

一种新的进化算法——种子优化算法*

张晓明^{1,2} 王儒敬¹ 宋良图¹

¹(中国科学院合肥智能机械研究所 智能决策与知识工程实验室 合肥 230031)

²(安徽省仿生感知与先进机器人技术重点实验室 合肥 230031)

摘要 受自然界种子传播方式的启发,提出一种进化算法——种子优化算法.该算法通过模拟植物生存的自适应现象,逐代进化,寻找最优结果,解决复杂的优化计算问题.对该算法的全局寻优性能进行分析证明.通过典型优化问题的实例仿真实验,表明该算法具有较好的寻优性能.

关键词 进化计算,种子优化算法(SOA),全局最优性,粒子群算法(PSO)

中图分类号 TP 181

A Novel Evolutionary Algorithm——Seed Optimization Algorithm

ZHANG Xiao-Ming^{1,2}, WANG Ru-Jing¹, SONG Liang-Tu¹

(Laboratory of Intelligent Decision and Knowledge Engineering, Institute of Intelligent Machines,
Chinese Academy of Sciences, Hefei 230031)

(Anhui Key Laboratory of Biomimetic Sensing and Advanced Robot Technology, Hefei 230031)

ABSTRACT

Inspired by the transmission of seeds in nature, an evolutionary algorithm, seed optimization algorithm (SOA), is proposed. The algorithm is designed by simulating the self-adaptive phenomena of plant and it can be used to resolve complex optimization problems with the evolution of plant. The global convergence analysis of SOA is made by using the Solis and Wets' research results. Finally, SOA is applied to three function optimization problems and compared with particle swarm optimization (PSO) algorithm. The experimental results show that SOA has stable and robust behaviour and it can be used as a promising alternative to existing optimization methods for engineering design.

Key Words Evolutionary Computation, Seed Optimization Algorithm (SOA), Global Optimality, Particle Swarm Optimization (PSO)

1 引言

从 20 世纪中期开始,生物模拟就构成了计算机

科学和人工智能的一个重要组成部分.自然界生物体通过自身演化和自然选择这种非定向机制往往能使问题得到完善的解决^[1].目前,人们开发出多种

* 国家自然科学基金资助项目(No. 60774096, 60472111)

收稿日期:2007-06-12;修回日期:2008-02-26

作者简介 张晓明,男,1979 年生,研究实习员,主要研究方向为智能计算、智能决策支持系统. E-mail: xmzhang@ustc.edu. 王儒敬,男,1964 年生,研究员,主要研究方向为智能决策、知识工程. 宋良图,男,1963 年生,研究员,主要研究方向为专家系统、复杂系统.

仿生进化算法,比如:借鉴自然界遗传机制的遗传算法^[1]、受鸟群觅食启发的粒子群优化算法^[2]、参照蚂蚁群体协作的蚁群算法^[3]、基于自然免疫系统中体液免疫反应的机制提出的免疫算法^[4]、针对动物行为建模的自由搜索算法^[5]、受人类社会舆论导向启发的人类进化模型算法^[6],等. 这些算法能够解决复杂和病态问题,许多改进类型已成功应用于优化、控制、机器学习等工程问题^[7-9].

2 种子优化算法

2.1 算法思想

众所周知,植物种子的传播方式有很多种,其中包括散射传播,比如大部分的豆科植物就采用这种传播方式. 当豆子成熟后,干燥而坚硬的外皮在太阳的照射下,发生爆裂,种子就会被弹射到植株周围,大部分种子会分布到植株的附近区域,也有个别会飞出很远或因为其他原因远离植株. 然后种子就会在所降落的区域发育成长. 有的会变得很茁壮,生出更多的种子,说明其所在地块很肥沃;有的可能很快被自然淘汰,说明其所在区域不适合植株生长. 久而久之,会出现大量植株聚集在肥沃的地块,而贫瘠之地则不会有植株. 本文受此启发,将待优化问题的问题域看作是土地,目标点所在的区域是最肥沃之地,根据目标函数值来确定该区域的肥沃程度,即越是背离最优目标值,其所代表的土地就越贫瘠,否则就越肥沃. 一包种子被随机撒到土地上面,如果种子落到肥沃区域,其成长的概率和繁衍后代的机会就会很大;否则,就很可能被淘汰. 经过多代繁衍,最终会有一颗植株生长在最富饶的土地上. 经过算法抽象,将演化过程大大压缩,表现为在某几个强大植株附近大量涌现后代植株,如此循环往复,因此本文称该算法为种子优化算法(Seed Optimization Algorithm, SOA). 该算法与粒子群优化算法等智能算法相比,主要区别在个体的移动方式不一样,SOA 中的个体对父种的响应程度更高,搜索效率也更高.

2.2 算法模型与算法实现

本算法中,种子个体用实值向量 $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ 来表示, n 的大小由问题本身决定. 大量种子组成种子群体,它用 sum 表示,大小由初始化设定. 种子被随机播撒到问题空间,其中适应度最大的几个种子称为父种,根据父种适应度的大小决定其后代的大小和分布情况. 后代种子的分布以父种周边为主,父种的适应度值越大,其后代种子的数量就越多;否则,其后代种子的数量就减少,而且种子的分

布更具有随机性. 后代种子基本播撒方程如下所示:

$$X[i] = \begin{cases} X[i], & \text{若 } X[i] \text{ 为父种} \\ X_{mb} + rand() \cdot Y(i), & \text{若 } X[i] \text{ 不为父种} \end{cases}$$

其中, $X[i]$ 是种子 i , X_{mb} 是种子 i 的父种, $Y(i)$ 是父种 X_{mb} 的传播范围向量函数,根据问题的情况进行定义和调整,表示如下:

$$Y(i) = \begin{cases} \alpha, & \text{若 } i \leq (sum \cdot a\%) \\ \beta, & \text{若 } (sum \cdot a\%) \leq i \leq (sum \cdot b\%) \\ \gamma, & \text{其他情况} \end{cases}$$

其中, $\alpha, \beta, \dots, \gamma$ 都是代表位置变动的常数向量,可根据实际问题进行设置, $a\%, b\% \dots$ 是指父种周围种子的分布比例,同样可根据实际问题进行设置. 其示意图如图 1 所示.

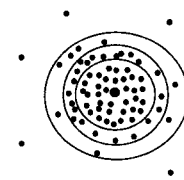


图 1 种子分布示意图

Fig. 1 Sketchmap for seed distribution

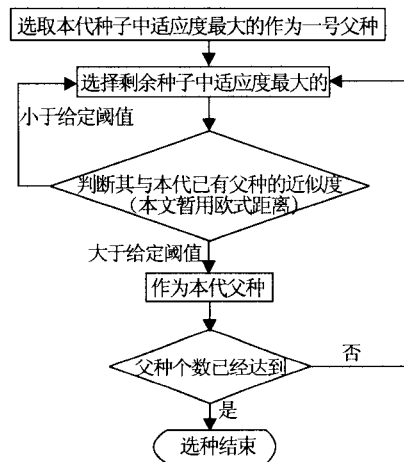


图 2 父种选择流程图

Fig. 2 Flow chart for father seed selection

另外,部分种子的分布不遵循该方程,而是根据父种的分布,选择父种之间的中间位置进行定位,用于增强算法的全局寻优性能.

后代种子生成后,对所有种子的适应度再进行评价,选择适应度最优的种子作为父种候选. 然后计算候选父种与同代其他父种的欧式距离是否满足设定的阈值,主要用来保证父种在空间上有较为合理的分布,避免算法过早收敛,提高算法的搜索效率和

全局寻优性能. 最后决定该种子是否作为本代的父种. 父种选择流程图如图 2 所示.

根据种子的进化方程, 在生成的每个父种的传播范围内生成相应的后代种子群体. 整个种子群体以此方式循环进化, 直到得到理想的优化结果或一号父种(每代中的最优种子)不再变化为止. 算法流程图如图 3 所示.

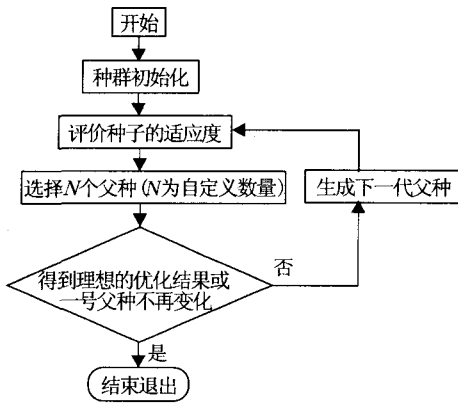


图 3 种子优化算法流程图

Fig. 3 Flow chart for seed optimization algorithm

3 算法分析

SOA 算法是一种随机优化算法, 关于随机优化算法以概率 1 收敛于全局最优解的条件, Solis 和 Wets 对其进行了证明^[10-11], 其主要结论如下.

假设 1^[10] 若 $f(D(z, \zeta)) \leq f(z), \zeta \in S$, 则 $f(D(z, \zeta)) \leq f(\zeta)$,

其中, D 为产生问题解的函数, ζ 为从概率空间 (\mathbf{R}^n, B, μ_k) 产生的随机向量, f 为目标函数, S 为 \mathbf{R}^n 的子集, 表示问题的约束空间, μ_k 为 B 上的概率度量, B 为 \mathbf{R}^n 子集的 σ 域.

假设 2^[10] 若对 S 的任意 Borel 子集 A , 有 $\nu(A) > 0$, 则

$$\prod_{k=0}^{\infty} (1 - \mu_k(A)) = 0,$$

其中, $\nu(A)$ 为子集 A 的 n 维闭包, $\mu_k(A)$ 为由 μ_k 产生 A 的概率.

定理^[10] 设 f 为一可测函数, S 为 \mathbf{R}^n 的一可测子集, $\{z_k\}_0^{\infty}$ 为随机算法产生的解序列, 则当满足假设 1 和假设 2 时, 有

$$\lim_{k \rightarrow \infty} P[z_k \in R_e] = 1,$$

其中 R_e 为全局最优点的集合.

由上述结论可知: 对于 SOA 算法, 只要能够满

足假设 1 与假设 2, 根据定理就可保证其以概率 1 收敛于全局最优解. 下面是我们对 SOA 算法的寻优分析.

在 SOA 算法中, 其返回的结果是第 t 代前的一号父种的位置 $X_g, X(t)$ 为第 t 代的某个种子的位置, $f(X)$ 是种子的适应度评价函数. 因此, 我们令假设 1 中的函数 D 为

$$D(X_g, X(t)) = \begin{cases} X_g, & f(X_g) \leq f(X(t)) \\ X(t), & f(X_g) > f(X(t)) \end{cases}$$

可以看出, 假设 1 是满足的.

对于假设 2, 只须证明规模为 S 的种子群体的样本空间的并集包含 S , 即

$$\bigcup_{i=1}^S M_{i,t} \supseteq S,$$

其中 $M_{i,t}$ 是 t 代时种子 i 的样本空间的支撑集.

设有 N 个父种, 其中 i 号父种的传播范围为 S_i , S_i 也就是 i 号父种后代种子的播撒范围, 即其支撑集. 所以种子群体的样本空间的并集为 $\bigcup_{i=1}^n S_i$.

由于父种的传播范围 S_i 是可以调节的, 当其覆盖范围是从其所在位置到问题域的边界时, 尽管边界所在环层的种子数很少, 但是却使得 $S_i = S$, 且搜索效率没有明显降低. 所以 $S = \bigcup_{i=1}^n S_i$. 假设 2 成立.

综上所述, 由定理可得 SOA 算法以概率 1 收敛于全局最优解.

4 实验结果与分析

4.1 标准函数介绍

实验函数采用 Rosenbrock、Rastrigin 和 Shaffer's F6 这 3 个标准测试函数. 这 3 个函数的优化求解问题都是 GA- 难度大的问题, 用它们来验证和比较算法的性能是比较合适的^[12].

1) Rosenbrock 函数的一般形式为

$$\min f(X) = \sum_{i=1}^{n-1} [100 \times (x_{i+1} - x_i^2) + (x_i - 1)^2],$$

$$x_i \in [-10, 10],$$

该函数具有一个全局极小点 $A = (1, 1, 1, \dots, 1)$, 最小值 $f(A) = 0$.

2) Rastrigin 函数的一般形式为

$$\min f(X) = \sum_{i=1}^n [x_i^2 - 10 \times \cos(2\pi x_i) + 10],$$

$$x_i \in [-5.12, 5.12],$$

该函数具有一个全局极小点 $A = (0, 0, 0, \dots, 0)$, 最小值 $f(A) = 0$.

3) Shaffer's F6 函数的一般形式为

$$\min f(x,y) = 0.5 + \frac{(\sin \sqrt{x^2 + y^2}) - 0.5}{(1.0 + 0.001(x^2 + y^2))^2},$$

$$x_i \in [-100, 100],$$

该函数具有一个全局极小点 $A = (0, 0)$, 最小值 $f(A) = 0$.

4.2 实验结果

算法中参数的设定采用手工调整方式, 以 Rastrigin 函数优化为例, 设定没带父种个数为 4, 父种的孩子数量分别设定为总数的 40%、30%、20%、10%; 以一号父种为例, 其孩子位置设置如下 ($\alpha, \beta, \eta, \gamma$ 的数量比例分别为 0.375、0.25、0.25、0.125):
 $\alpha = 1$ 号父种位置 + $0.0625 \times (2 \times \text{radom}(1) - 1)$,
 $\beta = 1$ 号父种位置 + $0.125 \times (2 \times \text{radom}(1) - 1) \times (\max - \min)$,
 $\eta = 1$ 号父种位置 + $0.25 \times (2 \times \text{radom}(1) - 1) \times (\max - \min)$,
 $\gamma = 1$ 号父种位置 + $0.5 \times (2 \times \text{radom}(1) - 1) \times (\max - \min)$.

PSORTP 表示带逆向思维粒子的 PSO 算法^[12]. 逆反比例指逆向思维粒子数在总群体中占的比例. Rosenbrock 函数和 Rastrigin 函数都是 30 维的, Rastrigin* 和 Shaffer's F6 是 2 维的. 针对 Rosenbrock 函数, PSO 中粒子的最大速度为 0.5, 粒子的初始位置域为 $[-10, 10]$. 针对 Rastrigin 函数, PSO 中粒子的最大速度为 0.5, 粒子的初始位置域为 $[-5.12, 5.12]$. 针对 Shaffer's F6 函数, PSO 中粒子的最大速度为 2.0, 粒子的初始位置域为 $[-100, 100]$.

结果的得出是针对每个函数每种算法各运行 50 次, 对它们取平均值. 具体结果见表 1.

表 1 实验结果

Table 1 Experimental results

| 函数名 | 方法 | 种子群体大小 | 进化代数 | 逆反比例 | 结果 |
|--------------|--------|--------|------|------|----------|
| Rosenbrock | PSO | 100 | 2000 | 0 | 28.87100 |
| | PSORTP | 100 | 2000 | 30% | 26.83900 |
| | SOA | 100 | 500 | 0 | 28.05600 |
| Rastrigin | PSO | 100 | 2000 | 0 | 32.47200 |
| | PSORTP | 100 | 2000 | 30% | 23.45600 |
| | SOA | 100 | 250 | 0 | 22.30100 |
| Rastrigin* | SOA | 100 | 15 | 0 | 0.00006 |
| | PSO | 100 | 2000 | 0 | 0.00012 |
| Shaffer's F6 | PSORTP | 100 | 2000 | 30% | 0.00000 |
| | SOA | 100 | 250 | 0 | 0.00000 |

本文实验中, SOA 求解 Rosenbrock 的最好结果是 20.696 027 132 395 486, 对应解为

- (1.0184607339306928, 1.0108132249017285, 0.9574780864453820, 0.9684803971250635, 0.9915819143318187, 0.9837862472093577, 0.9803886738311668, 0.9615056919806388, 0.9563634417494656, 0.9914332164344220, 0.9963984528123195, 1.0085983322660765, 0.9556990525848712, 0.8952534806662037, 0.7982357882424403, 0.7562201000234869, 0.8816340430466887, 0.9775939522254888, 0.9402642563148313, 0.9776336219711101, 1.0503389703178410, 1.0102839548255058, 1.0148857343606017, 0.9958210972173571, 0.9806207445536670, 0.9835426188752939, 0.9647751231812469, 0.9472537557683613, 0.9384226548583678, 0.9340397521948733).

SOA 求解 Rastrigin 的最好结果是 11.243 118 663 429 788, 对应解为

- (0.02141286694932882, 0.00837279196763443, 0.00995790097837716, 0.93470939555588120, 0.01448973370820166, 1.02159134951663440, -0.02782698169588149, 0.01558382225104612, -0.01970378708484297, 0.01779436386575305, 0.00683666105406823, -0.00020216164771703, -0.03475919700012636, 0.05538035732664422, 0.03252136215063677, -0.00062559580999301, 0.11362036932505770, 0.04370396161714160, 0.98652306946717530, 0.02265166956666811, -0.01922011329715650, 0.98159750296656600, -0.00132587742586738, 0.01021147216751365, 0.01318970647130352, 0.02122130004533483, 0.00445167869047168, -0.93841459058821460, -0.00236991069745793, 0.01058704069272726).

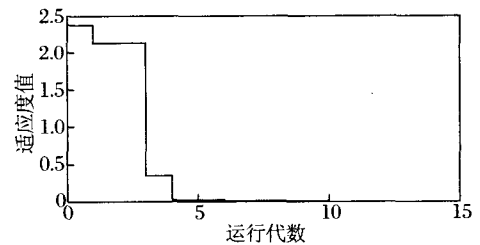


图 4 SOA 寻优收敛梯状图(2 维 Rastrigin)

Fig. 4 Stair chart for SOA optimization (2-dimension Rastrigin)

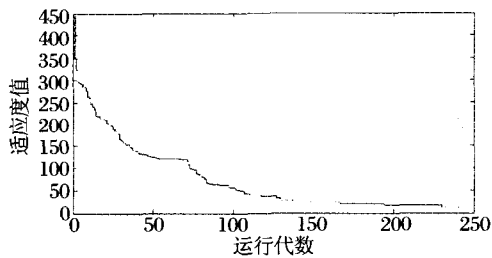


图 5 SOA 寻优收敛梯状图(30 维 Rastrigin)

Fig.5 Stair chart for SOA optimization (30-dimension Rastrigin)

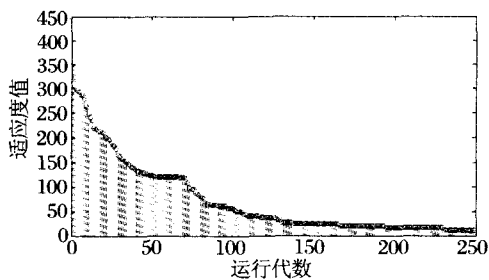


图 6 SOA 寻优收敛杆状图(30 维 Rastrigin)

Fig.6 Stem chart for SOA optimization (30-dimension Rastrigin)

以 Rastrigin 为例,分别绘制 SOA 优化 2 维 Rastrigin(图 4)和 30 维 Rastrigin(图 5、图 6)的收敛图。

4.3 结果比较分析

由上面的结果可看出,SOA 算法的性能总体上好于 PSO 系列算法.从数值优化结果来看,除了 Rosenbrock 函数中,SOA 结果逊于 PSORTP 算法外,其他的函数优化结果均好于 PSO 系列算法,Rastrigin 函数的结果尤其明显.在运行效率方面,SOA 优势更加明显,优化速度明显优于 PSO 系列算法.但是,SOA 的易用性不如 PSO 算法,对于不同的数值函数,都要对种子的播撒进行设置或调整,包括父种数量的确定、每个父种的孩子的数量、每个父种的孩子的分布等.通过实验发现:父种数量越多,算法的全局寻优性能越好,但是算法的计算复杂度越高.否则,如果父种数量过少,算法则很容易陷入局部最优的陷阱.每个父种的孩子的数量和分布也要遵循适度的原则,既要保证分配的覆盖范围,也要保证分配的重点突出.本文实验由于采用的是手工设置算法参数,所以无法保证参数设置是最优的,今后将对参数的自动调整进行研究,争取得到更好的结果.

5 结束语

本文提出一种进化算法——种子优化算法,并对其全局寻优性能进行分析证明,针对数值优化的实验结果也表明了该算法具有较好的寻优性能,在解的质量,尤其是求解效率上优于 PSO 系列算法.另外由于该算法刚刚开始研究,存在着参数设置较复杂,易用性相对较差的缺点.今后,对该算法的研究将重点针对父种传播范围和后代种子的群体分布,并尽快将其应用于智能决策支持系统中.

参 考 文 献

- [1] Chen Guoliang, Wang Xufa, Zhuang Zhenquan, *et al.* Genetic Algorithm and Its Application. Beijing, China: Posts & Telecommunications Press, 1996 (in Chinese)
(陈国良,王煦法,庄镇泉,等.遗传算法及其应用.北京:人民邮电出版社,1996)
- [2] Kennedy J, Eberhart R C. Particle Swarm Optimization // Proc of the IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942 - 1948
- [3] Ant Colony Optimization [DB/OL]. [2007 - 05 - 08]. <http://www.aco-metaheuristic.org>
- [4] Wang Lei, Pan Jin, Jiao Licheng. The Immune Algorithm. Acta Electronica Sinica, 2000, 28(7): 74 - 78 (in Chinese)
(王磊,潘进,焦李成.免疫算法.电子学报,2000,28(7): 74 - 78)
- [5] Penev K, Littlefair G. Free Search: A Comparative Analysis. Information Sciences, 2005, 172(1/2): 173 - 193
- [6] Montiel O, Castillo O, Melin P, *et al.* Human Evolutionary Model: A New Approach to Optimization. Information Sciences, 2007, 177(10): 2075 - 2098
- [7] Fan S K S, Zahara E. A Hybrid Simplex Search and Particle Swarm Optimization for Unconstrained Optimization. European Journal of Operational Research, 2007, 181(2): 527 - 548
- [8] Mukherjee V, Ghoshal S P. Intelligent Particle Swarm Optimized Fuzzy PID Controller for AVR System. Electric Power Systems Research, 2007, 77(12): 1689 - 1698
- [9] Zahiri S H, Seyedin S A. Swarm Intelligence Based Classifiers. Journal of the Franklin Institute, 2007, 344(5): 362 - 376
- [10] Solis F J, Wets R T B. Minimization by Random Search Techniques. Mathematics of Operations Research, 1981, 6(1): 19 - 30
- [11] Cui Zhihua, Zeng Jianchao. A Guaranteed Global Convergence Particle Swarm Optimizer // Proc of the 4th International Conference on Rough Sets and Current Trends in Computing. Uppsala, Sweden, 2004: 762 - 767
- [12] Zhang Xiaoming, Wang Rujing. Particle Swarm Optimization with Reverse Thinking Particles. Computer Science, 2006, 33(10): 156 - 159 (in Chinese)
(张晓明,王儒敬.一种带逆反的粒子群算法.计算机科学,2006,33(10): 156 - 159)