第46卷 第2期 西 安 交 通 大 学 学 报 Vol. 46 No. 2
2012年2月 JOURNAL OF XI'AN JIAOTONG UNIVERSITY Feb. 2012

求解约束优化问题的多成员人工蜂群算法

王翔[1,2]，郑建国[1]
（1. 东华大学旭日工商管理学院，200092，上海；2. 郑州航空工业管理学院土木建筑工程学院，450015，郑州）

摘要：针对约束优化问题提出了一种多成员人工蜂群算法. 新算法设计了一种多成员机制，增强了在可行域内的搜索能力. 在进行选择操作时，允许拥有较优目标函数的不可行解战胜可行解，增强了种群的分散性；在处理等式约束时，引入一种约束放松程度从大到小变化的机制，充分利用了等式约束周围不可行解的信息. 针对13个标准测试函数的仿真实验表明：当处理含有等式约束且可行域较小的问题g13和最优解位于可行域内部且可行域较大的问题g02时，与改进人工蜂群算法相比，新算法最优解的均值误差分别减小了76%和80%.

关键词：约束优化问题；优化；人工蜂群算法；分散性
中图分类号：TP18 **文献标志码**：A **文章编号**：0253-987X(2012)02-0038-07

A Multi-Member Artificial Bee Colony Algorithm for Constrained Optimization Problems

WANG Xiang[1,2], ZHENG Jianguo[1]
(1. Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China;
2. School of Civil Engineering, Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450005, China)

Abstract: A new multi-member artificial bee colony algorithm is proposed for constrained optimization problems. A multi-member mechanism is introduced in the algorithm to enhance the search capabilities in the feasible region, and infeasible solutions with better values of objective function are permitted to conquer the feasible solutions with worse objective function in selecting operators so that the population can be diversified well. Moreover, a mechanism of constraint relaxation is introduced to increase the probability of finding feasible solutions in dealing with linear equality constraints. The new algorithm is tested on thirteen well-known test problems. Comparison results with the modified artificial bee colony algorithm show that the mean errors of the new algorithm are reduced by 76% and 80% in handling the problem g13 with equality constraints and a small feasible region, and the problem g02 with the optimum inside a big feasible region, respectively.

Keywords: constrained optimization problems; optimization; artificial bee colony algorithm; diversity

在现实生活中，大部分工程和管理问题都可以转化成一个约束优化问题(constrained optimization problems, COPs)[1]，它的一般形式如下式所示

$$\left.\begin{aligned}&\min f(\boldsymbol{X}),\boldsymbol{X}\in\boldsymbol{D}\\&\text{s.t.}\begin{cases}g_i(\boldsymbol{X})\leqslant 0, & i=1,2,\cdots,q\\ h_k(\boldsymbol{X})=0, & k=q+1,q+2,\cdots,m\end{cases}\end{aligned}\right\}\quad(1)$$

式中：$\boldsymbol{X}=(x^1,x^2,\cdots,x^n)$表示一个可行解，其中$x^i\in[x^i_{\min},x^i_{\max}]$表示可行解的一个分量；$f$表示目标

收稿日期：2011-04-25. 作者简介：王翔(1976—)，男，博士生，郑州航空工业管理学院讲师；郑建国(通信作者)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(70971020).
网络出版时间：2011-12-20 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20111220.1805.006.html

函数；$\boldsymbol{D}=\prod_{i=1}^{n}[x_{\min}^{i},x_{\max}^{i}]\subseteq\mathbf{R}^{n}$ 表示定义域；$g_i(\boldsymbol{X})$表示不等式约束；$h_k(\boldsymbol{X})$表示等式约束.

人工蜂群算法(artificial bee colony, ABC)[2]是在 2005 年提出的一种进化算法，它具有寻优能力强、收敛速度快的特点，因此成为该领域新的研究热点. 目前关于 ABC 算法的研究主要集中在无约束优化问题[2-4]，针对 COPs 的研究很少. 2011 年 Karaboga 等[5]针对 COPs 提出一种改进的人工蜂群(modified artificial bee colony, MABC)算法，但是实验结果表明，该算法在处理含有等式约束问题 g13 时效果不佳.

本文针对约束优化问题，在 MABC 算法的基础上，提出一种多成员人工蜂群(multi-member artificial bee colony, Multi-ABC)算法. 新算法为了充分利用食物源信息以及搜索过程中不可行解的信息，进行了以下 3 点改进：①在一个食物源邻域内，雇佣蜂和观察蜂不是生成一个候选食物源，而是同时生成多个候选食物源；②选择操作时，新算法不是完全依据可行解优先的 DEB 准则，而是在一定概率下允许拥有较优目标函数的不可行解战胜可行解，从而利用不可行解的信息；③当处理等式约束时，新算法不是设置约束放松程度 ε 为一个固定数值，而是令其进行从大到小的变化，达到充分利用等式约束周围不可行解的目的. 针对 13 个 benchmark 函数的仿真实验结果表明，在处理含有等式约束且可行域较小的问题以及最优解位于可行域内且可行域较大的问题时，本文的 Multi-ABC 算法显著优于 MABC 算法.

1　Multi-ABC 算法

Multi-ABC 算法是一种在 MABC 算法基础上提出的改进算法. 假设问题规模是 n，蜂群大小为 s，雇佣蜂、观察蜂和食物源的数目相等，都为 $s/2$，则求解最小化的约束优化问题的 Multi-ABC 算法具体步骤如下.

1.1　初始化阶段

Multi-ABC 算法与 MABC 算法的初始化阶段完全相同，都包含初始化食物源种群和食物源指标两步操作.

(1)初始化食物源种群操作是指按照式(1)随机产生 $s/2$ 个 n 维食物源向量

$$\{\boldsymbol{X}_i=(x_i^1,x_i^2,\cdots,x_i^n)\mid i=1,\cdots,s/2\};$$
$$x_i^j=x_{\min}^j+\mathrm{rand}(0,1)(x_{\max}^j-x_{\min}^j)\qquad(2)$$

式中：$x_{\min}^j$和 $x_{\max}^j$分别是 $\boldsymbol{X}_i$ 中第 j 维的下限和上限.

(2)初始化食物源指标操作是指设定每个食物源的状态标识 $l_i=0$，其中 $i=1,\cdots,s/2$，l_i 表示食物源 $\boldsymbol{X}_i$ 没有得到提升的连续次数，它被作为侦查蜂进行随机搜索的判断指标.

1.2　雇佣蜂阶段

雇佣蜂阶段表示雇佣蜂针对每个食物源 $\boldsymbol{X}_i$ 在其邻域内寻找更优食物源的过程. 与 MABC 不同，Multi-ABC 算法在生成一个候选食物源操作的基础上，还设计了多成员机制和选择概率动态变化机制，达到增强种群分散性，充分利用一个食物源信息的目的.

1.2.1　*生成一个候选食物源的基本操作*　针对食物源 $\boldsymbol{X}_i$ 的变异操作是多成员机制和选择概率动态变化机制的基础，具体如下.

(1)首先按照下式产生新的食物源向量

$$\boldsymbol{C}_i=(c_i^1,\cdots,c_i^j,\cdots,c_i^n);$$
$$c_i^j=\begin{cases}x_i^j+\varphi_{ij}(x_i^j-x_k^j), & R_j<P_{\mathrm{MR}}\\ x_i^j, & \text{其他}\end{cases}\qquad(3)$$

式中：k 是从集合$\{1,2,\cdots,s/2\}$中随机抽取的元素且不等于 i；φ_{ij}是$[-1,1]$之间均匀分布的随机数；R_j 是$[0,1]$之间均匀分布的随机数；P_{MR}是变异概率.

(2)如果 $\boldsymbol{C}_i$ 与 $\boldsymbol{X}_i$ 相比没有改变，那么按照下式改变 $\boldsymbol{X}_i$ 的第 j 个元素，从而产生新的食物源

$$\boldsymbol{C}_i=(x_i^1,\cdots,x_i^{j-1},c_i^j,x_i^{j+1},\cdots,x_i^n);$$
$$c_i^j=x_i^j+\varphi_{ij}(x_i^j-x_k^j)\qquad(4)$$

式中：j 是$[1,n]$之间均匀分布的随机整数.

需要特别指出，与 ABC 算法不同，MABC 和 Multi-ABC 算法存在变异概率 P_{MR}，它表示食物源向量 $\boldsymbol{X}_i$ 中每个元素进行变异的概率，目的是加大变异程度.

1.2.2　*多成员机制*　当雇佣蜂在原食物源 $\boldsymbol{X}_i$ 的邻域内搜索时，MABC 算法只能生成一个候选食物源 $\boldsymbol{V}_i$，没有充分利用 $\boldsymbol{X}_i$ 的信息，故 Multi-ABC 算法设计了多成员机制来克服这个缺陷.

多成员机制也是一种生成候选食物源的方法，它包含两步操作：①针对一个食物源 $\boldsymbol{X}_i$，产生 N_0 个候选食物源$\{\boldsymbol{C}_m\mid m=1,\cdots,N_0\}$，其中每个候选食物源 $\boldsymbol{C}_m$ 按照式(3)和式(4)产生；②基于 DEB 准则[6]，从$\{\boldsymbol{C}_m\mid m=1,\cdots,N_0\}$中选出最优候选食物源 $\boldsymbol{V}_i$.

1.2.3　*选择概率动态变化机制*　当在原食物源 $\boldsymbol{X}_i$ 与候选食物源 $\boldsymbol{V}_i$ 之间进行选择操作时，MABC 算法仅仅利用了可行解优先的 DEB 准则，这不能保持

种群的分散性. 为了克服这个缺陷,Multi-ABC 算法中设计了一种选择概率动态变化机制,其中选择概率用 P_{SR} 表示,该机制包含以下两部分内容.

(1)在原食物源 $\boldsymbol{X}_i$ 与最优候选食物源 $\boldsymbol{V}_i$ 之间进行选择操作时,利用概率 P_{SR} 选择基于目标函数的比较准则[7],利用概率 $1-P_{SR}$ 选择 DEB 比较准则. 其中,DEB 比较准则的核心是可行解优先,故在进化过程中不能利用不可行解的信息;基于目标函数的比较准则的基本思想是不考虑约束违反情况,仅比较目标函数,故可在种群中可以保留拥有较优目标函数的不可行解,从而增强了种群的分散性.

(2)在进化过程中,概率 P_{SR} 不是固定不变的,而是按照式(5)由 P_{SR} 的初始值减小到 0.

$$P_{SR}=P_{SR,init}(1-c/C_{max}) \tag{5}$$

式中:$P_{SR,init}$ 表示概率 P_{SR} 的初始值;c 表示进化代数;C_{max} 表示最大进化代数.

1.3 观察蜂选择食物源的概率计算

Multi-ABC 算法与 MABC 算法的观察蜂选择食物源概率计算完全相同,都是利用每个食物源 $\boldsymbol{X}_i$ 的适应度 F_i 和 $\boldsymbol{X}_i$ 的约束违反程度 v_i,由式(6)计算其被观察蜂选中的概率

$$p_i=\begin{cases}0.5+F_i/\sum_{i=1}^{s/2}F_i\times 0.5, & \boldsymbol{X}_i \text{ 为可行解}\\ (1-v_i/\sum_{i=1}^{s/2}v_i)\times 0.5, & \boldsymbol{X}_i \text{ 为不可行解}\end{cases} \tag{6}$$

式中

$$F_i=\begin{cases}1/(1+f_i), & f_i\geqslant 0\\ 1+\mathrm{abs}f_i, & f_i<0\end{cases} \tag{7}$$

式中:f_i 表示食物源 $\boldsymbol{X}_i$ 的目标函数值.

由式(6)可知,可行解被选中的概率在[0.5, 1]之间,不可行解被选中的概率在[0, 0.5]之间,即观察蜂阶段可行解被选中的概率大于不可行解.

1.4 观察蜂阶段

观察蜂阶段是在较优食物源邻域内进一步的搜索. 一只观察蜂首先依据食物源被选中的概率$\{p_j\mid j=1,\cdots,s/2\}$选出一个食物源 $\boldsymbol{X}_i$;其次针对 $\boldsymbol{X}_i$ 执行与 1.2 部分雇佣蜂相同的操作.

1.5 侦察蜂阶段

Multi-ABC 算法与 MABC 算法的侦察蜂阶段完全相同,其基本思想都是如果一个食物源 $\boldsymbol{X}_i$ 在指定的迭代次数 L_{limit} 内没有提升,那么 $\boldsymbol{X}_i$ 的位置就被放弃,进而利用式(2)随机生成新位置. 在此阶段中,为了控制侦察蜂被调用的次数,引入参数侦察蜂工作频率 N_{spp},每执行 N_{spp} 次循环,就调用一次侦察蜂随机搜索操作.

1.6 多成员人工蜂群算法伪码

图 1 为本文 Multi-ABC 算法伪码. 与 MABC 算法相比,Multi-ABC 算法有 6 个步骤不同,分别为伪码的第 4、5、7、8、17 和 18 步.

```
    /*初始化阶段*/
1.  按照式(1)初始化食物源种群{X_i=(x_i^1, x_i^2, ⋯, x_i^n)| i=1, ⋯, s/2}
2.  初始化每个食物源的状态标识{l_i=0| i=1, ⋯, s/2}
3.  for c=1 to C_max do
4.  P_MR=rand(0.3, 0.8) /*设置变异概率*/
5.  按照式(5)生成选择概率 P_SR
    /*雇佣蜂阶段*/
6.  for i=1 to s/2 do
7.   利用多成员机制，针对食物源 X_i生成最优候选食物源V_i
8.   利用选择概率动态变化机制，在候选食物源V_i与食物源 X_i之间选出最优
9.  end for
    /*每个食物源被观察蜂选中概率的计算*/
10.  for i=1 to s/2 do
11.   按照式(6)计算食物源 X_i被观察蜂选中的概率 p_i
12. end for
    /*观察蜂阶段*/
13. t=0, i=1
14. repeat
15.  if rand(0,1) <p_i then
16.    t=t+1
17.    利用多成员机制，针对食物源 X_i生成最优候选食物源V_i
18.    利用选择概率动态变化机制，在候选食物源V_i与食物源 X_i之间选出最优
19.  end if
20.  i=i+1;
21.  i=i mod ((s/2)+1)
22. until t=s/2
    /*侦查蜂阶段*/
23. if mod(c, N_spp)=0 then
24.   if max(l_i)>L_limit then
25.    利用式(1)随机产生新的食物源替换 X_i
26.   end if
27. end if
28. end for
```

图 1 Multi-ABC 算法伪码

需要特别指出的是,式(3)中的变异概率 P_{MR} 在 MABC 和 Multi-ABC 中的设置并不相同. 在 MABC 中,P_{MR} 是固定数值,而在 Multi-ABC 中,每一次迭代的 P_{MR} 都不相同,它是(0.3, 0.8)之间均匀分布的随机数,这是文献[5]推荐的结果,详见图 1 的第 4 行.

1.7　等式约束处理机制

当处理含有等式约束的问题时，MABC 算法只是设置等式约束放松程度为一个固定值 $\varepsilon=10^{-3}$，不能充分利用等式约束周围的不可行解信息.

为了克服以上缺陷，Multi-ABC 算法在处理等式约束时采用与文献[8]同样的方法，引入参数 $\varepsilon_{\text{init}}$，按照公式(8)将等式约束放松程度 ε 设置成一个随着进化代数动态减小的数值，从而更有效利用不可行解

$$\varepsilon=\varepsilon_{\text{init}}/1.035^{c} \tag{8}$$

式中：c 表示进化代数；$\varepsilon_{\text{init}}$ 表示等式约束放松程度的初始值. 此外，需要指出的是 ε 的下限为 10^{-3}.

2　仿真实验

为了评价 Multi-ABC 算法性能，本文使用文献[8]提供的 13 个 benchmarks 函数进行测试，结果如表 1 所示.

表 1　13 个 benchmarks 函数的特点

问题	n	Type	ρ/%	N_{LI}	N_{NI}	N_{LE}	N_{NE}
g01	13	Quadratic	0.000 3	9	0	0	0
g02	20	Nonlinear	99.997 3	1	1	0	0
g03	10	Nonlinear	0.002 6	0	0	0	1
g04	5	Quadratic	27.007 9	0	6	0	0
g05	4	Nonlinear	0.000 0	2	0	0	3
g06	2	Nonlinear	0.005 7	0	2	0	0
g07	10	Quadratic	0.000 0	3	5	0	0
g08	2	Nonlinear	0.858 1	0	2	0	0
g09	7	Nonlinear	0.519 9	0	4	0	0
g10	8	Linear	0.002 0	3	3	0	0
g11	2	Quadratic	0.097 3	0	0	0	1
g12	3	Quadratic	4.769 7	0	9^{3}	0	0
g13	5	Nonlinear	0.000 0	0	0	0	3

注：n 表示问题维数；Type 表示目标函数类型；ρ 表示可行域占整个区域的比例；N_{LI} 表示线性不等式约束个数；N_{NI} 表示非线性不等式约束个数；N_{LE} 表示线性等式约束个数；N_{NE} 表示非线性等式约束个数.

2.1　参数设置

本文主要的比较对象是 MABC 算法，其参数设置与文献[5]完全相同，变异率 $P_{\text{MR}}=0.8$，种群大小 $s=40$，最大循环次数 $C_{\max}=6\ 000$，总的适应度函数评价次数 $N_{\text{fes}}=sC_{\max}=40\times 60\ 000=240\ 000$，食物源被放弃的迭代次数上限 $L_{\text{limit}}=0.5\ sn$，侦查蜂工作频率 $N_{\text{spp}}=0.5\ sn$.

与 MABC 算法相比，Multi-ABC 算法在参数设置时，不需要设置变异概率 P_{MR} 的数值，同时增加了 3 个参数. 种群大小 s、食物源被放弃的迭代次数上限 L_{limit} 以及侦查蜂工作频率 N_{spp} 与 MABC 算法的设置相同，最大循环次数 $C_{\max}=1\ 000$，多成员数目 $N_0=6$，利用目标函数比较准则的概率 P_{SR} 的初始值 $P_{\text{SR,init}}$ 为 0.225，当处理等式约束时，约束放松程度的初始数值 $\varepsilon_{\text{init}}=5$. 显然，Multi-ABC 算法的适应度函数评价次数 $N_{\text{fes}}=sN_0C_{\max}=40\times 6\times 1\ 000=240\ 000$，与 MABC 算法相同，从而保证了比较结果的公平性.

2.2　实验结果

表 2 显示了 Multi-ABC 算法与 MABC 算法在适应度函数评价次数相同(240 000 次)的条件下，30 次独立实验求出最优解 f^{*} 的均值与标准差，并利用 t 检验展示了 2 种算法求解 f^{*} 的均值是否存在显著性差异. 其中，MABC 算法的实验数据来自文献[5].

由表 2 可知，Multi-ABC 算法与 MABC 算法的实验结果在 6 个函数(g02、g03、g06、g09、g10、g11、g13)上存在显著性差异. 如果按照最优解的位置和约束类型，那么这 6 个函数可划分成 3 个类别：第 1 类是 g02 问题，它的特点是最优解位于可行域内且最优解周围存在多个局部极值，同时可行域面积非常大；第 2 类是 g06 和 g09 问题，它们的特点是最优解位于可行域边界上，且不存在等式约束；第 3 类是 g03、g11 和 g13 问题，它们的特点是包含等式约束.

按照以上的分类，由表 2 可知：Multi-ABC 算法在第 1 类问题 g02 上显著优于 MABC 算法，这表示新算法在可行域内寻优能力较强，值得一提的是在 30 次实验中，Multi-ABC 算法求解 f^{*} 均值的误差比 MABC 减小了 80%；在第 2 类问题 g06 和 g09 上，Multi-ABC 算法和 MABC 算法各有一次获胜，这表示两者在适应度函数评价次数相同的条件下，处理最优解位于可行域边界的问题各有优势；在第 3 类问题上，Multi-ABC 算法在 g03 和 g13 上优于 MABC 算法，在 g11 问题上劣于 MABC 算法，同时由表 1 可知，与 g03 和 g13 相比，g11 的可行域要大得多，达到了 9.73%，故可认为 Multi-ABC 算法处理含有等式约束且可行域较小问题时具有明显优势，特别是针对 g13 问题，在 30 次实验中，Multi-ABC 算法求解 f^{*} 均值的误差比 MABC 算法的误差减小了 76%.

2.3 最优参数分析

本文利用均值分析法(ANOM)寻找算法的最优参数. 由于 Multi-ABC 算法比 MABC 算法增加了 3 个参数 $P_{SR,init}$、N_0 和 ε_{init},故本文只讨论它们的最优数值. 需要指出的是,参数 ε_{init} 只针对存在等式约束的问题才有意义,故由表 1 可知只涉及 g03、g05、g11、g13 四个函数.

表 2 Multi-ABC 算法与 MABC 算法实验结果的比较

问题	已知最优解	f^* 均值		f^* 标准差		t 检验
		Multi-ABC	MABC	Multi-ABC	MABC	
g01	−15.000	−15.000	−15.000	0	0	
g02	−0.803 619	−0.801 434	−0.792 412	5.61×10^{-3}	1.20×10^{-2}	+
g03	−1.000	−1.002	−1.000	2.39×10^{-3}	0	+
g04	−30 665.539	−30 665.539	−30 665.539	6.29×10^{-9}	0	
g05	5 126.498	5 176.933	5 185.714	79.6	73.4	
g06	−6 961.814	−6 961.814	−6 961.813	7.1×10^{-10}	2.00×10^{-3}	+
g07	24.306	24.488	24.473	1.71×10^{-1}	1.86×10^{-1}	
g08	−0.095 825	−0.095 825	−0.095 825	0	0	
g09	680.63	680.67	680.64	2.03×10^{-2}	4.00×10^{-3}	+
g10	7 049.25	7 196.55	7 224.407	1.08×10^{2}	1.34×10^{2}	
g11	0.75	0.868	0.75	1.25×10^{-1}	0	+
g12	1.000	1.000	1.0	1.95×10^{-16}	0	
g13	0.053 950	0.272 680	0.968	2.62×10^{-1}	5.50×10^{-2}	+

注:f^* 为 30 次实验最优解;t 检验显著性水平为 0.05,如果 t 检验显著,则表示为+,否则表示为空.

以下针对 $P_{SR,init}$ 的 4 个水平数值{0, 0.225, 0.45, 0.675}、N_0 的 3 个水平数值{4, 6, 8}以及 ε_{init} 的 4 个水平数值{0.001, 1.25, 2.5, 5}探讨它们对算法性能的影响. 在进行实验时,只单独改变需要讨论的参数,其他参数与 2.1 节设置完全相同,每一组实验独立运行 30 次. 需要特别指出的是,表 3~表 5 只显示了具有参数敏感性的实验结果.

由表 3 可知 N_0 的最优参数设置为 6,由表 4 可知 $P_{SR,init}$ 的最优参数设置为 0.225,由表 5 可知 ε_{init} 最优参数设置是 1.25 和 2.5.

表 3 N_0 为 4、6、8 时的 ANOM 分析

问题	N_0	N_0^+	N_0^-	$\bar{N}_0$
g03	4,6,8	6,8	4	
g04	4,6,8		4	6,8
g06	4,6,8	6,8	4	
g11	4,6,8	4,6	8	

注:N_0^+、N_0^- 和 $\bar{N}_0$ 分别表示实验结果在平均水平之上、平均水平之下和处于平均水平的 N_0 值.

表 4 $P_{SR,init}$ 为 0,0.225,0.45,0.675 时的 ANOM 分析

问题	$P_{SR,init}$	$P_{SR,init}^+$	$P_{SR,init}^-$	$\bar{P}_{SR,init}$
g03	0,0.225,0.45,0.675	0,0.225	0.45,0.675	
g04	0,0.225,0.45,0.675		0.675	0,0.225,0.45
g05	0,0.225,0.45,0.675		0	0.225,0.45,0.675
g06	0,0.225,0.45,0.675		0.675	0,0.225,0.45
g07	0,0.225,0.45,0.675		0	0.225,0.45,0.675
g09	0,0.225,0.45,0.675		0	0.225,0.45,0.675
g10	0,0.225,0.45,0.675	0,0.225	0.675	0.45
g13	0,0.225,0.45,0.675		0	0.225,0.45,0.675

注:$P_{SR,init}^+$、$P_{SR,init}^-$ 和 $\bar{P}_{SR,init}$ 分别表示实验结果在平均水平之上、平均水平之下和处于平均水平的 $P_{SR,init}$ 值.

表 5　ε_{init} 为 0.001、1.25、2.5、5 时的 ANOM 分析

问题	ε_{init}	$\overset{+}{\varepsilon}_{init}$	$\overset{-}{\varepsilon}_{init}$	$\overset{-}{\varepsilon}_{init}$
g03	0.001,1.25,2.5,5	1.25,2.5,5	0.001	
g11	0.001,1.25,2.5,5		5	0.001,1.25,2.5
g13	0.001,1.25,2.5,5	5		0.001,1.25,2.5

注：$\overset{+}{\varepsilon}_{init}$、$\overset{-}{\varepsilon}_{init}$ 和 $\overset{-}{\varepsilon}_{init}$ 分别表示实验结果在平均水平之上、平均水平之下和处于平均水平的 ε_{init} 的值.

3　讨　论

3.1　多成员机制及其在 g02 函数中的作用

当处理最优解位于可行域内，且其周围有多个局部最优解的问题时，引入多成员机制的新算法优势明显，此类问题可以简化成为如下模型.

假定问题维度为 1，a 是当前个体，其下一步变异只有 2 种可能——进入全局性最优点 g，或进入局部极值点 l，而且一旦进入 g 或者 l 就无法通过变异操作逃出. 该模型如图 2 所示.

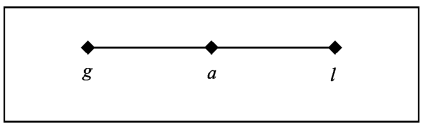

图 2　多成员机制模型机理

如果利用多成员机制，在 a 点同时变异产生 N_0 个候选子个体，那么至少有一个候选个体进入 global 的概率为 $1-0.5^{N_0}$. 此外，若利用 MABC 的变异机制，从 a 点开始进化 N_0 代，那么成功求解的概率为 0.5. 综上所述，只要 $N_0>1$，多成员机制成功求解此类问题的概率就会占优.

g02 问题可行域特别大，最优解位于可行域内部，且最优解周围有多个局部极值点，如果利用多成员机制进行变异操作，那么找到全局最优解的概率较大，如果利用 MABC 算法的变异机制，则找到全局最优解的概率相对较小. 综上所述，当解决此类问题时，采用多成员机制的 Multi-ABC 算法比 MABC 算法更有可能找到全局最优解，事实上，表 2 的实验数据也证明了这个结论.

3.2　选择概率动态变化机制及 g06 问题的解释

3.2.1　选择概率动态变化机制　采用 Muliti-ABC 算法在种群的个体及其所产生的候选子个体之间进行选择操作时，存在 2 种比较准则——基于目标函数的比较准则和 DEB 准则，P_{SR} 表示选择前者的概率. 当处理最优解位于可行域边界上的问题时，该机制可以有效地利用不可行解的信息，其原理如图 3 所示.

图 3 中，假设 g 表示可行域的最优解，位于可行域边界上，目标函数 $f(g)=0$，a 表示一个可行解，$f(a)=3$，b 表示 a 进行变异操作之后产生的候选子个体，它是一个不可行解，其目标函数 $f(b)=-1$. 与 a 相比，b 能为进化过程提供更多信息，一方面它距离最优解近，另一方面其目标函数优秀.

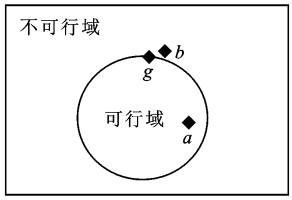

图 3　P_{SR} 动态变化模型 1

如果基于 DEB 准则，那么可行解 a 战胜不可行解 b 进入下一代种群，显然这不利于进化过程；如果基于目标函数的比较准则进行选择操作，那么 b 进入下一代种群. 故引入选择概率动态变化机制，可以利用拥有较优目标函数的不可行解信息.

3.2.2　g06 问题实验结果解释　g06 的最优解位于可行域边界上，且不存在等式约束. 如果选择操作只利用 DEB 准则，那么进化过程只会通过可行域一侧接近最优解 g；如果选择操作利用选择概率动态变化机制，那么进化过程不仅通过可行域一侧接近最优解 g，还会通过不可行域一侧接近最优解 g. 综上所述，当求解此类问题，利用选择概率动态变化机制比利用 DEB 准则多了一条从不可行域进化的路径，故表 2 中 Multi-ABC 算法求解 g06 的效果显著优于 MABC 算法.

3.3　等式约束放松机制及 g13 问题的解释

3.3.1　等式约束放松机制　进化算法在处理等式约束时，一般的思路都是将等式约束转化为不等式约束，并将约束放松程度设定一个固定不变的数值 ε. 本文为了充分利用等式约束周围不可行解的信息，利用式(8)实现 ε 随着进化代数增加从大变小、直到 10^{-3} 的过程.

等式约束放松机制的基本原理如图 4 所示. 其

中，为了简化问题，设定等式约束是一条直线 L，约束放松程度为 ε，L_1 和 L_2 是与 L 平行且距离为 ε 的2条直线. 引入约束放松机制后，问题的可行域从直线 L 变为 L_1 和 L_2 之间的区域，此区域内都是可行解，它们都可以用来引导进化过程，即该机制可以利用等式约束周围 ε 区域内的不可行解，从而引导进化过程.

按照式(8)，随着进化代数的增加，ε 从大变小，L_1 与 L_2 之间的区域面积也从大变小，这表示该机制利用的不可行解信息逐渐减少.

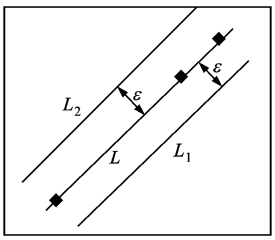

图4 等式约束放松机制的模型

3.3.2 g13问题实验结果的解释 针对g13函数，表2显示，当 $\varepsilon_{init}=5$ 时，Multi-ABC算法的结果显著优于MABC算法.

为了解释以上现象，本文给出了 $\varepsilon_{init}=5$ 时的典型进化过程，如图5所示.

图5表现了进化过程的最优解由小到大，这是合理的. 因为利用等式约束放松机制，初始阶段约束放松程度大，从而等式约束周围大量的不可行解被当做可行解，故初始阶段最优解小，而进化阶段后期，则正好相反.

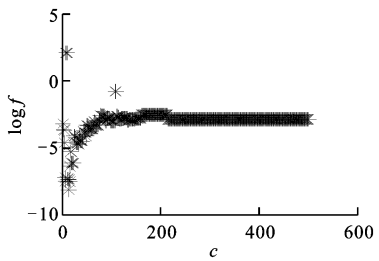

图5 $\varepsilon_{init}=5$ 时进化过程的散点图

4 结 论

本文针对约束优化问题提出了一种新颖的Multi-ABC算法. 新算法主要进行了以下3个方面的改进：①为了增强在可行域内的搜索能力，设计了一种多成员机制；②为了增强种群的分散性，允许以一定的概率 P_{SR} 进行基于目标函数的选择操作；③为了利用等式约束附近的不可行解，引入一种约束放松机制.

针对13个benchmarks函数仿真实验的结果表明：

(1)在处理最优解位于可行解内部且可行解区域较大的问题时，Multi-ABC算法显著优于MABC算法；

(2)在处理含有等式约束且可行域较小的问题时，Multi-ABC算法也显著优于MABC算法.

参考文献：

[1] 尚万峰，赵升吨，申亚京，等. 变容差遗传算法求解多约束问题的研究[J]. 西安交通大学学报，2007，41(11)：1267-1270.
SHANG Wanfeng，ZHAO Shengdun，SHEN Yajing，et al. Genetic algorithm and flexible tolerance algorithm hybridized for global optimization problems with multiple constraints[J]. Journal of Xi′an Jiaotong University，2007，41(11)：1267-1270.

[2] KARABOGA D，BASTURK B. On the performance of artificial bee colony (ABC) algorithm [J]. Applied Soft Computing，2008，8(1)：687-697.

[3] BANHARNSAKUN A，ACHALAKUL T，SIRINAOVAKUL B. The best-so-far selection in artificial bee colony algorithm[J]. Applied Soft Computing，2011，11(2)：2888-2901.

[4] GAO Weifeng，LIU Sanyang. Improved artificial bee colony algorithm for global optimization[J]. Information Processing Letters，2011，111(17)：871-882.

[5] KARABOGA D，AKAY B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems [J]. Applied Soft Computing，2011，11(3)：3021-3031.

[6] DEB K. An efficient constraint handling method for genetic algorithms [J]. Computer Methods in Applied Mechanics and Engineering，2000，186(13/4)：311-338.

[7] MEZURA-MONTES E，COELLO C A C. A simple multimembered evolution strategy to solve constrained optimization problems [J]. IEEE Transactions on Evolutionary Computation，2005，9(1)：1-17.

[8] RUNARSSON T P，YAO Xin. Stochastic ranking for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation，2000，4(3)：284-294.

(编辑 刘杨 赵大良)