

# 随机分形搜索算法

葛钱星,马良,刘勇

(上海理工大学,上海 200093)

**摘要:**现有的元启发式算法大多是模仿生物的群体运动来解决优化问题。为了进一步给优化算法的设计提供新的思路,受自然生长现象的启发,提出了一种新型的元启发式算法—随机分形搜索算法。该算法利用分形的扩散特性进行寻优,其优化原理完全不同于现有的元启发式算法。其中,算法的扩散过程采用高斯随机游走方式来开发问题的搜索空间,而更新过程则分别对个体的分量及个体本身采用相应的更新策略来进行更新,以此进行全局搜索和局部搜索,从而形成了一个完整的优化系统。通过对一系列典型的测试函数优化问题的求解实验并与其他算法进行比较,结果表明随机分形搜索算法不仅具有较高的计算精度,而且具有较快的收敛速度。

**关键词:**随机分形;随机分形搜索算法;扩散;更新;最优化

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2019)04-0001-06

doi:10.3969/j.issn.1673-629X.2019.04.001

## Stochastic Fractal Search Algorithm

GE Qian-xing, MA Liang, LIU Yong

(University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:** Most of the existing meta-heuristics are to imitate the biological group movement to solve the optimization problem. In order to provide a new idea for the design of the optimization algorithm, we propose a new meta-heuristic algorithm, stochastic fractal search (SFS), which is inspired by the natural growth phenomenon. The algorithm uses fractal diffusion to find the optimal, and its optimization principle is quite different from the existing meta-heuristic algorithms. In this algorithm, the diffusion processing adopts the Gauss random walk to exploit the search space of the problem, and the update processing separately updates the individual component and the individual itself with the corresponding update strategy to perform global search and local search, thus forming a complete optimization system. Series of computational experiments on typical benchmark functions are tested and the comparisons with that of other algorithms show that the SFS has both high computational accuracy and faster convergence rate.

**Key words:** random fractal; stochastic fractal search; diffusion; update; optimization

## 0 引言

近年来,元启发式算法取得了巨大的发展,出现了许多有代表性的方法。例如,遗传算法(genetic algorithm, GA)是基于生物进化论中“自然选择、适者生存”规律而提出的优化方法;粒子群优化算法(particle swarm optimization, PSO)是基于鸟群觅食行为规律而提出的群体智能优化方法<sup>[1]</sup>;人工蜂群算法(artificial bee colony, ABC)是基于蜜蜂群觅食行为特性而提出的优化方法<sup>[2]</sup>;蚁群算法(ant colony, AC)是基于蚁群在觅食过程中的行为特性而提出的仿生类算法<sup>[3]</sup>;引

力搜索算法(gravitational search, GSA)是基于万有引力定律而提出的智能优化算法<sup>[4-5]</sup>;布谷鸟搜索算法(cuckoo search, CS)是基于布谷鸟的寄生育雏的行为特性而提出的元启发式算法<sup>[6-7]</sup>。这些方法有助于许多复杂困难优化问题的求解<sup>[8-15]</sup>。但是,随着所研究的优化问题越来越复杂,具有大规模、不可微、非线性、不确定性、多目标等特征,这些算法也相继暴露出一些固有的缺陷,如易陷入局部极值和收敛速度慢等。因此,设计新型的优化算法仍然是值得关注的研究方向。

伊朗德黑兰大学学者 Hamid Salimi 于 2015 年提

收稿日期:2018-06-09

修回日期:2018-10-12

网络出版时间:2018-12-20

基金项目:教育部人文社会科学研究规划基金项目(16YJA630037);上海市“科技创新行动计划”软科学研究重点项目(17692109400, 18692110500);上海高校青年教师培养资助计划项目(ZZsl15018)

作者简介:葛钱星(1992-),女,硕士研究生,研究方向为智能优化、系统工程;马良,教授,博导,研究方向为智能优化、系统工程;刘勇,博士后,讲师,研究方向为智能优化、系统工程。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20181220.1109.072.html>

出了随机分形搜索算法 (stochastic fractal search, SFS)<sup>[16]</sup>, 该算法是基于分形的扩散性质进行优化搜索。其基本原理是, 引入分形的扩散过程作为其搜索机制, 并选择高斯分布作为扩散过程的随机游走方式。然后, 根据各个体的适应度函数值进行选择, 并对各个体的分量和位置进行更新, 最终求得问题的最优解。

Hamid Salimi 利用随机分形搜索算法对一系列标准无约束函数优化问题进行数值实验, 并与回溯搜索优化算法、差分进化算法和引力搜索算法等进行比较, 实验结果表明, 该算法具有较高的计算精度和较快的收敛速度。此外, 还利用随机分形搜索算法求解了部分工程设计优化问题, 包括弹簧设计问题、压力容器设计问题和焊接梁设计问题等, 并与遗传算法、共同进化粒子群优化算法、共同进化差分进化算法等进行比较。实验结果同样表明, 随机分形搜索算法具有较好的优化性能。

## 1 随机分形

随机分形是一种常见的分形, 它们的生成具有很大的随机性, 没有确定的数学法则, 其自相似性不是表现在形态上, 而是表现在结构或复杂度上, 这种相似性是近似的, 具有统计意义上的自相似性。因此, 又称为统计分形<sup>[17]</sup>。

随机分形可以通过随机规则来产生, 如莱维飞行, 高斯游走, 渗透集群, 自回避行走, 分形景观, 布朗运动和布朗树的轨迹 (即通过模拟受限扩散凝聚或受限反应聚集簇产生的树枝状分形)<sup>[18]</sup>。一些随机分形, 例如描述细菌菌落的簇, 可以通过称为“受限扩散凝聚” (diffusion limited aggregation, DLA) 的物理模型而生成<sup>[19]</sup>。为简单起见, 考虑在平面上形成这样的簇, 初始 (种子) 粒子位于原点。随后在原点附近产生其他粒子, 从而引起扩散。为了模拟扩散过程, 采用了随机游走的数学算法。扩散产生的粒子粘附在种子粒子上并重复该过程直到形成簇。在形成簇时, 与穿透内部的粒子相比, 扩散产生的粒子粘附于簇上的概率增加。因此, 这种性质会导致所形成的簇呈分支状结构, 如图 1 所示。

## 2 随机分形搜索算法

受随机分形扩散性质的启发所提出的随机分形搜索算法, 主要包括扩散过程和更新过程。在扩散过程中, 每个个体围绕其当前位置进行扩散, 从而增加了找到全局最优值的机会, 并且可以防止陷入局部最优值。另外, 在随机分形搜索中, 来自扩散过程的最佳生成个体是唯一被保留的个体, 剩下的个体均被丢弃, 这样就有效地避免了扩散过程导致个体数量急剧增加; 在

更新过程中, 群体中一个个体位置的更新是根据其他个体的位置来进行的。



图 1 通过受限扩散凝聚方法产生的一种简单分形

如图 2 所示, 个体扩散导致在其周围产生具有随机不同位置的新的个体。其中,  $E_i$  是进行扩散的原始个体, 扩散阶段完成后,  $E_i$  成为  $E_i^1$ , 而  $E_i^2, E_i^3, \dots, E_i^6$  均为扩散产生的新个体。并且, 扩散阶段的最佳个体, 也是唯一被保留进入下一阶段的个体, 其他个体均被丢弃。

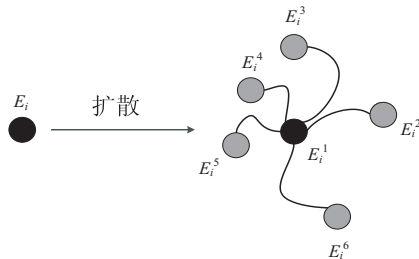


图 2 一个粒子的扩散过程

为了在扩散过程中产生新的个体, 文献[16]研究了莱维飞行和高斯分布两种统计方法<sup>[20]</sup>。对莱维飞行和高斯分布的初步研究显示, 尽管莱维飞行在几代中收敛速度高于高斯游走, 但高斯游走比莱维飞行更有希望找到全局最优解。因此, 选择高斯分布作为随机分形搜索的扩散过程中唯一的随机游走方式。通常, 参与扩散过程的一系列高斯游走如下所示:

$$GW_1 = \text{Gaussian}(\mu_{BP}, \sigma) + (\varepsilon \times BP - \varepsilon' \times P_i) \quad (1)$$

$$GW_2 = \text{Gaussian}(\mu_p, \sigma) \quad (2)$$

其中,  $\varepsilon$  和  $\varepsilon'$  是在区间  $[0, 1]$  上服从均匀分布的随机数;  $BP$  和  $P_i$  分别表示群体中的最佳个体和第  $i$  个个体的位置。式 1 中两个高斯参数是  $\mu_{BP}$  和  $\sigma$ , 其中  $\mu_{BP}$  正好等于  $|BP|$ ; 式 2 中两个高斯参数是  $\mu_p$  和  $\sigma$ , 其中  $\mu_p$  等于  $|P_i|$ 。高斯参数中的标准差  $\sigma$  为:

$$\sigma = \left| \frac{\log(g)}{g} \times (P_i - BP) \right| \quad (3)$$

随着迭代次数的增加, 为了增强个体的局部搜索能力, 并使得解越来越接近最优解, 可使用  $\frac{\log(g)}{g}$  来减小高斯跳跃的步长。假设存在一个  $D$  维的函数优

化问题,因此可设定每个搜索个体为一个  $D$  维向量。在初始化过程中,通过规定最小和最大边界,基于求解问题的约束条件随机初始化种群中的每一个个体。第  $j$  个个体的初始化方程为:

$$P_j = LB + \varepsilon \times (UB - LB) \quad (4)$$

其中,  $LB$  和  $UB$  分别是求解问题的向量的上下边界;  $\varepsilon$  是在区间  $[0, 1]$  上服从均匀分布的随机数。

在初始化所有个体之后,计算每个个体的适应度函数值以获得所有个体中的最佳个体 (BP)。根据扩散过程的开发性质,所有个体都围绕着当前的位置游走以开发问题的搜索空间。另一方面,由于探索属性,考虑了两个旨在进行更好的空间探索的统计过程作为更新过程。第一次更新过程对每个个体的分量索引执行,然后将第二个统计方法应用于所有的个体。

对于第一次更新过程,首先根据适应度函数值对所有的个体进行排序,然后给种群中的每个个体  $i$  设置性能级别  $P_{a_i}$ , 如下:

$$P_{a_i} = \frac{\text{rank}(P_i)}{N} \quad (5)$$

其中,  $\text{rank}(P_i)$  为个体  $P_i$  在种群中的排名;  $N$  为种群中个体的数量。

事实上,式 5 表明点越好,则其被选中的概率越高。该式用于在还未获得良好解决方案的情况下增加改变个体的位置的机会;另一方面,在下一代传递良好解决方案的机会将会增加。对于群体中的每个个体  $P_i$ , 判定条件  $P_{a_i} < \varepsilon$  是否满足,若满足,则根据式 6 更新点  $P_i$  的第  $j$  个分量;否则,保持不变。

$$P'_i(j) = P_r(j) - \varepsilon \times (P_i(j) - P_i(j)) \quad (6)$$

其中,  $P'_i$  是  $P_i$  更新后的位置;  $P_r$  和  $P_i$  是从种群中随机选择的个体;  $\varepsilon$  是在区间  $[0, 1]$  上服从均匀分布的随机数。

对于第一次更新过程,主要是针对个体的分量进行的,而第二次更新过程旨在通过考虑群体中其他个体的位置来改变一个个体的位置。这一特性提高了探索的质量,从而满足了多样性的属性。在开始第二次更新过程之前,从第一次更新过程获得的所有个体需要再次基于式 5 进行排序。类似于第一次更新过程,对于新的个体  $P'_i$ , 判定条件  $P_{a_i} < \varepsilon$  是否满足,若满足,则根据式 7 和式 8 修改  $P'_i$  的当前位置;否则,保持不变。

$$P_i = P'_i - \hat{\varepsilon} \times (P'_i - BP) \mid \varepsilon' \leq 0.5 \quad (7)$$

$$P_i = P'_i + \hat{\varepsilon} \times (P'_i - P'_i) \mid \varepsilon' > 0.5 \quad (8)$$

其中,  $P_r$  和  $P_i$  是从第一次更新阶段获得的个体当中随机选择的个体;  $\hat{\varepsilon}$  是由高斯正态分布生成的随机数;  $\varepsilon'$  是在区间  $[0, 1]$  上服从均匀分布的随机数。若

$P'_i$  对应的适应度函数值优于  $P_i$  对应的适应度函数值,则用  $P'_i$  去替换个体  $P_i$ 。

基于以上分析,随机分形搜索算法的主要步骤可描述如下:

Step1: 设置参数,并初始化种群。

Step2: 计算种群中每个个体  $P_i$  的适应度函数值,并找到最佳点 BP。

Step3: 执行扩散过程。对于每一个进行扩散的个体,根据所选择的高斯游走方式创建各个新个体的位置,并找到所有个体中的最佳个体,进行函数返回。

Step4: 执行第一次更新。对种群中所有的个体进行排序,根据式 5 得到个体  $P_i$  的概率值  $P_{a_i}$ 。判定条件  $P_{a_i} < \varepsilon$  是否满足。若满足,则根据式 6 更新个体  $P_i$  的各个分量  $P_i(j)$ , 得到  $P'_i$ ; 否则,保持不变。

Step5: 执行第二次更新。对第一次更新后种群中所有的个体进行排序,根据式 5 得到个体  $P'_i$  的概率值  $P'_{a_i}$ , 判定条件  $P'_{a_i} < \varepsilon$  是否满足。若满足,则根据式 7 和式 8 更新个体  $P'_i$ , 并且若  $f(P'_i)$  优于  $f(P_i)$ , 则用  $P'_i$  替换  $P_i$ ; 否则,保持不变。

Step6: 判定迭代次数  $G >$  最大迭代次数是否满足,若满足,则算法结束,处理并输出结果;否则,执行 Step3。

### 3 仿真实验及其结果

为验证算法的可行性和有效性,采用 CEC' 2010 标准测试函数库进行仿真实验,并与回溯搜索优化算法 (backtracking search optimization, BSA)<sup>[21]</sup>、协方差矩阵自适应进化策略 (covariance matrix adaptation evolution strategy, CMA-ES)<sup>[22]</sup>、差分进化算法 (differential evolution, DE)<sup>[23]</sup>、引力搜索算法 (gravitational search, GSA)<sup>[4-5]</sup>、人工蜂群算法 (artificial bee colony, ABC)<sup>[2,24]</sup>、动物迁徙优化算法 (animal migration optimization, AMO)<sup>[25]</sup> 进行比较。

对于 CEC' 2010 测试函数,统一设置维度为 30。所有算法的种群规模统一设置为 100,并且每种算法独立运行 25 次。对于随机分形搜索算法,最大扩散数量设置为  $q = 1$ ,其他几种算法的控制参数详见文献 [2, 4, 21-22, 25-26]。仿真结果如表 1 所示,可参考文献 [27] 查看函数详细信息。对于每个基准函数,表 1 给出了各个算法运行 25 次的平均解,并根据平均解从最低到最高对所有的算法进行排名,然后对这二十个函数的排名进行平均化处理,从而得到各算法的综合排名。

根据表 1 的结果可以很容易得出,随机分形搜索算法的性能要明显优于其他算法,在对所有基准函数求解结果的排名中,随机分形搜索算法始终排在第一

名,并且随机分形搜索算法可以找到大多数函数的全局最优解,如函数 $f_{01}^{CEC}$ , $f_{02}^{CEC}$ , $f_{05}^{CEC}$ , $f_{07}^{CEC}$ , $f_{08}^{CEC}$ , $f_{12}^{CEC}$ , $f_{13}^{CEC}$ , $f_{17}^{CEC}$ , $f_{19}^{CEC}$ , $f_{20}^{CEC}$ 。其中,对函数 $f_{04}^{CEC}$ - $f_{13}^{CEC}$ 的求解结果反映了除随机分形搜索算法以外的其他元启发式

算法易陷入局部最小值的缺点。通过分析表 1 可知,就整体优化能力而言,随机分形搜索算法远优于其他算法。

表 1 不同算法求解 CEC'2010 基准函数 $f_{01}^{CEC}$ - $f_{20}^{CEC}$ 的仿真结果比较

函数	算法	BSA	CMA-ES	DE	GSA	ABC	AMO	SFS
$f_{01}^{CEC}$	平均值	4.846 9E-06	6.863 2E+07	8.286 3E-17	7.209 2E+06	4.364 1E-16	0	0
	标准差	3.349 1E-06	3.205 6E+07	9.715 1E-17	1.432 1E+06	9.098 5E-17	0	0
	排名	5	7	3	6	4	1	1
$f_{02}^{CEC}$	平均值	5.963 5	223.547 7	159.794 4	6.666 2	0	6.003 6E-11	0
	标准差	2.408 6	84.819 2	24.546 5	1.628 1	0	1.894 0E-10	0
	排名	4	7	6	5	1	3	1
$f_{03}^{CEC}$	平均值	1.143 8E-09	9.465 5	5.329 1E-15	1.377 1E-09	3.019 8E-14	3.552 7E-15	3.552 7E-15
	标准差	6.019 7E-10	8.390 9	1.872 4E-15	1.686 0E-10	3.452 6E-15	0	0
	排名	5	7	3	6	4	1	1
$f_{04}^{CEC}$	平均值	1.215 6E+09	6.716 0E+09	1.441 3E+08	2.393 9E+10	2.703 2E+11	2.002 4E+10	4.328 1E-20
	标准差	8.709 4E+08	4.588 0E+09	3.266 6E+08	4.730 4E+09	9.993 0E+10	8.650 9E+09	1.671 1E-20
	排名	3	4	2	6	7	5	1
$f_{05}^{CEC}$	平均值	1.112 7E+06	3.220 7E+07	6.965 9E+05	1.874 4E+06	5.708 8E+06	6.799 8E+05	0
	标准差	1.060 4E+06	1.161 9E+07	6.715 5E+05	8.415 6E+05	1.867 1E+06	5.754 4E+05	0
	排名	4	7	3	5	6	2	1
$f_{06}^{CEC}$	平均值	5.437 4E+03	1.586 9E+07	20.857 2	4.134 6E+05	3.333 8E+06	1.966 9E+05	3.553 1E-09
	标准差	5.452 8E+03	1.414 1E+05	0.066 3	5.466 5E+05	6.796 1E+05	3.030 3E+05	1.123 5E-09
	排名	3	7	2	5	6	4	1
$f_{07}^{CEC}$	平均值	894.579 6	1.332 8E+06	1.163 6E-22	4.100 9E+04	6.549 8E+07	0	0
	标准差	1.317 3E+03	1.406 5E+06	1.073 0E-22	3.953 6E+03	4.164 9E+07	0	0
	排名	4	6	3	5	7	1	1
$f_{08}^{CEC}$	平均值	1.234 7E+05	2.135 1E+09	4.531 9E-21	1.847 8E+07	1.682 1E+06	1.380 1E+06	0
	标准差	1.834 3E+05	3.857 7E+09	8.237 4E-21	4.771 1E+07	1.068 2E+06	1.655 0E+06	0
	排名	3	7	2	6	5	4	1
$f_{09}^{CEC}$	平均值	4.568 8E+03	3.701 5E+04	2.094 2E+03	4.928 6E+04	2.931 2E+05	5.034 7E+04	5.510 8E-26
	标准差	2.809 0E+03	2.090 6E+04	1.845 0E+03	1.723 9E+04	6.906 7E+04	2.713 3E+04	1.672 4E-26
	排名	3	4	2	5	7	6	1
$f_{10}^{CEC}$	平均值	27.797 1	82.136 0	166.158 3	37.542 7	29.247 0	29.409 0	14.097 8
	标准差	3.071 3	24.745 4	9.968 5	4.478 6	4.124 2	2.739 5	3.829 7
	排名	2	6	7	5	3	4	1
$f_{11}^{CEC}$	平均值	1.920 8	52.743 0	20.625 7	2.325 4	4.845 3	2.918 7	7.105 4E-15
	标准差	1.140 9	0.178 5	0.115 9	0.990 0	0.809 0	0.400 7	0
	排名	2	7	6	3	5	4	1
$f_{12}^{CEC}$	平均值	0.003 0	2.084 3	4.914 6E-29	4.317 9E-18	14.515 1	0	0
	标准差	0.001 9	1.567 1	3.972 4E-29	8.785 5E-19	6.895 2	0	0
	排名	5	6	3	4	7	1	1
$f_{13}^{CEC}$	平均值	0.225 0	445.859 3	1.196 0	2.004 8E+03	5.136 1	2.653 9	0
	标准差	0.314 0	402.724 7	1.925 7	586.784 1	3.585 0	1.670 2	0
	排名	2	6	3	7	5	4	1
$f_{14}^{CEC}$	平均值	4.239 5E+04	1.743 2E+04	1.853 0E+04	8.206 2E+04	3.158 6E+05	1.151 0E+05	8.862 2E-25
	标准差	1.583 4E+04	1.111 7E+04	7.910 9E+03	4.035 1E+04	1.016 9E+05	4.761 0E+04	2.966 6E-25
	排名	4	2	3	5	7	6	1



续表 1

函数	算法	BSA	CMA-ES	DE	GSA	ABC	AMO	SFS
$f_{15}^{CEC}$	平均值	50.994 9	42.792 1	174.438 0	23.542 1	61.968 9	44.918 4	21.798 7
	标准差	6.279 8	4.962 6	8.765 2	4.187 1	4.283 4	5.382 5	3.865 2
	排名	5		7	2	6	4	1
$f_{16}^{CEC}$	平均值	3.280 6	55.720 7	39.558 1	0.088 9	12.987 2	8.796 9	1.314 5E-14
	标准差	2.103 2	32.919 0	20.753 0	0.012 8	3.105 7	2.055 3	1.716 1E-15
	排名	3	7	6	2	5	4	1
$f_{17}^{CEC}$	平均值	0.018 8	0.002 4	1.435 7E-28	5.446 8E-18	3.836 0	1.514 6E-29	0
	标准差	0.012 8	0.002 0	1.405 6E-28	9.431 1E-19	2.056 4	2.438 8E-29	0
	排名	6	5	3	4	7	2	1
$f_{18}^{CEC}$	平均值	7.550 3	3.678 6E+03	0.398 7	312.973 0	8.117 4	1.659 6	1.972 2E-29
	标准差	2.929 0	5.733 1E+03	1.260 7	705.732 4	3.536 5	3.368 9	0
	排名	4	7	2	6	5	3	1
$f_{19}^{CEC}$	平均值	0.474 9	0.007 4	7.648 0E-26	8.062 6E-18	542.082 4	1.162 0E-27	0
	标准差	0.294 8	0.012 0	1.395 3E-25	2.419 8E-18	259.903 1	5.445 5E-28	0
	排名	6	5	3	4	7	2	1
$f_{20}^{CEC}$	平均值	18.327 8	24.429 6	8.098 6E-26	16.235 6	0.121 2	0.002 4	0
	标准差	19.137 1	1.971 0	8.516 2E-26	0.230 9	0.211 5	0.003 6	0
	排名	6	7	2	5	4	3	1
平均排名		3.950 0	5.850 0	3.550 0	4.800 0	5.400 0	3.200 0	1
综合排名		4	7	3	5	6	2	1

除此之外,还通过实验研究了随机分形搜索算法的搜索行为。为评估随机分形搜索算法的搜索机制,设置了三种不同的搜索机制,分别为无扩散过程的机制,无第一次更新过程的机制,以及无第二次更新过程的机制。并且,在实验中选用来自 CEC’2010 基准函数的函数 $f_{04}^{CEC}$ 、 $f_{05}^{CEC}$ 、 $f_{06}^{CEC}$ 对三种搜索机制进行测试,表 2 给出了实验结果。

由表 2 中得到的结果表明,随机分形搜索算法中的每个过程都能显著影响最终结果的质量,并且随机

分形搜索算法的所有过程共同形成了一个完整的系统来解决优化问题。此外,通过大量的实验表明,对于大多数的函数优化问题,当 MDN (最大扩散数量)=1 时,随机分形搜索算法通常可以找到问题的全局最优解。一般来说,根据所研究的问题调整 MDN 这一参数肯定是有利的,但是其重要性是相对而言的;另一方面,增加 MDN 会面临要进行时间消耗和收敛速度之间的权衡。

表 2 三种搜索机制的仿真测试结果

函数	无扩散过程		无第一次更新过程		无第二次更新过程	
	平均值	标准差	平均值	标准差	平均值	标准差
$f_{04}^{CEC}$	1.064 7E+013	3.509 5E+012	5.203 7E-019	4.853 0E-020	2.231 2E+009	8.918 2E+008
$f_{05}^{CEC}$	2.465 1E+007	1.050 7E+007	144.672 9	27.026 3	2.689 9E+006	5.108 8E+005
$f_{06}^{CEC}$	1.077 9E+007	5.989 8E+005	9.955 5E+005	8.756 2E+005	1.492 5E+006	5.022 2E+005

4 结束语

随机分形搜索算法是一种在精度和时间消耗方面都很成功的新型元启发式算法。该算法基于分形的扩散特性进行优化搜索,为优化算法的设计提供了新思路。为验证算法性能,对 CEC’2010 基准函数进行了仿真实验,并与其他算法进行比较,实验结果表明该算法具有较好的可行性和有效性。基于算法良好的优化性能,其潜在的研究方向包括但不限于以下几点:

(1)算法的理论研究:当前对随机分形搜索算法的研究主要借助于数值实验,缺少严格的数学论证。因此,后续研究可以对此进行展开。针对收敛性的证明可以尝试利用马尔可夫链等进行推导。

(2)算法的设计研究:与大部分仿生智能优化算法不同,随机分形搜索算法是将分形扩散性质融入智能优化算法思想当中的新型算法。要充分吸收分形扩散的研究成果,从而更加有效地设计算法的搜索机制,进一步提高算法的优化性能。其次,还可以利用现有

的优化算法,包括经典优化算法和智能优化算法,结合随机分形搜索算法提出改进的混合优化算法。此外,还可以考虑引入自适应策略进行参数控制等。

(3)算法的应用研究:基于算法良好的优化性能,其应用前景将十分广阔。目前,随机分形搜索算法已经用于求解机械零部件设计等问题,并且取得较好的结果。算法的应用范围还可以进一步推广,例如 TSP 问题、0-1 背包问题和二次分配问题等组合优化问题以及路径规划、智能电网调度优化和应急选址等。

#### 参考文献:

- [1] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the sixth international symposium on micro machine and human science. Nagoya, Japan; IEEE, 1995: 39-43.
- [2] AKAY B, KARABOGA D. A modified artificial bee colony algorithm for real-parameter optimization[J]. Information Sciences, 2012, 192: 120-142.
- [3] 马良, 朱刚, 宁爱兵. 蚁群优化算法[M]. 北京: 科学出版社, 2010: 15-27.
- [4] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitational search algorithm[J]. Information Sciences, 2009, 179: 2232-2248.
- [5] 刘勇, 马良. 引力搜索算法及其应用[M]. 上海: 上海人民出版社, 2014: 10-49.
- [6] YANG X S, DEB S. Engineering optimisation by cuckoo search[J]. International Journal of Mathematical Modelling & Numerical Optimisation, 2010, 1(4): 330-343.
- [7] 兰少峰, 刘升. 布谷鸟搜索算法研究综述[J]. 计算机工程与设计, 2015, 36(4): 1063-1067.
- [8] ABDI M J, SALIMI H. Farsi handwriting recognition with mixture of RBF experts based on particle swarm optimization[J]. International Journal of Information Science and Computer Mathematics, 2010, 2(2): 129-136.
- [9] 杨振宇, 唐珂. 差分进化算法参数控制与适应策略综述[J]. 智能系统学报, 2011, 6(5): 415-423.
- [10] SALIMI H, GIVEKI D, SOLTANSHAHI M A, et al. Extended mixture of MLP experts by hybrid of conjugate gradient method and modified cuckoo search[J]. International Journal of Artificial Intelligence and Applications, 2012, 3(1): 1-12.
- [11] 李煜, 马良. 用量子蚁群算法求解大规模旅行商问题[J]. 上海理工大学学报, 2012, 34(4): 355-358.
- [12] AFLAKPARAST M, SALIMI H, GERAMI A, et al. Cuckoo search epistasis: a new method for exploring significant genetic interactions[J]. Heredity, 2014, 112(6): 666-674.
- [13] 彭晓华, 刘利强. 混沌搜索策略的改进人工蜂群算法[J]. 智能系统学报, 2015, 10(6): 927-933.
- [14] 王皓, 欧阳海滨, 高立群. 一种改进的全局粒子群优化算法[J]. 控制与决策, 2016, 31(7): 1161-1168.
- [15] 申艳光, 张玲玉, 刘永红. 基于混合遗传算法的物流路径优化方法研究[J]. 计算机技术与发展, 2018, 28(3): 192-196.
- [16] SALIMI H. Stochastic fractal search: a powerful metaheuristic algorithm[J]. Knowledge-Based Systems, 2015, 75: 1-18.
- [17] FALCONER K J. Random fractals[J]. Mathematical Proceedings of the Cambridge Philosophical Society, 1986, 100(3): 559-582.
- [18] VICZEK T. Fractal growth phenomena[J]. Computers in Physics, 1992, 3(5): 108.
- [19] WITTEN T A, SANDER L. Diffusion-limited aggregation[J]. Physical Review B, 1983, 27(9): 5686-5697.
- [20] YANG X S, DEB S. Cuckoo search via lévy flights[C]//World congress on nature & biologically inspired computing. Coimbatore, India; IEEE, 2009: 210-214.
- [21] CIVICIOGLU P. Backtracking search optimization algorithm for numerical optimization problems[J]. Applied Mathematics and Computation, 2013, 219(15): 8121-8144.
- [22] HANSEN N, MÜLLER S D, KOUMOUTSAKOS P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMAES)[J]. Evolutionary Computation, 2003, 11(1): 1-18.
- [23] CIVICIOGLU P. Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm[J]. Computers and Geosciences, 2012, 46: 229-247.
- [24] 张泰, 屠思远, 吴滨, 等. 增强寻优能力的自适应人工蜂群算法[J]. 计算机应用研究, 2016, 33(10): 2946-2948.
- [25] LI Xiangtao, ZHANG Jie, YIN Minghao. Animal migration optimization: an optimization algorithm inspired by animal migration behavior[J]. Neural Computing and Applications, 2014, 24(7-8): 1867-1877.
- [26] STORN R, PRICE K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [27] TANG Ke, LI Xiaodong, SUGANTHAN P N, et al. Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization[R]. [s. l.]: [s. n.], 2009.