

文章编号:1671-6833(2018)06-0030-06

一种应用于函数优化问题的多种群人工蜂群算法

王守娜^{1,2}, 刘弘^{1,2}, 高开周³

(1. 山东师范大学 信息科学与工程学院, 山东 济南 250014; 2. 山东师范大学 山东省分布式计算机软件新技术重点实验室, 山东 济南 250358; 3. 南洋理工大学 海事研究所, 新加坡 639798)

摘 要: 针对传统人工蜂群算法(ABC)收敛速度慢、易陷入局部最优解等不足,提出一种基于种群分割的多种群人工蜂群算法(MABC)应用于函数优化问题. 该算法利用 K 均值聚类算法对蜂群进行种群分割,在子种群中引入基于全局通信的蜜源位置更新方式加速算法收敛,同时引入基于局部通信的适应度函数扩展解方案的多样性. 通过对 6 个基准测试函数的实验表明, MABC 算法适应度高、收敛速度快,克服了 ABC 算法易陷入局部最优解等不足,在函数优化问题中表现出了更好的性能.

关键词: 人工蜂群算法; 种群分割; 蜜源位置更新; 适应度函数; 函数优化

中图分类号: TP183 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2018.06.019

0 引言

群体智能 (swarm intelligence, SI) 算法来源于自然界生物的觅食行为或进化过程的模拟,是一类新型的进化算法,主要特点是群体搜索策略和群体之间的信息交换. 典型的群体智能算法包括蚁群优化算法 (ant colony optimization, ACO)^[1]、粒子群优化算法 (particle swarm optimization, PSO)^[2] 和人工蜂群算法 (artificial bee colony, ABC)^[3] 等. 蚁群算法在求解性能上有很强的鲁棒性,但算法收敛速度慢,容易陷入局部最优,一般需要较长的搜索时间,而且该算法容易出现停滞现象,不利于发现更好的解. 粒子群算法具有相当快的逼近最优解的速度,其个体可以充分利用自身经验调整自身状态,缺点是容易产生早熟收敛,局部寻优能力较差.

人工蜂群算法 (artificial bee colony, ABC) 是 D. Karaboga^[4] 在 2005 年将蜜蜂的觅食行为应用到函数优化问题中而提出的,该算法计算简单、便于实现、鲁棒性强,在复杂组合优化问题中有明显的优势^[5],目前已经成功应用到模糊聚类^[6]、人工神经网络^[7]、传感器网络^[8] 等多个领域中. 与其他群体智能算法一样,ABC 算法也存在收敛速度慢、易陷入局部最优解等问题^[9]. 针对以上问

题,已有学者提出了相应的改进算法,这些改进算法在一定程度上提高了算法的收敛速度、寻优精度,改善了算法的性能,但是很难实现在避免算法早熟收敛的同时加快算法的收敛速度. 目前,大多数关于 ABC 算法的改进都是基于单一种群模型,没有考虑算法在基于多种群模型改进时相较于原始算法的优势.

聚类 (clustering)^[10] 是把所有数据对象划分到若干个子集的过程,每个子集是一个簇 (cluster),主要指导思想是尽量使相同簇内对象具有最大相似度. 在众多聚类方法中, K 均值算法^[11] 是一种非常重要的聚类算法,主要思想是依据参数 k 将给定数据集划分成 k 个簇,从而让各聚类中心的数据对象到其对应的中心点的距离平方和最小,该算法原理简单、易于实现.

笔者在原始人工蜂群算法的基础上,利用 K 均值算法的思想将蜂群划分为多个子蜂群,提出一种多种群人工蜂群算法 (multi-swarm artificial bee colony, MABC). 子种群将全局通信模式和局部通信模式相结合,基于局部通信的适应度函数扩展了解方案的多样性,基于全局通信的蜜源位置更新方式加速算法收敛,避免算法陷入局部最优. 在 Matlab 编程环境下选取 6 个标准测试函数对算法进行测试,证明了该算法能显著提高 ABC

收稿日期:2018-05-01;修订日期:2018-07-12
基金项目:国家自然科学基金资助项目(61472232,61272094)
通信作者:刘弘(1955—),女,山东济南人,山东师范大学教授,博士,博士生导师,主要从事分布式人工智能领域研究, E-mail:lhsden@126.com.

算法的收敛速度和寻优精度。

1 原始人工蜂群算法(ABC)

人工蜂群算法是模拟蜜蜂的采蜜行为提出的一种智能优化算法,由蜜源、被雇佣蜂(引领蜂)和未被雇佣蜂(跟随蜂、侦查蜂)3项基本元素构成。同时,引入3种基本的行为模式:搜索蜜源、招募蜜源和放弃蜜源。其中,蜜源的位置代表优化问题的可行解;蜜源的质量对应解的适应度。

最优蜜源的搜索过程如下:首先,算法随机产生含有 N 个侦查蜂的初始种群,初始时所有蜜蜂都为侦查蜂,同时也是蜜源的数量,每个解 \mathbf{X}_i 是一个 d 维向量, $\mathbf{X}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ 。计算解对应函数值,设定一个临界值,将大于临界值的解作为蜜源位置。一个引领蜂与一个蜜源是相对应的,与第 i 个蜜源相对应的引领蜂依据式(1)寻找新的蜜源。

$$V_{ij} = X_{ij} + R_{ij}(X_{kj} - X_{ij}), \quad (1)$$

式中: $i = 1, 2, \dots, N$, V_{ij} 代表更新后的蜜源; j 代表蜜源更新的维度, $j = 1, 2, \dots, d$; k 代表学习的蜜源位置, $k \neq i$, k, j 都是随机选择的,并且 $k \neq j$; R 为 $[-1, 1]$ 的随机数。跟随蜂采用轮盘赌策略选择蜜源,蜜源选择的概率公式如下:

$$p_i = \text{fitness}_i / \sum_{j=1}^N \text{fitness}_j, \quad (2)$$

式中: p_i 代表第 i 个蜜源被选择的概率; fitness_i 代表第 i 个蜜源的质量,即第 i 个解的适应度值,计算公式如式(3)所示; N 代表解的个数。

$$\text{fitness}_i = \begin{cases} \frac{1}{1 + f_i}, & f_i \geq 0; \\ 1 + \text{abs}(f_i), & f_i < 0. \end{cases} \quad (3)$$

式中: f_i 是第 i 个解的目标函数值,每个蜜源被选择的概率与适应度成正比。

所有引领蜂和跟随蜂经过一次邻域搜索后,检查蜜源未更新的次数 Limit , 如果一个蜜源经过 Limit 次循环后没有得到改善,则该蜜源将会被放弃,它所对应的引领蜂转为侦查蜂,侦查蜂通过公式(4)搜索新的蜜源 $V_{ij}^{[12]}$ 。

$$V_{ij} = X_j^{\min} + r_{ij}(X_j^{\max} - X_j^{\min}), \quad (4)$$

式中: r_{ij} 是区间 $[0, 1]$ 上的随机数; j 代表蜜源更新的维度, $j = 1, 2, \dots, d$; X_j^{\max} 和 X_j^{\min} 分别是第 j 维的上界和下界。

原始人工蜂群算法具有控制参数少、计算简单、便于实现等优点,已经被越来越多的研究者关注。但是该算法存在收敛速度慢、容易陷入局部最

优解等缺点,需要在收敛速度和寻优精度等方面加以改进。

2 多种群人工蜂群算法(MABC)

2.1 种群分割

首先,从蜜源 $X_i = \{x_1, x_2, \dots, x_i, \dots, x_N\}$ 中随机选取 \sqrt{N} (\sqrt{N} 取整数) 个蜜源作为初始聚类中心 C , 令 $C = \{C_1, C_2, \dots, C_j, \dots, C_{\sqrt{N}}\}^{[13]}$; 然后,利用基于 K 均值算法中基于欧氏距离的方式将剩余蜜源划分到每个聚类中心,即对于蜜源 x_i 按照 $\min\{D_{ij}(x_i, C_j)\}$ 的规则将人工蜂群划分为 \sqrt{N} 个子种群, D_{ij} 代表蜜源 x_i 到聚类中心 C_j 的距离。根据 ABC 算法思想,种群分割后的蜂群蜜源个数和引领蜂个数均为 \sqrt{N} 。

种群分割的具体步骤如下:

步骤1 随机初始化 \sqrt{N} 个聚类中心;

步骤2 将蜜源按照基于欧氏距离的方法依次划分到每个聚类中心;

步骤3 重新计算每类中所有节点的算数平均值,得到更新后的 \sqrt{N} 个簇的中心点;

步骤4 分别计算当前所有蜜源到相应的聚类中心的均方差之和;

步骤5 如果当前迭代次数的均方差之和与前次相同,结束蜜源聚类,转到步骤6,否则转到步骤2;

步骤6 输出蜜源聚类结果,即种群分割结果。

假设在边长为 $300 \text{ cm} \times 200 \text{ cm}$ 的空间中蜜源数量为 100,按照本文的种群分割方法将其划分为 10 个子种群,其中相同颜色的个体为同一个子种群,划分结果如图 1 所示。

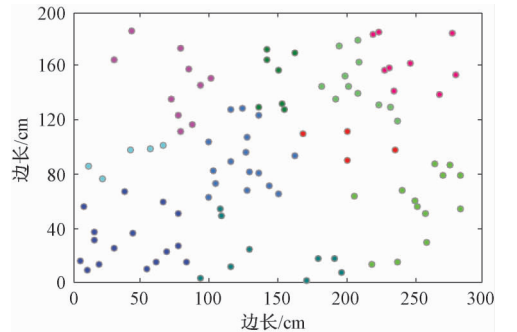


图1 种群分割结果

Fig. 1 The result of population segmentation

2.2 蜜源位置更新方式

ABC 算法利用原始的位置更新公式(1)时,引领蜂缺乏与整个蜂群的信息交流和共享,算法

易陷入局部最优.种群分割之后,提出基于全局通信的新蜜源位置更新公式(5):

$$V_{ij} = X_{ij} + R_{ij}(X_{kj} - X_{ij}) + \theta_{ij}(X_{Cbest,j} - X_{ij}), \quad (5)$$

式中: V_{ij} 代表更新后的蜜源; $\theta_{ij} \in [0,1]$ 是一个随机数; $X_{Cbest,j}$ 代表蜜源丰富度最高的子种群的聚类中心.

公式(1)只能权衡自己所在位置和历史最优位置的食物源的丰富度.公式(5)基于全局通信方式加入引导因子 $X_{Cbest,j} - X_{i,j}$,使引领蜂能够参考所有子种群中当前最优的位置,有很好的方向性.

2.3 适应度函数

种群分割之后以子种群 $C = \{C_1, C_2, \dots, C_j, \dots, C_{\sqrt{N}}\}$ 为单位进行引领蜂的选取和角色转换,同一种群内样本对象相似度最高.引领蜂被跟随蜂选择的概率公式(2)更改为公式(6):

$$p_{c,i} = fitness_{c,i} / \sum_{j=1}^{c_n} fitness_j, \quad (6)$$

式中: $p_{c,i}$ 代表第 c 组中第 i 个引领蜂被选择的概率,因为将人工蜂群划分为 \sqrt{N} 个子种群,所以 $c = \{1,2,\dots,\sqrt{N}\}$; $fitness_{c,i}$ 代表第 c 组中第 i 个蜜源的质量,即第 c 组中第 i 个解的适应度值; c_n 代表第 c 组中蜜源的数量.

基于局部通信的适应度函数由式(7)代替ABC算法中式(3):

$$fitness_i = \frac{1}{1 + F_i}. \quad (7)$$

改进的目标函数值 F_i 计算公式如式(8)所示:

$$F_i = \sum_{X_i \in C_j} D(X_i, C_j) / Num_{C_j}, \quad (8)$$

式中: $\sum_{X_i \in C_j} D(X_i, C_j)$ 代表种群 C_j 中所有蜜源到聚类中心的距离之和; Num_{C_j} 代表种群 C_j 中蜜源的数量.

公式(7)中的适应度函数同时考虑了子种群中的蜜源数量和距离,提高了算法的鲁棒性.

2.4 MABC 算法步骤

步骤1 根据ABC算法初始化蜂群,包括种群规模 N ,最大迭代次数 $MaxCycle$,适应度阈值 $Limit$,维度 Dim 等.

步骤2 利用 K 均值聚类算法中基于欧氏距离的方法对人工蜂群进行种群分割,得到 C 个子种群,其中 $C = \{C_1, C_2, \dots, C_j, \dots, C_{\sqrt{N}}\}$,每个子种群中分别独立运行.

步骤3 根据公式(7)分别计算 C_j 个子种群中蜜蜂的适应度值,适应度值高的一半作为引领

蜂,剩余的作为跟随蜂.

步骤4 C_j 中的引领蜂根据公式(5)搜索蜜源,根据公式(7)计算蜜源适应度值,若适应度值提高,则更新蜜源,否则保留原蜜源不变.

步骤5 引领蜂提供蜜源丰富度信息,跟随蜂根据公式(6)选择引领蜂,并采用公式(5)进行邻域蜜源搜索.

步骤6 更新 C_j 状态,迭代次数加1,若迭代次数达到 $Limit$ 后仍有未更新的蜜源,由侦查蜂根据公式(4)产生一个新蜜源.

步骤7 迭代次数达到最大迭代次数 $MaxCycle$ 时,算法结束;否则,返回步骤3.

3 实验仿真及分析

实验中采用操作系统是 Windows 10,处理器是 Intel(R)Core(TM) i5-6300HQ CPU @ 2.30 GHz,内存 4 GB 的计算机,编译软件为 Matlab-R2016a.

笔者使用 6 个典型的测试函数^[5]对 MABC 算法、ABC 算法^[14]及文献[15]中改进 ABC (CSABC)算法的鲁棒性、收敛速度和寻优精度进行对比试验.

3.1 测试函数

为了测试本文 MABC 算法在函数优化问题中的性能,选择 6 个常用的函数作为测试函数,见表 1.其中, $f_1 \sim f_3$ 是单峰函数,主要用来测试算法的收敛速度和寻优精度; $f_4 \sim f_6$ 是多峰函数,主要用来测试算法的全局寻优能力和避免早熟的能力.

表 1 测试函数
Tab.1 Testing functions

函数	名称	搜索空间	最小值
f_1	Sphere	$[-100,100]$	0
f_2	Rosenbrock	$[-30,30]$	0
f_3	Schwefel	$[-100,100]$	0
f_4	Rastrigin	$[-5.12,5.12]$	0
f_5	Ackley	$[-32,32]$	0
f_6	Griewank	$[-600,600]$	0

测试函数 $f_1 \sim f_6$ 表达式如下:

$$f_1(x) = \sum_{i=1}^n x_i^2;$$

$$f_2(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2];$$

$$f_3(x) = 4.1898n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|});$$

$$f_4(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10];$$

$$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) -$$

$$\exp\left(\frac{1}{n}\sum_{j=1}^m\cos\left(2\pi x_j\right)\right)+20+\exp(1);$$

$$f_6(x)=\sum_{i=1}^n\frac{x_i^2}{4\,000}-\prod_{i=1}^n\cos\left(\frac{x_i}{\sqrt{i}}\right)+1.$$

$$Cycle=1\,000, \text{ 适应度阈值 } Limit=50, \text{ 维度 } Dim=50.$$

3.2 测试结果对比分析

MABC、ABC 和 CSABC 3 种算法对比试验参数设置如下:种群规模 $N=100$, 其中引领蜂和跟随蜂的数目均为 50, 最大迭代次数 $Max-$

3 种算法对每个测试函数分别独立进行 50 次实验,得到对应的最优值、最差值、平均值和方差,测试结果见表 2. 其中,最优值和最差值反映算法的质量,平均值反映算法的收敛速度和寻优精度,方差反映算法的稳定性和数据的离散程度.

表 2 函数测试结果对比
Tab.2 Comparison of functions test results

函数	算法	最优值	最差值	均值	方差
Sphere	ABC	9.233 7	1.150 7E+02	1.186 0E+01	2.978 1E+02
	CSABC	7.697 1E-02	7.771 3	3.124 7E-01	1.513 3E-01
	MABC	1.985 6E-06	4.874 42E-04	3.744 5E-05	6.452 3E-05
Rosenbrock	ABC	7.092 0E+04	2.147 7E+05	1.207 3E+04	4.695 3E+02
	CSABC	4.141 4E+03	5.702 8E+05	6.724 2E+04	1.456 3E+02
	MABC	1.691 4	3.700 7E+02	7.566 3	1.142 5E-01
Schwefel	ABC	-7.092 0E+03	-2.147 7	-1.207 3E+01	4.695 3E+03
	CSABC	-4.141 4E+3.5	-5.702 8E+02	-6.724 2E+03	1.456 3E+02
	MABC	-1.691 4E+05	-3.700 7E+03	-7.566 3E+04	1.142 5
Rastrigin	ABC	1.085 7E+03	9.433 8E+06	4.736 9E+05	8.204 6E+06
	CSABC	7.401 4E+02	3.330 6E+05	1.184 2E+04	1.407 2E+04
	MABC	1.333 8	2.378 1E+02	8.166 7	7.243 8E+01
Ackley	ABC	9.233 7E+01	1.150 0	1.186 0E+01	2.978 19
	CSABC	7.697 1E-01	7.771 3E+01	3.124 7	1.513 3
	MABC	1.985 6E-04	4.874 4E-02	3.744 5E-04	3.744 5E-03
Griewank	ABC	1.480 3	1.691 4E+01	4.033 9	6.523 7E+01
	CSABC	3.700 7E-01	1.920 6	4.159 6	7.043 8
	MABC	5.341 1E-05	3.588 7E-03	7.332 8E-05	2.824 3E-03

通过表 2 可以看出,笔者提出的 MABC 算法在单峰函数和多峰函数测试中均有较高的寻优精度. 算法的最优值、最差值与 ABC 算法和 CSABC 算法相比有明显的提升,说明 MABC 算法稳定性更高;在相同的迭代次数下,MABC 算法在各测试函数上均得到了最小均值和最小方差,表明其收敛速度更快,寻优精度更高,性能明显优于 ABC 算法和 CSABC 算法.

图 2 中给出了 MABC 算法、ABC 算法和 CSABC 算法独立运行 50 次的平均适应度值进化曲线. 由图 2(a)、(b) 可以看出,MABC 算法达到稳定收敛的次数远小于 ABC 算法和 CSABC 算法;由图 2(c) 可以看出,ABC 算法和 CSABC 算法在搜索过程中陷入局部极值,种群分割策略使 MABC 算法分组寻优,收敛值明显小于其他两种算法;由图 2(d) 可以看出,ABC 算法和 CSABC 算法收敛较快,迭代次数较少,但过早停滞,寻优精度不如 MABC 算法;由图 2(e)、(f)

可以看出,MABC 算法在搜索后期表现出较强的优势,而 ABC 算法和 CSABC 算法则容易陷入局部最优.

由以上分析可以看出,MABC 算法达到稳定收敛时的寻优精度远高于 ABC 算法和 CSABC 算法,尤其对单峰函数而言收敛速度更加明显.

4 结束语

笔者针对原始人工蜂群算法存在的收敛速度慢、易陷入局部最优解等不足,提出了一种基于种群分割的多种群人工蜂群算法(MABC),该算法利用 K 均值聚类算法中基于欧氏距离的方式对人工蜂群进行种群分割,将大的种群划分为多个子种群,每个子种群分别独立运行,在子种群中引入基于全局通信的蜜源位置更新方式加速算法收敛,同时引入基于局部通信的适应度函数扩展解方案的多样性. 通过 6 组标准测试函数的仿真实验结果可知,笔者提出的 MABC 算法比 ABC 算法

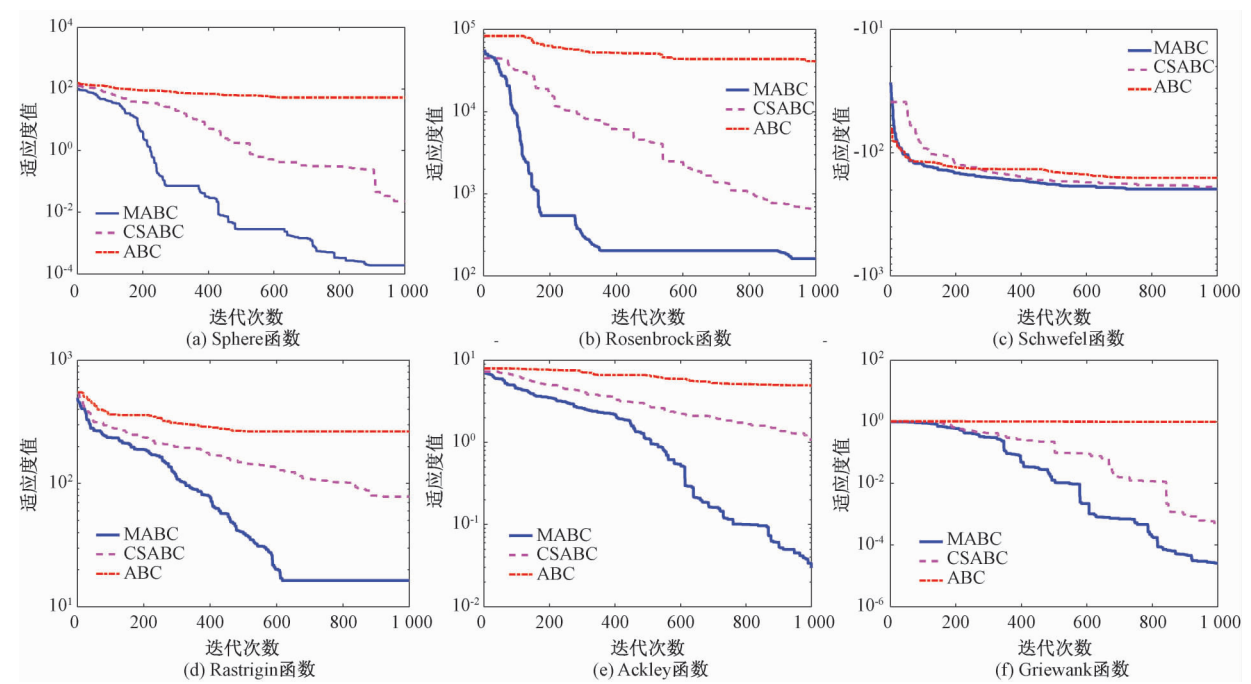


图 2 测试函数结果
Fig.2 The results of test functions

及 CSABC 算法收敛速度快、寻优精度高、鲁棒性高、适应度好。

在未来的工作中,我们计划进一步提高算法的性能,并通过更加多样性的测试问题进行验证,比如最优点不在原点或者各维度最优点不一样的测试函数进行测试,以证明没有偏差。另外笔者计划下一步将本文的 MABC 算法有效地应用到实际工程问题中,并采用数学方法使其表现出更多的价值。

参考文献:

[1] SAMÀ M, PELLEGRINI P, D'ARIANO A, et al. Ant colony optimization for the real-time train routing selection problem[J]. Transportation research, part B, 2016, 85:89-108.

[2] 梁静,宋慧,瞿博阳,等. 基于改进粒子群算法的路径优化问题研究[J]. 郑州大学学报(工学版), 2014,35(1):34-38.

[3] 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.

[4] KARABOGA D, OZTURK C. A novel clustering approach: artificial bee colony (ABC) algorithm[J]. Applied soft computing journal, 2011, 11(1): 652-657.

[5] NSEEF SK, ABDOLLAH S, TURKY A, et al. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems[J]. Knowledge-based systems, 2016, 104:14-23.

[6] KARABOGA D, OZTURK C. Fuzzy clustering with artificial bee colony algorithm[J]. Scientific research & essays, 2010, 5(14):1899-1902.

[7] 王允霞. 蜂群算法的研究及其在人工神经网络中的应用[D]. 广州:华南理工大学理学院, 2013.

[8] OZTURK C, KARABOGA D, GORKEMII B. Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm[J]. Sensors, 2011, 11(6):6056-6065.

[9] 徐斌,刘弘. 融合社会力与人工蜂群的人群疏散仿真方法[J]. 小型微型计算机系统, 2016, 37(8): 1725-1729.

[10] 胡燕,朱晓瑛,马刚. 基于 K-Means 和时间匹配的位置预测模型[J]. 郑州大学学报(工学版), 2017,38(2):17-20.

[11] KHANMOHAMMADI S, ADIBEING N, SHANEHBANOY S. An improved overlapping k-means clustering method for medical applications[J]. Expert systems with applications, 2016, 67:12-18.

[12] DERVIS D. Artificial bee colony algorithm[J]. Scholarpedia, 2010, 5(3):6915.

[13] 刘雷,王洪国,邵增珍,等. 一种基于蜂群原理的划分聚类算法[J]. 计算机应用研究, 2011, 28(5): 1699-1702.

[14] 秦全德,程适,李丽,等. 人工蜂群算法研究综述[J]. 智能系统学报, 2014,9(2):127-135.

[15] 毛力,周长喜,吴滨,等. 一种高效的求解函数优化问题的人工蜂群算法[J]. 小型微型计算机系统, 2016, 37(1):152-156.

A Multi-swarm Artificial Bee Colony Algorithm for Function Optimization

WANG Shouna^{1,2}, LIU Hong^{1,2}, GAO Kaizhou³

(1. School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China; 2. Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250358, China; 3. Maritime Institute, Nanyang Technological University, Singapore 639798, Singapore)

Abstract: A multi-swarm Artificial Bee Colony(MABC) algorithm based on the segmentation of population was proposed in this paper. It was applied to function optimization to overcome the drawbacks of slow convergence and low computational accuracy of conventional ABC algorithm. In this algorithm, K-means clustering algorithm based on Euclidean distance was introduced to divide the bee colony. In the subpopulation, a method was introduced to update the location of nectar based on global communication to accelerate the convergence of the algorithm; and the fitness function based on local communication was introduced to expand the diversity of the solution. The simulation results of six standard functions showed that the MABC algorithm could attain significant improvement on convergence rate and solution accuracy, and show better performance in function optimization problems when compared with the ABC algorithm.

Key words: Artificial Bee Colony algorithm; segmentation of population; nectar location updating; fitness function; function optimization

(上接第 22 页)

Improved Whale Swarm Algorithm and its Application in Steelmaking Continuous Casting Scheduling

ZENG Bing, WANG Mengyu, GAO Liang, DONG Haozhen

(School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: A new swam intelligent optimization algorithm named whale swarm algorithm (WSA) was studied. The principle and essential procedures of WSA were introduced; and the characteristics of WSA were presented through comparison with other classical swam intelligent optimization algorithms. For multimodal optimization, the iteration rule of WSA was improved, two parameters namely stability threshold and fitness threshold were introduced, and thus WSA with iterative counter (WSA-IC) was developed. The experimental results demonstrated that WSA-IC showed good performance in terms of the number and quality of optimal solutions and convergence speed. Then WSA-IC was applied to the steelmaking continuous casting scheduling problem, and proved to have good optimization ability and strong stability through the experiments. Finally, with the above research results, it was summarized that WSA had much value in practice, and further research of WSA could be carried out in theoretical study and practical application.

Key words: swam intelligent optimization algorithm; whale swarm algorithm; multimodal optimization; steel-making continuous casting scheduling