

文章编号:1001-9081(2010)10-2575-03

## 基于差分扰动的混合蛙跳算法

赵鹏军

(商洛学院 数学与计算科学系, 陕西 商洛 726000)

(pengjunzhao@126.com)

**摘要:**针对基本混合蛙跳算法在处理复杂函数优化问题时容易陷入局部最优、求解精度低的缺点,借鉴差分进化中的变异思想,提出了一种改进的混合蛙跳算法,利用子群中其他个体的有利信息,对其更新策略进行局部扰动。实验结果表明,改进的混合蛙跳算法对复杂函数优化问题具有较强的求解能力。算法寻优效率高、全局性能好、优化结果稳定,性能明显优于所比较的算法。

**关键词:**混合蛙跳算法;智能优化;早熟收敛;差分进化;扰动

**中图分类号:** TP18 **文献标志码:** A

## Shuffled frog leaping algorithm based on differential disturbance

ZHAO Peng-jun

(Department of Mathematics and Computational Science, Shangluo University, Shangluo Shaanxi 726000, China)

**Abstract:** Basic Shuffled Frog Leaping Algorithm (SFLA) algorithm easily traps into local optimum and has a low convergent precision when being used to address complex functions. To overcome these above shortcomings, an improved SFLA based on mutation idea in Differential Evolution (DE) was proposed. The proposed algorithm used beneficial information of the other individuals in sub-group to disturb updating strategy locally. The experimental results show that the improved SFLA has a better capability to solve complex functions than other algorithms. It has high optimization efficiency, good global performance, and stable optimization outcomes, and is superior to the other algorithms.

**Key words:** Shuffled Frog Leaping Algorithm (SFLA); intelligent optimization; premature convergence; Differential Evolution (DE); disturbance

### 0 引言

混合蛙跳算法(Shuffled Frog Leaping Algorithm, SFLA)源于对青蛙觅食行为的研究,是一种新的智能优化算法。其原理是由算法产生一组随机青蛙(个体或随机解),在每次进化中,根据个体的适应度来确定吸引域;以某种策略更新局部最差个体,使得搜索个体朝最优解移动;通过个体间的协作与竞争来实现在多维空间中对最优解的搜索。该算法继承了Memetic算法(Memetic Algorithm, MA)和粒子群优化算法(Particle Swarm Optimization, PSO)的优点<sup>[1]</sup>,具有概念简单、参数少、计算速度快、全局寻优能力强、易于实现等特点,并且简单易用,在多个领域取得了成功<sup>[1-8]</sup>。然而,与其他智能优化算法一样,SFLA同样存在早熟收敛现象,在求解部分函数优化问题时效果不够理想。为提高算法的性能,本文借鉴文献[9]中的思想,提出了一种改进的SFLA(记为DSFLA),可在一定程度上避免陷入局部最优,并通过实验验证了所给算法的可行性和有效性。

### 1 混合蛙跳算法

SFLA是一种基于群体的协同搜索的新算法,在SFLA中,整个青蛙群体被分为多个子群,不同的子群被认为是具有不同思想的青蛙的集合。子群中青蛙执行局部搜索策略,局部搜索使得思想在局部个体间传递,当子群进化到一定阶段以后,各个子群体之间的思想在混合过程中得到交换,局部搜索和混合过程一直持续到满足终止条件为止。

在连续空间坐标系下,SFLA的数学描述为:设青蛙群体规模为 $F$ ,其中第 $i$ 个个体在 $D$ 维空间中的坐标(问题的解)为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ,计算个体的适应度 $f(x_i)$ ,根据适应度将其按递减顺序排列。然后将整个群体划分为 $S$ 个子群,每个子群中包含 $N$ 个个体,即满足关系 $F = S \cdot N$ ,在进化过程中,第一个解放入第一个子群,第二个解放入第二个子群……第 $S$ 个解放入第 $S$ 个子群。然后,第 $S+1$ 个解又放入到第一个子群,第 $S+2$ 个解放入到第二个子群,这样循环分配下去,直到所有解分配完毕。

在每一个子群中,适应度最优、最差的解分别记为 $x_b = (x_{b1}, x_{b2}, \dots, x_{bD})$ 和 $x_w = (x_{w1}, x_{w2}, \dots, x_{wD})$ ;群体中适应度最优的解记为 $x_g = (x_{g1}, x_{g2}, \dots, x_{gD})$ 。在每次进化中,对 $x_w$ 进行更新操作,其搜索策略为:

$$D_j = r \cdot (x_b - x_w) \quad (1)$$

$$x_w' = x_w + D_j; -D_{\max} \leq D_j \leq D_{\max} \quad (2)$$

其中: $r$ 为 $[0, 1]$ 内的均匀随机数, $j = 1, 2, \dots, S, D_{\max}$ 为最大移动步长。如果 $x_w'$ 的适应度优于 $x_w$ 的适应度,则用 $x_w'$ 代替 $x_w$ ;如果没有改进,则用 $x_g$ 替换 $x_b$ ,重复执行式(1)、(2);如果仍没有改进,则从搜索空间中随机产生一个新解取代原来的 $x_w$ ,在指定迭代次数内继续执行以上操作,这样也就完成了SFLA的一次进化。

和遗传算法类似,SFLA也存在着群体的产生、进化、交叉和信息的传递和交换等操作,但进化机制更加灵活,算法是在进行局部搜索的基础上更新全局最优,具有全局优化与局部细致搜索的优点,可以用来优化连续和离散问题,并且具有

较强的鲁棒性<sup>[8]</sup>。

## 2 改进的混合蛙跳算法

为充分利用群体中的有利信息,受差分进化(Differential Evolution, DE)<sup>[9]</sup>中变异操作的启发,在SFLA的进化中引入差分向量,通过增加随机扰动,提高群体的多样性,从而改善全局搜索的能力。具体改进如下:

在子群 $j$ 中,随机选择两个解 $x_{r_1} = (x_{r_{11}}, x_{r_{12}}, \dots, x_{r_{1D}})$ 和 $x_{r_2} = (x_{r_{21}}, x_{r_{22}}, \dots, x_{r_{2D}})$  ( $r_1 \neq r_2 \neq w \neq b$ ),记差分向量 $Rx = (Rx_1, Rx_2, \dots, Rx_D) = x_{r_1} - x_{r_2}$ ,其值越小,扰动也越小,移动步长也会自动减小。相应地,式(1)变为:

$$D_{ji} = \begin{cases} r \cdot (x_{b_i} - x_{w_i}), & r_3 \geq CR \\ \alpha \cdot Rx_i + r \cdot (x_{b_i} - x_{w_i}), & r_3 < CR \end{cases} \quad (3)$$

其中: $i = 1, 2, \dots, D$ ,  $CR$ 是扰动概率,  $\alpha = (0.5 + \beta)/2$ ,  $r_3, \beta$ 为 $[0, 1]$ 内的均匀随机数。如果经由扰动得到的 $x_w'$ 的分量超出了搜索空间,那么将其限制在搜索空间内。若 $x_w'$ 的适应度优于 $x_w$ 的适应度,则用 $x_w'$ 代替 $x_w$ ;否则,用 $x_{g_i}$ 代替式(3)中的 $x_{b_i}$ ,重新执行式(3)、(2)。这样最差个体就可以向除自身之外的任意个体学习,成功地维持了群体的多样性,从而为进化操作提供更好、更有效的信息。因而DSFLA在保持其他步骤不变的基础上,既可以从局部最优的陷阱中跳出,更有可能求得优化问题的全局最优解。

综上所述,DSFLA的具体实现过程如下。

- 步骤1 初始化群体及参数:随机解,群体规模 $F$ ,子群数 $S$ ,子群内更新次数 $It$ ,混合迭代次数 $N_{\max}$ ,扰动概率 $CR$ ;
- 步骤2 评价群体,计算每个个体的适应度;
- 步骤3 根据适应度将 $F$ 个个体降序排列分为 $S$ 个子群;
- 步骤4 确定 $x_g, x_b$ 和 $x_w$ ,在迭代次数 $It$ 内按式(3)修正最差解;
- 步骤5 将各个子群重新混合构成新的群体;
- 步骤6 判断算法的终止条件是否满足,若满足,则结束寻优;否则,转至步骤2。

从以上的步骤可以看出,在进化初期,群体中个体的差异比较大,因此用来扰动的差分向量也较大,有利于算法的全局搜索;而随着进化的进行,算法趋于收敛的时候,群体中个体的差异也较小,从而用来扰动的差分向量也随之自适应地减小,有利于算法的局部收缩。也正是这简单又独具特色的扰动操作使DSFLA具有良好的全局和局部搜索能力,在整个进化过程中,能提高群体的多样性,可以确保群体持续寻优,有利于提高收敛速度,并能避免陷入局部最优。

## 3 仿真实验

为了验证DSFLA的性能,仿真实验选取单峰、多峰且存在许多局部最小的5个典型函数Sphere( $f_1$ )、Rosenbrock( $f_2$ )、Griewank( $f_3$ )、Ackley( $f_4$ )、Schaffer7( $f_5$ )作为测试对象<sup>[1,6,10]</sup>,所有测试函数的最小值均为0,5个测试函数的表达式如下。

$$f_1(x) = \sum_{i=1}^D x_i^2 \quad (4)$$

$$f_2(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (5)$$

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1 \quad (6)$$

$$f_4(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) -$$

$$\exp \left( \frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e \quad (7)$$

$$f_5(x) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} [\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1] \quad (8)$$

本文对SFLA、ISFLA<sup>[6]</sup>和DSFLA分别进行了测试,为了增强可比性,采用文献[1,6]中建议的参数设置, $F = 200$ ,  $S = 20$ ,  $It = 10$ ,  $N_{\max} = 500$ ,  $CR = 0.8$ 。为减少偶然性影响,三种算法分别运行30次,算法中其他参数设置见表1,仿真结果如表2所示,包括平均最优适应度、标准差和成功率。

表1 参数设置

函数	$D$	搜索空间	允许误差
$f_1$	30	$[-5.12, 5.12]^D$	0.01
$f_2$	30	$[-30, 30]^D$	100
$f_3$	30	$[-600, 600]^D$	0.01
$f_4$	30	$[-32, 32]^D$	0.5
$f_5$	30	$[-100, 100]^D$	10

表2 算法寻优能力比较

函数	算法	平均适应度	标准差	成功率/%
$f_1$	SFLA	0.012 868	0.008 834	43.33
	ISFLA	0	0	100.00
	DSFLA	0	0	100.00
$f_2$	SFLA	211.356 754	99.248 493	3.33
	ISFLA	70.242 521	56.478 120	83.33
	DSFLA	26.981 821	0.997 794	100.00
$f_3$	SFLA	0.884 869	0.211 365	0
	ISFLA	0.012 827	0.008 371	40.00
	DSFLA	0.000 247	0.001 350	100.00
$f_4$	SFLA	1.425 681	0.669 065	10.00
	ISFLA	0.003 870	0.001 181	100.00
	DSFLA	0	0	100.00
$f_5$	SFLA	28.420 643	8.270 940	0
	ISFLA	8.005 775	1.920 295	76.67
	DSFLA	0.709 944	0.124 716	100.00

从仿真数据可以看出,DSFLA由于采用了局部扰动策略,很大程度上改善了对测试函数的寻优能力,其寻优速度快,收敛精度高,鲁棒性好,对上述5个典型测试函数的优化结果明显优于SFLA和ISFLA,有效地改善了算法的搜索效率。

图1~5分别是上述5个函数采用3种算法运行30次后求得的平均最优适应度进化曲线。从图中可以看出,在进化早期,由于新算法有一个大的潜在搜索空间,个体要尽可能定位于较好搜索区域,因而全局收敛速度慢于SFLA和ISFLA;但是在进化中后期,由于新算法具有较强的局部搜索能力,因而收敛性能都优于SFLA和ISFLA。

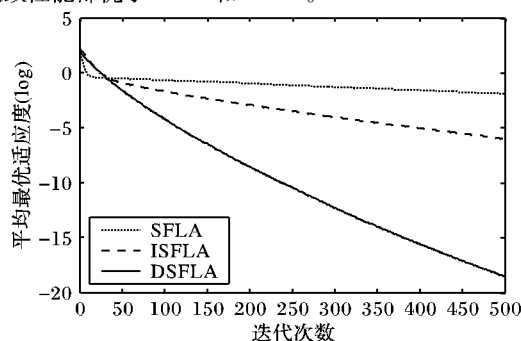
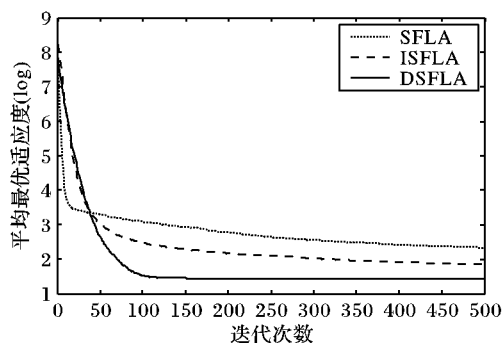
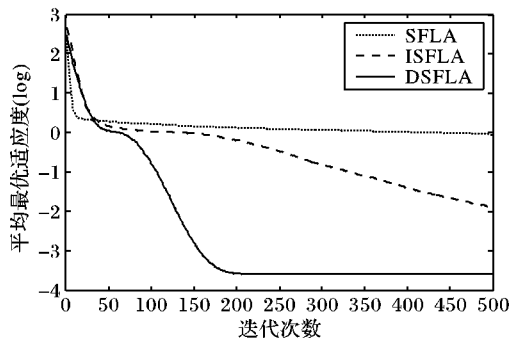
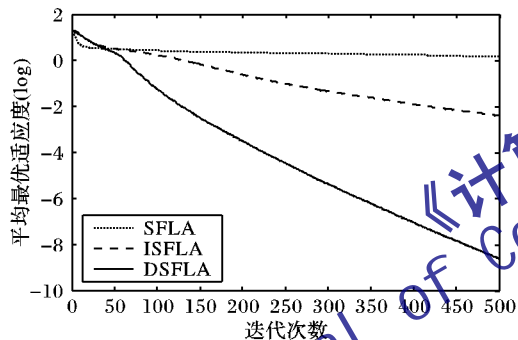


图1  $f_1$  (Sphere函数) 进化曲线

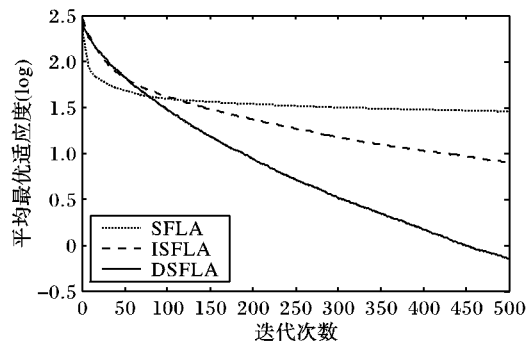
图2  $f_2$  (Rosenbrock 函数) 进化曲线图3  $f_3$  (Griewank 函数) 进化曲线图4  $f_4$  (Ackley 函数) 进化曲线

从表中的数据 and 进化曲线看,改进之后的收敛精度、后期收敛速度都得到了很大提高,能有效地避免算法陷入局部最优,具有更强的寻优能力,这给大量非线性、不可微和多峰值复杂优化问题的求解提供了一种新的思路和解决方法。

#### 4 结语

混合蛙跳算法是一种新的随机搜索算法,本文借鉴差分进化算法中的变异思想,修正了其更新机制,使算法在保证良

好收敛性的同时,提高了搜索效率。对5个测试函数的仿真结果表明 DSFLA 相对于 SFLA、ISFLA 的性能有了较大提高,有很强的全局搜索能力,能有效避免早熟收敛,而且算法的收敛速度快,稳定性好。接下来我们的研究将致力于本文算法的实际应用。

图5  $f_5$  (Schaffer7 函数) 进化曲线

#### 参考文献:

- [1] ELBELTAGI E, HEGAZY T, GRIERSON D. Comparison among five evolutionary-based optimization algorithms [J]. *Advanced Engineering Informatics*, 2005, 19(1): 43–53.
- [2] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using shuffled frog leaping algorithm [J]. *Journal of Water Resources Planning and Management*, 2003, 129 (3): 210–225.
- [3] EUSUFF M, LANSEY K, PASHA F. Shuffled frog-leaping algorithm: a meta-heuristic for discrete optimization [J]. *Engineering Optimization*, 2006, 38(2), 129–154.
- [4] SHIE Y H, ATIQUZZAMAN M. Optimal design of water distribution network using shuffled complex evolution [J]. *The Institution of Engineers*, 2004, 44(1): 93–107.
- [5] ELBERHAIRY H, ELBELTAGI E, HEGAZY T, *et al.* Comparison of two evolutionary algorithms for optimization of bridge deck repairs [J]. *Computer-Aided Civil and Infrastructure Engineering*, 2006, 21(8): 561–572.
- [6] 李英海, 周建中, 杨俊杰, 等. 一种基于阈值选择策略的改进混合蛙跳算法[J]. *计算机工程与应用*, 2007, 43(35): 19–21.
- [7] 吴华丽, 汪玉春, 陈坤明, 等. 基于混合蛙跳算法的成品油管网优化设计[J]. *石油工程建设*, 2008, 34(1): 14–16.
- [8] 王辉, 钱锋. 群体智能优化算法[J]. *化工自动化及仪表*, 2007, 34(5): 7–13.
- [9] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces [J]. *Journal of Global Optimization*, 1997, 1(4): 341–359.
- [10] 任小波, 杨忠秀. 一种动态扩散粒子群算法[J]. *计算机应用*, 2010, 30(1): 159–161.

(上接第 2574 页)

- [5] 杨佩, 陈春林, 周跃进, 等. 多 Agent 信念修正框架 BeReFrame [J]. *广西师范大学学报: 自然科学版*, 2008, 26(1): 139–142.
- [6] BALTAG A, SMETS S. Dynamic belief revision over multi-Agent plausibility models [C]// LOFT 07: Proceedings of the 7th Conference on Logic and the Foundations of Game and Decision Theory. Liverpool: University of Liverpool, 2006: 11–24.
- [7] van BENTHEM J. Dynamic logic for belief revision[J]. *Journal of Applied Non-Classical Logics*, 2007, 17(2): 1–27.
- [8] LIN FANGZHEN. On strongest necessary and weakest sufficient conditions [J]. *Artificial Intelligence*, 2001, 128(1): 143–159.
- [9] LIN FANGZHEN, REITER R. Forget it! [C]// Proceedings of AAAI Fall Symposium on Relevance. New Orleans: AAAI, 1994: 154–159.

- [10] EITER T, WANG KEWEN. Semantic forgetting in answer set programming[J]. *Artificial Intelligence*, 2008, 172(14): 1644–1672.
- [11] LIN YUNGUO, WANG XIULI. Semantics of knowledge forgetting in alternating-time temporal epistemic logic [C]// International Conference on Information Science and Engineering. Nanjing, China: [s. n.], 2009: 389–392.
- [12] 王国俊. 数理逻辑引论与归纳原理[M]. 北京: 科学出版社, 2000: 16–40.
- [13] LANG J, LIBERATORE P, MARQUIS P. Propositional independence: formula variable independence and forgetting [J]. *Journal of Artificial Intelligence Research*, 2003, 18(1): 391–443.
- [14] MOINARD Y. Forgetting literals with varying propositional symbols [J]. *Journal of Logic and Computation*, 2007, 17(5): 955–982.