

一种基于标杆管理的优化算法*

谢安世^{1,2}, 于永达¹, 黄思明²

¹(清华大学 公共管理学院, 北京 100084)

²(中国科学院 科技政策与管理科学研究所, 北京 100190)

通讯作者: 谢安世, E-mail: shermanxas@163.com

摘要: 借鉴标杆管理理念, 提出了一种基于标杆管理的优化算法(benchmarking-based optimization algorithm, 简称 BOA). 根据标杆管理的核心价值观, 设计了一套基于动态小生境的竞争性学习机制, 针对常用的编码方案, 设计出了具体可行的执行方法. 种群内个体执行方向明确的主动学习式搜索, 通过对标杆的模仿学习, 能够快速搜索到解空间内的目标区域内, 具有较好的智能性. 此外, 整个小生境种群系统通过自组织学习实现与环境的友好交互, 较好地解决了保持种群的多样性的难题. 分析了 BOA 算法与遗传算法等现代智能优化方法在搜索模式上的重要区别, 并通过对比仿真实验, 表明算法能够与环境进行稳定而友好的交互, 表现出较强的鲁棒性, 其搜索速度和寻优能力在实验中均有较好的表现.

关键词: 标杆管理; 优化算法; 智能计算; 搜索模式

中图法分类号: TP181

中文引用格式: 谢安世, 于永达, 黄思明. 一种基于标杆管理的优化算法. 软件学报, 2014, 25(5): 953-969. <http://www.jos.org.cn/1000-9825/4428.htm>

英文引用格式: Xie AS, Yu YD, Huang SM. Optimization algorithm based on benchmarking. Ruan Jian Xue Bao/Journal of Software, 2014, 25(5): 953-969 (in Chinese). <http://www.jos.org.cn/1000-9825/4428.htm>

Optimization Algorithm Based on Benchmarking

XIE An-Shi^{1,2}, YU Yong-Da¹, HUANG Si-Ming²

¹(School of Public Policy and Management, Tsinghua University, Beijing 100084, China)

²(Institute of Policy and Management, The Chinese Academy of Sciences, Beijing 100190, China)

Corresponding author: XIE An-Shi, E-mail: shermanxas@163.com

Abstract: Drawing on the benchmarking theory in the business management, a new search method, benchmarking-based optimization algorithm (BOA), is proposed in this paper. BOA provides a competitive learning mechanism based on dynamic niche according to the core values of benchmarking. Through imitation and learning, all the individuals within a population are able to approach to the high yielding regions in the solution space and seek out the optimal solutions quickly. Further, the formidable problem of maintaining the diversity of population is effectively resolved through the self-organizing learning process of the niche system and its friendly interaction with the environment. In this paper, the main differences between BOA and the existing intelligent optimization methods, such as genetic algorithm (GA), are analyzed. The comparative experiments show that BOA is robust and able to perform friendly interactive learning with the environment, and its search speed and optimization ability is far superior to the existing intelligent optimization methods.

Key words: benchmarking; optimization algorithm; intelligence computation; search mode

优化问题的求解技术一般以数学为基础来获取各种工程问题的最优解或者满意解. 在实践中, 绝大多数问题都可以转换为最优化问题来求解. 如 0-1 背包、组合优化、任务指派等许多存在于数学及工程领域中的 NP 难题, 一般很难找到精确的数学解法, 有的也不需要精确解, 只要寻找到近似最优解即可. 针对优化问题, 寻求除

* 收稿时间: 2012-05-29; 定稿时间: 2013-05-07

基于严格数学逻辑的运筹学方法之外的方法,国内外相关学者做出了很大努力,且成果显著。

在国外,1975年,美国的Holland教授提出了遗传算法。1982年,Kirkpatrick将模拟退火思想引入到了优化研究领域,提出了模拟退火算法(simulating annealing,简称SA)。1983年,Creutz提出了微正则退火算法(micro-canonical annealing,简称MA)^[1]。它和SA都是模仿退火机制,但在退火过程中的状态转移不是用Metropolis准则,而是用一种确定性方式。目前,关于MA的研究不太多,已有的研究主要应用在图像处理领域。1989年,Moscato首次提出了模拟文化进化过程的文化基因算法(memetic algorithm,简称MA)^[2]。1991年,意大利学者Dorigo根据蚂蚁觅食的群体行为,提出了蚁群算法的基本模型。1995年,美国心理学家Kennedy及其合作者Eberhart博士根据鸟类捕食行为模型提出了粒子群算法。1997年,Storn等人在遗传算法等进化算法思想上提出了差分进化算法。1999年,巴西的Castro最早在其论文中总结了人工免疫系统(artificial immune system,简称AIS)^[3]。后来,他基于克隆选择基本原理提出了著名的克隆选择算法(clonal selection algorithm,简称CSA)^[4]。进入新世纪以来,各类新式算法层出不穷,仅2000年就出现了3种新算法,如Eusuff等人提出了一种后启发式的混合蛙跳算法(shuffled frog leaping algorithm,简称SFLA)^[5],此后有若干改进。同年,日本的Murase提出了一种模拟植物光合作用原理的光合算法(photosynthetic algorithm,简称PA)^[6],在N皇后问题、有限元分析等问题上取得了较好的效果。在同一年,Zelinka和Lampinen基于物种合作竞争策略提出了自组织迁移算法(self-organizing migrating algorithm,简称SOMA)^[7]。2001年,Geem等人根据音乐演奏中的和声原理提出了一种和声搜索算法(harmony search,简称HS)^[8]。2002年,Passino提出了一种模拟人类大肠杆菌觅食行为的细菌觅食优化算法(bacteria foraging optimization algorithm,简称BFOA)^[9]。2004年,Nakrani首次根据蜜蜂的觅食原理提出了蜂群算法(bee algorithm,简称BA)^[10]。后来,Karabog根据蜂群的群居行为模型提出了系统的人工蜂群算法(artificial bee colony algorithm,简称ABC)^[11]。剑桥大学的Yang在2005年提出了一种模拟生化酶抑制和催化作用的酶算法(enzyme algorithm,简称EA)^[12]。同年,印度学者Krishnanand和Ghose根据萤火虫的求偶行为提出了萤火虫群优化算法(glowworm swarm optimization,简称GSO)^[13]。2007年,Mucherino等人根据猴子的攀树觅食模式提出了猴子搜索算法(monkey search,简称MS)^[14]。2008年,学者们又提出了3种新算法,如Havens等人提出了一种模拟蟑螂群居行为的蟑螂算法^[15];同年,剑桥大学的Yang又提出了萤火虫算法(firefly algorithm,简称FA)^[16];Simon根据生物物种迁移数学模型提出了一种生物地理学优化算法(biogeography-based optimization,简称BBO)^[17]。2010年1月,日本的Tero等人在《Science》上发表了使用粘菌(俗称鼻涕虫,英文名为Slime mold,一种粘菌门组织)设计连通东京及其附近城市的铁路网的研究成果,这项研究展现了仿生计算的若干特征^[18]。

在国内,2002年,清华大学的谢晓峰基于人类智能中的社会认知理论(social cognitive theory)提出了一种社会认知优化算法(social cognitive optimization,简称SCO)^[19]。2003年,浙江大学的李晓磊提出了模拟鱼群行为模式的人工鱼群算法(artificial fishswarm algorithm,简称AFSA)^[20]。同年,周永华等人提出了一种模拟人口迁移机理的人口迁移算法(population migration algorithm,简称PMA)^[21]。2005年,李彤等人提出了基于植物向光性机理的模拟植物生长算法^[22]。2008年,最早有国内匿名学者利用费马原理模拟自然界光线折射现象,提出了光线寻优算法(light ray optimization,简称LRO),后来有一部分学者进行了应用和发展^[23]。2009年,马海平等人提出了一种基于物种迁移优化的进化算法^[24]。2011年,国内相关学者提出了4种新颖的优化搜索算法,如:谢丽萍等人提出了一种基于拟态物理学方法的全局优化算法^[25];曹炬等人受烟花爆炸现象的启发,提出了一种新的并行弥漫式搜索的优化算法——爆炸优化算法^[26];谭世恒等人根据细胞膜的特性及其物质转动方式,提出了细胞膜优化算法(cell membrane optimization,简称CMO)^[27];台湾地区的潘文超根据果蝇的觅食原理,提出了果蝇优化算法(fruit fly optimization algorithm,简称FOA)^[28,29]。

这些方法一般受自然界规律和生物群体智能行为的启发,具有较新颖的设计思路,即将问题域中随机采样的样本点看作粒子,粒子具有适应值、速度和位置等属性。这些粒子在某种智能搜索策略的引导下,通过速度和位置的更新迭代逐步求得给定问题的最优解。虽然实践中这些方法都取得了令人满意的应用效果,但与基于严格数学逻辑的运筹学方法相比,目前尚无真正严格而完备的数理逻辑能够证明这些方法的全局收敛性。因此,可以将这些方法统一称为基于概率的智能算法。

与模拟自然界生物活动规律的做法不同,本文借鉴企业管理领域中的标杆管理方法,提出一种基于标杆管理的寻优算法(benchmarking-based optimization algorithm,简称 BOA).下面将介绍标杆管理的基本理念和基于这种理念的寻优方法,分析其不同于现有智能优化算法的特性,并通过实验验证这种新模式的有效性.

1 标杆管理优化算法的基本内容

标杆管理(benchmarking)一词来源于企业管理界,首次出现于施乐公司(Xerox corporation)的前身——位于美国康涅狄格州(Connecticut)罗切斯特市的哈洛伊德(Haloid)公司.标杆原意为“固定对象的标记,诸如用石柱来说明高出海平面之高度,作为调查中的参考点”,有基准之意,其构想是寻找学习的对象,以他们的既有成就为基准,透过合法管道学习,以“见贤思齐”的方式,达到改善自己经营品质的目的.换句话说,它是指一家公司可以就某一特定过程,将本身的绩效与其他公司的绩效相比较,然后学习其中绩效最佳者的做法,以达到提升自己绩效的目的,已经成为企业界通行的管理理念和管理工具.

然而,标杆管理并不只是简单地向他人学习,而是包括 4 项基本原则,即标杆管理的 4 项核心价值观:全面品质观、流程观、衡量标准观和学习观.全面品质观是指达成顾客的全面性满意;流程观是指标杆管理涵盖学习对象的运营流程及组织内部的计划和运作流程;衡量标准观是指标杆管理须订出某些组织功能上共同绩效衡量标准,作为比较的依据;学习观则同时强调向他人学习与自我学习的精神.在管理学中,标杆管理既是一种态度也是一种行动,表现为一个持续的学习过程,不断地向标杆迈进,不断地创新与改善,不断地提升发展优势,不断地提高组织效能.

通过 Web of knowledge, Google Scholar 等检索,目前尚未发现国内外有学者将标杆管理应用于优化算法领域.标杆管理,简而言之,即找出与最佳个案的差距,并通过模仿学习,快速缩小这个差距乃至超越对手.本文提出的基于标杆管理的寻优算法(BOA),其总体框架是:整个生态系统(解空间)内分布着若干小生境种群,相当于全球市场上各大企业法人主体,种群内的个体相当于企业内部各部门员工;根据优化目的,以目标值大小为衡量标准,找出各小生境种群内的最佳个体(即局部最佳个体)和整个生态系统内的最佳个体(即全局最佳个体),相当于树立内部标杆和外部标杆;小生境种群中的个体有选择地进行标杆学习,此外还会进行自我学习;通过对标杆进行模仿学习,迅速超越学习对象,进而成为其他个体学习的对象.因而,BOA 是一个学习性竞争和竞争性学习的寻优模型.

1.1 自组织学习设计

在搜索学习的过程中,各小生境种群的自组织学习是这样进行的:种群内个体首先进行外部标杆学习,即向整个生态系统内具有最佳目标值的个体学习,参照该最佳个体来调整自己的搜索方向和搜索步长,即主动拉近与外部标杆的距离;如果目标值没有得到改善,则该个体进行内部标杆学习,即向该个体所在种群内具有最佳目标值的个体学习,参照本种群内最优个体来调整自己的搜索方法和搜索步长,即主动拉近与内部标杆的距离;如果目标值仍然没有得到改善,则该个体继续进行自我学习.此外,各小生境种群在学习过程中会相互交换最佳个体,即各小生境种群内个体的学习对象(内部标杆)不断发生改变.个体的上述 3 个学习行为并不是按顺序执行的,而是有选择地执行的,只有在执行前一个学习策略没有得到改善时,才会执行下一个学习策略.

BOA 运用标杆学习的理念来获取候选解,如何实现这种学习理念,本文针对常用的浮点数(实数)编码方式和 0-1 编码方式(包括 Binary Code & Gary Code),分别设计了相应的实现方法.

1.1.1 外部标杆学习设计

设 X_E^{best} 是整个生态系统内具有最佳目标值(根据优化目的,取最大或最小,下同)的个体,即全局最佳个体,也即外部标杆,其对应的基因表达式为 G_E^{best} ; 种群 P_K 所属的第 i 个个体 X_K^i 所对应的基因表达式为 G_K^i , 则个体 X_K^i 的外部学习率为

$$\begin{cases} \max f(x): Grate_K^i = Grate' + f_K^i / \tilde{f}_K - 1 \\ \min f(x): Grate_K^i = Grate' + \tilde{f}_K / f_K^i - 1 \end{cases} \quad (1)$$

其中, $\max f(x)$ 表示求解目标函数的最大值, $\min f(x)$ 表示求解目标函数的最小值, $Grate'$ 表示外部学习率的初始值, f_K^i 表示个体 X_K^i 的目标值, \bar{f}_K 表示该个体所在种群 P_K 的平均目标值. 可以看到: 如果某个体的目标值越符合优化目的(根据优化目的, 以求解目标函数的最大值为例), 当其目标值大于所在种群的平均目标值时, 其学习欲望会变得强烈, 其外部学习率会增大. 这样, 生态系统内那些有前途的个体主动聚集到全局最佳个体所在的搜索邻域内, 因而能起到协助搜索的作用. 同理, 当其目标值小于所在种群的平均目标值时, 其学习欲望会降低, 其外部学习率会减小, 以等待下一个学习对象.

下面针对常用的两种不同编码方案, 设计了相应的学习方法:

如果采用 0-1 编码方案, 个体 X_K^i 进行外部标杆学习, 是指其基因表达式 G_K^i 中的基因位值以 $Grate_K^i$ 的概率被 G_E^{best} 中相应的基因位值所取代, 即个体 X_K^i 主动缩小与全局最佳个体 X_E^{best} 的海明距离(Hamming distance).

如果采用浮点数编码方案, 个体 X_K^i 进行外部标杆学习, 是指其对应的基因表达式 G_K^i 以 $Grate_K^i$ 的概率按公式(2)进行更新, 即个体 X_K^i 主动缩小与全局最佳个体 X_E^{best} 的欧氏距离(Euclidean distance):

$$G_K^i = G_K^i + \lambda(G_E^{best} - G_K^i) \quad (2)$$

其中, $\lambda \in [0, 1]$, 是 X_K^i 进行外部标杆学习时的移动步长因子. 实验结果表明: 当 λ 与搜索空间的大小成正比例时, 优化效果较好; 当然, 也可以引入目标值等要素, 使 λ 在学习过程中动态自适应地改变, 优化效果更好. 这不是本文研究重点, 留待以后再做深入研究.

1.1.2 内部标杆学习设计

设 X_E^{best} 是小生境种群 P_K 内具有最佳目标值的个体(即局部最佳个体, 也即内部标杆), 其对应的基因表达式为 G_E^{best} ; 该种群内第 i 个个体 X_K^i 所对应的基因表达式为 G_K^i , 则个体 X_K^i 的内部学习率 $Brate_K^i$ 为

$$\begin{cases} 0-1 \text{ 编码: } Brate_K^i = Brate' - HD_{k,h} / Length + 1 \\ \text{实数编码: } Brate_K^i = Brate' - ED_{k,h} / Radius + 1 \end{cases} \quad (3)$$

其中, $Brate'$ 表示内部学习率的初始值; $HD_{k,h}$ 为该个体与 X_E^{best} 的海明距离(Hamming distance), $Length$ 为种群内个体的基因表达式编码长度; $ED_{k,h}$ 为该个体与 X_E^{best} 的欧氏距离(Euclidean distance), 即 $ED_{k,h} = \sqrt{\sum_1^n (x_i^{best} - x_i)^2}$; $Radius$ 为搜索空间的直径, 即 $Radius = \sqrt{\sum_1^n (b_i - a_i)^2}$, 其中, x_i 是个体基因表达式中的第 i 维分量, 且 $x_i \in [a_i, b_i]$.

由公式(3)可知, 当小生境种群中某个体与该种群中最佳个体的海明距离或欧氏距离较小时, 其学习欲望会自动升高, 从而迅速聚拢到该局部最佳个体的搜索邻域内, 以协助其进行密集搜索.

与外部标杆学习类似, 当采用 0-1 编码方案时, 个体 X_K^i 进行内部标杆学习, 是指 G_K^i 中与 G_E^{best} 相异的基因位值以 $Brate_K^i$ 的概率被 G_E^{best} 中相应的基因位值替换, 即个体 X_K^i 主动缩小与局部最佳个体 X_E^{best} 的海明距离; 当采用浮点数编码方案时, 个体 X_K^i 进行内部标杆学习, 指 G_K^i 以 $Brate_K^i$ 的概率按公式(4)进行更新, 即个体 X_K^i 主动缩小与局部最佳个体 X_E^{best} 的欧氏距离.

$$G_K^i = G_K^i + \lambda(G_K^{best} - G_K^i) \quad (4)$$

同理, $\lambda \in [0, 1]$, 是 X_K^i 进行内部标杆管理时的移动步长因子.

这里的外部标杆学习和内部标杆学习都是缩小个体之间的海明(或欧氏)距离, 看似相同, 但实际上两者却有很大的差异, 这也是本算法的核心思想之一: 这种主动缩小与最优个体距离的行为, 既有利于种群进行密集搜索, 从而形成群集效应, 快速搜索到全局最优解, 同时也是保持种群多样性的最好方式——因为每个个体的学习对象是不断地动态变化的, 因此生态系统内个体群集的层次也是动态变化的(详见后文算法分析和仿真实验部分内容).

1.1.3 自我学习设计

如果采用 0-1 编码方案, 则个体的自我学习是指进行对偶映射(dual mapping)^[30], 即该个体的基因表达式中每个基因位都执行对偶映射, 即 $0 \leftrightarrow 1$, 如图 1 所示.

$$\begin{array}{l} \text{原基因表达式: } \frac{1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0}{} \\ \text{对偶基因表达式: } 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \end{array}$$

Fig.1 Dual mapping based on 0-1 encoding

图 1 基于 0-1 编码的对偶映射

如果采用浮点数编码方案,则个体的自我学习是指进行类逻辑斯蒂混沌映射(logistic chaos mapping),即个体 X_K^i 的基因表达式 G_K^i 按公式(5)进行更新.利用混沌运动对初始状态的敏感性和非重复的遍历性,使个体及时跳出当前所在区域,以便对解空间的其他区域展开搜索.设 $G_K^i = [x_1, x_2, \dots, x_{n-1}, x_n], x_i \in [a_i, b_i]$:

$$\begin{cases} \lambda_i(0) = \frac{x_i - a_i}{b_i - a_i} \\ \lambda_i(t+1) = \delta \lambda_i(t)(1 - \lambda_i(t)) \\ x_i(t) = a_i + \lambda_i(t)(b_i - a_i) \\ \delta \in [2, 4], i = 1, 2, 3, \dots, n \end{cases} \quad (5)$$

设 X_K^i 是隶属小生境种群 P_K 的个体,则个体 X_K^i 的自我学习率为

$$\begin{cases} \max f(x): Srate'_k = Srate' \times \tilde{f}_K / f_K^i \\ \min f(x): Srate'_k = Srate' \times f_K^i / \tilde{f}_K \end{cases} \quad (6)$$

其中, $\max f(x)$ 表示求解目标函数的最大值, $\min f(x)$ 表示求解目标函数的最小值, $Srate'$ 表示自我学习率的初始值, f_K^i 表示个体 X_K^i 的目标值, \tilde{f}_K 表示该个体所在种群 P_K 的平均目标值.由公式(6)可以看到:当优化的目的是实现目标最大化时,如果个体的目标值小于其所在种群的平均目标值,则其学习率将迅速增大,其学习欲望会迅速提高,于是有较多机会得到其对偶个体,以提高其自身的目标值;但如果该个体的目标值已经比其所在种群的平均目标值大了,则其学习欲望会快速消退,则其学习率将快速降低,这样可保护优良基因以免遭破坏.同理,当优化的目的是实现目标最小化时,其自学习欲望也会相应地自动调整,以符合优化目的.

1.2 BOA算法伪代码

设 $E = \{P_1, P_2, \dots, P_{np}\}$ 是包含 np 个小生境种群的生态系统, N_i 表示种群 P_i 中个体的数目, P_i^j 表示种群 P_i 中第 j 个个体, P_i^{best} 表示种群 P_i 中具有最佳目标值的个体, \tilde{f}_i 表示种群 P_i 在当前代的平均目标值, \tilde{f}_E 表示整个生态系统当前代的平均目标值, P_{best} 表示当前代整个生态系统中具有最佳目标值的个体, $maxgen$ 表示最大迭代次数.则 BOA 算法的伪代码可表示如下:

1. 种群等相关参数初始化
2. for $gen=1:maxgen$, do
 - (a) for $i=1:np$, do
 - i. 评估种群 P_i 中每个个体的目标值
 - ii. 计算种群 P_i 的平均目标值 \tilde{f}_i
 - iii. 找出并记录种群 P_i 中具有最佳目标值的个体 P_i^{best} (树立内部标杆)
 - (b) 找出并记录整个生态系统中具有最佳目标值的个体 P_{best} (树立外部标杆)
 - (c) 记录并更新整个生态系统中具有最佳目标值的个体,即全局最优解
 - (d) 评估整个生态系统在当前代的平均目标值 $\tilde{f}_E = (\sum \tilde{f}_i) / np$
 - (e) for $i=1:np$, do
 - i. 种群 p_i 中个体 P_i^j 进行外部标杆学习
 - ii. 如果个体 P_i^j 的目标值没有得到改善,则进行内部标杆学习
 - iii. 如果个体 P_i^j 的目标值仍然没有得到改善,则进行自我学习

- (f) if 与上一代相比,平均目标值 \bar{f}_E 没有得到改善或最佳个体 P_{best} 没有发生改变 do
- 各小生境种群之间相互交换具有最佳目标值的个体(即各种群重新树立新的内部标杆)

3. 对全局最佳个体解码并输出全局最优解

2 仿真实验

目前,在数值优化领域已经出现了多达几十个经典测试函数,这里摘取 15 个求解最小值的测试函数.考虑到实际优化问题中,最优解与所在搜索空间的相对位置是非确定性的,因此,借鉴文献[31]的思想设计 3 种位置关系:最优解恰好在搜索空间边界上(BOUNDARY)、最优解靠近搜索空间边界(CLOSE)、最优解在搜索空间正中心位置(CENTRE),具体设置见表 1.

Table 1 Testbed functions
表 1 标准测试函数列表

名称	函数	搜索空间			最优解的位置	全局最优解	目标值精度
		Boundary	Close	Centre			
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[0,200]^D$	$[-5, 100]^D$	$[-100, 100]^D$	0.0^D	0	0.01
Griewank	$f_2(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[0,1200]^D$	$[-5, 1195]^D$	$[-600, 600]^D$	0.0^D	0	0.01
Rastrigin	$f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[0,10]^D$	$[-1.12, 9.12]^D$	$[-5.12, 5.12]^D$	0.0^D	0	0.01
Rosenbroke	$f_4(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[0,60]^D$	$[0,60]^D$	$[-30, 30]^D$	1.0^D	0	0.01
Ackley	$f_5(x) = -20e^{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}} - e^{\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)} + e + 20$	$[0,64]^D$	$[-5, 59]^D$	$[-32, 32]^D$	0.0^D	0	0.01
Schwefel	$f_6(x) = 418.9829D - \sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	$[420.9687, 500]^D$	$[415, 500]^D$	$[340, 500]^D$	420.9687^D	0	0.01
Step	$f_7(x) = \sum_{i=1}^D (x_i + 0.5)^2$	$[0,200]^D$	$[-5, 195]^D$	$[-100, 100]^D$	-0.5^D	0	0.01
Schwefel's P2.22	$f_8(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[0,20]^D$	$[-1, 19]^D$	$[-10, 10]^D$	0.0^D	0	0.01
Quadric	$f_9(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[0,200]^D$	$[-5, 195]^D$	$[-100, 100]^D$	0.0^D	0	0.01
Quadric Noise	$f_{10}(x) = \sum_{i=1}^D ix_i^4 + rand[0,1]$	$[0,2.56]^D$	$[0.28, 2.28]^D$	$[-1.28, 1.28]^D$	0.0^D	0	0.01
Schwefel's P2.21	$f_{11}(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	$[0,200]^D$	$[-5, 195]^D$	$[-100, 100]^D$	0.0^D	0	0.01
Unnamed	$f_{12}(x) = \left(\sin^2 \left(50 \sqrt{\frac{D}{i=1} \sum_{i=1}^D x_i^2} + 1 \right) \sqrt{\sum_{i=1}^D x_i^2} \right)^2$	$[0,10]^D$	$[-1.12, 9.12]^D$	$[-5.12, 5.12]^D$	0.0^D	0	0.01
Schaffer	$f_{13}(x) = 0.5 + \frac{\sin^2 \sqrt{\sum_{i=1}^D x_i^2} - 0.5}{\left[1 + 0.001 \sum_{i=1}^D x_i^2 \right]^2}$	$[0,200]^D$	$[-5, 195]^D$	$[-100, 100]^D$	0.0^D	0	0.01

Table 1 Tested functions (Continued)

表 1 标准测试函数列表(续)

名称	函数	搜索空间			最优解的位置	全局最优解	目标值精度
		Boundary	Close	Centre			
Generalized Penalized01	$f_{14}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + (y_d - 1)^2 + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right\} + \sum_{i=1}^D (x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < a \end{cases}$	$[-1, 100]^D$	$[-5, 95]^D$	$[-50, 50]^D$	-1.0^D	0	0.01
Generalized Penalized02	$f_{15}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] \right\} + \sum_{i=1}^D (x_i, 5, 100, 4)$ $(x_i, a, k, m) = \begin{cases} k(x_i - a)_m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)_m, & x_i < a \end{cases}$	$[0, 100]^D$	$[-5, 95]^D$	$[-50, 50]^D$	1.0^D	0	0.01

为了验证 BOA 算法的性能,与近年来出现的几种有代表的算法进行对比,共设计了两组实验:一组测试算法的全局收敛能力,一组用来测试算法的综合性能。

分别使用近年来出现的几种典型的群体智能优化方法,如人工蜂群算法 ABC^[11]、人工鱼群算法 AFSA^[20]、改进的遗传算法 SAMGA^[32]、改进的粒子群优化算法 SzAPSO^[32]与本文所提出的标杆学习算法 BOA 进行测试比较。

2.1 全局收敛能力测试

算法的控制参数对算法性能影响非常大,各算法涉及到的控制参数众多且不尽相同,为了保证实验测试的客观、公正,除了相同的软硬件实验平台以外(硬件 CPU:AMD Athlon(tm) 64 X2 Dual Core Processor 3600+,1.91GHz; DDR:667MHz,1024MB,软件 Matlab7.12),在此实验中,各算法统一采用浮点数编码方案,种群中个体的总数目统一设为 100,进行一次搜索时迭代次数统一设为 3 000,每个实验重复执行 30 次;各算法的其他重要控制参数分别按照各自参考文献中提出的最佳方案配置。为了缩短篇幅,此实验只测试 15 个函数的最优解在搜索空间正中心位置的情形。另外,为求客观、公正,各算法在初始化时所有个体皆被置于边界上同一点,以消除初始化时个体的位置优劣对算法搜索性能的影响,同时,搜索空间维数设定为 50 维。

图 2 是由 5 种算法分别在 15 个测试函数上实验 30 次,取其中表现最好的一次和表现最差的一次,共两次实验结果平均值绘制的收敛曲线,其中,横轴是实验时的迭代次数,纵轴是对应的目标函数值取自然对数之后的函数值。观察其中各图可以发现:5 种算法都能搜索到各测试函数的全局最优解附近,在决策变量范围不大时,5 种算法搜索结果的精度相差不大;但当决策变量的变化范围较大时,5 种算法搜索结果的精度有较大差别,SzAPSO 和 BOA 表现较为突出,SzAPSO 有时比 BOA 表现更优异,如 f_4, f_7, f_8 和 f_{11} 。

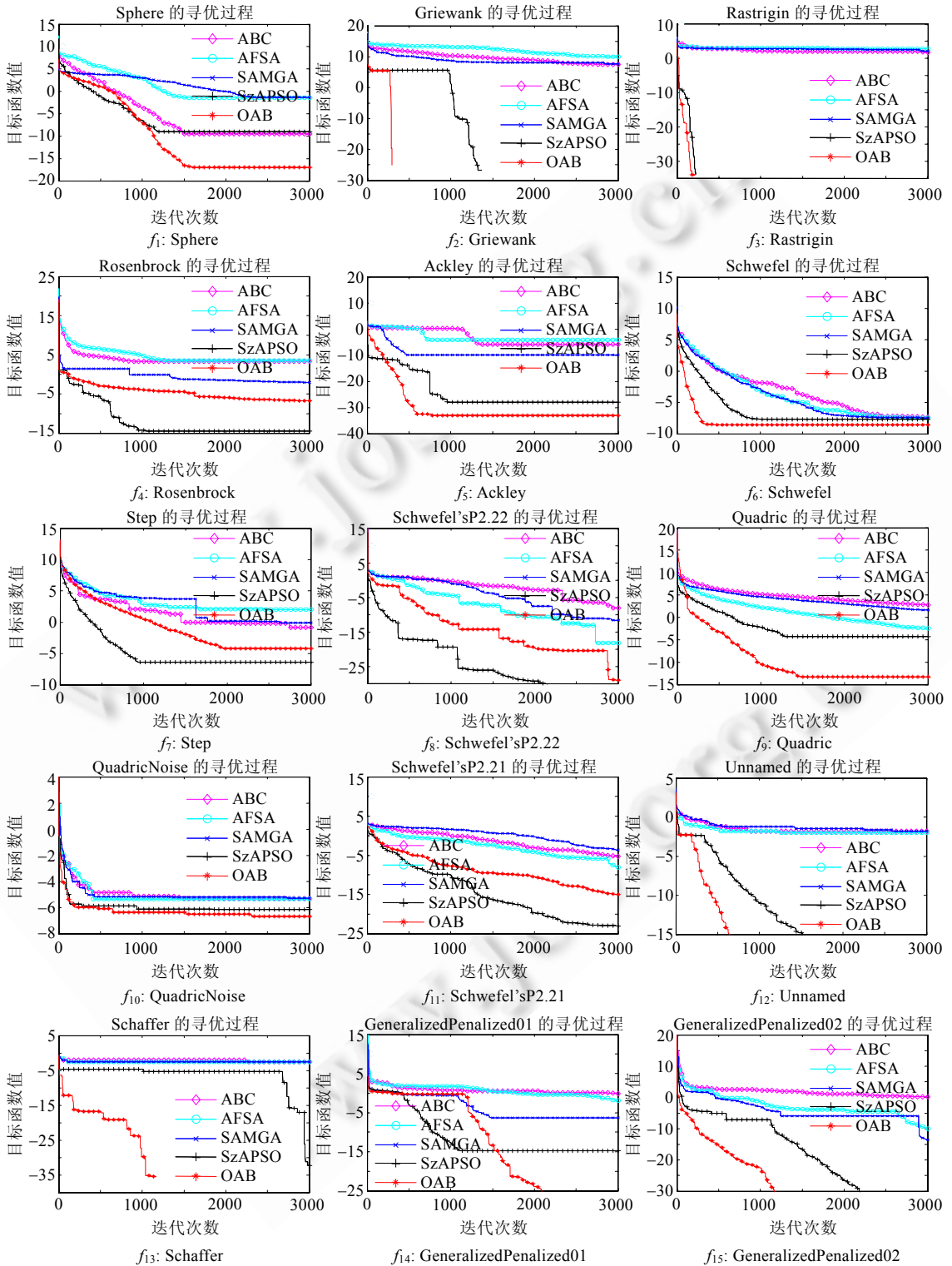


Fig.2 Test result of the 15 functions for the 5 algorithms while $D=50$

图2 $D=50$ 时,5种算法在15个标准测试函数上的收敛曲线

2.2 算法综合性能比较

在此实验中,各算法的参数设置与上一节实验相同。

表 2 是问题最优解与所在搜索空间处于 3 种相对位置、搜索空间维数为 50 时,5 种算法在以上 15 个测试函数上分别实验 30 次的平均目标值和方差.结合图 2,由表 2 中的数据可以看到最解位置对算法性能的影响:ABC,AFSA 和 SAMGA 对最优解的位置敏感程度不一,对有些问题,最优解在边界时优化效果较好,有时在中心位置时优化效果更好,从具体数据看,关系较为复杂;而 SzAPSO 和 BOA 总体上对最优解位置不敏感,无论最优解在何处,其收敛速度和收敛精度均表现较好,具有较强的鲁棒性.

Table 2 Mean and variance of test result of 5 algorithms for 3 situations while $D=50$

表 2 当 $D=50$ 时,5 种算法在 3 种位置关系下收敛精度测试结果的均值和方差

		f_1 : Sphere		f_2 : Griewank		f_3 : Rastrigin	
		M	SD	M	SD	M	SD
BOUNDARY	ABC	4.31E-02	8.53E-01	4.17E-01	7.80E-01	6.35E+00	5.47E-02
	AFSA	9.11E-01	6.22E-01	4.97E-01	3.90E-01	3.53E+00	2.96E-02
	SAMGA	8.18E-02	3.51E-01	9.03E-01	2.42E-01	8.21E+00	7.45E-03
	SzAPSO	2.64E-03	5.13E-02	9.45E-10	4.04E-03	1.54E-08	1.89E-04
	OAB	1.46E-07	4.02E-02	4.91E-10	9.65E-04	4.30E-10	6.87E-04
CLOSE	ABC	1.69E-02	1.84E-01	9.29E-02	6.44E-01	2.08E-03	3.11E-02
	AFSA	6.49E-01	3.68E-01	7.76E-01	3.79E-02	3.01E+00	9.23E-02
	SAMGA	3.17E-02	6.26E-02	4.87E-02	8.12E-02	4.71E+00	4.30E-02
	SzAPSO	6.48E-03	7.80E-01	4.36E-10	5.33E-04	2.30E-09	1.85E-05
	OAB	4.51E-07	8.11E-03	4.47E-12	3.51E-03	8.44E-10	9.05E-05
CENTRE	ABC	2.35E-03	5.24E-01	4.84E-01	3.53E-01	8.85E+00	1.20E-02
	AFSA	6.50E-01	3.88E-01	3.25E+00	5.42E+00	7.55E+00	1.21E-01
	SAMGA	7.70E-01	9.61E-01	6.65E-01	3.23E-01	8.03E+00	1.20E-02
	SzAPSO	1.55E-03	3.30E-03	8.20E-12	5.48E-04	3.62E-11	6.57E-06
	OAB	1.01E-08	5.80E-04	2.55E-12	9.95E-06	3.12E-11	4.54E-07
		f_4 : Rosenbrock		f_5 : Ackley		f_6 : Schwefel	
		M	SD	M	SD	M	SD
BOUNDARY	ABC	1.36E+01	7.60E+00	4.89E-02	1.32E-01	5.95E-02	8.55E-01
	AFSA	8.69E+01	2.40E+01	3.38E-03	9.42E-02	6.22E-02	2.62E-01
	SAMGA	5.97E-01	2.33E-01	9.00E-04	9.56E-02	6.03E-02	8.01E-01
	SzAPSO	5.50E-04	1.84E-04	3.69E-09	5.75E-04	7.11E-04	2.92E-02
	OAB	1.45E-04	2.40E-03	1.11E-09	5.98E-04	2.22E-03	9.29E-02
CLOSE	ABC	3.06E+01	9.39E-01	1.95E-03	9.80E-02	1.17E-02	7.30E-01
	AFSA	5.09E+00	8.76E-01	2.26E-02	4.39E-02	2.97E-02	4.89E-01
	SAMGA	5.11E-02	5.50E-01	1.71E-03	1.11E-01	3.19E-02	5.79E-02
	SzAPSO	8.18E-05	6.22E-04	2.28E-09	2.58E-04	4.24E-03	2.37E-02
	OAB	7.95E-03	5.87E-02	4.36E-10	4.09E-03	5.08E-03	4.59E-02
CENTRE	ABC	3.88E+02	5.64E+03	3.24E-03	8.60E-02	2.67E-02	5.20E-01
	AFSA	2.35E+02	1.20E+02	1.74E-02	4.50E-01	3.25E-02	3.84E-01
	SAMGA	1.07E-02	8.10E-01	1.03E-04	2.06E+00	2.25E-02	6.56E-01
	SzAPSO	1.22E-05	2.51E-03	2.20E-11	5.40E-04	3.51E-03	1.52E-02
	OAB	2.56E-03	3.65E-01	3.32E-12	6.24E-03	1.90E-03	6.52E-03
		f_7 : Step		f_8 : Schwefel'sP2.22		f_9 : Quadric	
		M	SD	M	SD	M	SD
BOUNDARY	ABC	9.63E-01	3.77E-02	1.07E-02	3.05E-01	1.83E+01	5.96E+00
	AFSA	5.47E+01	8.85E-01	6.54E-02	7.44E-01	2.40E+01	6.82E+00
	SAMGA	5.21E+00	9.13E-01	4.94E-03	5.00E-02	8.87E+01	4.24E-01
	SzAPSO	2.32E-05	7.96E-02	7.79E-10	4.80E-03	2.87E-02	7.14E-03
	OAB	4.89E-03	9.87E-03	7.15E-12	9.05E-05	4.90E-04	5.22E-03
CLOSE	ABC	6.24E+00	2.62E-01	9.04E-02	6.10E-02	1.68E+01	9.67E-02
	AFSA	6.79E+01	3.35E+00	8.91E-02	6.18E-03	9.79E+00	8.18E-01
	SAMGA	3.96E+00	6.80E+00	3.34E-03	8.59E-02	7.13E+01	8.18E-01
	SzAPSO	3.67E-05	1.37E-04	6.99E-10	8.05E-04	5.00E-02	7.22E-03
	OAB	9.88E-03	7.21E-02	1.98E-11	5.77E-03	4.71E-04	1.50E-04

Table 2 Mean and variance of test result of 5 algorithms for 3 situations while $D=50$ (Continued)

表 2 当 $D=50$ 时,5 种算法在 3 种位置关系下收敛精度测试结果的均值和方差(续)

		f_7 : Step		f_8 : Schwefel'sP2.22		f_9 : Quadric	
		M	SD	M	SD	M	SD
CENTRE	ABC	1.13E+00	3.87E-01	1.85E-02	2.45E-01	2.02E+02	1.10E+01
	AFSA	1.65E+02	3.50E+01	5.75E-03	5.42E-02	1.52E+01	8.00E+00
	SAMGA	2.35E+00	1.08E+01	1.22E-03	8.89E-02	1.98E+02	1.14E+01
	SzAPSO	7.54E-04	6.24E-03	6.45E-12	5.84E-03	2.00E-02	6.55E-02
	OAB	1.08E-02	1.35E-02	2.09E-13	1.02E-05	8.80E-05	5.45E-03
		f_{10} : QuadricNoise		f_{11} : Schwefel'sP2.21		f_{12} : Unnamed	
		M	SD	M	SD	M	SD
BOUNDARY	ABC	6.60E-02	1.73E-02	8.15E-02	1.58E-01	6.56E-02	7.06E-02
	AFSA	5.19E-02	3.91E-01	9.06E-01	9.71E-01	3.57E-01	3.18E-02
	SAMGA	9.73E-03	8.31E-01	1.27E-02	9.57E-02	8.49E-02	2.77E-02
	SzAPSO	6.49E-04	8.03E-02	9.13E-07	4.85E-03	9.34E-05	4.62E-04
	OAB	8.00E-05	6.05E-02	6.32E-05	8.00E-02	6.79E-07	9.71E-05
CLOSE	ABC	4.54E-02	3.99E-02	9.75E-02	1.42E-01	7.58E-02	8.23E-02
	AFSA	4.32E-03	5.27E-01	2.78E-02	4.22E-02	7.43E-02	6.95E-02
	SAMGA	8.25E-02	1.68E-01	5.47E-02	9.16E-02	3.92E-02	3.17E-02
	SzAPSO	8.35E-04	6.57E-02	9.58E-07	7.92E-03	6.55E-04	9.50E-03
	OAB	1.33E-03	6.28E-03	9.65E-04	9.59E-03	1.71E-03	3.44E-04
CENTRE	ABC	1.24E-02	2.54E-02	4.20E-01	8.40E-01	2.65E-01	6.24E-01
	AFSA	1.34E-02	2.15E-01	5.45E-01	1.01E+00	2.98E-01	8.54E-01
	SAMGA	5.54E-02	3.22E-01	3.07E-01	1.07E+00	1.89E-01	5.21E-01
	SzAPSO	7.50E-04	6.86E-02	1.04E-06	6.71E-02	5.33E-05	2.54E-03
	OAB	1.24E-04	4.44E-02	2.56E-04	8.46E-03	2.55E-06	8.87E-03
		f_{13} : Schaffer		f_{14} : GeneralizedPenalized01		f_{15} : GeneralizedPenalized02	
		M	SD	M	SD	M	SD
BOUNDARY	ABC	4.39E-01	2.76E-01	7.51E+00	8.41E-02	3.52E+00	7.59E-01
	AFSA	3.82E-01	6.80E-01	2.55E+00	2.54E-02	8.31E-01	5.40E-02
	SAMGA	7.66E-02	6.55E-02	5.06E-02	8.14E-01	5.85E-03	5.31E-03
	SzAPSO	2.80E-06	1.63E-04	6.99E-08	2.44E-03	5.50E-05	7.79E-04
	OAB	1.87E-08	1.19E-03	8.91E-07	9.29E-04	9.17E-06	9.34E-03
CLOSE	ABC	4.90E-01	4.98E-02	9.59E+00	3.50E-02	2.86E+00	1.30E-01
	AFSA	4.46E-01	9.60E-01	5.47E+00	1.97E-03	7.57E-02	5.69E-02
	SAMGA	6.46E-01	3.40E-02	1.39E-02	2.51E-02	7.54E-02	4.69E-02
	SzAPSO	7.09E-07	5.85E-04	1.49E-06	6.16E-03	3.80E-06	1.19E-04
	OAB	7.55E-07	2.24E-04	2.58E-07	4.73E-04	5.68E-05	3.37E-03
CENTRE	ABC	3.56E-01	5.64E-01	1.60E+01	6.41E-02	1.47E+01	2.45E-01
	AFSA	2.22E-01	4.76E-01	1.20E+01	5.41E-02	2.46E-03	8.99E-02
	SAMGA	4.26E-01	2.05E-01	5.66E-03	8.72E-02	8.81E-02	2.04E-01
	SzAPSO	8.89E-07	6.52E-03	7.57E-07	5.24E-03	4.15E-06	2.45E-03
	OAB	5.43E-08	8.82E-03	2.85E-06	7.65E-03	6.56E-07	2.36E-03

为了测试各算法的可靠性,每种算法在 15 个函数上分别实验 30 次,设定的终止条件是达到如表 1 中所示的指定精度或达到最大迭代次数 5×10^4 .表 3 是 5 种算法在指定精度下对表 1 中 15 个函数的测试结果的均值.由表 3 可以看到:BOA 对 15 个函数在 3 种位置关系下,大部分能以 100%的概率成功收敛到指定目标精度,因其收敛速度较快,因此所记录的成功达到指定目标精度时的迭代次数和 CPU 运行时间相对 ABC,AFSA 和 SAMGA 少了很多;SzAPSO 也有类似的结果,这是由于其空间缩放和吸引子策略的成功应用所致.

在上述对比实验中产生了大量的实验数据,有时并不利于用来分析对比算法的性能,因此采用配对双侧 T 检验方法对所得的实验数据进行量化统计分析,以便客观评价 BOA 算法与其他各算法之间的性能差别程度.在此次检验中,显著水平 α 设为 0.05,自由度 $V=30-1=29$,查表可知, $t_{0.05}(29)=1.699$,根据公式容易计算出统计量 t 值,其计算结果见表 4.根据双侧 T 检验规则,当 $|t| < t_{0.05}(29)$ 时,表明 BOA 算法与所配对算法无显著性差异,表 4 最后一列列出了在 15 个测试函数中,优化结果与 BOA 有显著差异的个数.由表中数据可知:BOA 与 ABC,AFSA 及 SAMGA 这 3 种算法对绝大多数测试函数在优化性能上有显著差异,而与 SzAPSO 在优化性能上差异并不显著,这与前面实验结果的若干直观印象是一致的.

Table 3 Mean value of the result of reliability test for the 5 algorithms under the specified target accuracy

表 3 在指定目标精度下,5 种算法可靠性测试结果的均值

		f_1 : Sphere			f_2 : Griewank			f_3 : Rastrigin		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
BOUNDARY	ABC	40 737	254.25	51	37 095	244.05	60	35 303	271.03	64
	AFSA	35 290	387.56	62	21 089	231.85	58	21 592	216.53	59
	SAMGA	46 350	398.61	56	35 787	289.11	59	43 847	304.52	67
	SzAPSO	1 869	59.41	97	611	87.45	100	309	42.98	100
	OAB	1 518	73.49	99	775	101.76	100	257	32.01	100
CLOSE	ABC	34 878	183.52	51	32 788	195.51	65	41 173	210.29	68
	AFSA	23 925	122.48	55	21 786	192.27	64	34 742	193.03	60
	SAMGA	27 345	184.1	68	42 457	298.1	57	15 855	498.34	64
	SzAPSO	876	96.08	100	700	49.04	100	512	48.35	100
	OAB	745	104.5	100	537	37.57	100	423	30.6	100
CENTRE	ABC	37 881	273.98	57	37 888	188.59	59	21 938	174.24	59
	AFSA	28 530	186.75	61	37 157	179.05	62	19 078	106.85	59
	SAMGA	27 859	256.64	56	19 612	123.19	62	38 276	188.81	67
	SzAPSO	1 869	67.03	98	2 174	48.1	97	760	57.16	100
	OAB	1 015	64.46	100	560	39.26	100	544	52.32	100
		f_4 : Rosenbrock			f_5 : Ackley			f_6 : Schwefel		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
BOUNDARY	ABC	24 489	266.5	61	37 564	220.71	57	17 500	218.86	63
	AFSA	32 280	287.51	72	32 755	196.05	59	19 830	122.21	57
	SAMGA	12 316	124.87	68	29 298	208.8	62	12 555	192.31	50
	SzAPSO	569	83.04	100	854	52.26	100	803	40.76	100
	OAB	1 535	64.68	97	546	51.41	100	665	35.6	100
CLOSE	ABC	23 802	274.9	58	47 965	236.93	65	17 583	139.11	63
	AFSA	23 986	220.5	67	27 361	159.34	57	21 542	200.51	67
	SAMGA	12 755	172.15	66	10 932	124.91	73	29 264	160.11	58
	SzAPSO	531	95.1	100	1 065	51.18	100	1 587	40.79	98
	OAB	1 450	101.76	98	876	57.04	100	860	42.82	100
CENTRE	ABC	24 919	258.29	62	42 036	223.51	64	14 292	185.43	60
	AFSA	27 988	215.47	71	12 715	120.97	61	37 861	115.17	64
	SAMGA	17 020	110.61	64	10 515	99.54	58	31 487	140.49	67
	SzAPSO	764	45.69	100	877	48.52	100	923	42.3	100
	OAB	1 101	47.58	98	664	40.12	100	892	40.16	100
		f_7 : Step			f_8 : Schwefel'sP2.22			f_9 : Quadric		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
BOUNDARY	ABC	43 793	257.54	55	30 100	162.81	64	35 333	237.59	59
	AFSA	42 698	228.08	57	23 149	221.03	69	38 095	222.38	68
	SAMGA	36 540	213.51	64	32 704	214.75	51	29 232	185.64	57
	SzAPSO	1 959	131.74	98	1 661	52.87	97	2 246	49.34	98
	OAB	1 701	119.36	97	908	50.05	100	2 166	50.95	99
CLOSE	ABC	46 496	312.35	64	22 528	120.33	55	43 435	266.38	64
	AFSA	38 442	290.71	63	24 192	132.24	60	42 322	253.3	53
	SAMGA	43 470	281.16	54	31 449	129.78	58	19 990	149.06	53
	SzAPSO	1 596	128.64	96	1 067	55.37	100	1 894	41.43	98
	OAB	1 257	110.94	97	2 119	63.83	99	1 704	45.23	98
CENTRE	ABC	38 110	215.73	62	41 291	163.39	53	21 571	197.17	57
	AFSA	39 715	208.73	60	26 918	133.78	64	35 533	206.09	51
	SAMGA	35 561	198.7	68	39 807	155.98	57	39 093	211.06	62
	SzAPSO	1 427	130.17	99	2 409	60.03	99	2 291	58.43	97
	OAB	2 283	123.93	97	2 734	56.57	98	1 977	43.05	99
		f_{10} : QuadricNoise			f_{11} : Schwefel'sP2.21			f_{12} : Unnamed		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
BOUNDARY	ABC	21 804	224.72	67	34 864	216.68	53	36 599	232.48	57
	AFSA	25 465	232.74	65	32 483	162.98	50	47 103	205.95	53
	SAMGA	31 986	228.43	59	45 136	135.54	61	47 807	212.93	55
	SzAPSO	2 694	95.15	99	1 240	59.96	97	761	50.32	100
	OAB	2 548	43.89	98	1 544	56.81	98	989	47.87	100

Table 3 Mean value of the result of reliability test for the 5 algorithms under the specified target accuracy (Continued)

表 3 在指定目标精度下,5 种算法可靠性测试结果的均值(续)

		f_{10} : QuadricNoise			f_{11} : Schwefel'sP2.21			f_{12} : Unnamed		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
CLOSE	ABC	29 652	259.31	67	34 463	185.09	60	31 739	131.59	53
	AFSA	29 103	189.55	70	36 886	219.83	56	28 658	154.7	55
	SAMGA	30 548	209.78	66	45 003	194.3	63	41 060	167.98	28
	SzAPSO	563	43.68	100	1 463	61.59	97	971	44.68	100
	OAB	691	42.67	100	1 261	59.38	97	952	47.86	100
CENTRE	ABC	30 799	148.77	56	39 013	178.03	53	28 450	198.71	58
	AFSA	21 996	152.71	66	39 487	175.52	60	32 456	154.42	63
	SAMGA	27 166	167.41	67	32 085	131.02	58	36 587	219.16	54
	SzAPSO	596	43.6	99	1 196	52.2	98	988	52.92	100
	OAB	598	41.45	100	923	50.36	100	947	46.15	100
		f_{13} : Schaffer			f_{14} : GeneralizedPenalized01			f_{15} : GeneralizedPenalized02		
		迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)	迭代次数	时间 (s)	成功率 (%)
BOUNDARY	ABC	27 351	192.34	50	25 318	194.08	50	30 388	194.62	60
	AFSA	24 817	247.93	48	25 426	164.68	61	25 063	157.14	54
	SAMGA	37 235	171.41	46	25 539	140.84	48	29 547	185.39	56
	SzAPSO	1 298	56.12	99	1 082	60.58	98	1 195	59.75	98
	OAB	939	52.14	100	952	54.6	100	1 216	56.76	99
CLOSE	ABC	29 176	208.28	50	32 216	166.21	49	29 739	145.97	55
	AFSA	28 425	167.29	49	38 931	181.31	50	31 297	216.5	61
	SAMGA	31 281	122.37	52	40 580	223.8	54	28 536	189.55	57
	SzAPSO	1 012	58.04	98	1 242	63.27	97	1 084	57.19	97
	OAB	1 057	46.52	98	1 137	59.5	95	1 285	50.25	97
CENTRE	ABC	46 470	221.1	50	46 951	201.57	49	25 556	149.08	54
	AFSA	38 786	194.97	52	43 798	220.63	52	36 169	167.54	59
	SAMGA	24 340	111.57	48	37 508	200.26	51	31 511	185.01	55
	SzAPSO	1 093	51.69	98	1 124	54.22	99	1 241	60.5	97
	OAB	940	40.78	100	853	51.78	100	945	59.06	100

Table 4 Test of the significance of difference of BOA with other 4 algorithms under 3 situations

表 4 BOA 与其他 4 种算法在 3 种位置关系下的差异显著性检验

		OAB 与 ABC 等 4 种算法的差异性检验								显著差异个数
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	
BOUNDARY	ABC	-1.3739	-31.6491	-28.6663	-154.58	-78.7985	-85.029	-39.3985	-1.5216	13
	AFSA	-14.3632	-47.8718	-70.8154	-110.953	-79.7309	-44.6566	-18.2948	-177.056	15
	SAMGA	-169.692	-73.6852	-29.9224	-62.267	-23.1887	-141.855	-152.937	-100.442	14
	SzAPSO	0	-0.9342	0	-0.6827	-0.2095	0	-1.6052	-0.5738	1
CLOSE	ABC	-46.6182	-103.581	-127.036	-161.765	-84.046	-53.8871	-137.538	-30.3748	15
	AFSA	-165.815	-58.2035	-74.7231	-65.1389	-78.4849	-58.9633	-138.102	-123.673	15
	SAMGA	-130.588	-30.8638	-182.512	-127.303	-66.7785	-121.39	-19.6693	-166.087	14
	SzAPSO	-1.4769	-0.2463	-0.5963	0	-0.3125	-0.0844	-2.834	0	1
CENTRE	ABC	-1.1852	-131.236	-13.7408	-140.781	-55.5658	-1.5374	-141.006	-32.4991	13
	AFSA	-103.2	-115.597	-74.6852	-54.6231	-103.679	-177.702	-126.565	-95.9243	15
	SAMGA	-34.2491	-183.459	-115.258	-98.5373	-137.89	-173.764	-86.027	-130.491	15
	SzAPSO	-0.9849	0	0	-0.4076	-0.4077	-1.5337	-0.8312	0	1
		f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	-	显著差异个数
BOUNDARY	ABC	-60.507	-173.185	-11.9195	-151.259	-117.211	-78.2001	-49.27	-	13
	AFSA	-101.475	-30.3641	-142.492	-146.53	-65.9368	-34.1872	-171.694	-	15
	SAMGA	-74.0718	-1.0468	-124.642	-158.265	-185.143	-134.776	-41.5539	-	14
	SzAPSO	-2.1889	-0.4106	-0.2242	-0.9773	-0.6	-0.0731	-1.2432	-	1

Table 4 Test of the significance of difference of BOA with other 4 algorithms under 3 situations (Continued)

表 4 标杆学习算法与其他 4 种算法在 3 种位置关系下的差异显著性检验(续)

		OAB 与 ABC 等 4 种算法的差异性检验								显著差异个数
		f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	-	
CLOSE	ABC	-6.7897	-33.5628	-107.222	-119.23	-118.042	-121.16	-156.277	-	15
	AFSA	-150.268	-47.4251	-81.7142	-77.8168	-176.58	-112.346	-136.346	-	15
	SAMGA	-139.018	-0.8135	-47.8386	-72.5641	-82.4437	-71.9111	-106.039	-	14
	SzAPSO	0	-0.6862	-1.6005	-0.1016	0	-0.2871	-1.1005	-	1
CENTRE	ABC	-126.226	-12.0824	-147.567	-132.814	-93.8962	-41.6284	-68.776	-	13
	AFSA	-22.3186	-171.526	-139.636	-151.556	-11.1454	-26.404	-32.8418	-	15
	SAMGA	-97.5004	-121.391	-42.6437	-58.946	-160.955	-4.6675	-177.785	-	15
	SzAPSO	-0.6622	-0.1639	-1.6245	0	-2.0262	-1.1028	-0.7661	-	1

3 简略分析

下面简单分析 BOA 算法的几个主要特点.

3.1 不同的搜索模式

如本文引言部分所述,目前国内外学者已经提出了 30 多种智能优化算法(不包括神经网络相关算法),典型的如遗传算法、蚁群算法、粒子群算法、人工鱼群算法和模拟退火算法等.它们之间的本质区别在于各自生成和选取候选解的指导思想和执行方法不同,也即搜索模式不同.

以遗传算法(GA)为代表的进化算法,一般通过选择、交叉和变异等遗传算子来生成和选取新的候选解,这个过程中遗传操作的随机性可以通过交叉率和变异率等参量来控制,但总体上还是智能性偏弱,且其搜索策略多是被动适应的.粒子群算法(PSO)中,通过粒子更新公式来产生候选解,粒子能够记住以前的位置和利用其他粒子的信息,其搜索行为具有一定智能性,一般只适用于浮点数编码;也有学者提出了基于 0-1 编码方案的粒子群算法^[33,34],却是以失去粒子搜索的方向性为代价的,牺牲了 PSO 搜索的智能性.人工蜂群算法(ABC)通过角色分工和转换的方式产生候选解,个体的迁移形成一种正反馈机制,还有一定的负反馈机制,蜂群行动的倾向性较为明显.人工鱼群算法(AFSA)与 PSO 类似,其候选解的产生机制模仿鱼群的觅食、聚群和追尾等行为模式来设计,一般只适用于浮点数编码.蚁群算法(ACO)利用正反馈机制,蚂蚁根据“信息素较浓的路线更近”的原则,采用概率算法选择出最佳路线,是一种比较独特的搜索模式,一般需要较多的迭代次数才能克服局部极值干扰,适用于像 TSP 一类的离散型问题,但当问题规模较大时,其收敛会变得很困难.模拟退火算法(SA)运用 Metropolis 接受准则来帮助跳出局部极值,本质上只是一个基于蒙特卡罗迭代求解法的启发式随机搜索过程,没有充分利用领域知识,且理论上需要无限多次迭代才能达到热平衡状态,但在实际运用过程中却无法达到这一要求.禁忌搜索算法(taboo search,简称 TS)和捕食搜索算法(predatory search,简称 PS)更多地表现为一种独特的搜索策略和搜索思路:前者利用禁忌表标记出解空间内已经搜索过的区域,能够帮助算法减小重复性的搜索动作;后者并没有具体的寻优计算方法,本质上只是一种平衡全局搜索和局部搜索的策略,即一种平衡算法开发性(exploration)与利用性(exploitation)的策略.多 Agent 系统用于优化问题时,一般用于分布式计算环境,其计算过程是异步并发的,每个成员 Agent 只有非完全的问题求解能力,并无全局信息,Agent 间基于合作的通信被限制在同一节点的 Agent 之间,其迁移机制和通信机制类似于并行遗传算法.

其他算法也有自己独特的搜索模式,这里不再赘述.

本文提出的基于标杆管理的优化算法(BOA),其候选解是通过标杆学习来产生和选取的,针对常见的浮点数编码方案和 0-1 编码方案,分别采用减少海明距离和欧氏距离的方法来实现这个学习过程.其过程为:生态系统内的所有个体都在有选择地进行标杆学习,并且学习对象在不断地变化.其搜索行动的目的性和方向性很明确,即通过对优秀个体的模仿学习,超越学习对象而使自己也成为其他个体的学习对象.随着学习过程的进行,当某个小生境种群发现了较好的全局解时(即重新树立了外部标杆),将会吸引其他小生境种群中的个体进入,以协助进行密集搜索.但如果某个小生境种群在进化过程中一直都没发现较好的全局解,则其中所有个体将

会逐渐被吸引向其他小生境种群,最终导致该小生境种群消亡.同理,如果某个体能够发现较好的全局解,将会从其他小生境种群中吸引更多的个体到其邻域内,从而形成新的小生境种群,这即是本文所讲的动态小生境技术.因此,BOA 算法是一个学习性竞争和竞争性学习的寻优模型.

本文提出的动态小生境,在前面的实验环境中,其并存小生境只是逻辑存在的,在单体机上的实验,本质上还是串行计算的.从上述原理可以看出:BOA 算法在本质上也适合并行计算环境,其中的通信机制还有待深入研究.

就智能优化算法本身来讲,目前国内外相关学者的的工作大致可以分为 3 个层次:一是对已有算法作局部改进,如设计出新的操作算子等等;二是提出新的算法,如在遗传算法之后,学者们提出的各种算法都有自己独特的搜索模式;三是智能计算思维的突破.遗传算法使用各种编码方案,本质上相当于将当前所在空间的问题映射到另外一个空间去解决;运用种群和个体进行搜索求解,本质上相当于用枚举法来求解问题答案,只是引入概率的思想之后,使得枚举答案的出现有了一定的倾向性,即所谓搜索行为的智能性.因此可以说,遗传算法开创了一种运用经过编码的种群和个体对问题答案进行枚举搜索求解的思路,这与传统的基于严格的数学逻辑的运筹学方法相比,是一种全新的智能计算的思维.如本文引言部分所提到的,后来出现的各种进化算法和群体智能算法,包括本文提出的 BOA 算法,都没有脱离这种思维.其他如人工神经网络(artificial neural network)、支持向量机(support vector machine)、小波分析(wavelet transformation)等都另辟蹊径,可以说是对人类思维的突破,因此能够开创一个研究领域,成为人类科学思维和科学方法史上的里程碑.

3.2 搜索过程中始终保持种群多样性

传统上,各种使用精英保留策略的算法多以种群内所有个体之间的距离以某种概率变为 0,也即当种群中所有个体都具有相同的基因型时获得最终解,这样就不可避免地存在着早熟现象.一般来讲,智能算法中的种群多样性是以个体之间的距离(海明距离或欧氏距离)来界定的,因此,算法保持种群多样性就是要保持解空间内有足够多的、相互之间距离不为 0 的个体.

在处理动态问题时,环境变量、约束条件等的改变会引起种群外部环境的变化,有时甚至引起搜索空间的变形(如维数大小的改变),从而问题的最优解不再固定不变.因此,针对这类动态问题的算法,其首要任务是随时侦测到环境的变化并做出快速反应,进而追踪到问题最优解在搜索空间内的漂移轨迹.基于这样的需求,对寻优算法来讲,在搜索过程中如何保持种群的多样性成为核心问题之一.

本文提出的 BOA 算法在运行过程中,设立 $np+1$ 个公告板(np 是生态系统内小生境种群的数目),分别记录各个小生境种群和整个生态系统的最佳个体,即内部标杆和外部标杆,也即问题的局部最优解和全局最优解,并随着迭代过程不断更新.当算法终止时,通过对公告板上的外部标杆进行解码,即可得到问题的全局最优解.可以说,BOA 算法使用了一种保留局部最佳个体和全局最佳个体的策略. BOA 算法在搜索学习的过程中,每个个体的学习欲望和学习强度各不相同,其大小是根据该个体自身的目标值与其学习对象的目标值及两者的距离的大小来决定的,差距愈小,学习欲望愈大,随之其学习强度也愈大.同时,每个个体的学习策略表现为:其根据学习对象(即标杆)的基因表达式来调整自己的基因表达式,因此,个体在进行外部标杆学习后可能缩小与当前生态系统中全局最佳个体的距离,在进行内部标杆学习后又可能缩小与其所属种群中局部最佳个体的距离.这种策略使得生态系统内有前途的个体快速聚集到最优个体的邻域之内,有助于其进行密集搜索,这个自组织过程体现了强者更强、弱者更弱的马太效应(Matthew effect).这种策略是否会使种群的多样性降低呢?答案是:在搜索空间的某一局部区域,即某个小生境种群内,多样性会有一定程度的降低,但这对整个生态系统(即整个解空间)并无太多消极影响,反而有着巨大的积极的影响.因为随着学习过程的进行,取得最优解的个体(包括全局最佳个体和局部最佳个体)也会不断发生改变,因此,生态系统内每个个体的学习对象也在不断变化着,因此不可能出现所有个体呈现相同基因型的情况.而且每个个体在进行自我学习后,会扩大相互之间的距离,因为本算法中的自我学习策略(尤其是当采用浮点数编码方案时的混沌映射)相当于个体重新初始化.这种组织策略,使得 BOA 算法具有天然的保持种群多样性的特性,从而保证了算法的在搜索过程中的探索性(exploration).

限于篇幅,有关动态优化问题的实验没有放入本文内容之中,有关 BOA 算法保持种群多样性能力的实验论

证,将另文阐述.

3.3 搜索过程中的重复性动作少

搜索过程中的重复性动作少,是 BOA 算法的又一个特色.以遗传算法为代表的进化算法在运行过程中会做大量重复性的遗传操作,如某个体可能会与多个个体做交叉运算,但并没有立即返回其对应的目标值,而是要等到完成规定的交叉次数之后才重新评估其目标值.而且,由于随机性选择的影响,在交叉操作完成之后,种群内有的个体很可能并没有参与交叉操作,其基因结构没有发生改变,而有的个体又因多次参与交叉操作,其在交叉过程中得到的优良基因结构可能因重复交叉而遭破坏.在标准遗传算法和当前已经出现的绝大多数改进版本中,3 个核心遗传算子(选择、交叉和变异)都是按顺序执行之后再评估个体的目标值,这实际上就等于在评估目标值之前,个体进行了一连串的重复性基因操作.此外,模拟退火算法在运行过程中也会做大量的试探性动作.粒子群算法虽然在搜索过程中,个体粒子没有重复性动作,但在完成一次漂移后,如果其目标值没有得到改善,则只能等到下一代才有机会修正.而 BOA 算法在搜索过程中,个体的学习策略是选择性的,只有当其执行前一个学习行为而目标值没有改善时才执行下一个学习策略,这样既不会产生大量无用的重复性动作,又有机会在一次搜索行动完成之前来修正基因结构,从而改善目标值,因此,BOA 算法的搜索效率较高,收敛速度较快.

3.4 搜索过程中没有无效性操作

所谓无效性操作,本文特指算法中某些搜索动作,这些搜索动作无益于所求解的优化问题.以决策变量为浮点数类型的函数优化问题为例,如果优化算法采用浮点数编码方案,则当前一些主流算法在搜索过程中会产生相当多的无效性操作.比如对于带边界约束的优化问题,算法的搜索动作常常使得个体飞出解空间之外,即决策变量在经过一系列操作之后往往超出了预先设定的变化范围而变成非法解,于是,这一系列操作变成了无效性操作.PSO,SA 和 AFSA 等都存在这种情况.文献[32,35]对这种现象分别做了理论分析和实验论证.针对这一问题,目前的处理方法一般是:如果检测到越界,则取决策变量的边界值,或对决策变量取模求余,或干脆在决策变量范围内重新取值等等.但这些操作实际上不仅是另一种形式的重复性动作,而且本质上更是一种无效性操作,从而破坏了算法的智能性,无益于对问题本身的求解.本文提出的 BOA 算法就不存在这种无效性操作的问题,因为本算法在运行过程中,个体进行的是以标杆(即最佳个体)为导向的、以减小与标杆的欧氏距离为手段的搜索行动,因为标杆是合法的,因此所有的个体都不可能飞出解空间以外,从而不会产生与 PSO,SA 和 AFSA 等同样性质的无效性操作,这在前面的实验中得到了验证.

4 结论和展望

当前,国际上已经提出了各种各样的智能优化算法,它们之间的本质区别在于各自选取候选的指导思想和执行方法不同.本文提出了一种新型的基于管理学理论的搜索模式,即提出利用标杆管理来获取候选解的理念,简略分析了这种搜索模式本身与现有的模仿生物界规律的搜索模式的区别与优胜之处.

综合上述,BOA 总体结构简单,易于编程,既是一种学习式的搜索策略,也是一种竞争性的寻优方法.当然,任何算法都不是完美无缺的,事实上,本文提出的标杆管理算法在许多方面还可以做进一步研究,比如:(1) 可以引入其他方法来改善种群初始化的效果;(2) 让个体的学习步长也实现自适应动态调整,以提高学习效率;(3) 可以研究除了 0-1 编码与浮点数编码之外,其他编码方式的标杆学习方法;(4) 与其他改进型智能算法在高维和动态优化问题处理上进行比较研究;(5) 对算法的时间复杂度和收敛性等,在数学机理方面进行深入的研究.

References:

- [1] Creutz M. Microcanonical Monte Carlo simulation. *Physical Review Letters*, 1983,50(19):1411-1414. [doi: 10.1103/PhysRevLett.50.1411]
- [2] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report, Report 826, Pasadena: California Institute of Technology, 1989.

- [3] De Castro LN, Nunes L, Von Zuben FJ. Artificial immune systems: Part I—Basic theory and applications. Technical Report, 210, Universidade Estadual de Campinas, 1999.
- [4] de Castro LN, Von Zuben FJ. The clonal selection algorithm with engineering applications. In: Proc. of the GECCO 2000. 2000. 36–39.
- [5] Eusuff MM, Lansey KE, Pasha F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 2006,38(2):129–154.
- [6] Murase H. Finite element inverse analysis using a photosynthetic algorithm. *Computers and Electronics in Agriculture*, 2000,29(1): 115–123.
- [7] Zelinka I. SOMA—Self-Organizing migrating algorithm. In: *New Optimization Techniques in Engineering*. Berlin, Heidelberg: Springer-Verlag, 2004. 167–217.
- [8] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: Harmony search. *Simulation*, 2001,76(2):60–68. [doi: 10.1177/003754970107600201]
- [9] Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 2002,22(3): 52–67. [doi: 10.1109/MCS.2002.1004010]
- [10] Nakrani S, Tovey C. On honey bees and dynamic server allocation in Internet hosting centers. *Adaptive Behavior-Animals, Animats, Software Agent, Robots, Adaptive Systems*, 2004,12(3-4):223–240. [doi: 10.1177/105971230401200308]
- [11] Karaboga D, Basturk B. On the performance of artificial bee colony algorithm. *Applied Soft Computing*, 2008,8(1):687–697. [doi: 10.1016/j.asoc.2007.05.007]
- [12] Yang XS. New enzyme algorithm, tikhonovregulation and inverse parabolic analysis. In: Simons T, Maroulis G, eds. *Advances in Computational Methods in Science and Engineering, Lecture Series on Computer and Computer Sciences (ICCMSE 2005)*. 2005. 1880–1883.
- [13] Krishnanand KN, Ghose D. Detection of multiple source locations using a glowworm metaphor with application to collective robotics. In: Proc. of the 2005 IEEE Swarm Intelligence Symp. (SIS 2005). IEEE, 2005. 84–91. [doi: 10.1109/SIS.2005.1501606]
- [14] Mucherino A, Seref O. Monkey search: A novel meta-heuristic search for global optimization. In: Proc. of the AIP Conf. on Data Mining, System Analysis and Optimization in Biomedicine, Vol.953. American Institute of Physics, 2007. 162–173. [doi: 10.1063/1.2817338]
- [15] Havens TC, Spain CJ, Salmon NG, Keller JM. Roach infestation optimization. In: Proc. of the Swarm Intelligence Symp. (SIS 2008). IEEE, 2008. 1–7. [doi: 10.1109/SIS.2008.4668317]
- [16] Yang XS. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2010.
- [17] Simon D. Biogeography-Based optimization. *IEEE Trans. on Evolutionary Computation*, 2008,6(12):702–713. [doi: 10.1109/TEVC.2008.919004]
- [18] Tero A, Takagi S, Saigusa T, Ito K, Bebbler DP, Fricker MD, *et al.* Rules for biologically inspired adaptive network design. *Science*, 2010,327(5964):439–442. [doi: 10.1126/science.1177894]
- [19] Xie XF, Zhang WJ, Yang ZL. Social cognitive optimization for nonlinear programming problems. In: Proc. of the 1st Int'l Conf. on Machine Learning and Cybernetics, Vol.2. Beijing, 2002. 779–783.
- [20] Li XL. A new intelligent optimization method-artificial fish school algorithm [Ph.D. Thesis]. Hangzhou: Zhejiang University, 2003 (in Chinese). Hangzhou: Zhejiang University, 2003 (in Chinese).
- [21] Zhou YH, Mao ZY. A new search algorithm for global optimization: Population migration algorithm (I). *Journal of South China University of Technology (Natural Science)*, 2003,31(3):1–5 (in Chinese with English abstract).
- [22] Li T, Wang CF, Wang WB, Su WL. A global optimization bionics algorithm for solving integer programming—Plant growth simulation algorithm. *Systems Engineering-Theory & Practice*, 2005,1(1):76–85 (in Chinese with English abstract).
- [23] Zhang Y. Research on optimization searching method of optical principle [Ph.D. Thesis]. Harbin: Harbin Engineering University, 2009 (in Chinese with English abstract).
- [24] Ma HP, Chen ZD, Pan ZX. Evolutionary algorithm based on species migration optimization. *Control and Decision*, 2009,24(1): 1620–1624 (in Chinese with English abstract).
- [25] Xie LP, Zeng JC. Physicomimetics method for global optimization. *Journal of Computer Research and Development*, 2011,48(5): 848–854 (in Chinese with English abstract).

- [26] Cao J, Hou XQ. Explosion search algorithm with neighborhood search mechanism. *Computer Engineering*, 2011,37(18):183–187 (in Chinese with English abstract).
- [27] Tan SH, Yu WY. New algorithm for global optimization: Cell membrane optimization. *Application Research of Computers*, 2011, 28(12):454–457 (in Chinese with English abstract).
- [28] Pan WT. A new evolutionary computation approach: Fruit fly optimization algorithm. In: *Proc. of the 2011 Conf. of Digital Technology and Innovation Management*. 2011. <http://www.oitecshop.byethost16.com/FOA.html>
- [29] Pan WT. A new fruit fly optimization algorithm: Taking the financial distress model as an example. In: *Proc. of the Knowledge-Based Systems, Vol.26*. 2012. 69–74. [doi: 10.1016/j.knosys.2011.07.001]
- [30] Yang S. Non-Stationary problem optimization using the primal-dual genetic algorithm. In: *Proc. of the 2003 Congress on Evolutionary Computation (CEC 2003), Vol.3*. IEEE, 2003. 2246–2253.
- [31] Chi YH, Sun FC, Wang WJ, Yu CM. An improved particle swarm optimization algorithm with search space zoomed factor and attractor. *Chinese Journal of Computers*, 2011,34(1):115–130 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2011.00115]
- [32] Srinivasa KG, Venugopal KR, Patnaik LM. A self-adaptive migration model genetic algorithm for data mining applications. *Information Sciences*, 2007,177(20):4295–4313. [doi: 10.1016/j.ins.2007.05.008]
- [33] Kennedy J, Eberhart RC. A discrete binary of the particle swarm optimization. In: *Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetic*. Piscataway: IEEE Service Center, 1997. 4104–4209.
- [34] Huang ZY, Gao Y. Boolean particle swarm optimization algorithm with disturbance factor. *Computer Engineering and Design*, 2011,32(11):3848–3852 (in Chinese with English abstract).
- [35] Sabine H, Rolf W. Theoretical analysis of initial particle swarm behavior. In: *Proc. of the 10th Int'l Conf. on Parallel Problem Solving from Nature—PPSN X*. Berlin, Heidelberg: Springer-Verlag, 2008. 889–898. [doi: 10.1007/978-3-540-87700-4_88]

附中文参考文献:

- [20] 李晓磊.一种新型的智能优化方法——人工鱼群算法[博士学位论文].杭州:浙江大学,2003.
- [21] 周永华,毛宗源.一种新的全局优化搜索算法——人口迁移算法(I).*华南理工大学学报:自然科学版*,2003,31(3):1–5.
- [22] 李彤,王春峰,王文波,宿伟玲.求解整数规划的一种仿生类全局优化算法——模拟植物生长算法.*系统工程理论与实践*, 2005,1(1):76–85.
- [23] 张毅,沈继红.基于光学原理的最优搜索方法研究[博士学位论文].哈尔滨:哈尔滨工程大学,2009.
- [24] 马海平,陈子栋,潘张鑫.一类基于物种迁移优化的进化算法.*控制与决策*,2009,24(1):1620–1624.
- [25] 谢丽萍,曾建潮.基于拟态物理学方法的全局优化算法.*计算机研究与发展*,2011,48(5):848–854.
- [26] 曹炬,侯学卿.具有邻域搜索机制的爆炸搜索算法.*计算机工程*,2011,37(18):183–187.
- [27] 谭世恒,余卫宇.一种新型的全局优化算法——细胞膜优化算法.*计算机应用研究*,2011,28(12):454–457.
- [31] 迟玉红,孙富春,王维军,喻春明.基于空间缩放和吸引子的粒子群优化算法.*计算机学报*,2011,34(1):115–130. [doi: 10.3724/SP.J.1016.2011.00115]
- [34] 黄志杨,高鹰.带扰动因子的布尔型粒子群优化算法.*计算机工程与设计*,2011,32(11):3848–3852.



谢安世(1983—),男,湖北鄂州人,博士,助理研究员,主要研究领域为智能优化技术,经济增长理论,政府治理研究.

E-mail: shermanxas@163.com



黄思明(1955—),男,博士,研究员,博士生导师,主要研究领域为数学规划,组合优化,投资组合与指数基金,数据挖掘.

E-mail: simhua@casipm.ac.cn



于永达(1952—),男,教授,博士生导师,主要研究领域为优势集聚理论与实务,国际经济战略与政策,国际经济,国际公共组织,领导科学与艺术.

E-mail: yuyongda@tsinghua.edu.cn