

基于刺激-响应分工机制的人工蜂群算法

王英聪¹, 刘军辉¹, 肖人彬^{2†}

(1. 郑州轻工业大学 电气信息工程学院, 郑州 450002; 2. 华中科技大学 人工智能与自动化学院, 武汉 430074)

摘要: 针对人工蜂群算法中探索与开采的不平衡以及由此导致的求解精度低、收敛速度慢等问题, 提出一种基于刺激-响应分工机制的人工蜂群算法. 将探索和开采看成两种不同的搜索任务, 令蜜蜂在雇佣蜂阶段执行探索, 在跟随蜂阶段执行开采. 根据种群多样性设计搜索任务的环境刺激, 利用搜索成功率设计蜜蜂个体的响应阈值. 在刺激-响应分工机制下, 蜜蜂在雇佣蜂和跟随蜂之间灵活转换, 从而实现探索和开采的平衡. 采用 22 个基准函数进行仿真实验, 实验结果验证了所提出算法的有效性.

关键词: 人工蜂群算法; 刺激-响应分工机制; 任务刺激; 响应阈值; 多样性; 成功率

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.1346

引用格式: 王英聪, 刘军辉, 肖人彬. 基于刺激-响应分工机制的人工蜂群算法[J]. 控制与决策, 2022, 37(4): 881-891.

Artificial bee colony algorithm based on stimulus-response labor division

WANG Ying-cong¹, LIU Jun-hui¹, XIAO Ren-bin^{2†}

(1. School of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China; 2. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: Aiming at the disadvantages of the imbalance between exploration and exploitation in the artificial bee colony algorithm, as well as the resulting low solution quality and slow convergence speed, this paper proposes an artificial bee colony algorithm based on stimulus-response labor division. Firstly, exploration and exploitation are regarded as two different search tasks, bees explore the search space in the employed bee phase and exploit the search space in the onlooker bee phase. Then, the environmental stimulus of the search task is designed according to the population diversity, and the response threshold of bees is designed according to the search success rate. Finally, under the mechanism of stimulus-response labor division, bees flexibly switch between the employed bee and the onlooker bee to achieve the exploration-exploitation balance. The experimental results on 22 benchmark functions show the high efficiency of the proposed algorithm.

Keywords: artificial bee colony algorithm; stimulus-response labor division; task stimulus; response threshold; diversity; success rate

0 引言

人工蜂群(artificial bee colony, ABC)算法是一种模拟蜜蜂协作觅食行为的群体智能算法^[1], 具有结构简单、易于实现、鲁棒性强、控制参数少等特点^[2]. 与其他基于群体的优化算法(如粒子群算法、遗传算法、蚁群算法、差分进化算法等)相比, ABC算法的寻优能力和算法精度都具有较强的竞争力^[3]. 目前, ABC算法已成功用于微网调度^[4]、车间调度^[5]、图像分割^[6]、负载均衡优化^[7]、天线阵列设计^[8]等领域.

平衡探索与开采之间的关系一直是基于群体的优化算法的研究热点^[9-11], 其中探索是指搜索解空间未知区域发现优良解的过程, 开采是指利用已知优良解信息搜索优良解的过程^[12]. ABC算法侧重探索而不善于开采, 导致其收敛速度慢且容易陷入局部最优^[13-14]. 针对这一不足, 学者们提出了很多改进方案. 例如, 文献[15-17]利用全局最优解信息提高算法的开采能力; 文献[18]在开采阶段采用 Rosenbrock 旋转方向法; 文献[19-20]分别为雇佣蜂和跟随蜂设计

收稿日期: 2020-09-27; 录用日期: 2021-01-08.

基金项目: 科技创新 2030——“新一代人工智能”重大项目(2018AAA0101200); 国家自然科学基金项目(61702463); 河南省科技攻关项目(192102210111); 郑州轻工业大学博士科研基金项目(2017BSJJ004).

责任编辑: 林崇.

†通讯作者. E-mail: rbxiao@hust.edu.cn.

了侧重探索和开采的搜索方程;文献[21]将种群分为多样性种群和收敛性种群分别进行探索和开采;文献[22]分别通过增大和减小种群规模加强探索和开采.此外,还有一些学者综合考虑多种具有良好探索能力或开采能力的搜索方程,并使用一定的选择策略来协调探索与开采,如符号函数选择^[23]、固定顺序选择^[24]、基于进化度的选择^[25]和基于成功率的选择^[26-27]等.

劳动分工是蜂群等社会性昆虫中的一种普遍现象,表现为不同的个体执行不同的任务.在劳动分工作用下,个体所执行的任务会随着环境动态调整,这种适应性经常被认为是社会性昆虫生态成功的首要原因^[28].本文将劳动分工融入ABC算法,提出一种基于刺激-响应分工机制的人工蜂群算法(artificial bee colony algorithm based on stimulus-response labor division, SRLDABC).新算法将探索和开采看成两种不同的搜索任务,令蜜蜂在雇佣蜂阶段执行探索任务,在跟随蜂阶段执行开采任务.每个任务都对应一个环境刺激,代表任务的紧急程度.蜜蜂对每个任务都存在一个响应阈值,代表蜜蜂执行该任务的倾向性.在环境刺激和响应阈值的共同作用下,蜜蜂在雇佣蜂和跟随蜂之间进行角色转换,以此来平衡算法的探索能力与开采能力.为验证本文算法的性能,在22个典型benchmark函数上进行仿真,并与基本ABC算法和其他改进ABC算法进行对比.

1 人工蜂群算法

ABC算法的灵感源于蜜蜂的摇摆舞和智能觅食行为,算法中包含雇佣蜂、跟随蜂和侦查蜂3种蜜蜂.雇佣蜂在食物源附近进行搜索,并将食物源信息分享给跟随蜂.根据共享信息,跟随蜂选择较优的食物源进行开采.当食物源枯竭时,对应的雇佣蜂转变为侦查蜂随机搜索食物源.求解优化问题时,食物源的位置代表优化问题的一个可行解,食物源的质量代表解的适应度.

初始时,食物源 $X_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$ 的位置在搜索空间内随机产生,有

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

其中: $i = 1, 2, \dots, \text{SN}$, SN为食物源(雇佣蜂或跟随蜂)的个数; $j = 1, 2, \dots, D$, D 为搜索空间的维数; x_j^{\min} 和 x_j^{\max} 分别为第 j 维分量的下界和上界; $\text{rand}(0, 1)$ 是 $[0, 1]$ 上均匀分布的随机数.

雇佣蜂在食物源 X_i 附近进行一维邻域搜索,生成新食物源 V_i ,有

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

其中: k, j 是随机选取的下标,且满足 $k \in \{1, 2, \dots, \text{SN}\}$, $j \in \{1, 2, \dots, D\}$, $k \neq i$; φ_{ij} 为 $[-1, 1]$ 上均匀分布的随机数.

雇佣蜂在 X_i 和 V_i 之间选择较优食物源,并向跟随蜂分享食物源信息.跟随蜂根据如下概率选择食物源:

$$P_i = \text{fit}_i / \sum_{m=1}^{\text{SN}} \text{fit}_m. \quad (3)$$

其中: P_i 是第 i 个食物源的选择概率, fit_i 是第 i 个食物源的适应度.食物源的适应度越高,它被跟随蜂选中的概率越大.

跟随蜂也根据式(2)在食物源附近进行邻域搜索,并通过贪婪选择更新食物源.如果某个食物源的质量连续 limit 次未提高,则该食物源处的雇佣蜂转变为侦查蜂,并根据式(1)在搜索空间内重新生成新食物源.

2 基于刺激-响应分工机制的人工蜂群算法

在蜂群等社会性昆虫中,不同的个体执行不同的任务,这种现象叫作劳动分工.劳动分工的显著特点是个体执行的任务会随着环境动态调整,表现出良好的适应性^[28].刺激-响应机制是一种典型的劳动分工机制,简要描述如下^[29]:族群中的每个任务都对应一个环境刺激 S ,刺激强度描述了任务的紧急程度;个体对于每个任务都存在一个响应阈值 θ ,阈值大小反映了个体执行任务的倾向性;当任务刺激 S 超过个体阈值 θ 时,个体执行任务的概率大,反之的概率小,具体概率计算如下:

$$\Gamma = \frac{S^2}{S^2 + \theta^2}. \quad (4)$$

此外,任务刺激 S 和个体阈值 θ 会随着族群需求和个体表现发生变化,使得个体可以灵活地执行任务.

针对ABC算法中探索与开采之间的不平衡,学者们提出了很多改进策略.本文将探索和开采看成两种不同的搜索任务,并采用劳动分工机制协调二者之间的关系.在社会性昆虫中,不同的角色通常对应不同的任务,劳动分工的适应性也经常被称为角色可塑性^[28].在ABC算法中,一般认为雇佣蜂负责探索,跟随蜂负责开采^[14,22].本文也保持这个原则,分别为雇佣蜂和跟随蜂设计具有较强探索能力和开采能力的搜索方程.通过将刺激-响应分工机制融入ABC算法,在劳动分工的角色可塑性特点下,蜜蜂能根据搜索环境的变化决定是转化为雇佣蜂进行探索,还是转化为跟随蜂进行开采.根据刺激-响应分工机制,本

文的SRLDABC算法主要包含搜索任务、环境刺激、响应阈值和执行概率4个部分。

2.1 搜索任务

为提高ABC算法的探索或开采能力,专家们提出了很多改进策略.一般而言,在搜索方程中引入随机解信息可加强探索,引入最优解信息可增强开采^[23-26].例如

$$v_{ij} = x_{rj} + \varphi_{ij}(x_{rj} - x_{kj}), \quad (5)$$

$$v_{ij} = \frac{x_{rj} + x_{kj}}{2} + \varphi_{ij}(x_{rj} - x_{kj}) + \phi_{ij}(x_{\text{best}j} - x_{rj}). \quad (6)$$

其中:式(5)表现出较强的探索能力,式(6)表现出较强的开采能力; r, k, j 是随机选取的下标,且满足 $r, k \in \{1, 2, \dots, SN\}, j \in \{1, 2, \dots, D\}, r \neq k \neq i$; x_{best} 是种群的全局最优解; φ_{ij} 和 ϕ_{ij} 分别为 $[-1, 1]$ 和 $[0, 1]$ 上均匀分布的随机数。

式(2)只进行一维邻域搜索,导致算法的搜索效率不高.为提高搜索效率,文献[30-31]引入参数MR来控制搜索的维度.具体的,对于每一维都生成一个 $[0, 1]$ 之间的随机数 R_{ij} ,若 $R_{ij} < MR$,则对应的这一维进行邻域搜索,否则不进行。

在本文算法中,雇佣蜂执行探索任务时,根据式(5)按变异率MR搜索食物源;跟随蜂执行开采任务时,根据式(6)按变异率MR搜索食物源.参照文献[30-31],MR在 $[0.3, 0.7]$ 之间随机取值。

2.2 环境刺激

在刺激-响应分工机制中,环境刺激描述了个体执行任务的外部驱动力.刺激越大,任务越紧急.在算法搜索过程中,探索与开采互相矛盾,可用种群多样性进行度量^[10].一般认为,算法探索时种群多样性好,算法开采时种群多样性差^[9].当种群多样性较差时应加强探索,当种群多样性较好时应加强开采^[32-33].因此,探索的刺激与多样性负相关,开采的刺激与多样性正相关。

多样性表现为个体之间的差异,反映了种群的分布情况.文献[34]定义了一种多样性度量方法,即

$$\text{diversity} = \frac{1}{SN \cdot L} \sum_{i=1}^{SN} \sqrt{\sum_{j=1}^D (x_{ij} - \bar{x}_j)^2}. \quad (7)$$

其中:SN为种群大小, L 为解空间大小, D 为解空间维度, \bar{x}_j 为种群中心.式(7)综合考虑且不依赖于种群大小、问题维度和每一维的搜索范围,能较好地度量种群多样性。

综上,将探索任务和开采任务的刺激分别定义为

$$S_{\text{explore}} = 1 - \sqrt[p]{\text{diversity}}, \quad (8)$$

$$S_{\text{exploit}} = \sqrt[p]{\text{diversity}}, \quad (9)$$

其中 p 为刺激对多样性的敏感系数.参考文献[34]的取值和仿真实验,文中 p 取50。

2.3 响应阈值

在刺激-响应分工机制中,响应阈值描述了个体执行任务的内部倾向性.阈值越小,倾向性越强.在族群奖励和惩罚机制下,个体成功执行任务时阈值会降低,未成功执行任务时阈值会增加^[28].个体对应任务的阈值可以用个体执行任务的成功率来度量,成功率越高,对应的阈值越低.因此,蜜蜂对应探索任务和开采任务的阈值分别与其进行探索和开采的成功率负相关。

在本文算法中,蜜蜂执行搜索任务成功指蜜蜂的食物源得到改进.令 n_t 表示执行某一任务的总次数, n_s 表示其中的成功次数,则蜜蜂执行该任务的成功率为

$$r = \frac{n_s}{n_t}. \quad (10)$$

综上,将蜜蜂对应探索任务和开采任务的阈值分别定义为

$$\theta_{\text{explore}} = 1 - r_{\text{explore}}, \quad (11)$$

$$\theta_{\text{exploit}} = 1 - r_{\text{exploit}}. \quad (12)$$

2.4 执行概率

根据刺激-响应分工机制,蜜蜂执行探索任务和开采任务的概率分别为

$$\Gamma_{\text{explore}} = \frac{S_{\text{explore}}^2}{S_{\text{explore}}^2 + \theta_{\text{explore}}^2}, \quad (13)$$

$$\Gamma_{\text{exploit}} = \frac{S_{\text{exploit}}^2}{S_{\text{exploit}}^2 + \theta_{\text{exploit}}^2}. \quad (14)$$

2.5 算法流程

基于上述描述,本文提出的SRLDABC算法流程如下:

step 1: 初始化算法参数,包括SN、limit、MR、 p 等.

step 2: 根据式(1)建立初始种群.

step 3: 根据式(7)计算种群多样性,根据式(8)和(9)计算探索任务和开采任务的环境刺激.

step 4: 对于每只蜜蜂,根据式(11)和(12)计算其对应探索任务和开采任务的响应阈值.

step 5: 根据式(13)和(14)计算蜜蜂执行探索任务和开采任务的概率,并依概率大小进行角色转换.

step 6: 若蜜蜂转化为雇佣蜂,则根据式(5)按变异率MR搜索食物源;若蜜蜂转化为侦查蜂,则根据

式(6)按变异率MR搜索食物源.

step 7: 根据式(10)更新蜜蜂执行探索任务或开采任务的成功率.

step 8: 若蜜蜂的食物源连续limit次没有改进,则蜜蜂转化为侦查蜂,根据式(1)重新初始化.

step 9: 记录全局最优值,若不满足算法终止条件,则返回step 3,否则输出最优解和最优值.

ABC算法中的蜜蜂在搜索时会依次经历雇佣蜂阶段和跟随蜂阶段,并在食物源枯竭后进入侦查蜂阶段.与ABC算法不同,SRLDABC算法中的蜜蜂会在雇佣蜂、跟随蜂和侦查蜂之间灵活转换,如图1所示.具体的,当探索的刺激 S_{explore} 超过相应阈值 θ_{explore} 时,蜜蜂以较大的概率转化为雇佣蜂执行探索任务;当开采的刺激 S_{exploit} 超过相应阈值 θ_{exploit} 时,蜜蜂以较大的概率转化为跟随蜂执行开采任务;当食物源枯竭时,蜜蜂转化为侦查蜂重新初始化.

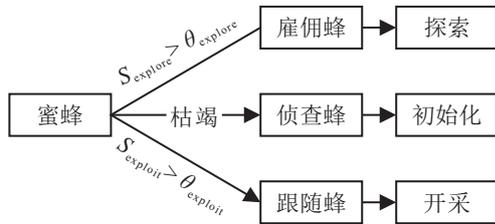


图1 蜜蜂的角色转换

2.6 算法复杂度分析

根据文献[35]的计算方法,比较算法SRLDABC和ABC的复杂度.设 N 为种群规模, D 为解的维度, M 为最大迭代次数.ABC算法初始化的复杂度为 $O(ND)$,雇佣蜂进行一维搜索的复杂度为 $O(MN)$,计算食物源被选中概率的复杂度为 $O(MN)$,跟随蜂进行一维搜索的复杂度为 $O(MN)$,侦查蜂进行 D 维搜索的复杂度 $O(MKD) < O(MND)$, K 为侦查蜂的数量.因此,整个迭代过程中ABC算法的复杂度大约为

$$O(ND + MN + MN + MN + MND) = O(ND + 3MN + MND).$$

与ABC算法相比,SRLDABC算法在迭代过程中引入了种群多样性、搜索成功率、种群最优解和多维度搜索,并省去了食物源概率的计算.计算种群多样性的复杂度为 $O(MND)$,计算搜索成功率的复杂度为 $O(2MN)$,计算种群最优解的复杂度均为 $O(MN)$,进行多维度搜索的复杂度 $O(MNR) < O(MND)$, R 为搜索的维度.所以,整个迭代过程中SRLDABC算法的复杂度大约为

$$O(ND + MND + 2MN + MN + MND + MND) = O(ND + 3MN + 3MND).$$

SRLDABC算法的计算复杂度高于ABC算法,其中复杂度的增加主要体现在种群多样性的计算上.文献[36]认为,通过计算个体之间的距离所增加的计算成本对于计算代价高昂的函数而言可以忽略不计.因此,对于函数评价代价较高的情况,SRLDABC算法和ABC算法的复杂度差异相对较小.此外,当仅考虑复杂度上限时^[19,35],SRLDABC算法和ABC算法的复杂度都为 $O(MND)$.

3 仿真实验

3.1 基准测试函数

为了测试SRLDABC的性能,选取22个基准测试函数进行仿真实验.表1给出了每个函数的搜索范围、最优值和可接受值,其中 D 表示函数的维度.在22个基准测试函数中, $f_1 \sim f_6$ 和 f_8 是连续单峰函数, f_7 是非连续单峰阶跃函数, f_{10} 在 $D \leq 3$ 时是单峰函数,而在 $D > 3$ 时是多峰函数, $f_{11} \sim f_{22}$ 是连续多峰函数.实验测试分别对函数在30维和100维下进行仿真,终止条件设置为函数评价次数达到5000 D .仿真实验环境为Windows 10操作系统,编程语言环境为Matlab R2012a,硬件环境为inter酷睿处理器i5-5350,主频为1.8 GHz,内存为8 GB.

表1 22个基准测试函数

序号	函数名称	搜索范围	最优值	可接受值
f_1	Sphere	$[-100, 100]^D$	0	1×10^{-8}
f_2	Elliptic	$[-100, 100]^D$	0	1×10^{-8}
f_3	SumSquare	$[-10, 10]^D$	0	1×10^{-8}
f_4	SumPower	$[-1, 1]^D$	0	1×10^{-8}
f_5	Schwefel 2.22	$[-10, 10]^D$	0	1×10^{-8}
f_6	Schwefel 2.21	$[-100, 100]^D$	0	1×10^0
f_7	Step	$[-100, 100]^D$	0	1×10^{-8}
f_8	Exponential	$[-10, 10]^D$	0	1×10^{-8}
f_9	Quartic	$[-1.28, 1.28]^D$	0	1×10^{-1}
f_{10}	Rosenbrock	$[-5, 10]^D$	0	1×10^{-1}
f_{11}	Rastrigin	$[-5.12, 5.12]^D$	0	1×10^{-8}
f_{12}	NCRastrigin	$[-5.12, 5.12]^D$	0	1×10^{-8}
f_{13}	Griewank	$[-600, 600]^D$	0	1×10^{-8}
f_{14}	Schwefel 2.26	$[-500, 500]^D$	0	1×10^{-8}
f_{15}	Ackley	$[-50, 50]^D$	0	1×10^{-8}
f_{16}	Penalized1	$[-100, 100]^D$	0	1×10^{-8}
f_{17}	Penalized2	$[-100, 100]^D$	0	1×10^{-8}
f_{18}	Alpine	$[-10, 10]^D$	0	1×10^{-8}
f_{19}	Levy	$[-10, 10]^D$	0	1×10^{-8}
f_{20}	Weierstrass	$[-1, 1]^D$	0	1×10^{-8}
f_{21}	Himmelblau	$[-5, 5]^D$	-78.332 36	-78
f_{22}	Michalewicz	$[0, \pi]^D$	-D	-D + 1

3.2 刺激-响应分工机制有效性分析

本文的重点是基于刺激-响应分工机制改进人工蜂群算法,本节先对该机制的有效性进行分析.从算法执行的角度来看,SRLDABC中的蜜蜂根据刺激-响应分工机制执行搜索任务,其要么转化为雇佣蜂执行探索任务,要么转化为跟随蜂执行开采任务,并在食物源枯竭后转化为侦查蜂重新初始化.刺激-响应分工机制的特点是:当任务刺激超过响应阈值时,蜜蜂执行任务的概率高;反之,执行任务的概率低.因此,刺激-响应分工机制是一种概率执行方式.为了分析其有效性,分别设计顺序执行和随机执行两种方式进行对比.在顺序执行方式中,蜜蜂依次经历雇佣蜂阶段和跟随蜂阶段(即雇佣蜂→跟随蜂→雇佣蜂→跟随蜂...).在随机执行方式中,蜜蜂每次都随机转化为雇佣蜂或跟随蜂进行搜索.在上述两种执行方式中,蜜蜂会在食物源枯竭后转化为侦查蜂重新初始化.

随机选取基准测试函数中3个单峰函数(f_1 、 f_4 、 f_5)和3个多峰函数(f_{10} 、 f_{15} 、 f_{17})进行对比实验,令PE、SE、RE分别表示基于刺激-响应机制的概率执行、顺序执行和随机执行,PE中相关参数设置如下: $SN = 50, limit = D \times SN, MR \in [0.3, 0.7], p = 50$.对于每个测试函数,独立运行3种搜索任务执行方式20次所得最优值的分布情况如图2和图3所示.可以看出,无论是低维还是高维,单峰还是多峰,PE的搜索效率均高于SE和RE.在刺激-响应分工机制中,任务刺激会随环境动态变化,个体阈值会随经验自适应调整.在刺激和阈值的共同作用下,个体能够执行符合族群需求的任务.根据算法搜索的特点,PE设计了基于种群多样性的任务刺激和基于搜索成功率的个体阈值,从而较好地继承了刺激-响应分工机制的柔性特点,对搜索环境表现出良好的适应性.相比之下,SE和RE在执行过程中没有考虑搜索环境的变化,导致适应性较差.

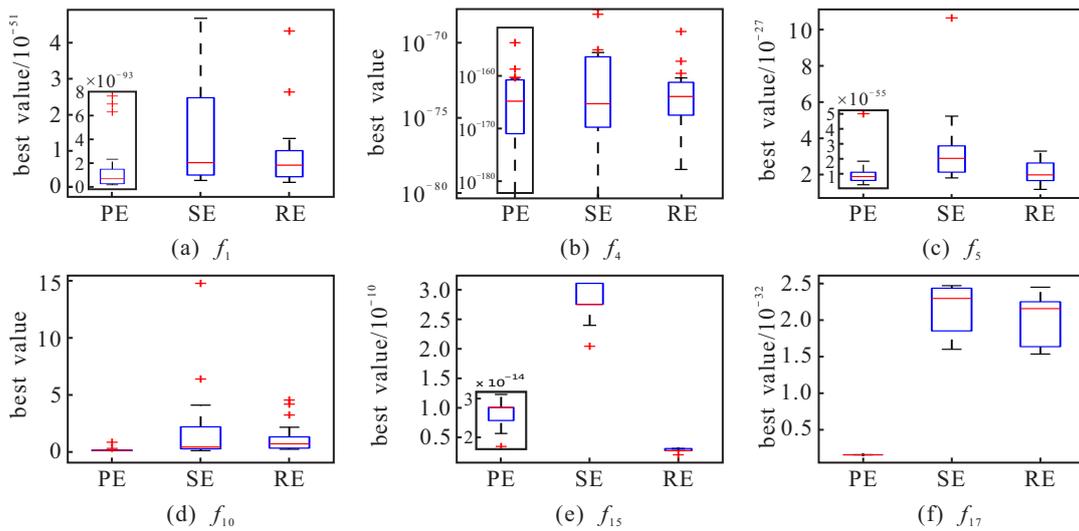


图2 3种执行方式求解不同测试函数的最优值分布情况($D = 30$)

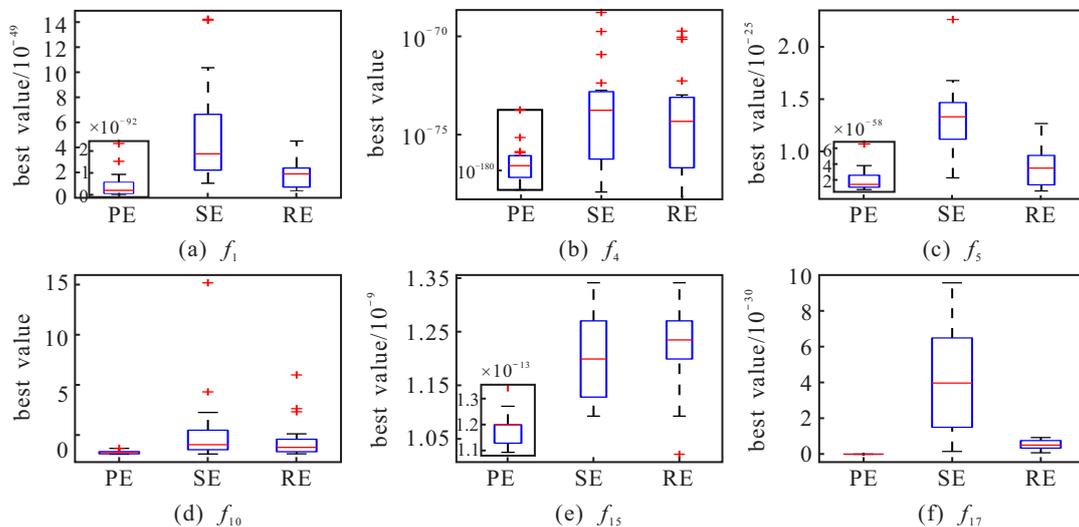


图3 3种执行方式求解不同测试函数的最优值分布情况($D = 100$)

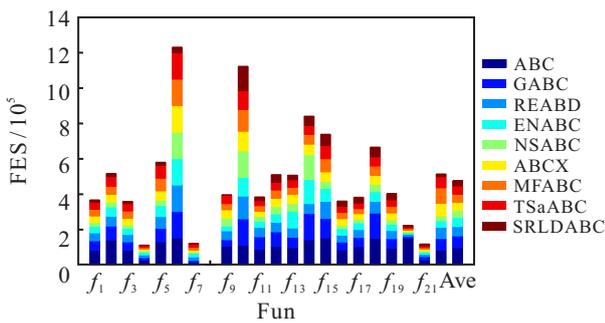
由图2和图3中还可以看出,RE的搜索效果优于SE. 通常情况下,算法在搜索时先探索有前景的区域,然后进行开采. 当搜索到局部最优值以后,继续探索其他有前景的区域,如此反复,从而逐渐逼近全局最优解. 在SE中探索(雇佣蜂)和开采(跟随蜂)依次进行,在RE中探索(雇佣蜂)和开采(跟随蜂)随机执行,前者探索与开采的转换频率比后者快,导致前者在搜索广度和搜索深度方面都不及后者.

3.3 与其他改进ABC算法比较分析

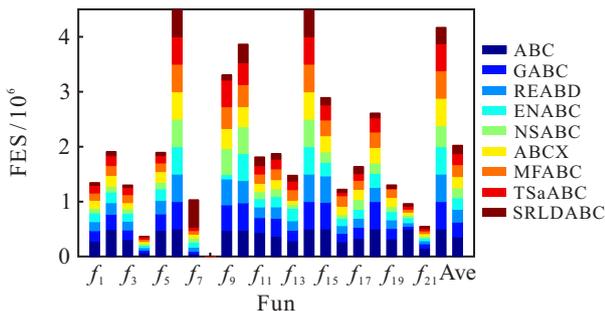
为进一步测试SRLDABC的性能,选择ABC^[1]、GABC^[15]、REABC^[13]、ENABC^[14]、NSABC^[17]、ABCX^[20]、MFABC^[25]和TSaABC^[27]这8种算法作对比,在22个基准测试函数上进行比较分析. 为了保证比较的公平性,所有算法在每个函数上独立运行30次,对比算法的参数参照文献设定,如表2所示. 基于30次独立实验,从平均值和标准差、非参数检验、

表2 各种ABC算法的参数设置

算法	参数设置
ABC	SN = 50, limit = $D \times SN$
GABC	SN = 50, limit = $D \times SN$, C = 1.5
REABC	SN = 50, limit = $D \times SN$, p = 0.1
ENABC	SN = 50, limit = $D \times SN$, p = 0.1
NSABC	SN = 50, limit = $D \times SN$, C = 1.5, k = 10
ABCX	SN = 50, limit = $D \times SN$
MFABC	SN = 50, limit = $D \times SN$, UR = 0.25
TSaABC	SN = 50, limit = $D \times SN$, a = 0.95
SRLDABC	SN = 50, limit = $D \times SN$, MR \in [0.3, 0.7], p = 50



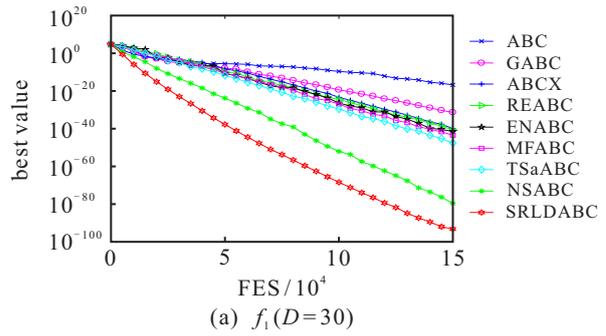
(a) $D=30$



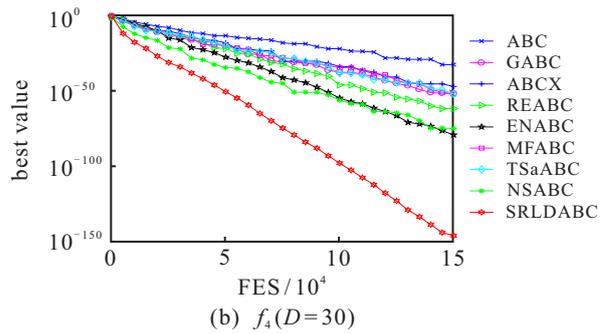
(b) $D=100$

图4 达到可接受值所需平均评价次数

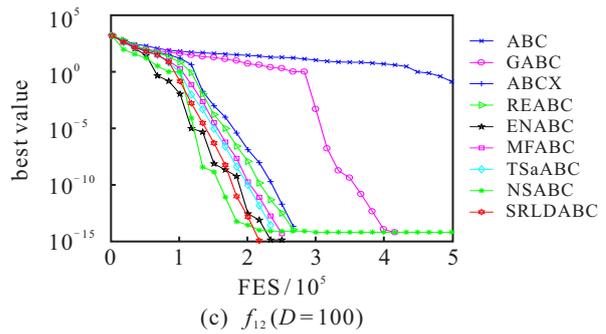
收敛速度3个方面展开对比,结果如图4和图5、表3和表4所示.



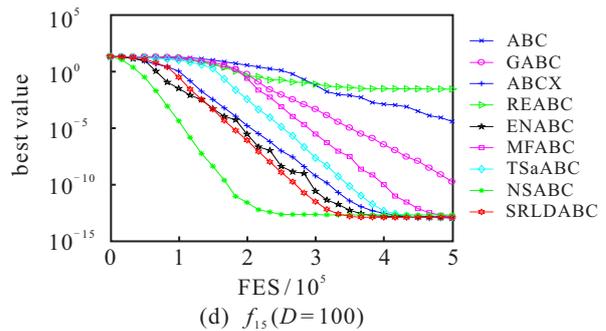
(a) $f_1(D=30)$



(b) $f_4(D=30)$



(c) $f_{12}(D=100)$



(d) $f_{15}(D=100)$

图5 函数收敛曲线

表3给出了各个算法在22个基准测试函数维度为30时的对比情况,包括基于平均值和标准差的解质量对比与基于非参数检验的综合对比. 表3中倒数第3行为解质量对比,其中“+”“=”“-”分别表示本文算法SRLDABC优于、等于、差于对比算法. 由表3可以看出,对于每一个对比算法, SRLDABC至少在11个函数上取得了更好的解,至多在2个函数上取得了较差解,在其余函数上取得了相同的解. 表3中最后两行为基于Friedman检验和Wilcoxon检验的

表3 SRLDABC与各个算法的比较 ($D = 30$)

函数		ABC	GABC	REABC	ENABC	NSABC	ABCX	MFABC	TSaABC	SRLDABC
f_1	Mean	4.33e-17	2.04e-32	7.45e-41	2.94e-42	5.40e-80	5.88e-40	5.29e-44	6.85e-48	3.88e-93
	Std	4.49e-17	1.28e-32	7.63e-41	6.26e-42	1.49e-79	7.89e-40	1.51e-43	7.63e-48	9.71e-93
f_2	Mean	1.55e-09	5.78e-26	8.18e-38	5.18e-39	4.45e-75	7.80e-32	8.11e-41	9.17e-44	2.56e-89
	Std	2.01e-09	4.69e-26	6.06e-38	9.68e-39	1.71e-24	8.08e-32	1.42e-40	1.02e-43	1.11e-88
f_3	Mean	9.12e-19	8.29e-34	6.08e-42	4.94e-42	6.38e-80	4.32e-38	1.55e-45	1.91e-48	5.72e-94
	Std	1.01e-18	5.60e-34	5.42e-42	1.45e-41	1.77e-79	1.98e-38	8.59e-46	3.76e-48	1.33e-93
f_4	Mean	7.51e-32	5.73e-53	3.80e-63	3.05e-80	1.39e-75	3.96e-48	5.68e-53	7.18e-53	2.60e-148
	Std	1.23e-31	2.13e-52	2.08e-62	9.57e-80	5.80e-75	7.70e-48	2.99e-52	2.19e-52	1.40e-147
f_5	Mean	1.67e-10	9.73e-18	1.27e-21	6.45e-23	2.57e-41	8.80e-22	2.98e-23	5.78e-25	1.40e-55
	Std	5.35e-11	2.75e-18	4.28e-22	1.20e-22	3.29e-41	5.38e-22	1.39e-23	2.34e-25	8.82e-56
f_6	Mean	8.91e+00	2.89e+00	7.57e+00	7.34e+00	3.14e+00	3.75e+00	2.92e+00	2.54e+00	5.91e-04
	Std	3.52e+00	6.77e-01	8.77e-01	1.05e+00	7.28e-01	3.08e-01	1.07e+00	4.81e-01	2.76e-03
f_7	Mean	0.00e+00								
	Std	0.00e+00								
f_8	Mean	7.18e-66								
	Std	3.22e-81								
f_9	Mean	5.83e-02	2.80e-02	3.69e-02	9.23e-03	7.57e-02	2.32e-02	2.22e-02	2.00e-02	3.07e-03
	Std	1.48e-02	5.33e-03	9.37e-03	3.32e-03	1.69e-02	6.95e-03	6.41e-03	7.97e-03	1.17e-03
f_{10}	Mean	6.31e-02	3.06e+00	1.45e-01	7.93e-01	4.98e+00	7.94e-02	1.54e-01	2.25e-01	1.52e-01
	Std	6.71e-02	7.20e+00	1.14e-01	9.96e-01	1.78e+01	9.96e-02	1.39e-01	3.83e-01	1.47e-01
f_{11}	Mean	3.46e-14	0.00e+00							
	Std	8.66e-14	0.00e+00							
f_{12}	Mean	4.88e-13	0.00e+00							
	Std	1.10e-12	0.00e+00							
f_{13}	Mean	2.03e-12	7.78e-09	0.00e+00	6.11e-04	3.09e-12	0.00e+00	0.00e+00	2.44e-10	0.00e+00
	Std	9.20e-12	4.26e-08	0.00e+00	2.37e-03	1.54e-11	0.00e+00	0.00e+00	1.06e-09	0.00e+00
f_{14}	Mean	1.58e+01	3.95e+00	1.76e-12	1.18e+01	2.76e+01	8.18e-05	8.18e-05	8.18e-05	1.21e-12
	Std	5.14e+01	2.16e+01	3.32e-13	3.61e+01	6.73e+01	6.88e-12	9.96e-13	8.68e-13	8.72e-13
f_{15}	Mean	1.63e-05	1.22e-11	1.41e-02	2.71e-14	2.82e-14	3.41e-14	3.91e-14	2.90e-14	2.75e-14
	Std	2.46e-05	3.67e-11	7.74e-02	2.91e-15	3.42e-15	4.77e-15	2.68e-13	4.03e-15	2.95e-15
f_{16}	Mean	5.32e-18	1.57e-32							
	Std	5.06e-18	5.57e-48							
f_{17}	Mean	8.87e-16	1.24e-31	1.50e-33	1.50e-33	1.50e-33	1.50e-33	1.50e-32	1.50e-32	1.50e-33
	Std	1.19e-15	9.32e-32	0.00e+00						
f_{18}	Mean	5.90e-06	9.03e-07	4.69e-22	5.06e-17	1.08e-15	1.30e-16	1.41e-23	1.34e-23	7.40e-18
	Std	5.02e-06	2.19e-06	4.19e-22	1.30e-16	2.55e-15	2.36e-16	4.51e-23	4.92e-23	4.05e-17
f_{19}	Mean	1.57e-13	1.65e-31	1.35e-31						
	Std	1.97e-13	6.43e-32	6.68e-47						
f_{20}	Mean	3.39e-03	0.00e+00							
	Std	2.90e-03	0.00e+00							
f_{21}	Mean	-7.83e+01								
	Std	6.98e-15	9.14e-15	9.51e-15	0.00e+00	0.00e+00	1.01e-14	0.00e+00	0.00e+00	0.00e+00
f_{22}	Mean	-2.94e+01	-2.95e+01	-6.85e+00	-2.96e+01	-2.96e+01	-2.84e+01	-2.84e+01	-2.86e+01	-2.96e+01
	Std	5.93e-02	2.63e-02	7.21e-01	4.96e-02	1.90e-02	3.71e-01	1.85e-01	2.12e-01	1.90e-02
+ / = / -		19/2/1	16/6/0	11/9/2	12/9/1	12/10/0	12/9/1	11/10/1	12/9/1	—
Friedman mean ranking		7.54	6.52	4.97	4.68	4.26	5.26	4.55	4.36	2.86
Wilcoxon p -value		4.95e-05	4.30e-05	1.28e-04	1.12e-04	3.13e-03	6.52e-05	1.66e-04	6.52e-04	—

表4 SRLDABC与各个算法的比较($D = 100$)

函数		ABC	GABC	REABC	ENABC	NSABC	ABCX	MFABC	TSaABC	SRLDABC
f_1	Mean	1.65E-15	6.52E-30	8.56e-39	7.26e-38	2.72e-77	1.70e-38	3.25e-43	1.43e-46	3.03e-93
	Std	1.31e-15	2.98e-30	4.06e-39	3.16e-37	8.08e-77	1.94e-38	2.18e-43	1.48e-46	2.52e-93
f_2	Mean	1.03e-08	2.13e-23	4.87e-35	7.02e-35	1.11e-72	1.48e-30	9.24e-39	3.05e-42	1.56e-88
	Std	8.92e-09	1.26e-23	3.23e-35	2.07e-34	4.46e-72	2.33e-30	1.14e-38	2.87e-42	3.62e-88
f_3	Mean	5.46e-16	2.38e-30	3.71e-39	8.71e-39	2.25e-77	7.95e-36	4.08e-43	5.32e-47	4.12e-93
	Std	5.50e-16	1.46e-30	2.22e-39	2.77e-38	5.39e-77	3.97e-36	3.08e-43	5.33e-47	7.64e-93
f_4	Mean	8.85e-31	4.68e-54	1.30e-63	7.76e-78	1.27e-40	1.05e-46	7.58e-52	1.26e-50	6.20e-170
	Std	2.58e-30	1.17e-53	6.88e-63	2.46e-77	3.80e-40	2.04e-46	2.73e-51	5.14e-50	0.00e+00
f_5	Mean	2.87e-09	3.42e-16	3.11e-20	4.53e-21	4.32e-40	9.85e-21	2.51e-22	4.21e-24	1.78e-58
	Std	6.87e-10	6.36e-17	7.83e-21	1.14e-20	5.00e-40	1.39e-20	1.26e-22	1.95e-24	1.07e-58
f_6	Mean	5.24e+01	3.88e+01	4.55e+01	4.79e+01	1.38e+01	3.37e+01	3.21e+01	2.94e+00	3.49e+01
	Std	2.32e+00	2.06e+00	1.42e+00	2.61e+00	1.30e+00	7.04e+00	2.38e+00	1.45e+00	6.44e+00
f_7	Mean	0.00e+00	1.99e+02							
	Std	0.00e+00	2.84e+02							
f_8	Mean	7.10e-218								
	Std	0.00e+00								
f_9	Mean	2.66e-01	1.69e-01	1.95e-01	3.39e-02	9.26e-01	1.38e-01	1.29e-01	1.11e-01	1.67e-02
	Std	6.07e-02	1.59e-02	2.38e-02	8.65e-03	1.17e-01	1.22e-02	1.67e-02	1.05e-02	3.84e-03
f_{10}	Mean	1.43e-01	2.13e+01	2.96e-01	1.20e+00	1.02e+01	3.85e-02	5.69e-01	1.04e+00	5.28e-01
	Std	1.75e-01	3.60e+01	2.60e-01	1.03e+00	2.44e+01	9.94e-03	7.02e-01	1.57e+00	6.00e-01
f_{11}	Mean	8.15e-09	0.00e+00	0.00e+00	0.00e+00	4.44e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Std	3.55e-08	0.00e+00	0.00e+00	0.00e+00	2.59e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_{12}	Mean	1.42e-01	0.00e+00	0.00e+00	0.00e+00	4.62e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Std	3.45e-01	0.00e+00	0.00e+00	0.00e+00	2.94e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_{13}	Mean	4.37e-16	0.00e+00	0.00e+00	8.13e-09	6.53e-14	0.00e+00	0.00e+00	2.78e-17	0.00e+00
	Std	6.60e-16	0.00e+00	0.00e+00	4.45e-08	1.71e-13	0.00e+00	0.00e+00	9.85e-17	0.00e+00
f_{14}	Mean	5.47e+02	1.58e+01	2.01e-01	3.76e+01	7.11e+01	2.72e-04	2.75e-04	2.73e-04	1.32e+01
	Std	2.13e+02	4.09e+01	1.10e+00	5.50e+01	9.12e+01	3.99e-12	7.28e-12	4.93e-04	3.64e+01
f_{15}	Mean	3.73e-05	3.03e-09	2.84e-02	1.23e-13	2.23e-13	1.35e-13	1.46e-13	1.25e-13	1.22e-13
	Std	2.70e-05	5.17e-09	1.02e-01	7.62e-15	1.82e-14	1.31e-14	3.23e-13	1.31e-14	9.55e-15
f_{16}	Mean	5.73e-17	1.07e-31	4.71e-33	4.71e-33	7.72e-33	4.71e-33	4.71e-33	4.71e-33	4.71e-33
	Std	6.99e-17	5.19e-32	1.39e-48	1.39e-48	3.05e-33	1.39e-48	1.39e-48	1.39e-48	1.39e-48
f_{17}	Mean	3.66e-14	3.86e-29	1.50e-33	1.50e-33	1.08e-32	1.50e-33	1.50e-33	1.50e-33	1.50e-33
	Std	2.63e-14	2.28e-29	0.00e+00	0.00e+00	6.64e-33	0.00e+00	0.00e+00	0.00e+00	0.00e+00
f_{18}	Mean	1.71e-03	5.71e-05	2.04e-17	1.29e-15	5.85e-15	5.28e-15	2.11e-22	4.39e-23	3.63e-25
	Std	2.85e-03	5.46e-05	1.11e-16	1.48e-15	5.84e-15	1.07e-14	3.10e-22	8.89e-23	9.45e-25
f_{19}	Mean	1.84e-13	1.47e-30	1.35e-31	1.35e-31	2.35e-31	1.35e-31	1.35e-31	1.35e-31	1.35e-31
	Std	2.40e-13	1.19e-30	6.68e-47	6.68e-47	6.88e-32	6.68e-47	6.68e-47	6.68e-47	6.68e-47
f_{20}	Mean	4.27e-02	0.00e+00							
	Std	1.50e-02	0.00e+00							
f_{21}	Mean	-7.83e+01								
	Std	2.97e-14	4.89e-14	5.24e-14	2.70e-14	3.29e-14	3.41e-14	2.93e-14	5.10e-14	2.40e-14
f_{22}	Mean	-9.76e+01	-9.86e+01	-1.72e+01	-9.95e+01	-9.92e+01	-8.74e+01	-8.85e+01	-8.89e+01	-9.95e+01
	Std	2.16e-01	9.61e-02	1.35e+00	4.16e-02	1.65e-01	6.77e-01	9.42e-01	7.15e-01	1.83e-02
+ / = / -		19/1/2	16/5/1	11/8/3	14/7/1	18/2/2	10/8/4	11/8/3	12/7/3	—
Friedman mean ranking		7.86	6.29	5.12	4.86	4.63	4.93	4.33	4.11	2.87
Wilcoxon p -value		4.95e-05	4.32e-05	4.31e-05	7.34e-04	1.46e-03	4.02e-04	4.31e-05	1.83e-03	—

综合对比. Friedman 检验时所有算法的排序结果为 SRLDABC < NSABC < TSaABC < MFABC < ENABC < REABC < ABCX < GABC < ABC, 平均排序值越小, 说明算法的整体优化性能越好. Wilcoxon 检验时显著性水平设置为 0.05, 8 个对比算法的 p 值均小于显著水平, 说明 SRLDABC 与这些算法之间有显著性差异, 即 SRLDABC 优于这些算法.

表 4 给出了各个算法在 22 个基准测试函数维度为 100 时的对比情况, 包括基于平均值和标准差的解质量对比与基于非参数检验的综合对比. 由表 4 可以看出, 函数维度由 30 增到 100 以后, 对 SRLDABC 与其他算法的性能对比影响不大. 在解质量对比方面, 对于每一个对比算法, SRLDABC 至少在 10 个函数上取得了更好的解, 至多在 4 个函数上取得了较差解, 在其余函数上取得了相同解. 在综合对比方面, Friedman 检验时所有算法的排序结果为 SRLDABC < TSaABC < MFABC < NSABC < ENABC < ABCX < REABC < GABC < ABC, Wilcoxon 检验时 8 个对比算法的 p 值均小于显著水平 0.05. 表 3 和表 4 的对比结果表明, 对于不同维度下的基准测试函数, SRLDABC 在优化时均表现出较好的收敛精度和稳定性(即所获得的解的质量较高), 以及良好的综合性能.

表 1 第 5 列给出了各基准测试函数的可接受值, 采用达到可接受值所需的平均评价次数评估算法的收敛速度. 评价次数指适应度函数的评价次数. 一般而言, 所需评价次数越小, 算法的收敛速度越快. 如果算法在运行过程中无法找到目标函数值小于可接受值的解, 则采用最大评价次数. 图 4 中 (a) 和 (b) 分别描述了基准测试函数维度为 30 和 100 时, 各个算法在 22 个函数上找到可接受值所需的平均评价次数. 从图 4 的 (a) 和 (b) 可以看出, SRLDABC 在 22 个基准测试函数上所需的平均评价次数整体较少. 进一步, 图 4 中最后一列 Ave 表示算法在所有函数上所需平均评价次数. 按照图例中的算法顺序, 图 4(a) 中 9 个算法在所有函数上所需平均评价次数分别为 94 753、67 044、53 308、50 866、40 764、44 907、47 357、46 056、28 985 (即 SRLDABC < NSABC < ABCX < TSaABC < MFABC < ENABC < REABC < GABC < ABC), 图 4(b) 中 9 个算法在所有函数上所需平均评价次数分别为 354 039、270 710、233 899、208 837、179 152、209 518、215 051、199 046、154 614 (即 SRLDABC < NSABC < TSaABC < ENABC < ABCX < MFABC < REABC < GABC <

ABC). 本文算法所需的评价次数整体偏少, 说明其收敛速度较快.

为了更直观地显示各算法的收敛速度, 图 5 给出了各算法在部分函数 $D = 30$ 和 $D = 100$ 下的收敛曲线. 由表 3 和表 4 可知, 本文算法在大部分函数上表现出较好或相似的优化能力. 限于篇幅, 部分函数选两个单峰函数 f_1 和 f_4 , 以及两个多峰函数 f_{12} 和 f_{15} . 对于其他大部分函数, 也能得到类似的结果. 由图 5 可见, SRLDABC 具有较快的收敛速度. 对于函数 f_1 和 f_4 , 相比于其他算法, SRLDABC 收敛到一定精度所需的评价次数最少, 且收敛精度最高. 对于函数 f_{12} , NSABC 在前期收敛速度最快, 在后期陷入局部最优. 在剩余的 8 个算法中, SRLDABC 的收敛速度最快, 且能找到函数的理论最优值(在评价次数达到 22 万次左右时搜索到 0, 由于图中 y 轴采用对数坐标轴, 所以没有显示 0). 对于函数 f_{15} , NSABC 的收敛速度最快, SRLDABC、ENABC、ABCX、TSaABC、MFABC 次之, 剩余 3 个算法较差.

4 结论

1) 在人工蜂群算法中, 探索指搜索解空间未知区域发现优良解的过程, 开采指利用已知解信息搜索优良解的过程, 平衡二者之间的关系一直是算法的研究热点之一. 在群智能劳动分工中, 个体能随着环境变化灵活地调整任务, 是族群生态成功的重要保障. 本文将探索与开采看成两种不同的搜索任务, 采用劳动分工机制协调二者之间的关系, 为人工蜂群算法及其他群智能优化算法的改进提供了新思路.

2) 本文将刺激-响应劳动分工机制融入人工蜂群算法, 根据种群多样性定义搜索任务的环境刺激, 根据搜索成功率设计蜜蜂个体的响应阈值. 通过刺激-响应劳动分工机制, 蜜蜂在探索任务和开采任务之间进行灵活转换, 较好地实现了探索与开采之间的平衡. 与顺序执行方式和随机执行方式展开对比, 表明了基于刺激-响应劳动分工机制的概率执行方式的优越性. 通过对 22 个测试函数进行仿真分析, 从解质量、非参数检验和收敛速度 3 个方面验证了所提出算法的有效性.

3) 本文算法在基准测试函数上取得了较好的效果, 但并未对实际问题进行测试, 下一步将利用所提出算法优化实际问题. 另外, 群智能劳动分工的两种代表性机制是刺激-响应机制和激发-抑制机制, 研究激发-抑制机制与人工蜂群算法的融合也是今后的一个研究方向.

参考文献(References)

- [1] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Erciyes University, 2005.
- [2] 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.
- [3] Karaboga D, Gorkemli B, Ozturk C, et al. A comprehensive survey: Artificial bee colony(ABC) algorithm and applications[J]. *Artificial Intelligence Review*, 2014, 42(1): 21-57.
- [4] 曹知奥, 汪晋宽, 韩英华, 等. 基于交叉-变异人工蜂群算法的微网优化调度[J]. *控制与决策*, 2020, 35(9): 2059-2069.
(Cao Z A, Wang J K, Han Y H, et al. Crossover-mutation based artificial bee colony algorithm for optimal scheduling of microgrid[J]. *Control and Decision*, 2020, 35(9): 2059-2069.)
- [5] 吴锐, 郭顺生, 李益兵, 等. 改进人工蜂群算法求解分布式柔性作业车间调度问题[J]. *控制与决策*, 2019, 34(12): 2527-2536.
(Wu R, Guo S S, Li Y B, et al. Improved artificial bee colony algorithm for distributed and flexible job-shop scheduling problem[J]. *Control and Decision*, 2019, 34(12): 2527-2536.)
- [6] Gao H, Shi Y J, Pun C M, et al. An improved artificial bee colony algorithm with its application[J]. *IEEE Transactions on Industrial Informatics*, 2019, 15(4): 1853-1865.
- [7] Xing H L, Song F H, Yan L S, et al. A modified artificial bee colony algorithm for load balancing in network-coding-based multicast[J]. *Soft Computing*, 2019, 23(15): 6287-6305.
- [8] Wang L, Zhang X, Zhang X. Antenna array design by artificial bee colony algorithm with similarity induced search method[J]. *IEEE Transactions on Magnetics*, 2019, 55(6): 1-4.
- [9] Crepinsek M, Liu S, Mernik M. Exploration and exploitation in evolutionary algorithms: A survey[J]. *ACM Computing Surveys*, 2013, 45(3): 1-33.
- [10] Lynn N, Suganthan P N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation[J]. *Swarm and Evolutionary Computation*, 2015, 24: 11-24.
- [11] Singh A, Deep K. Exploration-exploitation balance in artificial bee colony algorithm: A critical analysis[J]. *Soft Computing*, 2019, 23(19): 9525-9536.
- [12] Xu J Q, Zhang J H. Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis[C]. *Proceedings of the 33rd Chinese Control Conference*. Nanjing, 2014: 8633-8638.
- [13] 孔德鹏, 常天庆, 戴文君, 等. 基于排序选择和精英引导的改进人工蜂群算法[J]. *控制与决策*, 2019, 34(4): 781-786.
(Kong D P, Chang T Q, Dai W J, et al. An improved artificial bee colony algorithm based on the ranking selection and the elite guidance[J]. *Control and Decision*, 2019, 34(4): 781-786.)
- [14] 孟红云, 位冰可. 基于精英解和随机个体邻域信息的改进人工蜂群算法[J]. *控制与决策*, 2020, 35(9): 2169-2174.
(Meng H Y, Wei B K. An improved artificial bee colony algorithm based on elite solution and random individual neighborhood information[J]. *Control and Decision*, 2020, 35(9): 2169-2174.)
- [15] Zhu G P, Kwong S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. *Applied Mathematics and Computation*, 2010, 217(7): 3166-3173.
- [16] Cui L Z, Zhang K, Li G H, et al. Modified gbest-guided artificial bee colony algorithm with new probability model[J]. *Soft Computing*, 2018, 22(7): 2217-2243.
- [17] Wang H, Wang W J, Xiao S Y, et al. Improving artificial bee colony algorithm using a new neighborhood selection mechanism[J]. *Information Sciences*, 2020, 527: 227-240.
- [18] Kang F, Li J J, Ma Z Y. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions[J]. *Information Sciences*, 2011, 181(16): 3508-3531.
- [19] Gao W F, Liu S Y, Huang L L. Enhancing artificial bee colony algorithm using more information-based search equations[J]. *Information Sciences*, 2014, 270: 112-133.
- [20] Hakli H, Kiran M S. An improved artificial bee colony algorithm for balancing local and global search behaviors in continuous optimization[J]. *International Journal of Machine Learning and Cybernetics*, 2020, 11(9): 2051-2076.
- [21] Cui L Z, Li G H, Luo Y L, et al. An enhanced artificial bee colony algorithm with dual-population framework[J]. *Swarm and Evolutionary Computation*, 2018, 43: 184-206.
- [22] Cui L Z, Li G H, Zhu Z X, et al. A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization[J]. *Information Sciences*, 2017, 414: 53-67.
- [23] 王志刚, 王明刚. 基于符号函数的多搜索策略人工蜂群算法[J]. *控制与决策*, 2016, 31(11): 2037-2044.
(Wang Z G, Wang M G. Multi-search strategy of artificial

- bee colony algorithm based on symbolic function[J]. Control and Decision, 2016, 31(11): 2037-2044.)
- [24] 王志刚, 尚旭东, 夏慧明, 等. 多搜索策略协同进化的人工蜂群算法[J]. 控制与决策, 2018, 33(2): 235-241. (Wang Z G, Shang X D, Xia H M, et al. Artificial bee colony algorithm with multi-search strategy cooperative evolutionary[J]. Control and Decision, 2018, 33(2): 235-241.)
- [25] Song X Y, Zhao M, Xing S Y. A multi-strategy fusion artificial bee colony algorithm with small population[J]. Expert Systems with Applications, 2020, 142: 112921.
- [26] Kiran M S, Hakli H, Gunduz M, et al. Artificial bee colony algorithm with variable search strategy for continuous optimization[J]. Information Sciences, 2015, 300(1): 140-157.
- [27] Song X Y, Zhao M, Yan Q F, et al. A high-efficiency adaptive artificial bee colony algorithm using two strategies for continuous optimization[J]. Swarm and Evolutionary Computation, 2019, 50: 100549.
- [28] 肖人彬, 王英聪. 群智能自组织劳动分工研究进展[J]. 信息与控制, 2019, 48(2): 129-139. (Xiao R B, Wang Y C. Research progress of self-organized labor division in swarm intelligence[J]. Information and Control, 2019, 48(2): 129-139.)
- [29] Bonabeau E, Theraulaz G, Deneubourg J L. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies[J]. Proceedings of the Royal Society of London—Series B: Biological Sciences, 1996, 263(1376): 1565-1569.
- [30] Akay B, Karaboga D. A modified artificial bee colony algorithm for real-parameter optimization[J]. Information Sciences, 2012, 192(6): 120-142.
- [31] Kiran M S, Findik O. A directed artificial bee colony algorithm[J]. Applied Soft Computing, 2015, 26: 454-462.
- [32] Lin H S, Kang L S. Balance between exploration and exploitation in genetic search[J]. Wuhan University Journal of Natural Sciences, 1999, 4(1): 28-32.
- [33] Wang H, Sun H, Li C H, et al. Diversity enhanced particle swarm optimization with neighborhood search[J]. Information Sciences, 2013, 223: 119-135.
- [34] Ursem R K. Diversity-guided evolutionary algorithms[C]. Proceedings of Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Granada, 2002: 462-471.
- [35] Imanian N, Shiri M E, Moradi P. Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems[J]. Engineering Applications of Artificial Intelligence, 2014, 36: 148-163.
- [36] Cai Y Q, Wang J H. Differential evolution with neighborhood and direction information for numerical optimization[J]. IEEE Transactions on Cybernetics, 2013, 43(6): 2202-2215.

作者简介

王英聪(1987—), 男, 讲师, 博士, 从事群智能、布局优化等研究, E-mail: ying_cong_wang@163.com;

刘军辉(1991—), 男, 硕士生, 从事人工蜂群算法的研究, E-mail: tianxia768009668@163.com;

肖人彬(1965—), 男, 教授, 博士生导师, 从事群智能、涌现计算等研究, E-mail: rbxiao@hust.edu.cn.

(责任编辑: 孙艺红)