

随机聚焦搜索优化算法

郑永康¹, 陈维荣², 戴朝华², 王维博³

(1. 成都电业局修试所, 四川 成都 610021; 2. 西南交通大学 电气工程学院, 四川 成都 610031;

3. 西南交通大学 信息科学与技术学院, 四川 成都 610031)

摘要: 提出了一种新的优化算法: 随机聚焦搜索. 该算法属于群集智能, 它模仿了人类的搜索行为及其在搜索过程中的随机性, 算法简单并且计算复杂度小. 在对一系列典型复杂函数的优化测试中, 通过与差分进化算法和全面学习的粒子群算法进行对比, 验证了该算法性能. 仿真结果表明, 该算法能解决大多数benchmark函数问题, 并且有较快的寻优速度, 可以在一定程度上替代现有的优化算法.

关键词: 群集智能; 随机聚焦搜索; 人类随机搜索; 粒子群优化

中图分类号: TP18 **文献标识码:** A

Optimization algorithm with stochastic focusing search

ZHENG Yong-kang¹, CHEN Wei-rong², DAI Chao-hua², WANG Wei-bo³

(1. Test & Maintenance Department of Chengdu Electric Power Bureau, Chengdu Sichuan 610021, China;

2. School of Electrical Engineering, Southwest Jiaotong University, Chengdu Sichuan 610031, China;

3. School of Information Science and Technology, Southwest Jiaotong University, Chengdu Sichuan 610031, China)

Abstract: A novel optimization algorithm with stochastic focusing search(SFS) is proposed. This algorithm is a swarm-intelligence algorithm, which imitates the random action in human searching behaviors. The algorithm performance is studied by using a set of typical complex functions, and is compared with that of the differential evolution(DE) algorithm and the comprehensive learning-particle-swarm-optimizer(CLPSO) algorithm. The simulation results show that SFS solves most of the benchmark problems and can be considered a promising candidate of search algorithms when the existing algorithms have difficulties in solving some problems.

Key words: swarm intelligence; stochastic focusing search; human randomized searching; particle swarm optimization

1 引言(Introduction)

群集智能是指受到动物群体行为启发的用于解决分布式问题的算法或者设备^[1]. 这样的群体比传统的基于协商机制和中心控制的系统增加了: 鲁棒性、适应性、可量测性等. 许多群集智能算法在实际应用中已经显示出很好的效果, 但在可靠性和计算速度方面还存在一定的问题^[2].

人类生产生活过程中经常会遇到优化问题, 解决这些问题是人类的一个基本行为^[3]. 本文根据人类启发式随机搜索行为, 提出了一种优化算法: 随机聚焦搜索(stochastic focusing search, SFS).

2 SFS算法(SFS algorithm)

本文提出的SFS算法模拟了人基于自身记忆、经验、不确定性推理并相互交流的搜索行为. 在SFS中, 所有个体立足于自身的最好位置(个体极值), 随机

向群体中到目前为止获得最好搜索效果个体(全局极值)的一个邻域点搜索.

在SFS中, 每个搜索个体是作为 M 维搜索空间(解空间)中的一个点. 设 s 为群体大小, 则每个个体 i ($1 \leq i \leq s$)有如下属性: 第 t 次迭代(搜索)时个体在搜索空间中的位置 $\vec{x}_i(t) = [x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iM}]$ (j 表示变量 $\vec{x}_i(t)$ 的第 j 维分量, M 是维数); 所有个体到目前为止获得的全局极值的位置 \vec{g}_{best} ; 个体的速度 $\vec{v}_i(t) = [v_{i1}, v_{i2}, \dots, v_{ij}, \dots, v_{iM}]$. 假设待解决的优化问题为极小值问题, SFS中个体的位置按如下公式更新:

$$\vec{v}_i(t) = \begin{cases} \text{Rand}(\cdot) \times (R_t - \vec{x}_i(t-1)), \\ \quad \text{fun}(\vec{x}_i(t-1)) \geq \text{fun}(\vec{x}_i(t-2)), \\ \vec{v}_i(t-1), \\ \quad \text{fun}(\vec{x}_i(t-1)) < \text{fun}(\vec{x}_i(t-2)). \end{cases} \quad (1)$$

$$\vec{x}_i(t) = \vec{v}_i(t) + \vec{x}_i(t-1), \quad (2)$$

$$\vec{x}_i(t) = \vec{x}_i(t-1), (\vec{x}_i(t)) \geq \text{fun}(\vec{x}_i(t-1)). \quad (3)$$

其中: $\text{fun}(\vec{x}_i(t))$ 是个体*i*在第*t*次迭代时的搜索效果(目标函数值), R_t 是 \vec{g}_{best} 邻域空间*R*中随机选取的一个点.

$$R \in \left[\vec{g}_{\text{best}} - \frac{w(\vec{g}_{\text{best}} - \vec{x}_{\min})}{(\vec{x}_{\max} - \vec{x}_{\min})^{1-w}}, \vec{g}_{\text{best}} + \frac{w(\vec{x}_{\min} - \vec{g}_{\text{best}})}{(\vec{x}_{\max} - \vec{x}_{\min})^{1-w}} \right].$$

其中: \vec{x}_{\max} 和 \vec{x}_{\min} 是搜索空间的边界, w 是惯性权值. 当 w 从1逐渐减小为0时, R 就从整个解空间收敛到点 \vec{g}_{best} .

公式(1)的第1部分相当于PSO的“社会认知”部分, 也表示个体间的信息共享与相互合作; 公式(2)计算个体在当代更新后所在的位置; 公式(3)相当于PSO的“自我认知”部分.

从SFS算法的表达形式看, 它聚焦于全局最好位置点 \vec{g}_{best} , 在该点的一个动态收敛的邻域内随机进行搜索, 因此将其称为随机聚焦搜索优化算法.

R 的变化受到参数 w 的控制, 本文选择了如下的 w :

$$w = \left(\frac{G-t}{G} \right)^\delta. \quad (4)$$

其中 G 是最大进化代数, δ 是一个正实数. 随着迭代次数*t*的增加, w 将逐渐减小为0.

为了进一步避免搜索过程陷入局部最优, 提高全局搜索能力, 本算法对种群提出了一种分组策略, 如下所示:

$$w' = \left(\frac{G-t}{G} \right)^{\delta'}, \quad (5)$$

$$\mu = \lfloor w's + 1 \rfloor. \quad (6)$$

其中 w' 是惯性权值, δ' 是一个正实数, μ 是分组数, $\lfloor \cdot \rfloor$ 表示向下取整.

SFS算法相对于PSO算法有着自身的特点, 但是SFS有与PSO类似的群集智能算法特性: 无智能的主体通过合作表现出智能行为的特征. 因此将SFS算法归类于群集智能.

3 实验和分析(Experiment and analysis)

为了分析SFS算法的性能, 本文测试了文[4]提出的benchmark问题的23个函数系列, 搜寻这些函数的极小值点. 其中函数 $f_1 \sim f_7$ 是单峰的; $f_8 \sim f_{13}$ 是有很多局部极小值的多峰函数; $f_{14} \sim f_{23}$ 是只有很少局部极小值的多峰函数. 测试结果见表1~3.

表1 $f_1 \sim f_7$ 的测试结果

Table 1 Test result of $f_1 \sim f_7$

函数	允许误差	算法	平均计算时间/s	平均成功时间/s	成功次数
f_1 ($G=1500$)	1e-6	DE	5.4	3.7	30
		CLPSO	24.4	16.3	30
		SFS	14.1	6.7	30
f_2 ($G=2000$)	1e-6	DE	9.0	7.6	30
		CLPSO	33.6	21.3	30
		SFS	20.4	15.6	30
f_3 ($G=5000$)	1e-6	DE	32.8	25.9	30
		CLPSO	93.9	93.9	0
		SFS	63.5	33.5	30
f_4 ($G=5000$)	1e-6	DE	23.9	20.2	9
		CLPSO	83.9	83.9	0
		SFS	49.6	33.8	30
f_5 ($G=20000$)	1e-6	DE	84.1	16.2	30
		CLPSO	349.1	349.1	0
		SFS	189.7	151.8	30
f_6 ($G=1500$)	1e-6	DE	7.3	1.6	30
		CLPSO	25.7	12.5	30
		SFS	15.5	1.9	30
f_7 ($G=5000$)	1e-4	DE	29.5	29.5	0
		CLPSO	93.5	93.5	0
		SFS	61.5	48.4	18

表2 $f_8 \sim f_{13}$ 的测试结果

Table 2 Test result of $f_8 \sim f_{13}$

函数	允许误差	算法	平均计算时间/s	平均成功时间/s	成功次数
f_8 ($G=5000$)	-12000	DE	41.5	41.5	0
		CLPSO	92.1	28.4	30
		SFS	58.4	58.4	0
f_9 ($G=5000$)	1e-3	DE	36.1	36.1	0
		CLPSO	84.3	84.3	0
		SFS	54.7	54.7	0
f_{10} ($G=1500$)	1e-3	DE	7.7	4.4	30
		CLPSO	27.1	23.9	30
		SFS	16.7	7.6	30
f_{11} ($G=2000$)	1e-3	DE	10.8	4.0	30
		CLPSO	36.7	25.3	29
		SFS	24.3	5.7	30
f_{12} ($G=1500$)	1e-3	DE	9.5	3.8	30
		CLPSO	35.2	23.5	30
		SFS	17.7	3.7	30
f_{13} ($G=1500$)	1e-3	DE	9.8	5.0	30
		CLPSO	38.6	16.7	30
		SFS	17.7	4.4	30

表3 $f_{14} \sim f_{23}$ 的测试结果
Table 3 Test result of $f_{14} \sim f_{23}$

函数	允许误差	算法	平均计算时间/s	平均成功时间/s	成功次数
f_{14} ($G=100$)	0.998+ $1e-3$	DE	1.2	0.3	30
		CLPSO	2.1	0.8	30
		SFS	1.6	0.5	30
f_{15} ($G=4000$)	$3.175 \times 1e-4$	DE	31.5	29.9	2
		CLPSO	67.7	24.1	29
		SFS	44.5	8.9	30
f_{16} ($G=100$)	-1.0317	DE	0.6	0.1	30
		CLPSO	1.5	1.3	30
		SFS	0.9	0.2	30
f_{17} ($G=100$)	0.3981	DE	0.6	0.2	30
		CLPSO	1.5	1.4	30
		SFS	0.9	0.2	30
f_{18} ($G=100$)	3+ $1e-4$	DE	0.7	0.1	30
		CLPSO	1.6	0.8	30
		SFS	1.0	0.3	30
f_{19} ($G=100$)	-3.86+ $1e-4$	DE	0.7	0.1	30
		CLPSO	1.7	0.4	30
		SFS	1.0	0.2	30
f_{20} ($G=200$)	-3.32+ 0.01	DE	1.4	1.3	19
		CLPSO	3.5	2.1	13
		SFS	2.1	0.59	30
f_{21} ($G=100$)	-10	DE	1.0	0.5	30
		CLPSO	1.8	1.5	17
		SFS	1.3	0.3	30
f_{22} ($G=100$)	-10	DE	1.2	0.5	30
		CLPSO	1.9	0.8	23
		SFS	1.4	0.3	30
f_{23} ($G=100$)	-10	DE	1.3	0.5	30
		CLPSO	2.0	1.0	25
		SFS	1.6	0.3	30

用于与SFS做对比实验的算法为: DE, 全面学习的PSO(CLPSO)^[5]. 在实验中, 都选取种群数为100, 分别作30次独立实验. 种群的初始化是在解空间中随机产生的. DE的参数设置为: 尺度因子 $F = 0.5$, 交叉概率 $CR = 0.9$; CLPSO的参数根据文[5]的推荐设置, 但对于函数 $f_{14} \sim f_{23}$ 设置 $m = 2$; SFS的参数设置为: $\delta = \delta' = 12$. 算法程序都运行于一

台Pentium 4 CPU 2.4 GHz、内存512 M电脑. 结果见表1~3, 其中平均计算时间是30次独立实验中, 到达最大进化代数 G 时的平均时间; 平均成功时间是达到允许误差时平均需要的计算时间; 成功次数是30次实验中达到了允许误差的成功实验次数.

表1~3的测试结果表明SFS能够解决所选择的大多数单峰和多峰优化函数问题. 针对这些benchmark函数, 相对于CLPSO算法和DE算法, SFS有较好的全局搜索能力. 从这些表中还可以看出SFS的计算复杂度虽然高于DE, 但相对于CLPSO, SFS计算复杂度较低.

4 结语(Conclusion)

本文提出了一种基于模仿人类搜索行为的随机搜索优化算法SFS. 通过对benchmark函数问题的优化验证了SFS的性能. 接下来要进一步分析该算法的收敛性并研究参数的选取策略以提高算法的性能. 将来的研究需要将该算法应用于一些实际的复杂优化问题.

参考文献(References):

- [1] KENNEDY J, EBERHART R, SHI Y. *Swarm Intelligence*[M]. San Francisco: Morgan Kaufmann, 2001.
- [2] SCHOENAUER M, MICHALEWICZ Z. Evolutionary computation at the edge of feasibility[C] // *Proceedings of the 4th Parallel Problem Solving from Nature, Lecture Notes in Computer Science*. Berlin: Springer, 1996: 245 - 254.
- [3] DONALD A P. *Optimization Theory with Applications*[M]. New York: Dover Publications, 1986.
- [4] YAO X, LIU Y, LIN G M. Evolutionary programming made faster[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82 - 102.
- [5] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. *IEEE Transaction on Evolutionary Computation*, 2006, 10(3): 67 - 82.

作者简介:

郑永康 (1977—), 男, 博士, 研究方向为电力系统负荷预测、计算智能等, E-mail: zyk555@163.com;

陈维荣 (1965—), 男, 教授, 博士生导师, 主要从事牵引供电调度综合自动化、小波分析及其应用等;

戴朝华 (1973—), 男, 博士研究生, 研究方向为计算智能、多目标优化等;

王维博 (1977—), 男, 博士研究生, 研究方向为图像处理、优化算法等.