

蜂群—蚁群自适应优化算法 *

何宗耀¹, 王 翔²

(1. 河南城建学院 计算机科学与工程系,河南 平顶山 467036; 2. 郑州航空工业管理学院, 郑州 450015)

摘 要: 为了解决蚁群算法在求解连续函数优化问题时,存在局部搜索能力较差的缺陷,提出一种新颖的自适应蜂群—蚁群优化算法。新算法在蚁群优化算法的基础上,设计了一种参数 q 的自适应机制,进而减少了参数个数,提高了其鲁棒性;根据蜂群算法基本思想,利用雇佣蜂和观察蜂设计了高效的局部搜索算子,从而提升了算法的局部能力。针对五个标准测试函数的仿真实验结果表明:与蚁群优化算法相比,新算法的全局和局部寻优能力均得到了极大的提升。

关键词: 优化问题; 蚁群优化; 人工蜂群算法

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0130-05

doi:10.3969/j.issn.1001-3695.2012.01.036

Adaptive bee-ant colony optimization

HE Zong-yao¹, WANG Xiang²

(1. Dept. of Computer Science & Engineering, Henan University of Urban Construction, Pingdingshan Henan 467036, China; 2. Zhengzhou Institute of Aeronautical Industry Management, Zhengzhou 450015, China)

Abstract: This paper proposed a novel colony-ant colony optimization algorithm for continuous function optimization problems. The new algorithm was based on ant colony optimization algorithm. There were two improvements in the new algorithm. Firstly, it devised the adaptive mechanism for parameter q to reduce the parameters' number and improved the robustness of the new algorithm. Secondly, an efficient local search operator, which used employee bees and observed bees in the artificial bee colony, was devised to enhance the local searching capacity. The simulation results for five benchmark functions show that: compared with those of ant colony optimization, the global and local searching capability of colony-ant colony optimization has been greatly improved.

Key words: optimization problem; ant colony optimization(ACO); artificial bee colony algorithm

经过近二十年的发展,蚁群算法成为求解离散组合优化问题最经典的进化算法,其中的典型代表有三种:a) 评级蚂蚁算法(ant system rank, ASrank)^[1]; b) 最大最小蚂蚁算法(max-min ant algorithm, MMAA)^[2]; c) 蚁群算法(ant colony system, ACA)^[3]。

与求解离散组合优化问题相比,蚁群算法求解连续函数优化问题的文献并不多。1995 年 Bilechev 等人^[4]首先提出一种连续蚁群优化(continuous ant colony optimization, CACO),它直接扩展 ACO 算法到连续空间。Monarche 等人在 2000 年以及 Dero 在 2002 年提出了一种连续交互蚁群算法(continuous interacting ant colony, CIAC)^[5,6]求解连续函数优化问题。然而,以上这三种算法并不完全符合蚁群算法的基本概念。为了充分利用蚁群算法的基本思想,Socha 等人^[7]在 2008 年提出了一种基于 R 语言的求解连续函数优化的蚁群算法(ACO_R)。针对多个 benchmark 函数的测试结果表明,在解决连续函数优化问题时,ACO_R 算法是目前最有竞争力的蚁群算法。

ACO_R 算法虽然能够充分利用蚁群算法的优势,但是也存在收敛速度过快、求解精度不高的缺陷。本文在 ACO_R 算法的基础上,提出了一种求解精度较高的改进自适应蜂群—蚁群优

化(adaptive bee colony-ant colony optimization, ABC-ACO)算法。新算法针对 ACO_R 算法中参数 q 的特点,提出了一种自适应设置机制;针对 ACO_R 算法全局搜索能力强、局部搜索能力弱的特点,借鉴蜂群算法的基本思想引入了一种局部搜索算子。

1 蜂群—蚁群自适应算法

为了克服 ACO_R 算法局部搜索能力差的缺陷,本文提出一种新颖的 ABC-ACO 算法。新算法在 ACO_R 算法的基础上引入两种机制,首先借鉴蜂群算法的基本思想引入蜂群局部搜索算子,加强算法的局部搜索能力,同时引入一种参数自适应机制,减少了参数个数。

1.1 新算法的理论基础

1.1.1 高斯核函数

与 ACO_R 算法相同,ABC-ACO 算法的本质仍然属于分布式估计算法,其最重要的理论基础是基于高斯核的概率密度函数,此函数的具体形式如式(1)所示。与一般的高斯函数只能表达一个极值不同,高斯核函数能够同时表达拥有多个极值的复杂分布。

收稿日期: 2011-05-21; 修回日期: 2011-07-04

作者简介:何宗耀(1973-),男,河南平顶山人,副教授,硕士,主要研究方向为计算机教育和智能算法(shanghaiwx@live.cn);王翔(1976-),男,河南洛阳人,讲师,博士,主要研究方向为进化算法。

$$G^i(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^i{}^2}} \quad (1)$$

其中: $G^i(x)$ 表示 ABC-ACO 中第 i 个高斯核函数,每个高斯核函数由 k 个单独的高斯函数组成,其中第 l 个高斯函数又对应三个参数,分别是权重 ω_l 、均值 μ_l^i 、标准差 σ_l^i 。权重 ω_l 表示高斯核函数中第 l 个高斯函数的重要程度。

1.1.2 信息素表示

与 ACO_R 算法相同,ABC-ACO 算法仍然通过一个优势解集合 T 保存进化过程中较优解的信息,间接达到保存信息素的目的。假定要解决问题的维数为 n ,优势解集合元素数为 k ,则整个优势解集合的结构如图 1 所示。

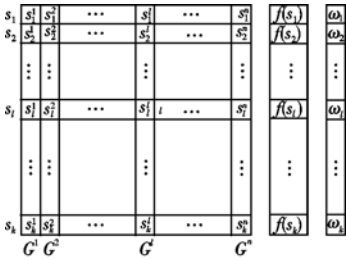


图1 ABC-ACO中优势解集合T

图 1 中: s_l 表示第 l 个解决方案, s_l^i 表示第 l 个解决方案的第 i 个变量; $f(s_l)$ 表示第 l 个解决方案的目标函数; ω_l 表示第 l 个解决方案对应的权重。当解决最小化问题时,图 2 中解决方案按照目标函数的顺序排序 $f(s_1) \leq f(s_2) \leq \dots \leq f(s_k)$,同时由目标函数的顺序可知权重大小顺序为 $\omega_1 \geq \omega_2 \geq \dots \geq \omega_k$ 。

1.1.3 基于信息素的高斯核函数参数

ABC-ACO 中共包含 n 个高斯核函数,它们是形成新解决方案的基础,其中第 i 个高斯核函数的形式如式(1)表示。在式(1)中 $G^i(x)$ 包含权重向量 ω 、均值向量 μ^i 和标准差向量 σ^i 三个参数,它们利用图 2 表示的优势解集合 T ,分别按照式(2)(3)和(5)确定。以下是三个公式的详细解释。

$$\mu^i = \{\mu_1^i, \mu_2^i, \dots, \mu_n^i\} = \{s_1^i, s_2^i, \dots, s_n^i\} \quad (2)$$

其中: μ^i 表示第 i 个高斯核函数 G^i 对应的均值向量,其具体数值是优势解集合 T 中所有 k 个解决方案第 i 个变量的集合。

$$\omega_l = \frac{1}{qk \sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \quad (3)$$

其中: ω_l 表示优势解集合 T 中第 l 个解决方案对应的权重, k 表示优势解集合规模, q 是算法参数。 ω_l 不是取决于解决方案目标函数的数值,而是取决于解决方案按照目标函数排序之后的名次 l 。具体计算时, ω_l 等于参数为 l 、均值为 1 且标准差为 $q \times k$ 的高斯函数数值。

需要特别指出的是式(3)的参数 q ,当 q 较小时,较优的解决方案权重比较差解决方案的权重大很多,则由式(4)可知选中较优解决方案的概率较大;当 q 较大时,所有解决方案的权重基本相同,此时由式(4)可知每个解决方案被选中的概率基本相同。

$$p_l = \frac{\omega_l}{\sum_{r=1}^k \omega_r} \quad (4)$$

其中: p_l 表示第 l 个解决方案被选中的概率。

高斯核函数式(1)中标准差 σ_l^i 的具体形式如下:

$$\sigma_l^i = \xi \sum_{e=1}^k \frac{|s_e^i - s_l^i|}{k-1} \quad (5)$$

其中: σ_l^i 表示 ACO_R 算法的第 i 个高斯核函数 $G^i(x)$ 中第 l 个高斯函数的标准差,参数 ξ 是算法的一个参数。较小的 ξ 使得算法的收敛速度较慢,较大 ξ 使得算法的收敛速度较快。

1.2 新算法的高层框架

为了解决 ABC-ACO 算法的缺陷,在原有框架基础上主要进行了两方面的改进:a) 引入了一个守护进程 daemonActions() 的概念,事实上在通常的 ACO 中都存在这个概念,它的目的是引入局部搜索机制;b) 设计参数 q 的自适应机制,并将其引入原框架的第三步 antBasedSolutionConstruction() 和第四步 pheromoneUpdate() 中。

图 2 是 ABC-ACO 算法的高层框架,在框架中使用斜粗体表示出了与原框架的不同。

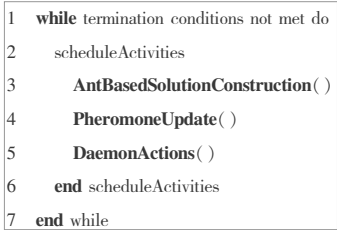


图2 ABC-ACO 算法高层框架

AntBasedSolutionConstruction() 表示蚁群根据归档 T 利用高斯核函数抽样形成新解决方案的过程,其中引入了参数 q 的自适应机制。此函数会在初始化部分根据算法进化过程中以前的信息自适应地生成一个 q 值,即每次调用此函数 q 值都有可能不同。

PheromoneUpdate() 表示利用新产生的解决方案集合更新归档表 T 中的元素,其中也引入了参数 q 的自适应机制。此机制根据元素替换的情况增强或者减弱相应 q 值被选中的概率。

DaemonActions() 函数通常存在于解决离散组合优化问题的 ACO 算法中,目的是用于局部搜索。然而,在解决连续函数优化的 ACO_R 算法却没有这种机制,从而造成了算法局部搜索能力差的缺陷,故改进算法增加此函数完成局部搜索。

1.3 新算法的细节解释

与 ACO_R 算法的核心思想一样,ABC-ACO 算法也是不断利用高斯核函数抽样形成解决方案的一个组件,最终形成新的解决方案。

1.3.1 参数 q 的自适应

文献[7]中 q 的取值范围是 $[0, 1]$,事实上为了保证收敛速度,原文献在实验中取 $q = 1E-4$ 。本文认为 q 的最优取值具有问题敏感性,而且其取值决定了算法的收敛速度以及求解精度。因此本文提出一种自适应机制,通过进化过程的信息保证 q 的取值最优。

该机制基本思想如下:首先根据 q 的范围,设置其最可能取值集合为 q_values ;其次当每次迭代时,利用轮盘赌法从集合 q_values 中选出 q 值;再次,根据选出的 q 值生成新解决方案的情况,增强或者减弱此 q 值被再次选中的可能。

该机制中 q 最可能的取值集合为 $q_values = \{1E-4, 1E-3, 1E-2, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ 。另外设置一个与之对应的集合 $q_probability$ 表示集合 q_values 中每个数值被选中的可能,其中每个元素 $q_probability(i) \geq 1$,初始时

每个元素为 1。参数 q 的自适应机制如图 3 所示。

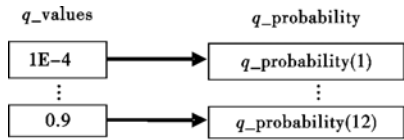


图3 参数 q 的自适应机制

该自适应机制包含两部分内容:a)根据 $q_probability$ 数值,利用轮盘赌法从集合 q_values 中选出 q 值,它被包含在 $AntBasedSolutionConstruction()$ 函数中;b)根据生成的新解对于归档 T 的更新情况,增加或者减少此 q 值被选中的概率,它被包含在 $PheromoneUpdate()$ 中。以下假定求解问题的维数为 n 。

1.3.2 蚁群构建解决方案函数

$AntBasedSolutionConstruction()$ 表示蚁群基于参数 q 的自适应机制,根据归档 T ,利用高斯核函数抽样形成新解决方案的过程。其伪码如下:

- a) 根据集合 $q_probability$,按照轮盘赌法从集合 q_values 选出第 u 个值 $q = q_values(u)$
- b) 按照式(3)形成归档 T 中 k 个解决方案的权重 $\omega = \{\omega_1, \omega_2, \dots, \omega_k\}$
- c) 依据权重 ω ,按照式(4)计算每个解决方案被选中的概率 p
- d) for $j = 1$ to m
- e) 依据概率 p ,按照轮盘赌法选中第 t 个解决方案作为抽样的依据
- f) for $i = 1$ to n
- g) 按照式(2)设置均值 μ_i
- h) 按照式(5)计算方差 σ_i
- i) 从均值 μ_i 方差 σ_i 的正态分布抽样形成解决方案的第 i 个元素 x_i
- j) end
- k) $X_j = (x_1, x_2, \dots, x_n)$ 是抽样形成的第 j 个解决方案
- l) end
- m) return $\{X_1, X_2, \dots, X_m\}$

新的蚁群算法构建函数的主要创新点在于第一步。当蚁群开始构建 m 个新的解决方案之前,该步骤根据进化过程中每个 q 值被选中次数的集合 $q_probability$,利用轮盘赌法从 q_values 集合中选出一个新的 q 值。

1.3.3 信息素更新函数

$PheromoneUpdate$ 表示利用 $AntBasedSolutionConstruction$ 生成的 m 个新解决方案集合,更新归档表 T ,与此同时根据归档表 T 是否被更新,进行参数 q 值的计数工作。 q 值的计数规则有两条,若归档表 T 被更新,则表示生成的新解决方案集合中存在较优秀的个体,从而表示这个 q 值有效,故使得该 q 值对应的计数加 1;依据这个推理,若归档表 T 不被更新,则 q 值对应的计数减 1。

- 信息素更新 $PheromoneUpdate$ 伪码如下所示:
- a) 利用生成新解集 $\{X_1, X_2, \dots, X_m\}$ 更新归档表 T ,同时保持归档表尺寸为 k ,假设共有 i 个元素被替换
 - b) if $i \geq 1$ then
 - c) $/ * u$ 表示上一步中集合 q_values 的第 u 个值被选中 $*/$
 $q_probability(u) = q_probability(u) + 1;$
 - d) else

- e) if $q_probability(u) \geq 1$ then
- f) $q_probability(u) = q_probability(u) - 1;$
- g) end
- h) end

1.3.4 蜂群局部搜索算子

ACO_R 算法全局搜索能力强、局部搜索能力弱,因此本文在新算法中引入局部搜索算子增强其搜索能力。受到人工蜂群算法启发,本文设计的局部搜索算子包含了其中局部搜索能力较强的雇佣蜂和观察蜂相关操作,进而构成了新算法的守护进程 $DaemonActions()$ 。

人工蜂群算法是一种典型的受到蜜蜂采蜜行为启发的智能算法,它在 2005 年由 Karaboga 等人^[8]提出,主要解决无约束优化问题。人工蜂群的基本思想是模拟三种蜜蜂,即雇佣蜂、观察蜂和侦察蜂在食物源周围的运动行为,达到寻优的目的。其中食物源与进化算法中种群个体相对应,同时也与 ACO_R 算法归档 T 中的较优解等价;雇佣蜂在食物源周围进行局部搜索操作;观察蜂针对适应度较高的食物源继续进行局部搜索操作;侦察蜂是在食物源历经 $limit$ 次局部搜索没有提升的情形下,进行一次随机搜索操作。

蜂群算法中雇佣蜂和观察蜂实现局部搜索操作,因此本文将引入守护进程 $DaemonActions()$ 。其基本思想是将 ACO_R 算法归档表 T 中每一个解当做一个食物源,进而利用蜂群算法的雇佣蜂和观察蜂对其进行局部搜索。

- 引入局部搜索的守护进程伪码如下:
- a) 针对归档表 T 中每个元素,进行雇佣蜂局部搜索
 - b) 针对归档表 T 中每个元素,进行观察蜂局部搜索
 - c) return T
- 雇佣蜂和观察蜂的详细操作参见文献[8]。

2 仿真实验及讨论

本章利用 ACO_R 和 ABC-ACO 算法针对五个基本的单模和多模函数进行测试,比较了两算法的求解效果和收敛速度。

2.1 测试函数选择

为了说明改进算法的有效性,本文选择了五个基本的测试函数进行了实验,这五个函数的基本特征如表 1 所示。

表 1 基础的单模和多模测试函数

f	$f(x^*)$	搜索范围	函数形式
sphere	$f(0) = 0$	$[-100, 100]^n$	$f(x) = \sum_{i=1}^n x_i^2$
rosenbrock	$f(1) = 0$	$[-2.048, 2.048]^n$	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$ $f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$
ackley	$f(0) = 0$	$[-32.768, 32.768]^n$	
griewank	$f(0) = 0$	$[-600, 600]^n$	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$
rastrigin	$f(0) = 0$	$[-5.12, 5.12]^n$	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$

表 1 中的五个经典测试函数中,sphere 和 rosenbrock 函数是单模函数,分别用于测试算法的求解精度和收敛速度。函数 arckly、griewank 和 rastrigin 都是多模函数,拥有多个局部极值点,用于测试算法的全局寻优能力。

2.2 算法参数设置

为了比较时的公平性,两算法参数设置基本相同,具体设

置如表 2 所示。

表 2 两算法参数设置			
算法参数	变量	ACO _R	ABC-ACO
蚂蚁数目	m	10	10
收敛速度	ξ	0.85	0.85
搜索过程局部性	q	10^{-4}	—
归档表尺寸	k	50	50
蜜蜂失败上限	limit	—	100

ACO_R 算法与文献[7]中参数设置唯一的不同在于蚂蚁数目 m 从 2 改为 10,这是因为原文献中主要解决测试函数的维数较低,而本文解决测试函数的维数相对较高。

ABC-ACO 与 ACOR 算法相比,由于引入了 q 的自适应机制,因此不需要设置参数 q ;因为改进算法引入了蜂群算法的雇佣蜂和观察蜂的操作,所以同时引入了与这两种蜜蜂相关的参数 limit,它表示两种蜜蜂局部寻优失败的上限。

2.3 五个实验的实验结果

在适应度函数评价次数最大 30 000 次的条件下,两算法针对 30 维的五个测试函数进行了 30 次独立实验。表 3 展示了相应实验的统计结果,括号外是均值、括号内是标准差。

表 3 30 次独立运行实验的均值(标准差)
(30 维,最大适应度函数评价次数 30 000)

Function	ACO _R	ABC-ACO	t -test
sphere	9.48E-39 (2.48E-38)	7.19E-37 (2.91E-36)	0.19
rosenbrock	1.77E+00 (0.00E+00)	1.60E+00 (6.43E-02)	0.00
ackley	1.65E+00 (7.62E-01)	1.50E+00 (6.52E-01)	0.42
griewank	2.48E-02 (2.51E-02)	2.32E-02 (2.67E-02)	0.82
rastrigin	1.20E+02 (0.00E+00)	4.92E+01 (1.60E+01)	0.00

由表 3 的 t 检验可知,两算法在 sphere、ackley 和 griewank 三个测试函数上没有明显差异。ABC-ACO 算法在 rosenbrock 和 rastrigin 函数上明显占优,这说明了改进算法的有效性。

2.4 收敛性分析

为了进一步说明改进算法的优势,利用图 4 展示了两算法的收敛曲线。在图 4 中,横坐标表示适应度函数评价次数 FES,它的上界是 30 000;纵坐标表示 30 次实验适应度函数的均值。这里需要特别指出的是,为了更清晰地显示趋势,本文使用实验适应度函数的均值对数形式 $\log_{10}(\text{FITNESS})$ 取代了适应度函数的均值 FITNESS。

五张图的收敛曲线可以分成三类:第一类是 sphere 和 griewank 函数的收敛曲线;第二类是 ackley 函数的收敛曲线;第三类是 rosenbrock 与 rastrigin 函数的收敛曲线。

第一类中,ABC-ACO 与 ACO_R 算法的收敛曲线基本重合,但是改进算法波动更大。这说明两点:蜂群局部搜索算子没有起到作用;参数 q 的自适应机制造成了收敛曲线的波动。

第二类中,ABC-ACO 比 ACO_R 算法在更小的适应度函数值处收敛。在 ACO_R 算法中 $q = 10^{-4}$,这并不是处理 ackley 函数的最佳参数。与之相比,ABC-ACO 算法由于引入了参数 q 的自适应机制,进而可以自适应地找到最优的 q 值,故在更优的适应度函数处收敛。同时也应该注意到,在适应度函数评价次数 30 000 次的条件下,蜂群局部搜索算子没起到作用。

第三类中,ACO_R 算法出现早熟现象。与之相比,ABC-ACO 算法却能够逃离局部极值,进一步向全局最优解靠近。

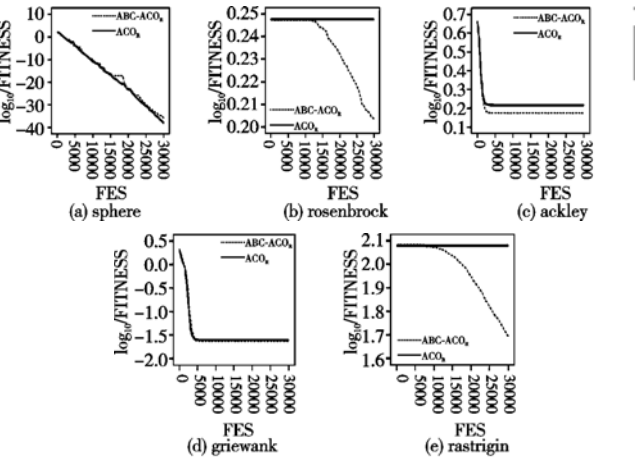


图 4 五个函数的收敛曲线

2.5 讨论

本节从五个实验函数特征角度分析 ABC-ACO 和 ACO_R 算法的优势与不足;从 ABC-ACO 算法创新的角度分析五个函数的实验结果。

2.5.1 ABC-ACO 和 ACO_R 算法优势及不足

2.3 和 2.4 节的实验函数分为单模和多模函数两类。

由图 4 可知,在处理单模函数时,ACO_R 算法在 sphere 函数上收敛速度快、求解效果好,整个收敛曲线都呈现线性下降趋势。然而,该算法求解三个多模函数时,全部快速陷入局部最优解。故可认为 ACO_R 算法不适合解决多模函数。

由图 4 可知,ABC-ACO 算法能够有效解决单模函数 sphere 和 rosenbrock。此外,在解决多模函数时,新算法在 ackley 和 rastrigin 函数上具有明显优势。故可得到如下结论,ABC-ACO 算法能够有效解决单模函数问题,同时在解决多模函数问题时,对 ACO_R 算法也有一定提升。

2.5.2 ABC-ACO 算法的创新对于实验结果的影响

与 ACO_R 算法相比,ABC-ACO 算法的创新点主要体现在两个方面:a)引入参数 q 的自适应机制,自适应调整 q 的数值;b)引入蜂群局部搜索算子,加强算法的局部寻优能力。分别从这两个创新点角度解释表 3 和图 4 中的实验结果:

a)在处理单模函数 rosenbrock 函数时,由表 3 和图 4 可知:ABC-ACO 算法的实验结果在统计意义上显著优于 ACO_R 算法,这种现象是合理的。由于 rosenbrock 函数的最优解位于一个平滑、狭长的抛物线形山谷内,这需要非常精细的局部搜索才能找到全局最优解;与此同时,ACO_R 算法的基本思想是根据整个种群的分布进行抽样形成新解,且抽样只能根据种群分布进行导向型搜索而无法进行精细的搜索,故如图 4(b)所示,在进化后期,ACO_R 算法的整个种群基本都收敛到一个局部极值点。与之相比,ABC-ACO 算法虽然同样无法通过 ACO_R 算法的抽样机制推进整个种群朝着全局最优解方向进化,但是在这个局部极值点周围雇佣蜂和观察蜂机制会推进种群朝着 rosenbrock 函数平滑的山谷底部进化,这种现象在图 4(b)也有显示。综上所述,ABC-ACO 算法处理此类最优解位于平滑曲线尾部的单模函数具有优势。

b)在处理多模函数 ackley 时,由表 3 可知 ABC-ACO 和 ACO_R 算法的实验结果不存在显著性差异,然而由图 4 显示 ABC-ACO 算法的收敛数值要优于 ACO_R 算法。事实上,这是

ABC-ACO 算法中参数 q 自适应机制作用的结果。由图 4 中 ABC-ACO 和 ACO_R 算法关于 ackley 函数的收敛曲线形状相同可知,针对此问题局部搜索机制并不能帮助算法提升效果;同时考虑到新算法是在 ACO_R 算法基础上只引入了两项改进机制(局部搜索机制和参数 q 的自适应机制),故新算法收敛数值优于原算法只能是参数 q 的自适应机制作用的结果。此外文献[7]也指出, ACO_R 算法的参数 q 最优取值与问题相关,这也从一个侧面佐证了本文算法参数 q 自适应机制的必要性。

c)在处理多模函数 rastrigin 时,由表 3 和图 4 可知 ABC-ACO 算法的实验结果在统计意义上显著优于 ACO_R 算法,这种现象也是合理的。Rastrigin 是典型的多模函数,存在多个局部极值,由于 ACO_R 是分布式估计算法,故其很快陷入局部极值,这由图 4 可证明。由图 4 可知,ABC-ACO 算法加入雇佣蜂和侦查蜂机制后,能够将种群从局部极值点拉出,向着更优解方向进化。这证明了局部搜索机制在解决多模函数时的有效性。

3 结 束 语

本文针对连续函数优化问题,提出了一种新颖的蚁群算法。新算法在 ACO_R 算法的基础上进行了两方面的改进:a)根据 ACO_R 算法中参数 q 的特点提出了一种 q 的自适应机制设置机制;b)利用 ACO_R 算法全局搜索强、局部搜索能力差的特点,借鉴蜂群算法的基本思想引入了一种局部搜索机制,进而提升了算法的搜索能力。仿真实验结果表明,新算法能够有效地提升 ACO_R 算法的搜索能力。

(上接第 87 页)

6 结 束 语

本文提出的降低计算复杂度的方法,最大化地去除计算过程中不必计算的冗余,较大降低了计算复杂度。相比传统的穷尽搜索方法,该方法并没有减少搜索相位旋转序列的数目,因此运用该方法降低峰均比可以获得与传统的穷尽搜索方法降低峰均比的效果是相同的,但计算量大大减少,展现了该方法的所在,为实际应用计算提供了参考。

参考文献:

- [1] 周恩,张兴,吕召彪.下一代宽带无线通信 OFDM 与 MIMO 技术[M].北京:人民邮电出版社,2008.
- [2] WANG Y C, LUO Z Q. Optimized iterative clipping and filtering for PAPR reduction of OFDM signals[J]. IEEE Trans on Communications, 2011, 59(1): 33-37.
- [3] FUJII T, NAKAGAWA M. Adaptive clipping level control for OFDM peak power reduction using clipping and filtering[J]. IEICE Trans on Fundamentals of Electronics Communications and Computer Sciences, 2002, E85A(7): 1647-1655.
- [4] JIANG Tao, YANG Yang, SONG Yong-hua. Exponential companding technique for PAPR reduction in OFDM systems[J]. IEEE Trans on Broadcasting, 2005, 51(2): 244-248.
- [5] DAOUD O, ALANI O. Reducing the PAPR by utilisation of the LD-PC code[J]. IET Communications, 2009, 3(4): 520-529.
- [6] AISUSA E, YANG L. Selective post-IFFT amplitude randomising for

参考文献:

- [1] BULLNHEIMER B, HARTL R F, STRAUSS C. A new rank based version of the ant system; a computational study[R]. Vienna: WU Vienna University of Economics and Business, 1997: 25-38.
- [2] STÜTZLE T, HOOS H H. MAX-MIN ant system[J]. Future Generation Computer Systems, 2000, 16(9): 889-914.
- [3] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.
- [4] BILCHEV G, PARMEE I. The ant colony metaphor for searching continuous design spaces[C]//Lecture Notes in Computer Science, vol 993. Berlin: Springer-Verlag, 1995: 25-39.
- [5] MONMARCHÈ N, VENTURINI G, SLIMANE M. On how Pachycondyla apicalis ants suggest a new search algorithm[J]. Future Generation Computer Systems, 2000, 16(9): 937-946.
- [6] DREO J, SIARRY P. A new ant colony algorithm using the heterarchical concept aimed at optimization of multimimima continuous functions[C]//Lecture Notes in Computer Science, vol 2463. Berlin: Springer-Verlag, 2002: 216-221.
- [7] SOCHA K, DORIGO M. Ant colony optimization for continuous domains[J]. European Journal of Operational Research, 2008, 185(3): 1155-1173.
- [8] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.
- [9] peak-to-average power ratio reduction in orthogonal frequency-division multiplexing-based systems[J]. IET Communications, 2008, 2(4): 553-561.
- [7] Le GOFF S Y, AL-SAMAH S S, KHOO B K, et al. Selected mapping without side information for PAPR reduction in OFDM[J]. IEEE Trans on Wireless Communications, 2009, 8(7): 3320-3325.
- [8] HOU Jun, GE Jian-hua, LI Jing. Peak-to-average power ratio reduction of OFDM signals using PTS scheme with low computational complexity[J]. IEEE Trans on Broadcasting, 2011, 57(1): 143-148.
- [9] GHASSEMI A, GULLIVER T A. A low-complexity PTS-based radix FFT method for PAPR reduction in OFDM systems[J]. IEEE Trans on Signal Processing, 2008, 56(3): 1161-1166.
- [10] LIM D W, HEO S J, NO J S, et al. A new PTS OFDM scheme with low complexity for PAPR reduction[J]. IEEE Trans on Broadcasting, 2006, 52(1): 77-82.
- [11] 王灵根,曹叶文.降低 OFDM 系统峰均功率比的低计算复杂度 PTS 方法[J]. 通信学报, 2009, 30(10): 51-57.
- [12] 王进祥,吴新春,毛志刚,等.降低 OFDM 信号 PAPR 的低复杂度 PTS 方法[J]. 西安电子科技大学学报, 2010, 37(2): 326-333.
- [13] WANG L, CAO Y. Sub-optimum PTS for PAPR reduction of OFDM signals[J]. Electronics Letters, 2008, 44(15): 921-922.
- [14] GUEL D, PALICOT J, LOUET Y. Tone reservation technique based on geometric method for orthogonal frequency division multiplexing peak-to-average power ratio reduction[J]. IET Communications, 2010, 4(17): 2065-2073.