

基于种群进化速度的动态烟花算法

杜振鑫

(韩山师范学院 计算机与信息工程学院, 广东 潮州 521041)

摘 要: 为了提高烟花算法的性能, 先分析目标函数中的两种欺骗性进化方向, 接着给出烟花算法的一个改进版本: 基于种群进化速度的动态烟花算法。在改进算法中, 参数 z 根据种群进化速度动态改变。改进算法只改进了烟花算法中参数 z 的更新方式, 没有改变烟花算法的结构, 因此那些基于烟花算法改进的算法, 可以方便的移植到本文的改进算法中。在 9 个标准测试函数上的实验表明, 该改进算法在搜索精度方面超过原始烟花算法。

关键词: 烟花算法; 爆炸半径; 群体智能; 函数优化

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2016)10-0024-04

Dynamic Fireworks Algorithm Based on Population Evolution Velocity

DU Zhen-xin

(School of Computer Information Engineering, Hanshan Normal University, Chaozhou 521041, China)

Abstract: In order to enhance the performance of FA, this paper firstly analysed the two deceptive directions in objective function, then accordingly proposed an enhanced version of FA named Dynamic Fireworks Algorithm based on population evolution velocity (DFA). The parameter z in DFA changes dynamically according to the population evolution velocity. DFA only changed parameter z 's updating strategy without changing the structure of FA, thus the improved algorithm based on FA can be transplanted to DFA conveniently. Experimental evaluation on nine benchmark functions shows that DFA outperforms conventional FA in global solution accuracy capabilities.

Key words: fireworks algorithm; explosion amplitude; swarm intelligence; function optimization

1 引言

烟花算法是在 2010 年提出的一种新的群体智能优化算法^[1], 其优化效率较高。Pholdee 和 Bureera^[2]系统比较了 24 种元启发算法的优化性能, 他们主要是针对具有动态约束的桁架(truss)质量最小化问题, 在不同问题规模的多种情况下仔细地分析比较了这些算法的性能, 给出了客观排名, 其中烟花算法的综合优化能力处于中上游。国内外对烟花算法的研究刚刚逐渐增多。2013 年, 文献[3]分析了烟花算法的一个重要参数爆炸半径对算法的影响, 该文认为: 对于群体中的最优烟花, 因为其爆炸半径趋近于 0, 使得算法中存在冗余计算, 并给出了爆炸半径线性递减的解决方案。随后, Zheng Shao-

qiu^[4]等也指出了爆炸半径的问题并给出了线性递减的方案, 同时改进了爆炸火花产生方式、高斯算子和精英选择策略。Pei Y^[5]等提出了一种基于搜索空间估计法烟花算法, 文献[6]研究了如何更好控制烟花算法中的勘探与开采。文献[7]系统总结了目前烟花算法的最近进展。

2 进化算法的两种搜索策略

进化算法的关键步骤有两步: (1) 按照某种策略搜索到解空间中的某个点 P ; (2) 评价 P 点的函数值 f , 如果 f 比以前搜索到的全局最优值更好, 则算法得到一个正向激励, 相应的调整算法的搜索目标, 向 P 点靠近。但是, 在进化过程中, 进化方向经常会被误导, 即搜到的点可能存在这种情况: 函数值变

好了,其位置却可能变差了.

Rosenbrock 函数是常用的测试函数,其形式如下:

$$f(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad (1)$$

式中, D 是目标函数的维数. 函数在 $(1,1)$ 点取最小值 0. 以该函数为例,来说明两种错误的进化方向:

(1) 每次更新某点的一维或者少数维即评价函数值,如果新的函数值比当前全局最优值更小,则向该点方向进化. 假设在 t 时刻函数的全局最优值的坐标为 $(3,2)$,函数值为 4 904,采用每次更新少数维策略,设在 $t+1$ 时刻该点的第二维发生变化,函数坐标变成 $(3,3)$,函数值更新为 3 604,函数值变小了,这对算法而言是一个正向激励,将促使算法向坐标 $(3,3)$ 移动,然而这个移动方向恰恰是错误的,已知函数最小值点在 $(1,1)$,算法却向着背离最优点的方向进化. 这个例子告诉我们,只更新目标函数的一维或少数维即进行函数评价,依据该评价判断整个函数是否被“进化”,有可能是一种误导,造成算法向错误的方向进化.

(2) 多维或全部维同时进化,得到的函数值作为进化方向. 仍然以 Rosenbrock 函数为例,假设某时刻函数位于 $(5,5)$ 点,函数值是 40 016,另一时刻算法搜索到 $(8,45)$ 点,函数值是 36 149,函数值变小了,这对于进化算法而言,是一个正向激励,鼓励算法向 $(8,45)$ 点进化,然而很明显,该点距离最小值点 $(1,1)$ 更远了,所以这个方向也是错误的. 这说明,函数的多数维同时进化,也会误导算法向错误的方向进化. 显然,两种进化策略各有优缺点. 总体而言,算法应该在初期以全局搜索为主,尽量搜索广大区域,后期应该以局部搜索为主,提高搜索精度. 根据这一思想,本文提出动态更新维数的进化算法,初期以全局搜索为主,更新较多的函数维数以后进行函数评价;后期以局部搜索为主,更新较少的函数维数即进行函数评价,以利于精细调整. 本文提出的改进算法,可以应用于多种进化算法,如粒子群算法,烟花算法等,本文仅以烟花算法为例来验证改进算法的正确性.

3 烟花算法的原理

烟花算法是一种模拟烟花爆炸过程的算法^[1,3]. 烟花爆炸会产生大量火花,散布于局部空间,而火花又可以作为新的烟花继续爆炸,逐步使得烟花(火花)充满整个空间. 烟花算法模拟了现实生

活中的烟花爆炸过程,图 1 给出了烟花算法的流程图.

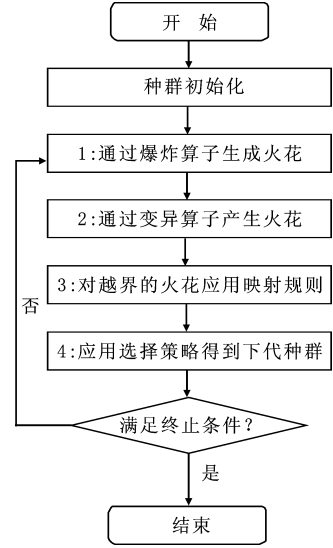


图 1 烟花算法流程图

每个烟花爆炸以后产生的火花总数用下式表示:

$$s_i = m \cdot \frac{y_{\max} - f(x_i) + \xi}{\sum_{i=1}^n (y_{\max} - f(x_i)) + \xi} \quad (2)$$

式中, $i(i = 1, 2, \dots, n)$ 表示烟花的序号, $y_{\max} = \max(f(x_i))(i = 1, 2, \dots, n)$ 表示所有烟花对应的目标函数的最坏值,即最大值. ξ 用于防止上式出现除零错误,一般可取计算机所能表示的最小数值. m 是所有 n 个烟花产生的火花的总数.

烟花 $i(i = 1, 2, \dots, n)$ 的爆炸的半径:

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

式中, $y_{\min} = \min(f(x_i))$ 表示 n 个烟花代表的目标函数的最好值(即最小值). A 是最大爆炸半径, ξ 是计算机所能表示的最小数值.

烟花 i 产生火花的过程可以表述如下:假定待优化的函数是 D 维,现随机取其中的 z 维坐标更新,烟花 $i(i = 1, 2, \dots, n)$ 的第 $j(j = 1, 2, \dots, s_i)$ 个火花的第 $k(k = 1, 2, \dots, z)$ 维坐标按照下式更新:

$$\tilde{x}_{i,k}^j = \tilde{x}_{i,k}^j + A_i \cdot \text{rand}(-1, 1) \quad (4)$$

式中, $\text{rand}(-1, 1)$ 是在区间 $[-1, 1]$ 的随机数. 上面的过程中, n 个烟花产生的火花总数是 m .

为了增加多样性,另外 $\hat{m}$ 个火花按照高斯分布产生,其中第 j 个火花的第 $k(k = 1, 2, \dots, z)$ 维坐标按照下式:

$$\hat{x}_k^j = \hat{x}_k^j \cdot \text{Gaussian}(1, 1) \quad (5)$$

式中, Gaussian(1,1) 表示高斯分布随机数, 均值和方差为 1.

前面的过程共产生的火花数是 $n+m+\hat{m}$ 个, 需从中选择 n 个烟花(火花) 进入下一轮爆炸. 其中, 首先选择最优烟花进入下一代, 剩下的 $n-1$ 个烟花(火花) 按照下面的概率选择:

$$R(x_i) = \sum_{j \in K} d(x_i - x_j) \quad (6)$$

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)} \quad (7)$$

式中, d 表示两个烟花(火花) 之间的距离. 根据上式, 烟花(火花) 被选中的概率与浓度成反比, 有利于种群的多样性.

4 烟花算法的分析与改进

定义 1 烟花算法的一轮进化: 完成图 1 中步骤 1 ~ 4, 则完成烟花算法的一轮进化.

4.1 爆炸半径的改进

式(3) 的主要意图是烟花的爆炸半径与函数值成反比, 这样可以保证好的烟花加强局部搜索. 但是对于最好的一个烟花, 带入公式(3):

$$A_i = \hat{A} \cdot \frac{\xi}{\sum_{i=1}^n (f(x_i) - y_{\min}) + \xi} \quad (8)$$

因为分子 ξ 是计算机表示的最小常数, 趋近于 0, 分母远大于分子, 因此上式在计算机的表示范围内是与 0 相同的, 这明显与算法的最初设计意图不符. 而根据式(2), 最好的一个烟花产生的火花数目最多, 这意味着最好的一个烟花产生的大量火花, 没有进行任何搜索, 白白增加了计算量.

针对这一问题, 文献[3] 给出了线性递减更新最优爆炸半径的方案:

$$A_i = \hat{A} - (\hat{A} - \hat{A}_{\min}) \times t/T \quad (9)$$

$$A_i = A_i/T \quad (10)$$

其中 T 是一个常数, 表示预先设置的烟花算法的总的进化轮数, t 表示当前进化的轮数. 改进算法的效率虽然得到提高, 但是引入了新的参数 T , 需要人工设定.

文献[4] 也给出了类似的改进, 但是引入了更多需要人工设置的参数.

为了简化参数设置, 并达到类似的作用, 本文改用下式更新:

$$A_i = A_i/t$$

式中, t 表示当前算法进化的轮数. 与文献[7-8] 相比, 公式没有引入需要人工设置的参数, 保持了烟

花算法的简单性.

4.2 基于进化速度动态更新维数

根据烟花算法的原理, 烟花算法在进化时, 每轮迭代中都要随机选择 D 维中的 z 维更新. 根据前面的分析, 无论取较大的数值, 还是较小的数值, 都存在缺陷. 本文提出在每轮进化中, 根据算法进化速度动态确定每轮更新维数 z 的策略. 一个直观的想法是, 当种群仍然有较强的进化能力的时候, 说明每轮进化时烟花的更新维数 z 是合适的, 下一轮仍然保持不变, 以利于全局搜索. 反之, 如果烟花种群失去进化能力, 应该适当减少 z , 以增强局部搜索能力. 那么如何衡量烟花种群是否仍然有进化能力呢? 本文引进了烟花进化速度 k 的概念来判断烟花种群是否有进化能力.

定义 2 烟花种群的进化速度 k : 设本轮进化是第 t 轮, 取 $t, t-1, t-2$ 三轮进化中得到的全局最优值进行线性拟合, 得拟合直线为:

$$y = \tilde{k}x + \tilde{b} \quad (12)$$

令: $k = |\tilde{k}|$, 则 k 为烟花种群的进化速度.

在改进算法中, 参数 z 按照式(13) 更新:

$$z = \begin{cases} z & \text{if } k > k_0 \\ z * \mu & \text{if } k < k_0 \end{cases} \quad (13)$$

式中, k_0 是预先设定的一个进化速度阈值. 如果 $k > k_0$ 说明烟花种群仍然有进化能力, 所以保持 z 不变; 如果 $k < k_0$ 说明种群已经失去或者近似失去进化能力, z 按照递减系数 μ 缩减. 式(13) 中, z 的取值范围在 $[1, D]$, 算法初始时, $z = D$. z 的取值根据进化速度 k 逐步动态递减, 直至最小值 1. 递减系数 μ 的取值范围一般可取 $[0.9, 1)$.

5 实验

本文的算法是基于文献[1] 改进的, 为了验证改进算法的有效性, 本文同样在文献[1] 采用的 9 个测试函数上进行了测试. 实验采用的参数与文献[1] 相同. 即 $n = 5, m = 50, a = 0.04, b = 0.8, \hat{A} = 40, \hat{m} = 5$. CPSO、SPSO 的参数设置参照文献[1].

实验给出了四个算法在 9 个测试函数上独立运行 20 次的统计结果, 本文提出的基于进化速度动态更新的烟花算法数据与进行比较 FA、CPSO、SPSO 的数据(来自文献[1]), 可以看出, 本文提出的 DFA 算法在所有测试函数上的表现均好于或等于另外三种算法, 并全部求得测试函数的理论最小值.

从函数评价次数 $F_{es} = 10\ 000$ 时的 20 次独立

运行的统计结果,可以看出,DFA 在较短的时间($Fes=10\ 000$)、在所有测试函数上的结果也都好于或者等于其他算法.除 Rosenbrock 尚未收敛到理论最小值点之外,其余函数都已求得理论最小值,这反映 DFA 的收敛速度较快.在两个很难优化的函数 Rosenbrock 和 Schwefel 上,DFA 比 FA 有显著的优势.

6 结束语

本文基于烟花进化速度改进了烟花算法中每次火花的更新维数 z ,使得算法在接近陷入局部最优的时候,通过动态的调整 z ,增大算法跳出局部极小值的机会.本文提出的改进算法没有改变原始烟花算法的结构,仅改变烟花算法的参数设置方式,因此没有增加原始烟花算法的计算复杂度.所以,对于基于原始烟花算法所做的改进算法,都可以移植到本文改进算法中.实验表明,改进算法提高了搜索精度.本文提出的动态更新维数的思想,也可以推广到其他进化算法中,例如粒子群、蜂群算法,期待更有效的改进.

参考文献:

[1] Tan Y,Zhu Y C. Fireworks algorithms for optimization [C]//Proceedings of Int. Conf. on Swarm Intelligence (ICSI2010), Part II. Beijing, China : LNCS 6145, 2010:355-364.

[2] Eun Hee Lee,Je Hoon Lee,Il Hwan Park,et al. Implementation of high -speed SHA-1 architecture[J]. IE-ICE Electronics Express,2009,16(6):1174-1179.

[3] 黄淳,白国强,陈弘毅. 快速实现 SHA-1 算法的硬件结构[J]. 清华大学学报,2005,45(1):123-125.

[4] SHA1. Secure hash algorithm core[EB/OL]. [2016-02-20]. <http://www.cast-inc.com/ip-cores/encryption/sha-1/index.html>.

[5] Helion Technology. Fast Hash IP cores for ASIC[EB/OL]. [2014-11-07]. http://www.heliontech.com/fast_hash.htm.

[6] William Stallings. 密码编码学与网络安全:原理与实践[M]. 5 版. 北京:电子工业出版社,2013.

[2] Pholdee N,Bureerat S. Comparative performance of meta-heuristic algorithm for mass minimission of trusses with dynamic constraints[J]. Advances in Engineering Software,2014,75(9):1-13.

[3] 杜振鑫. 烟花算法中爆炸半径的改进研究[J]. 计算机时代,2013,30(1):28-29.

[4] Zheng S,Andreas J, Tan Y. Enhanced fireworks algorithm[C]// IEEE International Conference on Evolutionary Computation. Cancun, MEXICO :IEEE,2013:2069-2077.

[5] Pei Y,Zheng S,Tan Y,et al. An empirical study on influence of approximation approaches on enhancing fireworks algorithm[C]// Proceeings of the 2012IEEE congress on System. Man and Cybernetics, Seoul, South Korea;IEEE,2012: 1322-1327.

[6] Tan Y, Shi Y, Mo H. The improvement on controlling exploration and exploitation of firework algorithm [C]// ICSI 2013, Part I, LNCS 7928. Berlin Heidelberg: Springer_Verlag,2013: 11-23.

[7] 谭莹. 烟花算法引论[M]. 北京:科学出版社,2015:3-12.

作者简介:

杜振鑫 男,(1976-),博士研究生,讲师.研究方向为数据挖掘、计算智能. E-mail:gladmail@163.com.

[7] He J,Chen H,Huang H. A compatible SHA series design based on FPGA[C]//Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on. Thailand,IEEE, 2010:380-384.

作者简介:

杜晓婧 女,(1990-),硕士研究生.研究方向为信息安全算法及其大规模集成电路设计与实现. E-mail:duxiaojing1727@163.com.

李树国 男,(1963-),教授,博士生导师.研究方向为信息安全算法及其大规模集成电路设计与实现.

(上接第 23 页)