

PSO 和 SFLA 混合优化算法

代永强,王联国

(甘肃农业大学 信息科学技术学院,甘肃 兰州 730070)

摘要:各种智能优化算法由于进化原理不同,优化性能各异,将不同种类的智能优化算法混合起来,往往能够取长补短,互相促进,提高混合算法的优化性能。利用粒子群优化(PSO)算法的快速收敛特性和混合蛙跳算法(SFLA)突出的全局协同搜索能力,提出了一种 PSO-SFLA 混合优化算法。该混合算法在执行过程中将种群分为 2 个子群体,一个子群体采用 PSO 算法进化寻优,另一个子群体采用改进的 SFLA 进化寻优,2 个子群体共享整个种群极值信息。通过对 3 个标准函数进行实验并与基本 PSO 算法进行比较,实验结果表明混合算法获得了更好的解,具有更好的优化性能。

关键词:粒子群优化算法;混合蛙跳算法;混合算法;优化性能

中图分类号:TP18

文献标识码:A

文章编号:1673-629X(2014)04-0077-03

doi:10.3969/j.issn.1673-629X.2014.04.019

Hybrid Optimization Algorithm of PSO and SFLA

DAI Yong-qiang, WANG Lian-guo

(College of Information Science & Technology, Gansu Agricultural University, Lanzhou 730070, China)

Abstract: All kinds of intelligent optimization algorithm show different optimized performance because of different evolution principle. To mix different kinds of intelligent optimized algorithm can complement and promote each other, and improve the optimization performance of hybrid algorithm. A hybrid algorithm combined Particle Swarm Optimization (PSO) and Shuffled Frog Leaping Algorithm (SFLA) is proposed by using the rapid convergence properties of PSO algorithm and the outstanding global cooperative search ability of SFLA. The algorithm divides the swarm into two sub-groups. In each iteration, one sub-group evolves using PSO algorithm, the other sub-group evolves using SFLA, and two algorithms share the information of groups extremum. The algorithm is experimented for three standard functions and compared with basic PSO algorithm, results show that PSO-SFLA hybrid algorithm outperforms PSO algorithm.

Key words: Particle Swarm Optimization (PSO) algorithm; Shuffled Frog Leaping Algorithm (SFLA); hybrid algorithm; optimization performance

0 引言

Kennedy 和 Eberhart 于 1995 年提出粒子群优化 (Particle Swarm Optimization, PSO) 算法^[1]。PSO 算法原理简单、易于实现,具有深刻的智能背景,国内外科研人员对该算法进行了研究分析和改进^[2-5],这些改进算法进一步提高了 PSO 算法的优化性能。

2000 年 Eusuff 和 Lansey 首次提出了一种新的基于全局协同搜索的智能优化方法——混合蛙跳算法 (Shuffled Frog Leaping Algorithm, SFLA) 并首次成功地解决了组合优化问题^[6],该算法参数较少、概念简单、全局搜索能力强、容易实现,近十年来,SFLA 不断得到改进及应用^[7-12]。目前该算法在解决多目标混

合型装配线排序问题、管道网络扩充中的管道尺寸最小化问题、旅行商问题、调度问题、配料问题、齿轮问题等方面,进一步证明了其在组合优化方面的优越性。但是,与其他优化算法一样,混合蛙跳算法也容易陷入局部最优值,对于部分多峰函数,优化效果不够理想。

文中将粒子群优化算法和改进后的混合蛙跳算法结合起来,提出了 PSO-SFLA 混合优化算法。该算法利用了 PSO 算法的快速收敛性和混合蛙跳算法突出的全局协同搜索能力,扩展了算法的搜索区域,有效克服了 PSO 算法易陷入局部最优和 SFLA 算法收敛速度较慢的缺点。仿真结果说明,这种混合算法具有更好的优化性能。

收稿日期:2013-06-05

修回日期:2013-09-16

网络出版时间:2014-01-28

基金项目:国家自然科学基金资助项目(61063028);甘肃省科技支撑计划(1011NKCA058);甘肃农业大学科技创新基金(GAU-CX1119)

作者简介:代永强(1981-),男,讲师,硕士,研究方向为计算智能、智能信息处理;王联国,教授,硕士生导师,博士,通讯作者,研究方向为农业信息技术、智能计算。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140128.1144.034.html>

1 基本算法

1.1 基本 PSO

PSO 思想源于对鸟类等生物种群觅食行为的研究和仿真,算法描述了数量为 N 的鸟群在 D 维空间中以一定速度飞行,每只鸟的飞行速度和位移根据自身和鸟群的飞行经验调整,在 D 维子空间中的飞行速度和位置按下式调整:

$$v_{i+1} = wv_i + c_1r_1(p_i - z_i) + c_2r_2(p_g - z_i) \quad (1)$$

$$z_{i+1} = z_i + v_{i+1} \quad (2)$$

$$\begin{cases} v_i = v_{\max} & \text{if } v_i > v_{\max} \\ v_i = v_{\min} & \text{if } v_i < v_{\min} \end{cases}$$

式中, $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ 为鸟类个体 i 在 D 维空间的运行速度; $z_i = (z_{i,1}, z_{i,2}, \dots, z_{i,D})$ 为鸟类个体 i 在 D 维空间的当前位置; $Q_i = (Q_{i,1}, Q_{i,2}, \dots, Q_{i,D})$ 为鸟类个体 i 迄今为止找到的最优位置; $Q_g = (Q_{g,1}, Q_{g,2}, \dots, Q_{g,D})$ 为整个鸟群到目前为止找到的最好位置。其中 $i = 1, 2, \dots, N$; b 为权重系数, c_1 和 c_2 是加速因子; r_1 和 r_2 是 $[0, 1]$ 范围内的随机数; $[v_{\min}, v_{\max}]$ 是鸟类个体运动速度的下限和上限。通常情况下,速度更新公式中的 b 随着进化过程,由最大加权因子 b_s 线性变化到最小加权因子 b_e ,即:

$$b_t = b_s - t \frac{b_s - b_e}{T} \quad (3)$$

其中, t 是当前的进化次数; T 为总的进化次数; $b_s = 0.9$, $b_e = 0.4$ 。

鸟类个体在搜索区域根据来自自身信息和鸟群信息的指导,不时调整运动速度和方向,接近最优值。

1.2 改进的混合蛙跳算法 (ISFLA)

改进的混合蛙跳算法最差的个体在更新时不仅向当前子群最优值或全局最优值学习,同时由于延续了上次更新的部分惯性步长,能够向记忆中个体邻域历史最好值学习,这种进化策略扩大了搜索范围,加快了收敛速度,提高了算法搜索性能。更新策略如公式 (4)。

$$\begin{aligned} \text{Dis}^{k+1} = & w * (R_1 * \text{Dis}^k + R_2 * h(P_{\text{worst}})) + \\ & R_3 * (P_{\text{best}} - P_{\text{worst}}) \\ & (-\text{Dis}M \leq \text{Dis} \leq \text{Dis}M) \end{aligned} \quad (4)$$

其中, R_1, R_2, R_3 都是 $[0, 1]$ 范围内的随机数; P_{best} 是子群中适应值最好个体的解; P_{worst} 是适应值最差个体的解; Dis 是 P_{worst} 个体移动步长; $\text{Dis}M$ 是允许个体移动的最大步长; w 是惯性权重因子,取值与公式 (3) 一致; $h(P_{\text{worst}})$ 是 P_{worst} 的个体邻域中的历史最好值。

图 1 为个体邻域结构,有 N 个存储空间,分别记为 $\{C(i), i=1, 2, \dots, N\}$,存放每个个体进化时的历史最好值。个体邻域指的是当前个体存储区域半径范围内包含的存储区域,如半径为 1, $C(2)$ 个体邻域包含的存

储区域为 $\{C_1, C_2, C_3\}$ 。 $C(1)$ 包含在 $C(P)$ 的右邻域范围内, $C(P)$ 包含在 $C(1)$ 的左邻域范围内。

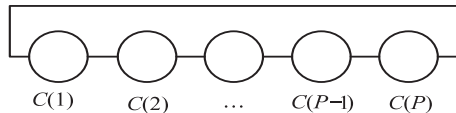


图 1 邻域存储示意图

2 PSO 和 SFLA 混合优化算法

图 2 为混合优化算法 (PSO-SFLA) 的流程图,混合算法进化过程中将种群分为 2 个子种群,分别采用 PSO 算法和改进的 SFLA 进化,2 个子种群受到全局最优极值信息指导。

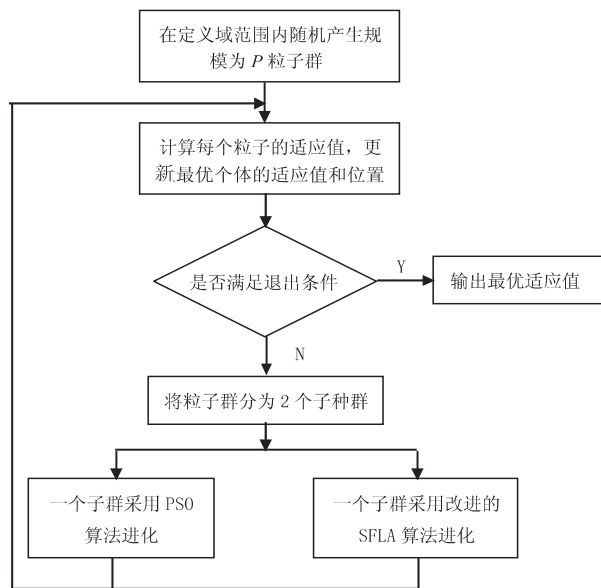


图 2 PSO-SFLA 混合算法流程

PSO 算法中各微粒通过自身的最好位置和种群最好位置,动态地调整当前的状态,在算法初期保持了快速的收敛速度,但是在算法后期由于粒子出现同一化现象,算法易陷入局部极值。而混合蛙跳算法将种群分成独立的若干子种群,经过多次子种群迭代更新,合并、排序后再分组,各个子种群之间得到充分交流,保持了整个种群的多样性,同时也保证了算法的全局收敛性。因而将 PSO 和 SFLA 两种算法混合使用,能够进一步扩大算法的搜索范围,加快收敛速度,克服 SFLA 算法运行速度慢和 PSO 易陷入局部极值点等缺点,从而兼顾了算法的优化精度和效率二者之间的关系,提高了优化性能。

3 实验方法,结果及说明

3.1 实验设计

对 3 个标准测试函数采用 PSO 和 PSO-SFLA 两种算法进行极小值寻优,测试平台为 VC++, CPU 为 P4 (2.2 G),内存大小为 1 G,测试函数的参数如表 1。

$$f_1(x) = \sum_{i=1}^{n-1} (100 (x_{i+1}^2 - x_i)^2 + (1 - x_i)^2)$$
$$f_2(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$$
$$f_3(x) = \frac{1}{4\,000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$$

实验中算法具体参数设置:种群规模大小为 200, PSO-SFLA 算法中 PSO 种群和 SFLA 种群各为 100,将 SFLA 种群分为 10 个子群,每个子群规模大小为 10,子群内更新迭代次数为 10。算法的进化迭代次数为 500。PSO-SFLA 算法中的个体邻域半径为 1,最终的结果选两种算法独立运行 30 次的平均值,参数和精度设置见表 1。

表 1 参数表

函数	维数	搜索范围	理论最优值	目标精度
f_1	30	$[-30,30]$	0	3×10^1
f_2	30	$[-5.12,5.12]$	0	3×10^1
f_3	30	$[-600,600]$	0	1×10^{-2}

3.2 结果及分析

3.2.1 迭代进化次数不变,算法收敛速度和精度分析

从表 2 可以看出,PSO-SFLA 的最小值、最大值和平均最优值均明显好于基本 PSO 算法。

表 2 进化次数不变的实验结果

函数	算法	平均最优值	最小值	最大值
f_1	PSO	4.210 1e+001	2.072 6e+001	9.493 7e+001
	PSO-SFLA	2.889 8e+001	2.063 2e+001	8.265 4e+001
f_2	PSO	2.563 7e+001	1.094 5e+001	4.079 3e+001
	PSO-SFLA	4.775 8e+000	9.949 6e-001	1.194 0e+001
f_3	PSO	1.803 1e-002	2.837 8e-004	8.216 9e-002
	PSO-SFLA	3.036 1e-003	0.000 0e+000	2.457 3e-002

图 3~图 5 是三个函数分别采用两种算法运行 30 次后平均进化曲线,纵坐标表示函数平均优化极值的常用对数,横坐标为进化迭代次数。从这些图中可以

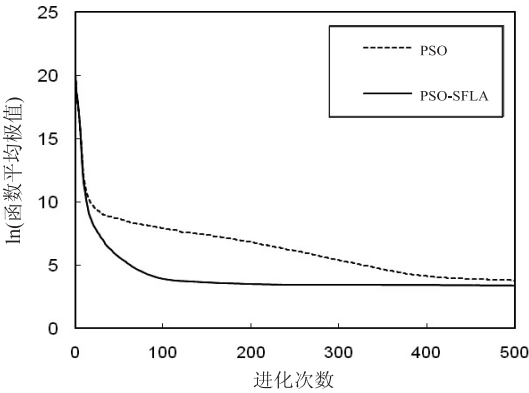


图 3 函数 f_1 平均进化曲线

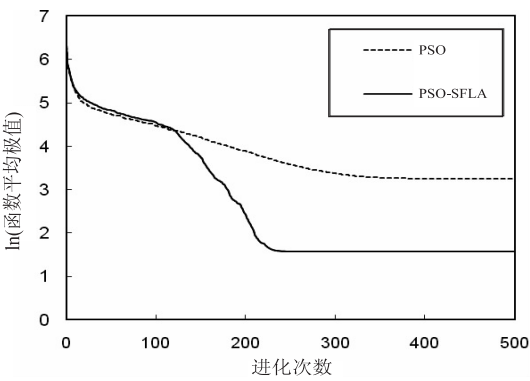


图 4 函数 f_2 平均进化曲线

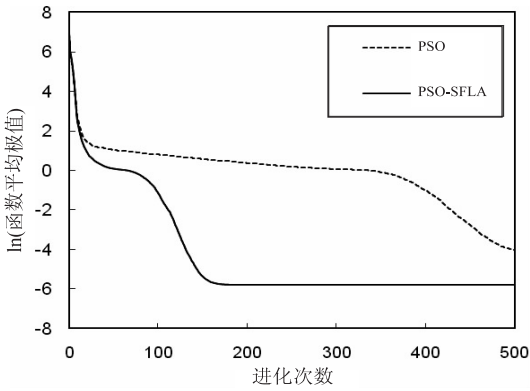


图 5 函数 f_3 平均进化曲线

看出,PSO-SFLA 的收敛精度和速度较好。

3.2.2 收敛精度不变,算法迭代进化次数分析

表 3 成功率为达到要求精度的迭代次数与实验总次数(30)的比值。从中看到 PSO-SFLA 对 3 个函数的优化成功率,达到目标优化精度的平均迭代次数、最小迭代次数以及最大迭代次数较好。说明 PSO-SFLA 收敛速度更快,优化成功率更高,稳定性更好。

表 3 收敛精度不变的实验结果

函数	算法	成功率 /%	平均迭 代次数	最小迭 代次数	最大迭 代次数
f_1	PSO	57	439	378	499
	PSO-SFLA	93	124	87	374
f_2	PSO	100	305	273	328
	PSO-SFLA	100	61	51	74
f_3	PSO	57	465	432	486
	PSO-SFLA	87	136	122	166

4 结束语

PSO-SFLA 充分利用了 SFLA 全局搜索能力和 PSO 算法的快速收敛性,提高了算法在解空间的搜索能力和收敛精度,有效克服了基本 PSO 算法容易陷入局部最优的缺陷。是一种融合了两种算法各自优点的混合优化算法,具有较好的优化性能。

种算法都会达到一个最优解,只是大小不同,求最大值的,在可行域范围内,越大越好。相反,求最小值的,越小越好。对于收敛速度,代数越少,说明收敛速度越快。文中的算法收敛速度较其他三种快,且最值较其他三种更优且更接近理论值。其中,LGA、GGA 又比 GA 的收敛速度快,且搜索到的最值更优。标准差能反映一个数据集的离散程度,离散度是评价方法的好坏的最重要也是最基本的指标。在此,标准差代表的是此代的种群个体间的离散程度,从图 3 中 f_2 函数的标准差变化图来看,LGGA 比 GGA 变化平稳,而 GGA 比 LGA、GA 较平稳得多。总体来说,文中提出的算法是可行有效的。

3 结束语

虽然 GA 能寻找到问题的最优解,但它仍然有一些不尽如人意的地方,最主要的是早熟现象以及收敛速度慢的问题。为了避免这种缺陷,文中针对交叉算子与变异算子在整个 GA 中的特点,对其加以改进,然后应用于 3 个多峰值函数优化。结果显示,LGGA 的寻优能力优于其他三种算法。LGGA 的全局搜索能力有所提高,减轻了早熟现象,收敛速度也加快了,表现出了较强的寻优性能。

参考文献:

[1] Dejong K A. An analysis of the behavior of a class of genetic adaptive systems [D]. Michigan: University of Michigan, 1975.

[2] Holland J H. Adaptation in natural and artificial systems[M].

(上接第 79 页)

参考文献:

[1] Kenned J,Eberhart R. Particle swarm optimization[C]//Proc of IEEE international conference on neural networks. Perth, Australia;[s. n.],1995:1942-1948.

[2] Eberhart R,Kennedy J. A new optimizer using particle swarm theory[C]//Proc of the 16th international symposium on micromachine and human science. Nagoya, Japan; IEEE Press, 1995:39-43.

[3] Shi Yuhui,Eberhart R C. A modified particle swarm optimizer [C]//Proc of IEEE international conference on evolutionary computation. Piscataway, USA; IEEE Press,1998:69-73.

[4] Higashi N,Iba H. Particle swarm optimization with Gaussian mutation[C]//Proc of the IEEE swarm intelligence symposium. Indianapolis, Indiana, USA; IEEE Press,2003:72-79.

[5] 胡建秀,曾建潮. 具有随机惯性权重的 PSO 算法[J]. 计算机仿真,2006,23(8):164-167.

[6] Eusuff M M,Lansey K,Pasha F. Shuffled frog leaping algorithm; A memetic meta- heuristic for combinatorial optimiza-

Ann Arbor:The University of Michigan Press,1975.

[3] 刘红,韦穗. 遗传算子的分析[J]. 计算机技术与发展, 2006,16(10):80-82.

[4] Ichikawa Y,Ishii Y. Retaining diversity of genetic algorithms for multivariable optimization and neural network learning [C]//Proc of IEEE international conference on neural network. [s. l.]:[s. n.],1993:1110-1114.

[5] Srinivas M,Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithm[J]. IEEE trans on systems, man and cybernetics,1994,24(4):656-667.

[6] 杨启文,蒋静坪,张国宏. 遗传算法优化速度的改进[J]. 软件学报,2001,12(2):270-275.

[7] 张晋,李冬黎,李平. 遗传算法编码机制的比较研究 [J]. 中国矿业大学学报,2002,31(6):637-640.

[8] 陈国龙,陈火旺,郭文忠,等. 基于随机错位算术交叉的遗传算法及其应用[J]. 模式识别与人工智能,2004,17(2): 250-256.

[9] Deep K,Thakur M. A new crossover operator for real coded genetic algorithms[J]. Applied mathematics and computation, 2007,188(1):895-911.

[10] 李小宁. 关于混合遗传算法改进的研究[D]. 西安:西北大学,2008.

[11] 金芬,孙春华,钟鸣. 遗传算法中适应度函数的改进术 [J]. 机械设计与制造,2010(3):218-219.

[12] 杨平,郑金华. 遗传选择算子的比较与研究[J]. 计算机工程与应用,2007,43(15):59-62.

[13] 何琳,王科俊,李国斌,等. 最优保留遗传算法及其收敛性分析[J]. 控制与决策,2000,15(1):63-66.

[14] 王小平,曹立明. 遗传算法理论、应用与软件实现[M]. 西安:西安交通大学出版社,2002.

tion[M]. London;J. Heuristics,2000.

[7] Zhen Ziyang,Wang Daobo,Liu Yuanyuan. Improved shuffled frog leaping algorithm for continuous optimization problem [C]//Proc of 2009 IEEE congress on evolutionary computation. Trondheim,Norway;[s. n.],2009.

[8] Zhang Xuncai,Hu Xuemei,Cui Guangzhao,et al. An improved shuffled frog leaping algorithm with cognitive behavior[C]// Proceedings of the 7th world congress on intelligent control and automation. Chongqing, China;[s. n.],2008.

[9] 赵鹏军,刘三阳. 求解复杂函数优化问题的混合蛙跳算法 [J]. 计算机应用研究,2009,26(7):2435-2437.

[10] 赵守法. 蛙跳算法的研究与应用[D]. 上海:华东师范大学,2008.

[11] Emad E,Tarek H,Donald G. A modified shuffled frog-leaping optimization algorithm: Applications to project management [J]. Structure and Infrastructure Engineering,2007,3(1):53-60.

[12] 罗雪晖,杨焱,李霞. 改进混合蛙跳算法求解旅行商问题[J]. 通信学报,2009,30(7):130-135.

PSO和SFLA混合优化算法

作者：[代永强](#)，[王联国](#)，[DAI Yong-qiang](#)，[WANG Lian-guo](#)

作者单位：[甘肃农业大学 信息科学技术学院](#)，[甘肃 兰州](#)，[730070](#)

刊名：[计算机技术与发展](#)

英文刊名：

Computer Technology and Development

ISTIC

年，卷(期)：

2014(4)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201404019.aspx