

Enhanced Fireworks Algorithm with Adaptive Explosion Amplitude and Differential Variation

Wenwen Ye, Jiechang Wen

School of Applied Mathematics, Guangdong University of Technology, Guangzhou Guangdong
Email: yww0314@126.com, wjcpig@126.com

Received: Jan. 18th, 2018; accepted: Feb. 3rd, 2018; published: Feb. 13th, 2018

Abstract

Enhanced fireworks algorithm improves the performance defect of the fireworks algorithm, and shows good effect in many optimization problems. However, it still has low precision and is easy to fall into local optimal solution prematurely in some functions. In order to improve the above mentioned problems, the maximum explosion amplitude is set as the simulated annealing factor to accelerate its search process, and we learn from the idea of differential variation to increase the diversity of the population, so as to jump out of the local optimal solution. The proposed algorithm is tested on 10 benchmark functions and the experimental results show that the proposed algorithm significantly outperforms the enhanced fireworks algorithm and the fireworks algorithm.

Keywords

Enhanced Fireworks Algorithm, Simulated Annealing, Adaptive Explosion Amplitude, Differential Variation

自适应爆炸半径和差分变异的增强烟花算法

叶雯雯, 温洁嫦

广东工业大学应用数学学院, 广东 广州
Email: yww0314@126.com, wjcpig@126.com

收稿日期: 2018年1月18日; 录用日期: 2018年2月3日; 发布日期: 2018年2月13日

摘要

增强烟花算法对烟花算法存在的性能缺陷进行改进, 并在许多优化问题中显示出不错的效果。然而, 增

强烟花算法在某些函数上寻优精度低、且容易过早地陷入局部最优解。为了改善这种缺陷, 将其最大爆炸半径设置为模拟退火因子来自适应地加速其搜索过程, 并借鉴差分变异中的思想以增加种群的多样性, 达到跳出局部最优解的目的。改进后的算法在10个Benchmark函数上进行测试, 实验结果表明改进后的算法效果明显优于增强烟花算法。

关键词

增强烟花算法, 模拟退火, 自适应爆炸半径, 差分变异

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

烟花算法(Fireworks Algorithm, FWA) [1]是近年来新提出的群体智能算法, 其思想来源于生活中的烟花爆炸过程。FWA 同其他群智能算法类似, 产生初始种群后, 经过爆炸、变异、选择等操作产生下一代个体。

烟花算法自 2010 年谭营教授提出来就吸引了越来越多的科研工作者的研究, 其中对传统烟花算法的改进分为两个方面: 一是对烟花算法的四个算子进行分析和改进; 二是和其它进化算法互补结合成混合算法。Pei [2]等人利用烟花算法迭代过程中产生的爆炸火花和高斯变异火花等的种群位置和适应值信息对目标函数的搜索空间形状进行估计, 为算法提供启发式加速信息。Tapas Si [3]等人构造了一种新的计算烟花爆炸数目和幅度的方法, 达到控制烟花爆炸数量、算法的开发性和探索性等效果。Zheng [4]等人针对烟花算法性能上存在的缺陷, 给出了对应的改进方法和策略, 提出了增强烟花算法(Enhanced Fireworks Algorithm, EFWA)。Zheng [5]等人提出动态烟花算法, 即将烟花种群按适应度值的优劣划分成核心烟花和非核心烟花, 并对核心烟花的爆炸半径进行自适应地调整来改善搜索性能。Li [6]等人提出自适应烟花算法, 即一种自适应爆炸半径机制, 能够根据搜索结果进行自适应调节。Yu [7]等人提出带有差分变异算子的新型烟花算法, 以增加种群的多样性。Zheng [8]等人将烟花算法和差分算法结合起来形成一种有效的混合算法, 种群多样性增加达到跳出局部最优的效果。Zheng [9]等人为了克服核心烟花比非核心占有绝对优势以及高斯变异算子不够有效的缺陷, 通过提出合作框架即一种独立选择的方法提高开发能力, 结合避免拥挤的合作策略提高搜索能力。Zhang [10]等人提出基于生物地理学优化的烟花算法, 即在三个方面提高烟花算法的交互能力: 新的变异算子提高烟花的学习能力; 爆炸操作中引入 BBO 的偏移算子, 增强信息共享能力; 新的选择策略使好的个体有更大的几率被选择到下一代。Li [11]等人在烟花算法中引入引导火花(GS), 增强算法信息的利用率, 同时构建一个有方向和有自适应长度的指导向量(GV)形成精英解决方案, 大大加强了烟花算法的探索和开发能力。Gao [12]等人提出文化烟花算法在烟花爆炸中获取相应的知识, 保存在知识库中, 使个体能够学习知识库中的经验。

增强烟花算法是一种已被证明能够解决优化问题的不错算法, 但在某些函数上寻优精度低、且容易过早地寻找到局部最优解, 本文对增强烟花算法进行改进, 提出一种自适应爆炸半径和差分变异的增强烟花算法(SAEFA)。将最大爆炸半径设置为模拟退火因子, 来自适应地调节其搜索过程, 使前期有利于全局搜索而后期有利于局部搜索。并在变异的过程中借鉴差分变异中的思想以增加种群的多样性, 达到跳出局部最优解的目的。

本文的组织结构如下: 第 2 章介绍增强烟花算法; 第 3 章详细讨论最大爆炸半径为模拟退火因子和变异方式为差分变异以及算法的总流程; 第 4 章为实验设置和结果分析; 第 5 章总结全文。

2. 增强烟花算法

假设增强烟花算法[4]所求解的单目标无约束优化问题式子如下:

$$\min f(x) \in R, x \in \Omega \quad (1)$$

其中 Ω 为可行域。假设烟花的个数为 N , 求解的函数为 d 维, 则初始化时在 Ω 内随机产生 N 个烟花作为初始解, 其中 $x_i = (x_i^1, x_i^2, \dots, x_i^d)$ 表示第 i 个烟花的位置。通过对初始烟花进行爆炸操作、变异操作和选择操作得到下一代个体, 直到找到函数的全局最优解。

2.1. 爆炸算子

增强烟花算法产生子代的方式是, 在 N 个烟花附近的规定区域内使用爆炸操作生成新的火花。先确定第 i 个爆炸火花数量 S_i 和爆炸的半径 A_i :

$$S_i = m \frac{y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^N (y_{\max} - f(x_i)) + \varepsilon} \quad (2)$$

$$A_i = A \frac{f(x_i) - y_{\min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - y_{\min}) + \varepsilon} \quad (3)$$

其中 m 和 A 为常数; ε 为一个非常小的数; 若 $f(x_i)$ 表示个体 x_i 的适应函数值; 则:

$$y_{\max} = \max(f(x_i)), (i = 1, 2, \dots, N), \quad y_{\min} = \min(f(x_i)), (i = 1, 2, \dots, N)$$

为了避免爆炸操作产生烟花的数量太多或太少, 限制每个烟花产生的数量:

$$\hat{s}_i = \begin{cases} \text{round}(am), S < am \\ \text{round}(bm), S > bm, a < b < 1 \\ \text{round}(S_i), \text{其它} \end{cases} \quad (4)$$

其中, \hat{s}_i 是第 i 烟花最终产生的烟花个数; $\text{round}()$ 表示四舍五入取整函数; a 和 b 为给定的常数。

为了避免适应度值最小的烟花由于爆炸半径太小而发挥不出挖掘的作用。增强烟花算法引入最小爆炸半径检测策略, 则在维度 k 上烟花 i 的爆炸半径设置为:

$$A_{ik} = \begin{cases} A_{\min,k}, A_{ik} < A_{\min,k} \\ A_{ik}, \text{其它} \end{cases} \quad (5)$$

其中, $A_{\min,k}$ 是在第 k 维上的爆炸半径最低的阈值, 有两种不同的选择策略, 其中非线性递减爆炸半径检测策略为:

$$A_{\min,k}(t) = A_{\min,k} - \frac{A_{\min,k} - A_{\min,k}^{\text{final}}}{\text{evals}_{\max}} \sqrt{(2\text{evals}_{\max} - t)t} \quad (6)$$

烟花的爆炸首先要确定每个烟花爆炸的半径和产生火花的数量。同时在爆炸的过程中, 烟花会进行相应的位移操作。不同于传统烟花算法产生火花时在每个维度具有相等的位移偏移。增强烟花算法在烟花产生火花的过程中在各个维度各个不同位置产生位移偏移来增加种群的多样性。

2.2. 变异操作

为了避免高斯变异时在原点位置附近产生很多火花, 导致最优值为原点的测试函数占优, EFWA 提出的变异操作为在当前解 x_{ik} 和当前种群中最好的个体 x_{Bk} 之间进行变异, 其计算公式为:

$$x_{ik} = x_{ik} + (x_{Bk} - x_{ik}) \times e \quad (7)$$

其中, e 为高斯随机变量, 其均值和方差分别为 0 和 1。

2.3. 映射规则

EFWA 为了避免原始烟花算法造成智能加速收敛的假象, 提出一种随机映射规则, 其公式为:

$$x_{ik} = x_{ib,k} + U(0,1) \cdot (x_{ub,k} - x_{ib,k}) \quad (8)$$

其中, $U(0,1)$ 是在 0 到 1 区间上的均匀分布随机数。

2.4. 选择策略

增强烟花算法提出一种精英-随机选择策略, 即首先从烟花、爆炸火花、高斯变异火花中选择一个适应度值最低的个体, 再从剩下的烟花中随机选择 $N-1$ 个烟花, 共同作为下一代烟花种群的烟花。

3. 自适应爆炸半径和差分变异的增强烟花算法

SAEFA 是在 EFWA 的基础上改进的。它们的不同之处是改进的算法将最大爆炸半径设置为模拟退火因子, 并且借鉴了差分变异的思想将新型高斯变异方式变为差分变异方式。改进的算法由爆炸算子、变异操作和选择操作组成, 同时为了防止越界也使用了映射规则。

3.1. 最大爆炸半径的设置

EFWA 中将最大爆炸半径设置为常数, 这将造成算法后期精细搜索能力被减弱, 使算法的求解精度不高。受到动态搜索思想的启发, 若最大爆炸半径总体上呈现出非线性递减的变化趋势, 算法前期将有利于全局搜索, 后期则有利于局部搜索, 达到自适应加速的效果。

本文将最大爆炸半径设置为模拟退火因子。模拟退火[13]的思想受固体退火过程的启发, 先让固体内部的温度升至充分高, 当温度升高时, 固体内的颗粒会随着温度上升变成无序状态, 内能变大; 再让固体逐渐冷却, 颗粒随着温度降低渐趋有序而达到平衡态, 最终下降到常温时变成基态, 内能降到最小。其算法步骤为:

- Step1: 初始温度 T , 初始解状态 S , 每个 T 值的迭代次数 L ;
- Step2: 对 $k=1, \dots, L$ 执行 Step3 至 Step6;
- Step3: 产生新解 S' ;
- Step4: 计算增量 $\Delta t' = C(S') - C(S)$, 其中 $C(S)$ 为评价函数;
- Step5: 若 $\Delta t' < 0$, 则新解为 S' , 否则以概率 $\exp(-\Delta t'/kT)$ 接受 S' 为新的当前解;
- Step6: 终止条件满足则输出 S' 作为最优解;
- Step7: T 逐渐减少趋于 0, 转至 Step2。

设 R 为初始最大爆炸半径, Gen 为当前迭代次数, $MaxGen$ 为最高的迭代次数, 则第 i 个烟花最大爆炸半径公式为:

$$R_i = R \left(1 - \frac{Gen}{MaxGen} \right)^5 \quad (9)$$

因此最大爆炸半径为模拟退火因子的烟花爆炸半径公式为:

$$A_i = R_i \frac{f(x_i) - y_{\min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - y_{\min}) + \varepsilon} \quad (10)$$

3.2. 变异操作

在 EFWA 中, 变异算子非常重要。通过变异操作, 一个烟花不仅可以在其附近的位置搜索, 甚至可以在其较远的位置产生火花进行搜索。受到差分进化(Differential Evolution, DE) [14]中变异算子的启发, 在 SAEFA 的进化中引入差分向量, 再增加随机扰动, 具体如下。

在子群中随机选择两个解 $x_{j_1}^k$ 和 $x_{j_2}^k$, 其差分向量记为 $x_{j_1}^k - x_{j_2}^k$, 其值越小, 扰动越小, 每次变异进行如下操作:

$$x_i^k = c_1 x_{best}^k + c_2 (x_{j_1}^k - x_{j_2}^k) \quad (11)$$

其中 x_{best}^k 为当前代中适应度值最优的个体, c_1 和 c_2 为缩放因子, 本文实验中取 c_1 为 1, c_2 为 0.03。

3.3. 基于模拟退火和差分变异的搜索策略

SAEFA 沿用了 EFWA 计算爆炸火花的数量公式, 采用同样的爆炸火花产生方式和选择策略, 而采用公式(10)计算爆炸半径和采用公式(11)进行变异操作。SAEFA 的基本流程如下:

- Step1: 初始时随机生成 n 个烟花的位置;
- Step2: 计算所有烟花的适应度值;
- Step3: 循环执行 a 至 e, 直到满足终止条件;
 - a: 使用公式(2)和(4)计算烟花个数;
 - b: 使用公式(10)计算烟花的爆炸半径;
 - c: 生成爆炸火花;
 - d: 使用公式(11)生成变异火花;
 - e: 用选择策略选择 N 个个体进入下一代;
- Step4: 输出最优个体的位置和其适应度值。

4. 实验设置和结果分析

为了比较分析 SAEFA 的性能, 本文将 SAEFA 算法、FWA 算法和 EFWA 算法用于 10 个典型的 Benchmark 函数[15] [16]上进行测试, 并对实验结果进行对比。Benchmark 函数如表 1 所示。由于一些函数的名称太长, 这里用符号表示。其中 $F_1(x)$ 表示 Six-Hump-Camel-back 函数, $F_2(x)$ 表示 Axis parallel-hyper-ellipsoid 函数, $F_3(x)$ 表示 Rotated-hyper-ellipsoid 函数。*代表此函数有好几个最优点。

4.1. 实验设置

由于选取的 Benchmark 函数中的许多函数的全局最优点在原点附近, 烟花算法由于高斯变异在原点附近有很强的搜索能力。SAEFA 因此为测试函数集使用了位置偏移, 不同优化问题的搜索区域位置偏移也不同, 如表 2 所示, 其中 *Max* 与 *Min* 是优化问题的搜索范围。

仿真实验使用的平台是 MATLAB2014a, 为了使对比实验更加有效, 三种算法采用相同的参数设置: 初始时生成 5 个烟花, 种群规模为 50, 变异个数为 5, 函数的评估次数最大设置为 300,000, 在同一条件下独立运行 50 次。

Table 1. Benchmark functions
表 1. Benchmark 函数

函数名	维数	搜索范围	最优点	最优值
Sphere	30	[-100,100]	(0,...,0)	0
Schwefel	30	[-100,100]	(0,...,0)	0
Ackley	30	[-32,32]	(0,...,0)	0
Griewank	30	[-600,600]	(0,...,0)	0
Penalized	30	[-50,50]	(1,...,1)	0
$F_1(x)$	2	[-5,5]	*	-1.0316285
Goldstein-Price	2	[-2,2]	(0,-1)	3
Schaffer	2	[-100,100]	(0,0)	0
$F_2(x)$	30	[-5.12,5.12]	(0,...,0)	0
$F_3(x)$	30	[-65.535,65.536]	(0,...,0)	0

Table 2. The position of the Benchmark functions
表 2. Benchmark 函数的位置偏移

SI (偏移索引)	SV (偏移值大小)
0	0
1	$0.05 \times \frac{Max - Min}{2}$
2	$0.1 \times \frac{Max - Min}{2}$
3	$0.2 \times \frac{Max - Min}{2}$
4	$0.3 \times \frac{Max - Min}{2}$
5	$0.5 \times \frac{Max - Min}{2}$
6	$0.7 \times \frac{Max - Min}{2}$

4.2. 实验结果分析

为了分析 SAEFA 的效果, 在同一条件下, 三个算法对表 1 中的 10 个 Benchmark 函数进行实验仿真, 对每个位置偏移索引, 下面给出函数平均最优值的表格, 如表 3 到表 12 所示。

由表 3 至表 7, 表 11, 表 12 可知, FWA 在无位置偏移的时候均能取得最优值, 但是随着位置发生偏移, 函数的仿真值越来越偏离最优值。而 SAEFA 和 EFWA 在所有测试函数中随着位置的偏移而不会发生太大影响, 均能保持稳定。而在表 4, 表 6 和表 12 中, EFWA 没有找到测试函数的最优解, 进入局部最优。表 8 和表 9 显示 SAEFA、EFWA 和 FWA 三个算法均能够随着位置的偏移保持非常好的性能, 都寻找到 $F_1(x)$ 函数和 Goldstein Price 函数的全局最优值。

Table 3. Comparison of global optimal mean of Sphere function (SI is offset index)

表 3. Sphere 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	0.259016	1.896952	18.041002	43.875453	121.01039	198.58711
EFWA	0.080583	0.078879	0.07936	0.079119	0.0771	0.081641	0.077676
SAEFA	0	0	0	0	0	0	0

Table 4. Comparison of global optimal mean of Schwefel function (SI is offset index)

表 4. Schwefel 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	567.41627	1954.4070	4273.8175	5040.0143	5515.6789	4823.1137
EFWA	0.327639	0.303336	0.318864	0.32688	0.313821	0.316406	0.309674
SAEFA	0.014518	0.013493	0.012076	0.012857	0.012698	0.015377	0.027172

Table 5. Comparison of global optimal mean of Ackley function (SI is offset index)

表 5. Ackley 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	3.667334	3.919792	3.510247	3.815807	16.435764	19.608222
EFWA	10.209394	14.788456	17.501181	19.537629	19.976466	20.00438	20.010942
SAEFA	16.995873	19.043891	18.904865	19.86018	19.95723	19.988987	19.997008

Table 6. Comparison of global optimal mean of Griewank function (SI is offset index)

表 6. Griewank 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	0.034059	0.054588	0.205115	0.683953	0.9717	1.021696
EFWA	0.134513	0.131399	0.13835	0.135506	0.135921	0.132523	0.133103
SAEFA	0.009604	0.007967	0.011242	0.013357	0.012885	0.012676	0.012015

Table 7. Comparison of global optimal mean of Penalized function (SI is offset index)

表 7. Penalized 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0.018924	0.029145	5.502877	157.05845	130.89703	1756.6532	25977.161
EFWA	0.003541	0.002475	0.003581	0.00263	0.002854	0.002545	0.002874
SAEFA	0.007463	0.008789	0.010211	0.008468	0.011909	0.008992	0.009087

Table 8. Comparison of global optimal mean of $F_1(x)$ function (SI is offset index)

表 8. $F_1(x)$ 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628
EFWA	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628
SAEFA	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628

Table 9. Comparison of global optimal mean of Goldstein-Price function (SI is offset index)
表 9. Goldstein-Price 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	3	3	3	3	3	3	30.000037
EFWA	3	3	3	3	3	3	30
SAEFA	3	3	3	3	3	3	31.058824

Table 10. Comparison of global optimal mean of Schaffer function (SI is offset index)
表 10. Schaffer 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	0.000762	0.002286	0.001906	0.00248	0.001905	0.003049
EFWA	0	0	0	0	0	0	0
SAEFA	0.001905	0.001905	0.001524	0.001334	0.001524	0.000572	0.001715

Table 11. Comparison of global optimal mean of $F_2(x)$ function (SI is offset index)

表 11. $F_2(x)$ 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	0.078245	0.308127	1.532319	4.571801	15.182234	55.750759
EFWA	0.003003	0.002899	0.003049	0.002853	0.003034	0.002859	0.002851
SAEFA	0.000002	0.000001	0.000002	0.000002	0.000002	0.000003	0.000004

Table 12. Comparison of global optimal mean of $F_3(x)$ function (SI is offset index)

表 12. $F_3(x)$ 函数全局最优平均值的对比(SI 为偏移索引)

算法	SI = 0	SI = 1	SI = 2	SI = 3	SI = 4	SI = 5	SI = 6
FWA	0	48.232453	316.91266	1260.4623	2487.7016	5142.8280	6808.9586
EFWA	0.477472	0.47585	0.487051	0.479914	0.492438	0.48436	0.486137
SAEFA	0.00026	0.000354	0.000333	0.000292	0.00038	0.000585	0.000574

上图的 10 个小图分别给出了 3 个算法对于 10 个 Benchmark 函数的进化曲线。为了更加清晰地展示各种算法的性能差异, 图像纵坐标为函数适应值差取以 10 为底的对数值。如图 1 所示, SAEFA 在 Sphere 函数中快速找到最优值, 而在 Schwefel 函数中取得比 EFWA 和 FWA 较精确的值; 如图 2 所示, SAEFA 在 Generalized Griewank 函数中取得比 EFWA 和 FWA 较精确的值, 在 Ackley 函数中虽然寻找的最优值与函数实际最优值有些差距, 但是从迭代趋势来看, 如果给出足够的迭代次数, 最终还是可以找到最优值; 如图 3、图 4 和图 5 所示, SAEFA 在 $F_1(x)$ 和 Goldstein 函数中均能够快速找到最优值, 而在 penalized、 $F_2(x)$ 和 $F_3(x)$ 函数取得比 EFWA 和 FWA 较精确的值。总体来看, SAEFA 在 10 个测试函数中均表现出不错的效果。

5. 总结

增强烟花算法对某些函数寻优精度低, 容易过早地陷入局部最优解, 本文提出一种自适应爆炸半径和差分变异的增强烟花算法(SAEFA)。通过将最大爆炸半径设置为模拟退火因子自适应地加速算法搜索

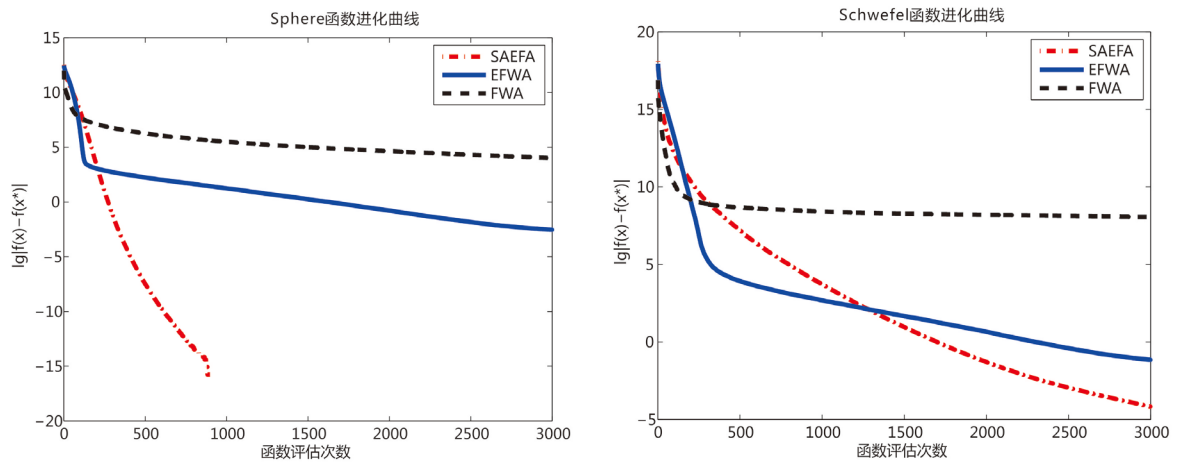


Figure 1. Convergence curves of Sphere and Schwefel functions
图 1. Sphere 和 Schwefel 函数进化曲线

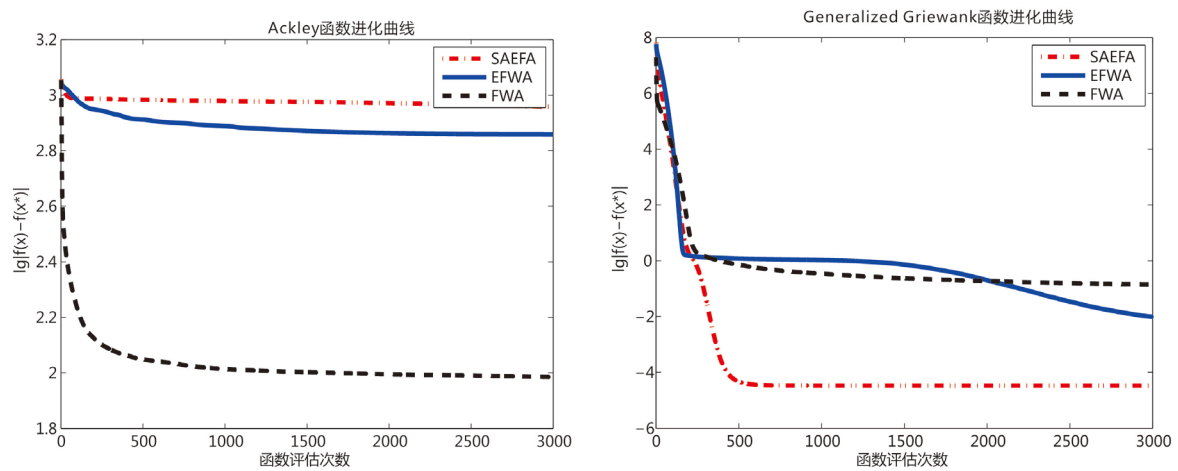


Figure 2. Convergence curves of Ackley and Generalize-Griewank functions
图 2. Ackley 和 Generalize-Griewank 函数进化曲线

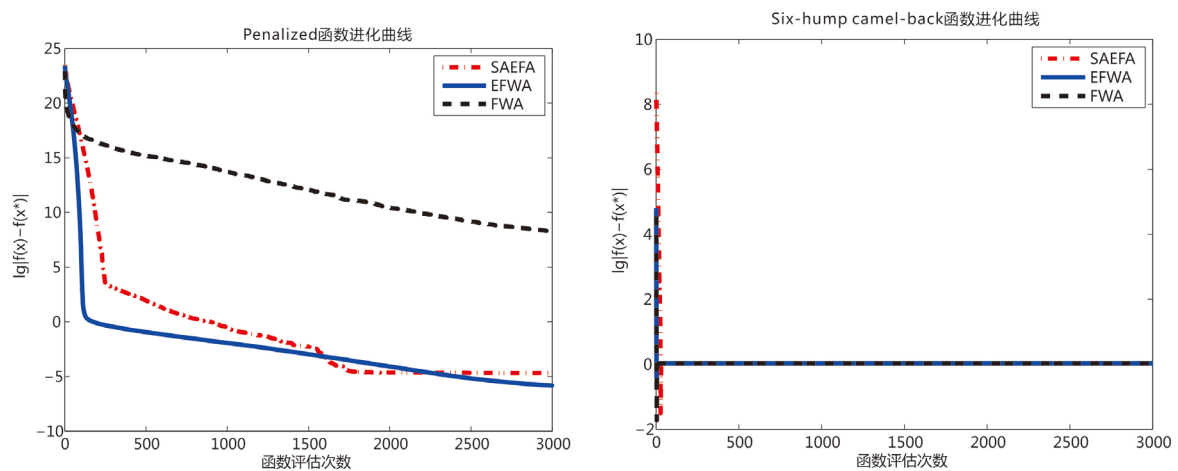


Figure 3. Convergence curves of Penalized and $F_1(x)$ functions
图 3. Penalized 和 $F_1(x)$ 函数进化曲线

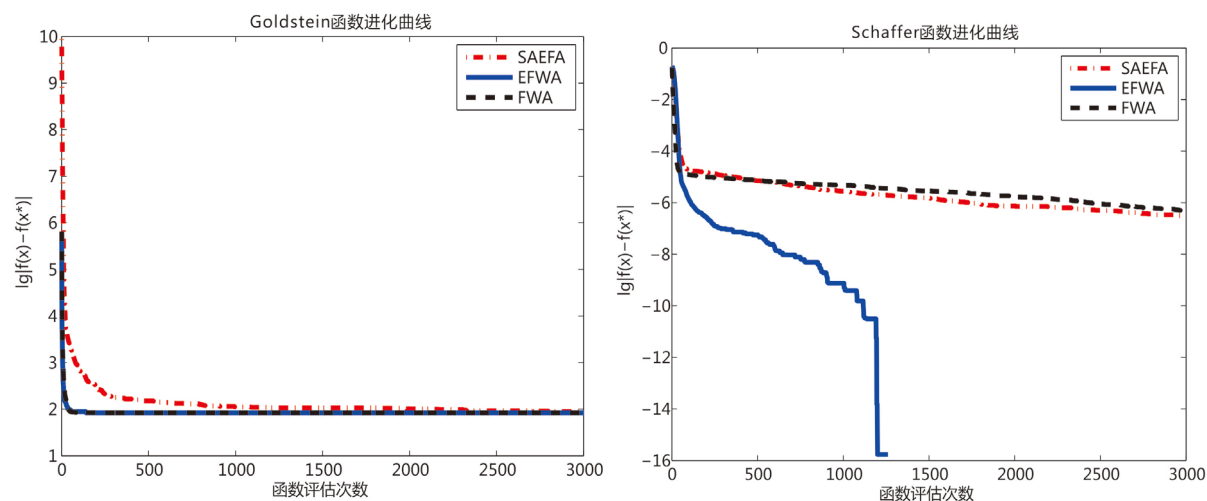


Figure 4. Convergence curves of Goldstein and Schaffer functions

图 4. Goldstein 和 Schaffer 函数进化曲线

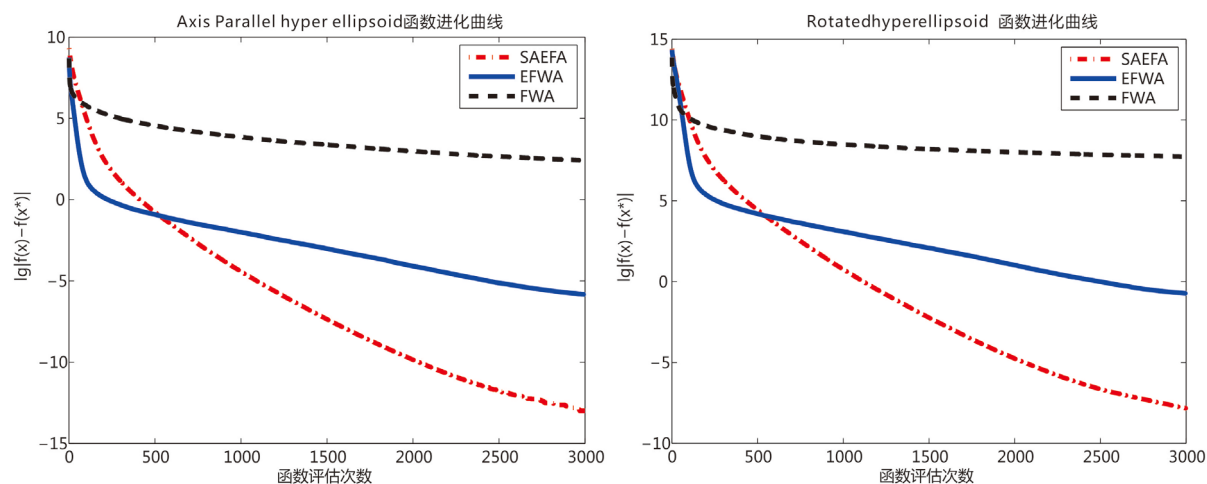


Figure 5. Convergence curves of $F_2(x)$ and $F_3(x)$ functions

图 5. $F_2(x)$ 和 $F_3(x)$ 函数进化曲线

过程, 使算法迭代过程中全局搜索和局部搜索能力之间能够平衡, 提高搜索性能。同时变异操作时借鉴差分变异的思想引入差分变异, 使种群的多样性增加更加有机会跳出局部区域。并在 10 个 Benchmark 函数中进行仿真实验, 与 EFWA 和 FWA 进行对比, 表明 SAEFA 能够随着位置偏移保持良好的性能, 并且取得不错的搜索效果。

致谢

感谢导师温洁嫦老师在写作期间给予我的指导和鼓励, 还有感谢刘海林老师的指导, 感谢同门的陪伴和关怀。

基金项目

广州市科技计划项目(502150155)。

参考文献 (References)

- [1] Tan, Y. and Zhu, Y. (2010) Fireworks Algorithm for Optimization. *International Conference on Advances in Swarm Intelligence*. Springer-Verlag, 355-364.
- [2] Pei, Y., Zheng, S., Tan, Y. and Hideyuki, T. (2012) An Empirical Study on Influence of Approximation Approaches on Enhancing Fireworks Algorithm. 2014 *IEEE International Conference on Systems*, 1322-1327. <https://doi.org/10.1109/ICSMC.2012.6377916>
- [3] Si, T. and Ghosh, R. (2015) Explosion Sparks Generation Using Adaptive Transfer Function in Firework Algorithm. 2015 *IEEE International Conference on Signal Processing, Communication and Networking*, 1-9. <https://doi.org/10.1109/ICSCN.2015.7219917>
- [4] Zheng, S., Janecek, A. and Tan, Y. (2013) Enhanced Fireworks Algorithm. 2013 *IEEE Congress on Evolutionary Computation (CEC)*, 2069-2077. <https://doi.org/10.1109/CEC.2013.6557813>
- [5] Zheng, S.Q., Janecek, A., Li, J.Z. and Tan, Y. (2014) Dynamic Search in Fireworks Algorithm. 2014 *IEEE Congress on Evolutionary Computation (CEC)*, 3222-3229. <https://doi.org/10.1109/CEC.2014.6900485>
- [6] Li, J., Zheng, S. and Tan, Y. (2014) Adaptive Fireworks Algorithm. 2014 *IEEE Congress on Evolutionary Computation (CEC)*, 3214-3221. <https://doi.org/10.1109/CEC.2014.6900418>
- [7] Yu, C., Li, J. and Tan, Y. (2014) Improve Enhanced Fireworks Algorithm with Differential Mutation. *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 264-269.
- [8] Zheng, Y.J., Xu, X.L., Ling, H.F. and Chen, S.Y. (2015) A Hybrid Fireworks Optimization Method with Differential Evolution Operators. *3rd International Conference on Swarm Intelligence (ICSI)*, 75-82
- [9] Zheng, S.Q., Li, J.Z., Janecek, A. and Tan, Y. (2016) A Cooperative Framework for Fireworks Algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1-14.
- [10] Zhang, B., Zheng, Y.J., Zhang, M.X., et al. (2017) Fireworks Algorithm with Enhanced Fireworks Interaction. *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, **14**, 42. <https://doi.org/10.1109/TCBB.2015.2446487>
- [11] Li, J., Zheng, S. and Tan, Y. (2017) The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm. *IEEE Transactions on Evolutionary Computation*, **21**, 153-166. <https://doi.org/10.1109/TEVC.2016.2589821>
- [12] Gao, H. and Diao, M. (2011) Cultural Firework Algorithm and Its Application for Digital Filters Design. *International Journal of Modelling Identification & Control*, **14**, 324-331. <https://doi.org/10.1504/IJMIC.2011.043157>
- [13] 赵晶, 唐焕文, 朱训芝. 模拟退火算法的一种改进及其应用研究[J]. 大连理工大学学报, 2006, 46(5): 775-780.
- [14] Storn, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. <https://doi.org/10.1023/A:1008202821328>
- [15] Liang, J., Nsuganthan, P. and Ghernandez Diaz, A. (2013) Problem Definitions and Evaluation Criteria for the CEC2013 Special Session on Real-Parameter Optimization.
- [16] Liang, J.J., Qu, B.Y. and Nsuganthan, P. (2014) Problem Definitions and Evaluation Criteria for the CEC2013 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2324-7991, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: aam@hanspub.org