值	最小值	平均最优值	方差	最大值	最小值
f_1	20	78. 59984	7. 75E-6	78. 6	78. 5997	78. 599304	2. 02E-4	78. 6	78. 5919
	40	78. 59985	9. 52E-6	78. 6	78. 5997	78. 59967	9. 63E-5	78. 6	78. 5962
	80	78. 59985	9. 62E-6	78. 6	78. 5997	78. 59993	1. 79E-5	78. 6	78. 5994
f_2	20	3905. 906	0. 0049	3905. 929	3905. 687	3905. 914	0. 0059	3905. 93	3905. 6714
	40	3905. 924	0. 0015	3905. 93	3905. 799	3905. 928	5. 61E-4	3905. 93	3905. 9117
	80	3905. 928	3. 28E-4	3905. 93	3905. 910	3905. 929	6. 07E-5	3905. 93	3905. 9281
f_3	20	54. 91	0. 0289	55. 0	54. 0	54. 96	0. 0282	55. 0	54. 0
	40	55. 0	0. 0	55. 0	55. 0	55. 0	0. 0	55. 0	55. 0
	80	55. 0	0. 0	55. 0	55. 0	55. 0	0. 0	55. 0	55. 0
f_4	20	1252. 444	0. 3925	1262. 354	1241. 622	1251. 145	0. 4853	1255. 484	1246. 9507
	40	1252. 520	0. 3929	1260. 239	1241. 583	1251. 320	0. 7388	1257. 173	1245. 4572
	80	1252. 747	0. 3799	1260. 495	1241. 842	1251. 295	0. 6821	1259. 157	1247. 0757
f_5	20	498. 7968	0. 0454	499. 2699	497. 7629	499. 1133	0. 04605	499. 2699	497. 8197
	40	498. 9140	0. 0392	499. 2699	498. 1080	499. 2603	0. 0095	499. 2699	498. 7974
	80	499. 1333	0. 0211	499. 2699	498. 1080	499. 2699	2. 60E-11	499. 2699	499. 2699

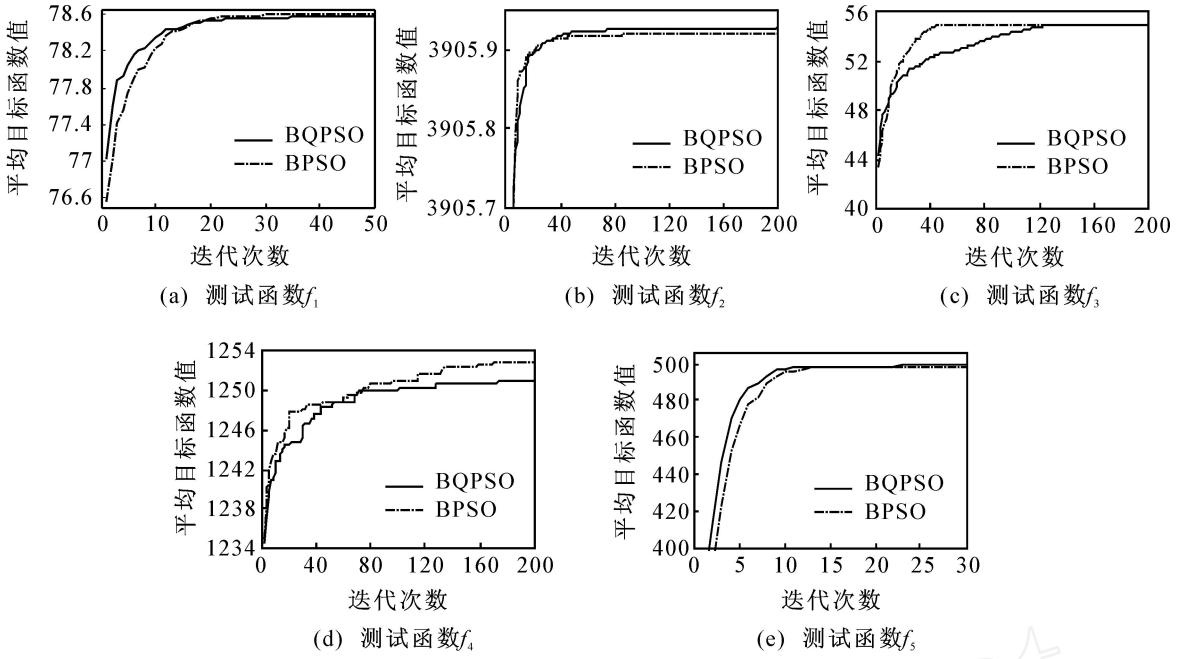

图 4 两种算法优化不同测试函数的目标函数值的收敛曲线比较

取得了较为明显的优势. 由图 4 的收敛曲线也可以知道,BQPSO 算法在测试函数的仿真中要优于 BPSO 算法.

5 结 论

本文从 QPSO 算法不能优化离散空间的优化问题角度出发,提出了针对离散空间优化问题的二进制编码的量子粒子群优化(BQPSO)算法. 详细地分析了 BQPSO 算法的进化过程,给出了算法的迭代方程,分析了迭代方程的内在含义,并给出了算法解决优化问题的步骤;接着使用泛函分析的方法证明了 BQPSO 算法的收敛性;最后通过测试函数进行了仿真测试,结果表明 BQPSO 算法在绝大多数情况下都要优于 BPSO 算法,验证了 BQPSO 算法的有效性.

参考文献(References)

[1] Kennedy J, Eberhart R C. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Network. Piscataway: IEEE, 1995: 1942-1948.

[2] Kennedy J, Eberhart R C. A discrete version of the particle swarm algorithm[C]. Proc of the 1997 Conf on System, Man and Cybernetics. Piscataway: IEEE, 1997: 4104-4109.

[3] Khanesar M A, Teshnehlab M, Shoorehdeli M A. A novel binary particle swarm optimization [C]. 15th Mediterranean Conf on Control and Automation. Piscataway: IEEE, 2007: 1-6.

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

[5] Fang W, Sun J, Xu W B. Design IIR digital filters using quantum-behaved particle swarm optimization[C]. Int Conf on Natural Computation. Zurich: Springer-Verlag, 2006: 637-640.

[6] Sun J, Liu J, Xu W B. Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems [J]. Int J of Computer Mathematics, 2007, 84(2): 261-272.

[7] Chen W, Sun J, Ding Y R. Clustering of gene expression data with quantum-behaved particle swarm optimization[C]. IEA/AIE, Zurich: Springer-Verlag, 2008: 388-396.

[8] Sun J, Choi H L, Xu W B. A novel and more efficient search strategy of quantum-behaved particle swarm optimization[C]. ICANNGA, Zurich: Springer-Verlag, 2007: 394-403.

[9] Sun J, Choi H L, Xu W B. A modified quantum-behaved particle swarm optimization[C]. Int Conf on Computational Science. Zurich: Springer-Verlag, 2007: 294-301.

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