

# 基于交叉突变算子的人工蜂群算法及其应用\*

邱剑锋<sup>1</sup>, 谢娟<sup>2</sup>, 汪继文<sup>1</sup>

(1. 安徽大学 计算机科学与技术学院, 合肥 230601; 2. 安徽建筑大学 数理学院, 合肥 230022)

**摘要:** 人工蜂群 (artificial bee colony, ABC) 算法在解决多峰函数优化问题时经常会陷入局部最优, 使得算法过早停滞, 而在解决单峰问题时往往出现收敛速度过慢的问题。针对上述不足, 为了进一步提高算法的优化性能, 提出了一种基于交叉突变的人工蜂群 (intersect mutation ABC, IMABC) 算法。IMABC 算法将整个蜂群依据其适应度值优劣进行划分, 引入种群划分参数, 对不同种群中的个体运用交叉突变算子, 有效地平衡了种群的局部开采与全局探测能力, 避免早熟收敛和提高收敛速度。从对基本函数的测试上可以看出, IMABC 相对于 GABC、IABC、ABC/best 等改进的 ABC 算法, 优化能力有了较大的提高。最后, 将 IMABC 用于优化 K-means 算法, 验证了该方法具有一定的实用性。

**关键词:** 人工蜂群算法; 交叉突变算子; 差分进化; 函数优化; K-均值

**中图分类号:** TP391      **文献标志码:** A      **文章编号:** 1001-3695(2014)05-1336-06

**doi:** 10.3969/j.issn.1001-3695.2014.05.012

## Improved artificial bee colony based on intersect mutation operator and its application

QIU Jian-feng<sup>1</sup>, XIE Juan<sup>2</sup>, WANG Ji-wen<sup>1</sup>

(1. School of Computer Science & Technology, Anhui University, Hefei 230601, China; 2. School of Mathematics & Physics, Anhui Jianzhu University, Hefei 230022, China)

**Abstract:** It is easier to get trapped in the local optima in optimizing complex multimodal functions while slow convergence speed in solving unimodal functions using ABC algorithm. In order to overcome these shortcomings and improve the optimization performance, this paper introduced an improved ABC algorithm based on intersect mutation strategies. According to the IMABC algorithm, the whole bee colony could be divided into two sub-populations by introducing the parameter  $M$ . An intersect mutation operator could be applied to the individuals belong to different sub-populations and got a balance between local exploitation and global exploration among the populations avoiding premature convergence and enhancing convergence speed. The experiments test the performance improvement comparing the other improved ABC algorithm, such as, GABC, IABC, ABC/best. At last, the IMABC algorithm can be applied to optimize K-means algorithm and a better experimental result validate its practicability and effectiveness.

**Key words:** artificial bee colony algorithm; intersect mutation operator; differential evolution; function optimization; K-means

## 0 引言

群智能 (swarm intelligence, SI) 是受蚂蚁、蜜蜂、鸟群和鱼群等动物种群表现出来的智能行为启发而形成的一种新的演化计算技术, 目前已成为国内外许多学者关注的热点。SI 当中的“群”可以看做是由一些无智能或仅具有简单智能的个体组成, 其个体之间通过自组织、信息交换等方式使整个种群表现出智能行为。1999 年 Bonabeau 等人<sup>[1]</sup>给出群智能的定义: 群智能是指任何一种受社会昆虫群体和其他动物种群的社会行为启发而设计出的算法或以分布式方式解决问题的策略。由于基于 SI 技术的算法在解决优化问题时相对于传统的优化算法普遍具有实现简单、收敛速度快、控制参数少的优势, 一些基于 SI 技术的优化算法应运而生, 其中典型的算法包括模拟

蚂蚁在寻找食物过程中发现最佳路径的蚁群优化 (ant colony optimization, ACO) 算法<sup>[2]</sup>、受鸟群觅食行为的启发并加以抽象而形成的粒子群优化 (particle swarm optimization, PSO) 算法<sup>[3]</sup>等, 并得到了广泛深入的应用。

基于种群的优化算法由于其自身的优势 (劳动分工、自组织、信息交互、并行操作等), 使得其在优化问题领域将继续成为一个研究的热点。近年来, 蜜蜂在觅食过程中通过劳动分工 (采蜜蜂、观察蜂和侦察蜂) 和不同工种的蜜蜂之间信息交换 (摇摆舞) 所表现出来的群智能现象引起了中外学者的关注。Karaboga<sup>[4]</sup>于 2005 年首次提出人工蜂群 (artificial bee colony, ABC) 算法, 该算法具有实现简单、灵活、参数少的优点, 实验结果表明在解决单峰及多峰的数值优化问题时具有一定的优势。此后, Karaboga 等人<sup>[5,6]</sup>将 ABC 算法用于优化不同维数数值问题, 并与进化算法 (evolutionary algorithm, EA)、遗传算法 (ge-

**收稿日期:** 2013-07-29; **修回日期:** 2013-09-02      **基金项目:** 国家自然科学基金资助项目 (61075049); 安徽省自然科学基金资助项目 (KJ2013A009, KJ2012B038); 安徽省优秀青年人才基金资助项目 (2011SQRL018); 安徽大学青年科学研究基金资助项目 (KJQN1015)

**作者简介:** 邱剑锋 (1979-), 男, 安徽合肥人, 讲师, 博士研究生, 主要研究方向为自然计算、群智能算法 (qiujianf@ahu.edu.cn); 谢娟 (1980-), 女, 副教授, 硕士, 主要研究方向为人工智能、统计计算; 汪继文 (1958-), 男, 教授, 博士, 主要研究方向为人工智能。

netic algorithm, GA)、PSO、差分进化(differential evolution, DE)算法等其他群智能进化算法进行了分析比较。然而中外不少研究者通过对ABC算法性能的分析研究发现,与其他进化算法类似,ABC算法在解决单峰问题时往往会出现收敛速度较慢,而在优化多峰问题时又易于陷入局部最优,出现“早熟”现象。针对上述问题,许多学者提出了不同的解决方案,代表性的算法包括基于PSO的启发、Zhu等人<sup>[7]</sup>提出的GABC算法。GABC算法是在整个蜂群的搜索策略中引入全局最优解,以期提高算法的收敛速度,改进其局部开采能力获得较好的优化能力。类似地,受DE算法启发,Gao等人<sup>[8]</sup>提出了改进的ABC算法——ABC/best。与GABC类似,在其搜索策略中,每次的迭代总是向当前的最优解趋近,从而使得单峰问题快速趋向最优解,提高收敛速度。胡珂等人<sup>[9]</sup>提出改进的人工蜂群算法,利用数学中的外推技巧定义了新的位置更新公式,由此构造出一种具有引导趋势的蜂群算法,提高算法在可行区域内的局部搜索能力。上述这些改进ABC算法的提出,改进了收敛速度,尤其是其局部开采能力得到了提高,但也不同程度地存在着陷入局部最优的可能,尤其是在解决多峰优化问题。同时,ABC算法在优化神经网络参数<sup>[10,11]</sup>、数据挖掘<sup>[12,13]</sup>、图像处理<sup>[14]</sup>多个不同的应用领域取得了卓有成效的应用。

众所周知,在基于SI的优化算法中,局部开采与全局探测能力的平衡是影响优化算法性能的重要因素。过度强调局部开采能力将使优化算法过早地停滞,陷入局部最优;反之,将使算法收敛速度降低。针对上述问题,本文在ABC算法的基础上,根据每个蜜源的适应度值的优劣进行种群划分,引入种群划分参数 $M$ ,并在种群之间运用交叉突变算子,通过对标准测试函数的测试验证该方法的有效性,最后将改进的ABC算法用于优化K-均值聚类算法,说明该方法具有一定的实用性。

## 1 基本的人工蜂群算法及研究现状

人工蜂群算法模拟蜜蜂在觅食过程中所产生的智能行为。根据劳动分工的不同,整个蜂群由承担三种不同任务的蜜蜂组成,即采蜜蜂、观察蜂和侦察蜂。其中采蜜蜂和观察蜂各占整个蜂群的50%,对每个蜜源仅允许一个蜜蜂去采蜜,即对应一个采蜜蜂,当一个蜜源因含蜜量殆尽,其对应的采蜜蜂转换为侦察蜂,重新选取新的蜜源。周而复始,最终使得整个蜂群找到最佳蜜源。在ABC算法中,蜜源对应优化问题的解,不同的问题其编码方式不同,通常表示为一个 $n$ 维向量。蜜源的优劣往往与其含蜜量的多少、距离蜂巢的远近等因素相关。这些因素在ABC算法中通常抽象地用适应度值来衡量。整个算法由采蜜蜂、观察蜂和侦察蜂三个阶段组成<sup>[4]</sup>。整个算法的框架如下所示:

种群初始化,形成 $SN$ 个蜜源, $x_i = 1, \dots, SN$

评估每个蜜源的优劣

repeat

a) 采蜜蜂阶段:对每个蜜源在其邻域附近实施局部搜索,评估搜索结果的好坏,根据贪婪算法更新蜜源;

b) 观察蜂阶段:依据蜜源适应度值的好坏,随机选取一个蜜源执行上述更新过程;

c) 侦察蜂阶段:从整个蜜源中淘汰一个含蜜量即将殆尽的蜜源(或者说该蜜源经过多次搜索仍无法改进),进而对应的采蜜蜂转为侦察蜂,继续随机地从整个种群中选择新的蜜源;

until 满足预先设定的条件

### 1.1 初始化

基于种群的优化算法希望初始种群尽可能均匀地分布在搜索空间以增加种群的多样性,其分布情况对整个算法的收敛速度和最优解的质量密切相关。ABC算法中每个蜜源对应搜索空间的一个可行解,采用均匀分布的方式产生初始蜜源:

$$x_{ij} = x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min})$$

$$i = 1, \dots, SN; j = 1, \dots, D \quad (1)$$

其中: $SN$ 代表蜜源的数量, $D$ 代表每个蜜源(即候选解)的维数。初始化之后,整个ABC算法进入前述三个阶段的循环,直至满足某种条件,完成搜索任务。在种群初始化研究方面,混沌映射和基于反向学习的种群初始化方法有助于提高种群的多样性和算法的收敛速度<sup>[15-17]</sup>。受此启发,Gao等人<sup>[18]</sup>将上述两种方法结合用于ABC算法的初始化,取得了较好的实验结果。

### 1.2 采蜜蜂阶段

每个蜜源仅由一只采蜜蜂进行开采,即蜜源的数量由采蜜蜂的数量决定。对蜜蜂采蜜过程加以抽象,在ABC算法中,每个采蜜蜂对其所在的蜜源处附近产生一个随机的扰动,产生一个候选解 $v_i$ ,其产生过程可由式(2)确定:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (2)$$

其中: $x_i$ 为搜索空间中第 $i$ 个蜜源; $x_k$ 为其邻域内随机选取的另一个候选解; $j$ 表示第 $j$ 个分量,其中 $k \neq i \in \{1, \dots, SN\}$ ;  $\varphi_{ij} \in [-1, 1]$ 为服从均匀分布的随机数。在ABC算法中利用随机选取的蜜源 $x_k$ ,由 $x_i - x_k$ 差值作为对当前蜜源 $x_i$ 的扰动项,以生成新的候选解。在式(2)中,当搜索过程接近问题的最优解时,扰动项 $x_i - x_k$ 会自适应地减小。

ABC算法优化能力与式(2)密切相关,通过借鉴相关领域的研究成果,出现了许多改进的ABC算法。Zhu等人<sup>[7]</sup>为了增强ABC算法的局部开采能力,在式(2)中引入最佳蜜源,以引导整个蜂群的寻优能力,提高收敛速度,但这种最佳蜜源的引入未必是全局最佳,这就导致了在解决多峰问题时出现过早收敛,即搜索停滞现象。DE算法因其在解决优化问题时的简单、有效,已成为进化计算领域的研究热点。借鉴DE算法的思想,Gao等人<sup>[18]</sup>提出了相应的ABC改进算法,如ABC/best、ABC/rand等。种种改进算法的提出,大大改进了ABC算法的性能,扩大了其应用范围。

### 1.3 观察蜂阶段

采蜜蜂阶段,对每个蜜源利用式(2)产生的候选解进行评估,根据贪婪算法确定是否需要更新蜜源。在蜂群觅食过程中,观察蜂驻留在巢穴周围,通过观察采蜜蜂所携带的各自蜜源信息飞往优质蜜源处,这种信息交互是蜂群觅食中智能行为的重要体现。在ABC算法中,对采蜜蜂阶段的每个蜜源,引入一个概率计算公式:

$$P_i = \frac{\text{fitness}_i}{\sum_{i=1}^{SN} \text{fitness}_i} \quad (3)$$

其中: $\text{fitness}_i$ 表示第 $i$ 蜜源的适应度值;对应的 $P_i$ 为该蜜源在整个蜂群中的优劣程度, $P_i$ 越大说明相应的蜜源好,对优化问题来说其解的质量更高。对 $P_i$ 高者,能吸引更多的观察蜂前往采蜜,以期发现更多的潜在优质蜜源,此时观察蜂变为采蜜蜂。

### 1.4 侦察蜂阶段

当蜜源的含蜜量即将殆尽时将被遗弃,进而开发新的未开采的蜜源。ABC 算法中,当一个蜜源经过采蜜蜂和观察蜂阶段多次开采之后,其解的质量仍得不到改进,该蜜源将被抛弃,进而随机选取新的蜜源。这个开采的上限是 ABC 算法中特有的一个参数,记为 limit。如果开采次数 counter > limit,该蜜源将被淘汰,进而根据式(1)选取新的蜜源。重复上述过程,直至满足预先设定的结束条件。

### 1.5 ABC 算法的参数设置

ABC 算法的参数主要包括种群大小(采蜜蜂 + 观察蜂)、开采次数的上限 limit、最大循环次数等。其中 limit 的取值过大降低收敛速度,取值过小会错失发现最优解的机会。对此,研究者给出各自的研究成果<sup>[19, 20]</sup>。

## 2 基于交叉突变算子的人工蜂群(intersect mutation ABC, IMABC)算法

基于进化或群智能的算法中,个体的社会学习能力是形成群体智能的因素之一。在 ABC 算法中,在采蜜蜂和侦察蜂阶段对其每个蜜源邻域的突变是其局部学习能力的表现,也是形成 ABC 算法智能的关键。从突变式(2)可以看出,每次只能随机选取一个蜜源,且信息的更新只能发生在其中一个方向上,这种随机性和更新的有限性限制了 ABC 算法的寻优能力,影响了算法的收敛速度。近年来 DE 已成为进化算法领域的研究热点,其中突变、交叉和选择算子的不同改进方法大大增强了算法的优化能力及应用范围<sup>[21-23]</sup>。交叉算子是将两个父代个体的部分成分加以交换重新结合成新个体,新个体中继承了父代优良基因,提高了种群的多样性,增强了算法的优化能力,在 DE 算法中起到了核心作用。本文同样借鉴 DE 算法的基本思想,在 ABC 算法的基础上引入交叉算子,以期提高 ABC 算法的整体寻优能力。算法的具体步骤描述如下:

a) 与 ABC 算法类似,执行初始化,采蜜蜂阶段和侦察蜂阶段。

b) 在侦察蜂阶段之后,引入交叉突变,即根据蜜源情况不同进行划分,针对不同的种群采用不同的交叉算子:

(a) 首先随机地从当前的蜜源中选取两个蜜源,计算其对应的适应度值,根据适应度值的大小,优者胜出,进入父代种群  $P_o$ ,直至父代种群的个数满足种群划分规模参数  $M$ ,完成种群划分。

(b) 对父代种群  $P_o$  中的每一个蜜源  $x_i$ ,本文采用的突变方式是随机地从剩下蜜源中选取一个蜜源  $x_{poor1}$ ,再从当前的父代种群  $P_o$  中随机选取两个蜜源  $x_{better1}$ 、 $x_{better2}$ ,按式(4)进行突变运算:

$$v_i = x_{poor1} \varphi_1 + (x_{better1} - x_{better2}) \varphi_2, \text{poor1} \neq \text{better1} \neq \text{better2} \quad (4)$$

其中:  $\varphi_1$ 、 $\varphi_2$  是在  $[0, 1]$  上的随机数。

(c) 评估  $x_i$  和  $v_i$  的适应度值,如果  $v_i$  的适应度值优于  $x_i$ ,则由  $v_i$  更新  $x_i$ 。

(d) 类似地,对剩下的蜜源采用式(5)进行突变运算:

$$v_i = x_{better1} \varphi_1 + (x_{poor1} - x_{poor2}) \varphi_2, \text{better1} \neq \text{poor1} \neq \text{poor2} \quad (5)$$

其中:  $x_{better1}$  随机地取自父代种群  $P_o$ ,  $x_{poor1}$ 、 $x_{poor2}$  取自剩下的蜜源中。

(e) 最后,评估  $x_i$  和  $v_i$  的适应度值,如果  $v_i$  的适应度值优

于  $x_i$ ,则由  $v_i$  更新  $x_i$ 。

c) 侦察蜂阶段,淘汰无开采价值的蜜源,继续随机生成新的蜜源。

IMABC 算法中的交叉算子在 ABC 算法的基础上,针对不同类型的优化问题作了一定程度上的平衡。算法中对父代种群  $P_o$  中的个体,利用与适应度较差的个体交叉,避免在解决多峰优化问题时算法过早停滞的现象,而对剩下的蜜源引入父代种群中较优的个体和当前种群中的个体进行交叉,扩大种群的多样性,使得算法能快速发现最优值,解决了在单峰优化问题时 ABC 算法收敛过慢的问题。IMABC 算法的伪代码如下所示:

初始化: 设置种群大小  $SN$ 、参数 limit、种群划分参数  $M$ ,基于混沌和反向学习的方法生成初始蜜源,计算各个蜜源的适应度值,形成对初始蜜源的评估。

```

while 循环条件不满足
    采蜜蜂阶段
    观察蜂阶段
    交叉阶段/* 引入交叉算子,提高算法的性能 */
    a) 划分种群,形成由  $M$  个蜜源组成的父代种群;
    b) 对父代种群中的每个个体,根据式(4)进行交叉、评估、更新;
    c) 对剩余的  $(SN - M)$  个个体,根据式(5)进行交叉、评估、更新;
    侦察蜂阶段
end while

```

## 3 仿真实验及分析

IMABC 中引入种群划分参数  $M$ 。为了验证该方法的有效性和实用性并确定  $M$  的有效取值范围,本文实验分为两组,第一组将 IMABC 方法用于一组标准的测试函数寻优问题,并与 ABC 及一些具有代表性的改进 ABC 算法在相同实验环境下进行比较;第二组是将 IMABC 算法用于 K-means 聚类算法的优化问题,说明该算法能较好地解决 K-means 算法受初始中心影响且易陷于局部最优的问题。

### 3.1 基本函数测试

表 1 给出了六个基本的测试函数,包括单峰、多峰等不同类型。

表 1 基本测试函数(UM:单峰;MM:多峰)

function	formulation	search range
Sphere ( $f_1$ )(UM)	$f(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]$
Rosenbrock ( $f_2$ )(UM)	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-50, 50]$
Ackley ( $f_3$ )(MM)	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i) + 20 + e$	$[-32, 32]$
Griewank ( $f_4$ )(MM)	$f(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$
Rastrigin( $f_5$ ) (MM)	$f(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$

不同的学者在不同的实验环境下分别比较了各自的改进 ABC 算法。为了充分研究 IMABC 算法性能,本文分别针对比较的对象不同而设置不同的实验环境。将 Zhu 等人<sup>[7]</sup>提出的 GABC 算法与基本 ABC 算法作出比较。本文采用与文献<sup>[7]</sup>相同的设置,即种群大小为 80,最大循环次数为 5 000,实验独立运行 30 次,记录运行的平均最优值和标准偏差。实验结果如表 2~6 所示。

表 2 Sphere 函数(UM)

algorithm	sphere function			
	D = 30		D = 60	
	mean	std	mean	std
ABC	6.379110e-16	1.203e-16	2.277413e-15	3.178e-16
GABC(C = 1.5)	4.176106e-16	7.365e-17	1.433867e-15	1.375e-16
IMABC(M = 0.1)	0	0	0	0
IMABC(M = 0.3)	0	0	0	0
IMABC(M = 0.5)	0	0	0	0
IMABC(M = 0.7)	0	0	0	0
IMABC(M = 0.9)	0	0	0	0

表 3 Rosenbrock 函数(UM)

algorithm	sphere function			
	D = 2		D = 3	
	mean	std	mean	std
ABC	9.931357e-03	8.143e-03	6.449468e-02	4.852e-02
GABC	1.684969e-04	1.454e-04	2.635891e-03	2.115e-03
IMABC(M = 0.1)	0	0	0	0
IMABC(M = 0.3)	0	0	0	0
IMABC(M = 0.5)	0	0	0	0
IMABC(M = 0.7)	0	0	0	0
IMABC(M = 0.9)	0	0	0	0

表 4 Ackley 函数(MM)

algorithm	ackley function			
	D = 30		D = 60	
	mean	std	mean	std
ABC	4.695503e-14	5.954e-15	1.660893e-13	2.217e-14
GABC	3.215205e-14	3.252e-15	1.000088e-13	6.089e-15
IMABC(M = 0.1)	8.88178e-16		0.88178e-16	0
IMABC(M = 0.3)	8.88178e-16		0.88178e-16	0
IMABC(M = 0.5)	8.88178e-16		0.88178e-16	0
IMABC(M = 0.7)	8.88178e-16		0.88178e-16	0
IMABC(M = 0.9)	8.88178e-16		0.88178e-16	0

表 5 Griewank 函数(MM)

algorithm	ackley function			
	D = 30		D = 60	
	mean	std	mean	std
ABC	1.273055e-15	1.464e-15	2.510399e-13	7.514e-13
GABC	2.960594e-17	4.993e-17	7.549516e-16	4.127e-16
IMABC(M = 0.1)	0	0	0	0
IMABC(M = 0.3)	0	0	0	0
IMABC(M = 0.5)	0	0	0	0
IMABC(M = 0.7)	0	0	0	0
IMABC(M = 0.9)	0	0	0	0

表 6 Rastrigin 函数(MM)

algorithm	rastrigin function			
	D = 30		D = 60	
	mean	std	mean	std
ABC	1.345294e-13	7.966e-14	2.064794e-08	1.121e-07
GABC	9.473903e-15	2.154e-14	3.524291e-13	1.243e-13
IMABC(M = 0.1)	0	0	0	0
IMABC(M = 0.3)	0	0	0	0
IMABC(M = 0.5)	0	0	0	0
IMABC(M = 0.7)	0	0	0	0
IMABC(M = 0.9)	0	0	0	0

从上述实验结果可以发现,IMABC 算法在优化多峰及单峰函数时其性能优于 GABC 算法。GABC 算法是在当前最优值的引导下进行寻优,理论上讲对于单峰函数可以加快其收敛,而对于多峰函数来说,优于这个当前最优值也可能是局部最优,因此可能会使寻优过程陷入停滞,从 GABC 的实验结果也证明了这一点。IMABC 算法通过引入种群划分,对单峰函数优化,可以利用与较优种群的交叉,提高整个种群的快速发现最优解的能力,提高收敛速度;而对多峰函数优化,通过交叉

算子,使可能已经陷入局部最优的个体脱离局部束缚,提高种群优化的能力。同时实验结果也表明,在种群数量足以满足优化问题需要时,其种群划分参数  $M$  对算法的影响并不明显。

受 DE 算法的启发,Gao 等人<sup>[8,18]</sup>提出了基于 DE 算法的 ABC/best/1、ABC/best/2 及 IABC 等改进 ABC 算法。为了与 IMABC 算法作比较分析,分别设置参数如下:

a)IMABC 与 IABC、DE、PSO

种群大小为 50(蜜源数 foodnumber 为 25),limit = foodnumber  $\times$  D,每个测试函数取相同的评估次数  $5.0e + 4$ ,实验独立运行 30 次,记录运行的平均最优值和标准偏差。实验结果如表 7 所示。

表 7 IMABC 与 IABC、DE、PSO 性能比较

function	DE <sup>[19]</sup>	PSO <sup>[19]</sup>	ABC <sup>[19]</sup>	IABC <sup>[18]</sup>	IMABC
	mean $\pm$ std				
Sphere (D = 30)	3.43e-14 $\pm$ 7.51e-14	2.13e-16 $\pm$ 3.21e-16	2.41e-09 $\pm$ 2.73e-09	6.75e-57 $\pm$ 1.72e-56	0
排名	4	3	5	2	1
Rosenbrock (D = 100)	4.71e+03 $\pm$ 3.76e+03	5.74e+03 $\pm$ 1.19e+04	4.06e+02 $\pm$ 5.10e+02	2.83e+02 $\pm$ 8.94e+01	9.63e+001 $\pm$ 3.32e-001(M = 0.3)
排名	4	5	3	2	1
Ackley (D = 30)	3.99e-08 $\pm$ 2.65e-08	3.23e-01 $\pm$ 6.70e-00	1.71e-05 $\pm$ 6.71e-06	3.87e-14 $\pm$ 8.52e-15	8.88e-016 $\pm$ 0.00e+000
排名	3	5	4	2	1
Griewank (D = 30)	6.15e-04 $\pm$ 2.68e-03	1.34e-02 $\pm$ 1.65e-02	2.59e-04 $\pm$ 1.12e-03	0	0
排名	3	4	2	1	1
Rastrigin (D = 30)	1.47e+02 $\pm$ 2.96e+01	3.85e+01 $\pm$ 9.68e-00	1.45e-01 $\pm$ 4.16e-01	0	0
排名	4	3	2	1	1

在表 7 中,相同的实验环境,算法的优化性能 IMABC 优于其他的进化算法,其中,Sphere、Griewank、Rastrigin 均达到最优值 0,有效地改进了多峰函数的寻优问题。

b)IMABC 与 ABC/best/1、ABC/best/2

种群大小为 100,foodnumber = 50,limit = 0.5  $\times$  foodnumber  $\times$  D,实验独立运行 30 次,记录运行的平均最优值和标准偏差。实验结果如表 8 所示。

表 8 IMABC 与 ABC/best/1、ABC/best/2 性能比较

function	D	G	ABC/best/2 <sup>[8]</sup>	ABC/best/1 <sup>[8]</sup>	IMABC
			mean $\pm$ std	mean $\pm$ std	mean $\pm$ std
Sphere	30	1000	1.57e-27 $\pm$ 1.14e-27	4.37e-22 $\pm$ 2.14e-22	0.00e+000 $\pm$ 0.00e+000
			1.57e-20 $\pm$ 4.90e-21	2.42e-25 $\pm$ 1.09e-25	0.00e+000 $\pm$ 0.00e+000
Rosenbrock	3	1000	1.95e-03 $\pm$ 1.52e-03	9.06e-06 $\pm$ 1.41e-05	0.00e+000 $\pm$ 0.00e+000
			1.86e-03 $\pm$ 1.03e-03	1.29e-07 $\pm$ 3.83e-07	2.30e-004 $\pm$ 8.76e-005
Ackley	30	1000	1.89e-11 $\pm$ 4.75e-12	1.26e-13 $\pm$ 3.48e-14	8.88e-016 $\pm$ 0.00e+000
			1.08e-10 $\pm$ 4.97e-12	3.40e-13 $\pm$ 6.35e-14	8.88e-016 $\pm$ 0.00e+000
Griewank	30	1000	4.47e-08 $\pm$ 1.05e-07	4.23e-11 $\pm$ 2.16e-11	0.00e+000 $\pm$ 0.00e+000
			2.18e-10 $\pm$ 4.33e-10	0 $\pm$ 0	0.00e+000 $\pm$ 0.00e+000
Rastrigin	30	1000	0 $\pm$ 0	0 $\pm$ 0	0.00e+000 $\pm$ 0.00e+000
			60	2000	0 $\pm$ 0

ABC/best/算法是在当前最优值的引导下,加速了 ABC 算法在解决单峰问题时的收敛速度问题。对单峰函数 Sphere 和 Rosenbrock,IMABC 算法达到了函数的最优值,明显优于 ABC/best 算法。在多峰函数 Ackley、Griewank 和 Rastrigin 优化时,IMABC 优于或保持与 ABC/best 的优化能力,减少了在优化多峰问题时过早陷入局部最优的问题。

针对表 5 的五个基本测试函数,分别采用 IMABC、ABC、ABC/best/1 和 ABC/best/2 算法对其进行优化。为了使算法具有可比性,基本的参数设置如下:种群大小为 80,最大迭代次数为 1 000,  $lim\ it = foodnumber \times D$ ,收敛性能分析如图 1 ~ 5 所示。

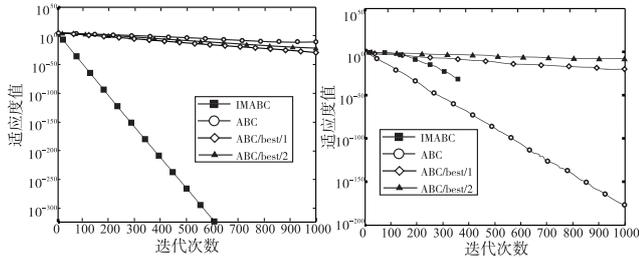


图1 Sphere函数(D=30)

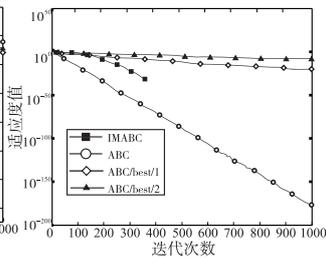


图2 Rosenbrock函数(D=30)

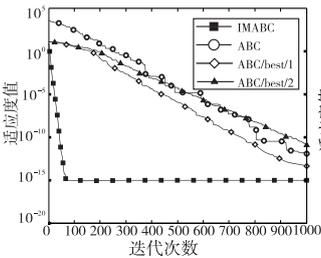


图3 Ackley函数(D=30)

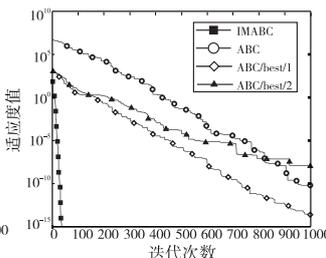


图4 Griewank函数(D=30)

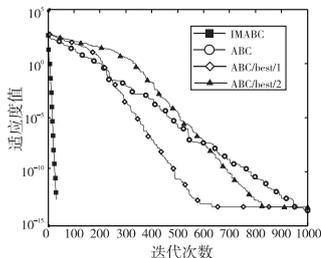


图5 Rastrigin函数(D=30)

在表 5 中的基本测试函数,其最优值均为 0,故算法的迭代终止条件设为超过最大迭代次数或函数值为 0。从上述实验中可以看出,图 1、2、4、5 中 IMABC 算法均在不超过最大迭代次数的情况下结束,即达到是函数的最优值,明显优于其他几种改进 ABC 算法。图 3 中的 Ackley 函数虽然在最大迭代次数内没有达到最优值,但收敛速度和性能上仍优于其他改进 ABC 算法。

IMABC 算法根据适应度值大小对种群进行划分,父代种群中的个体具有较好的适应度值,但其中的个体可能是全局最优,也可能是局部最优。为了减少陷入局部最优的风险,对父代中的个体,利用剩余个体中适应度值相对较差的个体进行交叉突变,以减少陷入早熟,算法停滞的现象;而对剩余个体组成的子群,利用父代种群中具有较好适应度值的个体对其进行交叉突变,增强种群的多样性,提高算法的收敛速度。从上述实验结果也可以看出,IMABC 算法在优化单峰和多峰问题时均有较好的表现,有效地平衡了算法的局部开采与全局探测能力。

### 3.2 数据聚类

聚类分析是在没有先验知识的情况下,能够按属性对各类事物进行合理分类的基本方法,广泛地应用于数据挖掘<sup>[24]</sup>、模式识别<sup>[25]</sup>、图像分析<sup>[26]</sup>等多个领域。数据聚类就是将一组数据按预先设定的相似性度量标准,将其分类到不同类的一个过程。聚类的结果使得同一类中的对象有很大的相似性,而类之

间的对象具有很大的相异性。

K-means 算法因其概念简单、计算方便已经成为聚类分析中最常用的算法之一,然而 K-means 算法对初始聚类中心的依赖性以及容易陷于局部最优的问题已经影响到聚类质量的进一步提高。为了克服上述问题,许多学者从不同的角度提出了一些改进措施。Krishna 等人<sup>[27]</sup>引入基于距离的突变算子用于优化 K-means 算法,提出遗传 K-均值算法;Omran 等人<sup>[28]</sup>将 PSO 与 K-means 结合用于提高聚类分析的质量;此外,许多学者也提出将 ABC 或者改进的 ABC 算法用于 K-means 聚类,其中具有代表性的包括 Karaboga 等人<sup>[13]</sup>提出的 ABCK-means、Zou 等人<sup>[29]</sup>提出的 CABCK-means、Yan 等人<sup>[30]</sup>提出的 HABCK-means。

#### 3.2.1 基于 IMABC 的 K-means 算法

将 IMABC 算法用于优化 K-means 算法,需要解决以下方面的问题:

a) 初始化。IMABC 算法中每个蜜源代表优化问题的一个候选解,具体到 K-means 问题中,其每个蜜源代表一种聚类的划分方案,即由一组聚类中心组成。假设样本集合  $\{x_i\}_{i=1}^N$ ,其中  $N$  为样本点个数。每个样本由  $p$  个属性组成  $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ ,整个数据集划分成  $k$  个子类。每个蜜源可由  $k$  个聚类中心组成,其中第  $i$  个蜜源可表达为  $C_i = (c_{i1}, c_{i2}, \dots, c_{ip}, c_{p+1}, \dots, c_{k \times p})$ 。

b) 评估蜜源的质量,即评估每一种聚类方案的优劣。聚类的目标就是找到一组最佳的划分方案,使得在该划分方案下,每个子类的内部耦合性最好,类间的耦合性最差。这种耦合性的好坏通常采用欧几里德距离作为度量标准。

首先定义样本集中任意两个样本点之间的距离公式为

$$T = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N d(x_i, x_j) \tag{6}$$

式(6)可改写为

$$\begin{aligned} T &= \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} (\sum_{C(j)=k} d(x_i, x_j) + \sum_{C(j) \neq k} d(x_i, x_j)) = \\ &= \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j)=k} d(x_i, x_j) + \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(j) \neq k} d(x_i, x_j) = \\ &= W(C) + B(C) \end{aligned} \tag{7}$$

其中:  $W(C)$  定义了在当前聚类划分方案  $C$  下的所有子类内部之间的耦合性,  $B(C)$  定义了类之间的耦合性。聚类方案的优劣  $\Leftrightarrow$  最小化  $W(C) \Leftrightarrow$  最大化  $B(C)$ 。如果距离  $d$  采用欧几里德距离公式,则  $W(C)$  可写为

$$W(C) = \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - m_k\|^2 \tag{8}$$

其中:  $m_k$  为第  $k$  个子类的中心,  $N_k$  为第  $k$  类中样本的个数。因此,将式(8)作为蜜源的适应度值,以此来评价每个蜜源(一种聚类方案)的好坏。

#### 3.2.2 IMABCK-means 仿真实验及结果

a) 数据集及算法的参数设置

为了说明 IMABC 算法在解决 K-means 算法时的有效性,本文选取了五个国际标准测试数据集 Iris、Wine、CMC、Glass 和 LD<sup>[31]</sup>(关于数据集的详细描述,可见参考文献[30]),并与 ABC、CABC、HABC、PSO 等其他算法<sup>[30]</sup>在优化 K-means 时性能作分析比较。为了便于分析比较,本文采用与文献[30]相同的 IMABC 参数设置,种群大小 100, limit = 100, 每个实验独立运行 30 次,并记录多次运行后平均值和标准偏差。具体的实验结果如表 9 所示。

表 9 IMABC、HABC、ABC、CABC、PSO、K-means 在五个数据集上的类内部距离之和的平均值和标准偏差

datasets		IMABC	HABC	ABC	CABC	PSO	K-means
Iris	mean	9.46024e+001	9.46034e+001	9.46106e+001	9.46039e+001	9.52257e+001	1.08303e+002
	std	2.65149e-006	2.86938e-006	9.68198e-003	5.62618e-004	6.56006e-001	1.95741e+001
Wine	mean	1.65313e+004	1.62977e+004	1.63060e+004	1.62992e+004	1.63371e+004	1.87911e+004
	std	7.61425e+000	5.27099e+000	3.64531e+001	8.10866e+000	6.36983e+001	7.45520e+002
CMC	mean	5.68368e+003	5.69486e+003	5.69567e+003	5.69628e+003	5.71398e+003	5.95809e+003
	std	3.45127e-001	9.01368e-001	1.96629e+000	5.49646e+000	2.4463e+001	1.82840e+002
Glass	mean	2.1534e+002	2.21660e+002	2.30550e+002	2.23393e+002	2.54035e+002	2.55300e+002
	std	2.384e-001	4.06821e+000	1.14880e+002	6.17616e+000	1.0107e+001	1.6353e+001
LD	mean	9.84229e+003	9.85178e+003	9.85175e+003	9.85179e+003	9.90415e+003	1.2063e+004
	std	7.13248e-001	1.12273e-002	8.25978e-002	1.46685e-001	2.34147e+002	6.53482e+002

从上述实验结果看出,IMABC 算法用于参与 K-means 过程,提高了聚类效果。与其他群智能算法相比,其优化性能 IMABC > ABC、CABC、PSO;仅仅在对 Wine 数据集聚类时,IMABC 略差于 HABC,其他数据集上 IMABC 仍表现出较好的优化能力。

### 4 结束语

本文针对 ABC 算法在解决单峰和多峰函数优化时的不足,利用种群划分参数对整个蜂群进行划分,在种群的个体之间运用交叉突变算子,在种群的局部开采和全局搜索作了较为有效的平衡。数值仿真实验验证了该方法具有较佳的寻优能力、较快的收敛速度、较好的鲁棒性等优点,最后将该方法用于 K-means 算法中距离优化问题,提高了聚类的效果。后期进一步将该算法拓展到诸如解决多目标优化等其他优化问题中,提高了算法的应用范围。

### 参考文献:

[1] BONABEAU E, DORIGO M, THERAULAZ G. Swarm intelligence: from natural to artificial systems [M]. [S. l.]: Oxford University Press, 1999.

[2] COLORNI A, DORIGO M, MANIEZZO V. Positive feedback as a search strategy [R]. 1991.

[3] KENNEDY J F, EBERHART R C. Particle swarm optimization [C] // Proc of IEEE International Conference of Neural Network. 1995.

[4] KARABOGA D. An idea based on honey bee swarm for numerical optimization, TR06 [R]. [S. l.]: Erciyes University Press, 2005.

[5] 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): 459-471.

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

[7] ZHU Guo-pu, KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization [J]. *Applied Mathematics and Computation*, 2010, 217(7): 3166-3173.

[8] GAO Wei-feng, LIU San-yang, HUANG Ling-ling. A global best artificial bee colony algorithm for global optimization [J]. *Journal of Computational and Applied Mathematics*, 2012, 236(11): 2741-2753.

[9] 胡珂, 李迅波, 王振林. 改进的人工蜂群算法性能 [J]. *计算机应用*, 2011, 31(4): 1107-1110.

[10] OZTURK C, KARABOGA D. Hybrid artificial bee colony algorithm for neural network training [C] // Proc of IEEE Congress on Evolutionary Computation. [S. l.]: IEEE Press, 2011: 84-88.

[11] 向娜. 基于神经网络和人工蜂群算法的水质评价和预测研究 [D]. 广州: 华南理工大学, 2012.

[12] 毕晓君, 宫汝江. 一种结合人工蜂群和 K-均值的混合聚类算法 [J]. *计算机应用研究*, 2012, 29(6): 2040-2042, 2046.

[13] KARABOGA D, OZTURK C. A novel clustering approach: artificial bee colony (ABC) algorithm [J]. *Applied Soft Computing*, 2011, 11(1): 652-657.

[14] CUEVAS E, SENCION-ECHAURI F, ZALDIVAR D, et al. Multi-cir-

cle detection on images using artificial bee colony (ABC) optimization [J]. *Soft Computing*, 2012, 16(2): 281-296.

[15] 王珊, 顾幸生. 基于混沌优化的双种群人工蜂群算法 [J]. *上海电机学院学报*, 2012, 15(1): 11-17.

[16] ALATAS B. Chaotic bee colony algorithms for global numerical optimization [J]. *Expert Systems with Applications*, 2010, 37(8): 5682-5687.

[17] RAHNAMAYAN S, TUZGUIS G H R, SALAMA M M A. Opposition-based differential evolution [J]. *IEEE Trans on Evolutionary Computation*, 2008, 12(1): 64-79.

[18] GAO Wei-feng, LIU San-yang. Improved artificial bee colony algorithm for global optimization [J]. *Information Processing Letters*, 2011, 111(17): 871-882.

[19] KARABOGA D, AKAY B. A comparative study of artificial bee colony algorithm [J]. *Applied Mathematics and Computation*, 2009, 214(1): 108-132.

[20] AKAY B, KARABOGA D. A modified artificial bee colony algorithm for real-parameter optimization [J]. *Information Sciences*, 2012, 192(1): 120-142.

[21] QIN A K, SUGANTHAN P N. Self-adaptive differential evolution algorithm for numerical optimization [C] // Proc of IEEE Congress on Evolutionary Computation. 2005: 1785-1791.

[22] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(2): 398-417.

[23] TINOCO J C V, COELLO C A C. hypDE: a hyper-heuristic based on differential evolution for solving constrained optimization problems [M] // EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II. Berlin: Springer, 2013: 267-282.

[24] JAIN A K. Data clustering: 50 years beyond K-means [J]. *Pattern Recognition Letters*, 2010, 31(8): 651-666.

[25] KUO R J, CHAO C M, LIU C Y. Integration of  $\langle i \rangle K \langle /i \rangle$ -means algorithm and AprioriSome algorithm for fuzzy sequential pattern mining [J]. *Applied Soft Computing*, 2009, 9(1): 85-93.

[26] LI Jun-qing, PAN Quan-ke, XIE Sheng-xian, et al. A hybrid artificial bee colony algorithm for flexible job shop scheduling problems [J]. *International Journal Computers Communications & Control*, 2011, 6(2): 286-296.

[27] KRISHNA K, NARASIMHA M M. Genetic K-means algorithm [J]. *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics*, 1999, 29(3): 433-439.

[28] OMRAN M, ENGELBRECHT A P, SALMAN A. Particle swarm optimization method for image clustering [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2005, 19(3): 297-321.

[29] ZOU Wen-ping, ZHU Yun-long, CHEN Han-ning, et al. A clustering approach using cooperative artificial bee colony algorithm [J]. *Discrete Dynamics in Nature and Society*, 2010, 2010: 1-16.

[30] YAN Xiao-hui, ZHU Yun-long, ZOU Wen-ping, et al. A new approach for data clustering using hybrid artificial bee colony algorithm [J]. *Neurocomputing*, 2012, 97: 241-250.

[31] BLAKE C, MERZ C J. UCI repository of machine learning databases [EB/OL]. <http://archive.ics.uci.edu/ml/datasets.html>.