

DOI:10.3969/j.issn.1673-4785.201405070
网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.tp.20150611.0902.001.html>

广义中心混合蛙跳算法

赵嘉, 吕莉, 樊棠怀
(南昌工程学院 信息工程学院, 江西 南昌 330099)

摘 要:为解决标准混合蛙跳算法族群之间信息共享能力差的问题,加强族群内蛙的学习能力,利用各族群最优蛙位置的平均中心,构造一个与各族群最优蛙都有关联的虚拟广义中心蛙,提出广义中心混合蛙跳算法。该算法在进化过程中,首先蛙群最优蛙在原有位置及广义中心蛙的位置上进行“贪婪”选择,选择最好位置作为新的族群最优蛙位置;其次将广义中心蛙的优势运用于蛙跳规则中,在标准混合蛙跳算法的蛙跳规则中加入族群最差蛙向广义中心蛙学习的能力。将本文算法与不同维度下的标准混合蛙跳算法及新近提出的知名群智能算法进行比较,实验结果表明,本文算法在解的精度、收敛速度及解的稳定性等方面具有更优的性能。

关键词:蛙跳算法;混合蛙跳算法;广义中心;蛙跳规则;群智能算法

中图分类号:TP301 **文献标志码:**A **文章编号:**1673-4785(2015)03-0414-08

中文引用格式:赵嘉,吕莉,樊棠怀. 广义中心混合蛙跳算法[J]. 智能系统学报, 2015, 10(3): 414-421.
英文引用格式:ZHAO Jia, LYU Li, FAN Tanghuai. Shuffled frog-leaping algorithm based on the general center[J]. CAAI Transactions on Intelligent Systems, 2015, 10(3): 414-421.

Shuffled frog-leaping algorithm based on the general center

ZHAO Jia, LYU Li, FAN Tanghuai
(School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China)

Abstract: In this paper, a shuffled frog-leaping algorithm based on general center (GC-SFLA) is proposed to solve the problem of weak information sharing between memplexes in the shuffled frog leaping algorithm (SFLA) to enhance the learning ability and use the average center of optimal frog. The proposed GC-SFLA generates a virtual general center frog from the optimal frog of each memplex. Firstly, the optimal frog selects the best location among the original location and general center greedily as new location of new memplex. After that, the advantage of general center frog is applied to the frog-leaping rule, which enable the worst frog to learn from the general center frog. Experiments are conducted on a set of swarm intelligence algorithms to verify that the new approach outperforms SFLA in different dimensions. The experiment results present promising performance of the GC-SFLA on convergence velocity, precision and stability of solution.

Keywords: frog-leaping algorithm; shuffled frog leaping algorithm (SFLA); general center; frog leaping rule; swarm intelligence algorithms

混合蛙跳算法 (shuffled frog leaping algorithm, SFLA)^[1]是一种基于群体智能的亚启发式协同搜索计算技术,最早由 M. M. Eusuff 和 K. E. Lansey 于 2000 年提出。它结合了基于基因进化的模因演算法(memetic algorithm, MA)^[2]和基于群体行为的粒子群优化算法(particle swarm optimization, PSO)^[3]

收稿日期:2014-06-03. 网络出版日期:2015-06-11.
基金项目:国家自然科学基金资助项目(61261039,61263029);江西省自然科学基金资助项目(20132BAB211031);江西省科技厅科技支撑项目(20142BBG70034);南昌市科技计划项目(2013HZCG006,2013HZCG011,2014HZCC008).
通信作者:赵嘉. E-mail: zhaojia925@163.com.

的优点^[4],具有概念简单、参数设置少、计算速度快、全局寻优能力强、易于实现等特点^[5],并在无线传感器网络覆盖优化^[6]、函数优化^[7]、经济负荷分配^[8]、生产调度组合优化^[9]等领域取得较好应用,正成为智能计算领域的研究热点。

与其他群智能算法相似,混合蛙跳算法也存在易陷入局部极值、进化后期收敛速度慢、计算精度低等缺点。为此,研究人员在不断深入分析和研究后,提出了多种不同思想的改进混合蛙跳算法,比较有代表性的有罗雪晖等^[10]在混合蛙跳算法中加入调整序思想设计了局部搜索策略,并在全局信息交换中加入变异算法,提出一种改进的混合蛙跳算法并应用于求解 TSP 问题;T. Niknam 等^[11]利用混沌局部搜索策略提出改进的混沌混合蛙跳算法;借鉴分子动力学模拟的思想,张潇丹^[12]提出基于分子动力学模拟的改进混合蛙跳算法;Sun 等^[13]提出一种基于粒子共享的粒子群蛙跳混合优化算法,算法利用粒子群具有良好全局搜索性能与混合蛙跳算法具有较强局部搜索能力的特点,克服了群体智能算法后期易陷入局部最优及“早熟”收敛的缺点。这些算法都在标准 SFLA 基础上进行不同程度的改进,但其改进也不同程度增加了算法的复杂性。

在 SFLA 中,青蛙的跳跃主要经历局部搜索和全局信息交换 2 个阶段,局部搜索使模因信息在局部个体间进行传递,全局信息交换使得局部间的模因信息得到交换,这在很大程度上决定了算法的收敛速度与解的质量。但青蛙在进化过程中,族群中的最差青蛙只向自身蛙群和最优青蛙所在蛙群的最好青蛙学习,族群之间的相互学习不够、共享能力不强。为此,利用各族群最优蛙的优势,构造广义中心青蛙,并改进蛙群的进化策略,使蛙群在原有学习策略的基础上,增加向广义中心青蛙学习的能力,提出广义中心混合蛙跳算法(shuffled frog leaping algorithm based on general center, GC-SFLA)。通过对 8 个标准测试函数的实验仿真,将提出的广义中心混合蛙跳算法与标准混合蛙跳算法及新近提出的知名群智能算法比较算法收敛速度和全局寻优能力。

1 混合蛙跳算法

混合蛙跳算法是一种基于群体智能的生物进化算法。该算法模拟湿地中一群青蛙按族群划分的思想进行觅食。在算法执行时,首先产生 F 只青蛙组成蛙群,对于 N 维优化问题,群体中第 i 只青蛙表示为 $\mathbf{X}_i = (x_i^1, x_i^2, \dots, x_i^N)$,将蛙群内的青蛙个体按照适应值进行降序排列,找到全局最好青蛙(解) \mathbf{P}_g 。

将蛙群分成 m 个族群,每个族群包含 n 只青蛙,满足关系 $F = m \times n$ 。族群的划分规则是:第 1 只青蛙进入第 1 个族群,第 2 只青蛙进入第 2 个族群,第 m 只青蛙进入第 m 个族群,第 $m + 1$ 只青蛙又进入第 1 个族群,第 $m + 2$ 只青蛙进入第 2 个族群,以此类推,直至所有青蛙划分完毕。第 k 个族群中最好的一个青蛙,记为 \mathbf{P}_k^b ,相应地可以得到一个最差的青蛙,记为 \mathbf{P}_k^w 。

为了获得更多的食物,较差的蛙受较好蛙的影响而跳向较好的蛙。根据上述初始蛙跳规则,第 k 个青蛙族群中最差蛙的更新公式为:

$$\mathbf{D}_k = r \times (\mathbf{P}_k^b - \mathbf{P}_k^w) \tag{1}$$

$$\mathbf{P}_k^{\text{new-w}} = \mathbf{P}_k^w + \mathbf{D}_k, -D_{\max} \leq D_i \leq D_{\max} \tag{2}$$

式中: k 表示是第几个族群,且 $1 \leq k \leq m$; r 为 $[0, 1]$ 的随机数; \mathbf{D}_k 为第 k 个族群最差蛙移动的步长; $\mathbf{P}_k^{\text{new-w}}$ 为第 k 个族群最差蛙更新后新位置; D_{\max} 为蛙的最大跳动步长。

若执行更新策略(1)和(2)后, $\mathbf{P}_k^{\text{new-w}}$ 对应的适应值优于 \mathbf{P}_k^w 对应的适应值,则用 $\mathbf{P}_k^{\text{new-w}}$ 取代 \mathbf{P}_k^w ; 否则采用 \mathbf{P}_g 代替式(1)中的 \mathbf{P}_k^b ,重新更新 $\mathbf{P}_k^{\text{new-w}}$ 。若重新执行更新后, $\mathbf{P}_k^{\text{new-w}}$ 对应的适应值优于 \mathbf{P}_k^w 对应的适应值,则用 $\mathbf{P}_k^{\text{new-w}}$ 取代 \mathbf{P}_k^w ; 否则采用式(3)随机产生一个新的蛙代替 \mathbf{P}_k^w 。

$$\mathbf{P}_k^w = r \times (\mathbf{O}_{\max} - \mathbf{O}_{\min}) + \mathbf{O}_{\min} \tag{3}$$

式中: \mathbf{O}_{\max} 和 \mathbf{O}_{\min} 分别表示算法搜索范围的最大值和最小值。

重复以上的更新操作,直至满足事先设定的族群内的算法迭代次数。当所有族群的局部深度搜索完成以后,进行全局信息交换。局部深度搜索和全局信息交换两阶段交替进行,直到满足相应的结束条件。

2 广义中心混合蛙跳算法

2.1 广义中心策略

在群智能算法中,随着进化的进行,全局极值将会越来越接近最优解。搜索结束后,全局极值将位于最优解的邻近区域。与此同时,由于群智能算法的随机性,每个个体也分布在全局极值的邻近区域。为了改善全局极值,使其更快向最优解靠近^[14],Liu 等^[15]引入中心粒子。中心粒子由群体的中心位置形成,伴随于整个搜索过程,除不具有速度外,该粒子具有与其他普通粒子相同的所有性质。其产生的方式如式(4)所示:

$$x_d^{\text{sc}} = \sum_{i=1}^F x_i^d \tag{4}$$

式中, F 为个体总数, $d(1 \leq d \leq N)$ 为个体维数, N 为个体总维数, x_i^d 为第 i 个体第 d 维位置。

汤可宗等^[16]通过实验进一步发现,在粒子群优化算法中,所有个体极值形成的中心相比群体的中心更能趋近于最优解。为此,将 Liu 等^[15]提出的中心定义为狭义中心(special center, SC),见式(4);将个体极值形成的中心定义为广义中心(general center, GC),并提出双中心粒子群优化算法。广义中心粒子产生的方式如式(5):

$$x_d^{\text{gc}} = \sum_{i=1}^F p_{i,\text{best}}^d \quad (5)$$

式中: $p_{i,\text{best}}^d$ 为第 i 个体第 d 维极值位置。

混合蛙跳算法的基本原理是每只青蛙在族群内最优青蛙和全群最优青蛙的引导下,“积极”向着最优解靠近,青蛙会被吸引到全群最优蛙和族群最优蛙的邻域。混合蛙跳算法的核心思想是族群划分,每个族群均有族群内最优粒子。搜索结束后,每个族群的最优蛙位置位于最优解或其邻近区域,相比全群最优蛙,族群最优蛙的中心或许会更接近最优解,这一启示给改善混合蛙跳算法提供了思路,为加速算法的收敛速度提供了非常有用的信息。借鉴文献[16]广义中心思想,但混合蛙跳算法中无个体极值概念。为此,混合蛙跳算法中的广义中心青蛙定义如下。

定义 1 广义中心青蛙(general center frog, GCF) 在混合蛙跳算法的进化过程中,各族群的最优蛙位置表示为 P_k^b , 其中 k 表示是第几个族群。则广义中心青蛙 P^{gcf} 定义为

$$P^{\text{gcf}} = \frac{1}{m} \left(\sum_{k=1}^m P_k^b \right) \quad (6)$$

P^{gcf} 可能会跳出边界 $[O_{\min}, O_{\max}]$ 成为非可行解,此时,按照式(7)进行重置。

$$\begin{cases} P^{\text{gcf}} = O_{\max}, & P^{\text{gcf}} > O_{\max} \\ P^{\text{gcf}} = O_{\min}, & P^{\text{gcf}} < O_{\min} \end{cases} \quad (7)$$

对形成的广义中心青蛙,评估其适应值,从 P_g 和 P^{gcf} 选择较优的解作为新的 P_g ,其公式描述为

$$P_g = \begin{cases} P^{\text{gcf}}, & f(P^{\text{gcf}}) < f(P_g) \\ P_g, & f(P_g) \leq f(P^{\text{gcf}}) \end{cases} \quad (8)$$

式中: $f(\cdot)$ 为适应值函数。

2.2 蛙跳规则的改进

标准 SFLA 算法蛙跳规则过于简单,族群最差蛙向本族群最优蛙学习,最差蛙的可能新位置被限定在当前蛙与最好蛙位置的线段上^[6],限制了模因进化的搜索区域,且青蛙的学习能力不强,随着迭代

次数的增加,各族群的性能将趋同,多样性降低。

本文提出一种新的蛙跳规则,该策略借助广义中心青蛙的特性,使族群内最差粒子在进化过程中,能够学习其他族群内最优粒子。首先,此蛙跳规则会增加最差粒子向全局最优优点运动的可能;其次,根据蛙跳规则的数学表述可知,该操作扩大了最差粒子的搜索范围;再次,各族群在进化过程中形成了自己族群特色,也保证各族群的多样性。其位置更新公式与式(2)一致,最差蛙的移动步长更新公式为

$$D_k = r_1 \times (P_k^b - P_k^w) + r_2 \times (P^{\text{gcf}} - P_k^w) \quad (9)$$

式中: $j \in 1, 2, \dots, m$, r_1, r_2 为 $[0, 1]$ 的随机数。

若执行更新策略(9)和(2)后, $P_k^{\text{new-w}}$ 对应的适应值优于 P_k^w , 则用 $P_k^{\text{new-w}}$ 取代 P_k^w , 否则,最差蛙的移动步长更新公式为

$$D_k = r_1 \times (P_g - P_k^w) + r_2 \times (P^{\text{gcf}} - P_k^w) \quad (10)$$

若执行更新策略(10)和(2)后, $P_k^{\text{new-w}}$ 对应的适应值优于 P_k^w , 则用 $P_k^{\text{new-w}}$ 取代 P_k^w , 否则采用式(3)随机生成 1 个新的个体位置代替原来的 P_k^w 。

2.3 算法流程

GC-SFLA 算法的流程如图 1 所示。

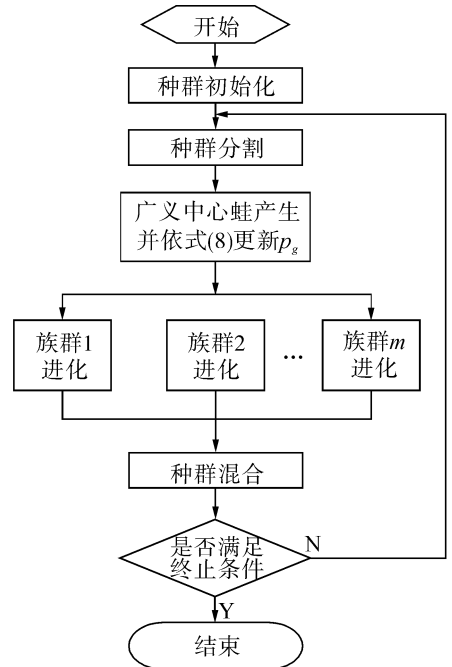


图 1 GC-SFLA 算法流程

Fig. 1 Flowchart of GC-SFLA

1) GC-SFLA 算法参数设置,包括族群数、族群内更新次数 L_{\max} 、族群内青蛙数、混合迭代次数 G_{\max} 等的设置。

2) 蛙群初始化。初始化蛙群中青蛙位置并评估其适应值,记录蛙群最优蛙为 P_g 。

3) 族群初始化。将蛙群分成族群,每个族群包含相同数量的蛙,第 k 个族群内的最优蛙和最差蛙分别记为 \mathbf{P}_k^b 与 \mathbf{P}_k^w 。

4) 广义中心蛙的生成。利用式(6)计算 \mathbf{P}^{scf} 并评估其适应值,并根据式(8)更新 \mathbf{P}_g 。

5) 族群进化。对第 k 个族群中的 \mathbf{P}_k^w 根据式(9)和(2)进行更新并计算其适应度值,若更新后的蛙优于原来的蛙,则取代原来族群中的蛙;如果没有改进,则根据式(10)和(2)进行更新并评估其适应度值,若更新后的蛙优于原来的蛙,则取代原来族群中的蛙;否则根据式(3)随机产生 1 个新蛙直接取代原来的 \mathbf{P}_k^w ;重复上述局部搜索 L_{\max} 次。

6) 族群混合。将更新后的各族群内蛙重新混合,对更新后的蛙群中的蛙排序,记录更新后的蛙群最优蛙为 \mathbf{P}_g 。

7) 检验是否满足终止条件,若满足,则停止迭代,输出全局最优粒子位置 \mathbf{P}_g 及其对应的适应值,否则转到步骤 2)。

3 仿真实验

3.1 测试函数

本文选用 8 个基准测试函数^[17]来测试算法的性能,见表 1,其中 $f_1 \sim f_4$ 是单模态函数,在给定搜索范围内只有 1 个极值点,主要检验算法的收敛速度和寻优精度, $f_5 \sim f_8$ 是多模态函数,在给定搜索范围内有多个局部极值点,主要考察算法的全局搜索能力和逃离局部最优能力。

表 1 8 个基准测试函数

Table 1 Eight benchmark functions used in this paper				
函数序号	函数名	范围	最优值坐标	最优值
f_1	Sphere	$[-100,100]$	$(0,0,\cdots,0)$	0
f_2	Schwefel.2.22	$[-10,10]$	$(0,0,\cdots,0)$	0
f_3	Schwefel.1.2	$[-100,100]$	$(0,0,\cdots,0)$	0
f_4	Quadric Noise	$[-1.28,1.28]$	$(0,0,\cdots,0)$	0
f_5	Rastrigin	$[-5.12,5.12]$	$(0,0,\cdots,0)$	0
f_6	Ackley	$[-32,32]$	$(0,0,\cdots,0)$	0
f_7	Griewank	$[-600,600]$	$(0,0,\cdots,0)$	0
f_8	Penalized.1	$[-50,50]$	$(1,1,\cdots,1)$	0

3.2 与标准混合蛙跳算法在不同维度下的比较

维度差异对算法性能有显著性影响。为验证改进算法的寻优效果和稳定性,将 GC-SFLA 算法与标准 SFLA 算法在不同维度下进行实验。实验参数设置为:最大函数评估次数 5.0×10^5 ,青蛙个体总数 200,族群数为 20,每个族群的青蛙个体数 10,族群内的迭代次数 10,最大蛙跳步长为最大搜索范围的 0.4 倍。

考虑篇幅限制,选取 1 个单峰函数 f_1 和 3 个多峰函数 $f_5 \sim f_7$ 进行不同维度下的实验,为消除算法的随机性影响,算法独立运行 50 次,以最终的平均值作为算法的最后寻优结果,实验结果见表 2。表中 Mean、Std.Dev 表示在限定的评估次数下算法的平均最优适应值及标准差,平均最优适应值反映了限定的评估次数下算法的寻优精度,标准差反映了算法的稳定性和鲁棒性。

表 2 2 种混合蛙跳算法在不同维度下的寻优结果

Table 2 The optimization results of two shuffled frog-leaping algorithms under different dimensions

函数 序号	Mean±Std.Dev					
	D=10		D=30		D=50	
	SFLA	GC-SFLA	SFLA	GC-SFLA	SFLA	GC-SFLA
f_1	6.655 0e-007±	0.000 0e+000±	6.130 5e+000±	0.000 0e+000±	9.609 3e+001±	0.000 0e+000±
	8.774 4e-006	0.000 0e+000	7.965 1e+001	0.000 0e+000	5.304 9e+001	0.000 0e+000
f_5	5.12e+000±	0.000 0e+000±	5.12e+000±	0.000 0e+000±	5.12e+000±	0.000 0e+000±
	1.256 1e-014	0.000 0e+000	1.256 1e-014	0.000 0e+000	1.256 1e-014	0.000 0e+000
f_6	6.154 9e-005±	5.887 2e-016±	7.759 0e-001±	5.887 2e-016±	2.082 8e+000±	9.428 1e-002±
	5.719 5e-004	0.000 0e+000	3.912 7e+000	0.000 0e+000	2.640 8e+000	1.278 1e-001
f_7	5.480 9e-002±	0.000 0e+000±	1.611 5e-001±	0.000 0e+000±	9.167 3e-001±	6.505 6e-012±
	1.712 0e-001	0.000 0e+000	5.920 5e-001	0.000 0e+000	9.664 5e-001	2.277 8e-011

由表 2 可以看出,在不同的实验维度下,无论是解的质量还是算法的稳定性,GC-SFLA 算法较标准 SFLA 算法均有较大提高。 f_1 函数为单峰函数,在搜索区域内只有 1 个极值点,无论在何测试维度下,GC-

SFLA 算法均能寻找到最优解,但 SFLA 算法在不同维度下的测试结果差异大。针对多峰函数,标准 SFLA 算法的寻优结果均不理想,对 f_5 函数,不论在何测试维度下,算法的均值和方差都一样,且效果非常不

理想,但 GC-SFLA 算法不仅算法稳定性好,且都能寻找到最优解;对 f_7 函数,在不同测试维度下,标准 SFLA 算法寻优结果与最优结果差距大,但 GC-SFLA 算法均能在误差允许的范围内(如设置允许的误差范围为 10^{-10})达到最优; f_6 函数是一个带有指数项的连续、多峰值函数,对 f_6 函数,虽然 GC-SFLA 算法的寻优结果与最优解之间有差距,但较标准 SFLA 算法,寻优能力确有明显提高。

图 2 给出了 GC-SFLA 算法与 SFLA 算法在不同维度下对上述 4 个基准测试的进化过程曲线,图中横坐标计算次数的范围为 $0 \sim 5 \times 10^5$,从图 2 可以看出,GC-SFLA 算法不仅寻优能力强,且收敛速度快,每种算法在较少次迭代后均达到较理想的寻优结果,而 SFLA 算法在较少的迭代后陷入局部极值。

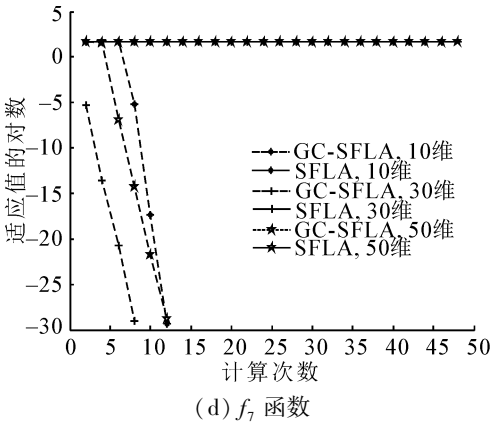
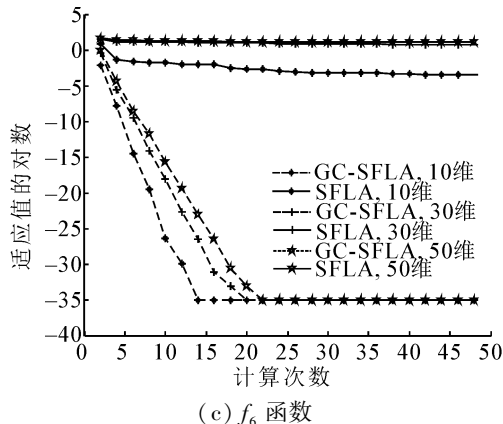
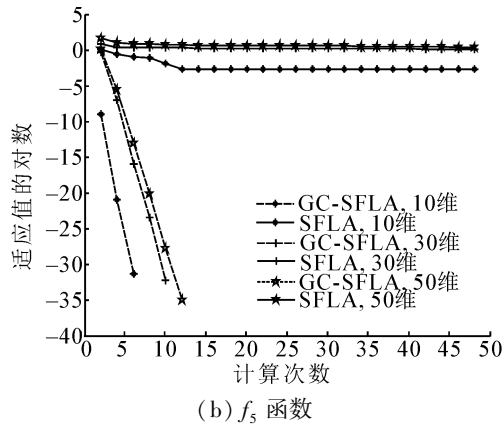
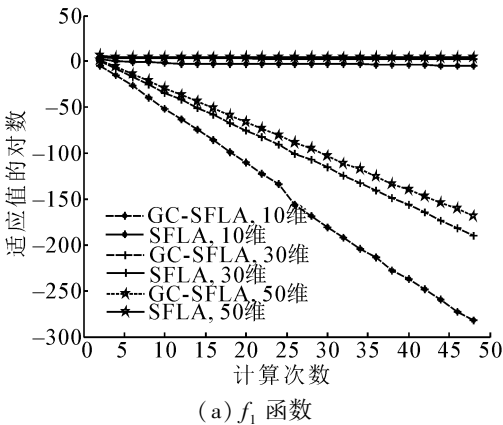


图 2 4 种测试函数的进化曲线
Fig. 2 The evolution curves of four benchmark functions

3.3 与新近提出的知名群智能算法进行比较

为进一步验证广义中心混合蛙跳算法的进化效果,将广义中心混合蛙跳算法和新近提出的知名群智能算法,如 M. M. Eusuff 等^[1]提出的标准混合蛙跳算法(shuffled frog leaping algorithm, SFLA)、Zhan 等^[18]提出的自适应粒子群优化算法(adaptive particle swarm optimization, APSO)、Zhu 等^[19]提出的全局最优引导的人工蜂群算法(Gbest-guided artificial bee colony algorithm, GABC)、汤可宗等^[16]提出的双中心粒子群优化算法(double center particle swarm optimization, DCP-SO)和 Wang 等^[20]提出的多策略集成的人工蜂群算法(multi-strategy ensemble artificial bee colony algorithm, MEABC)等进行比较。GC-SFLA 与 SFLA 算法参数设置参见 3.2 节,其他群智能算法的参数设置参见对应文献,最大函数评估次数 2.0×10^5 。表 3 给出了 GC-SFLA 与其他群智能算法在 30 维时的寻优结果对比。表 3 的数据显示,在 8 个测试函数中,GC-SFLA 算法相比于 SFLA、DCPSO、GABC、MEABC 等 4 种算法得到的种群均值和方差均有明显的优势;与 APSO 算法相比仅在 f_8 函数上表现较 APSO 差,在另外 7 个测试函数中,均有较好的表现。

为了进一步比较这 6 种算法的测试结果,对 6 种算法进行 t 检验,表 4 是 GC-SFLA 算法和其他 5 种算法在 8 个函数的 t 检验结果。 t 检验的分位数为单侧 0.05,自由度为 30,查表得到 t 检验的临界值为 1.697,即当 t 值大于这个值时,2 种算法存在显著性差异。用“ $w/t/l$ ”表示 GC-SFLA 算法与所选算法相比在 w 个函数上优于该算法, t 个函数上无明显差异, l 个函数上差于该算法。

从表 4 中数据可知,GC-SFLA 算法与标准 SFLA 算法、DCPSO 算法相比,在 5 个测试函数上表现出较明显的优势,在 3 个测试函数上无显著差异;与 AP-

SO 和 GABC 算法在 8 个函数的仿真实验相比,除了在 f_1 函数上无显著差异,在 f_8 上有明显劣势外,在其他 6 个函数上,GC-SFLA 算法有着很大的优势。另外 GC-SFLA 算法与 MEABC 相比,除在 f_8 上有明显劣势,在 f_5 和 f_7 上无显著差异外,在其他 5 个函数上均有明显的优势。

表 3 GC-LSFLA 与新近的知名群智能算法寻优结果对比

Table 3 Comparison between the GC-LSFLA and the recently proposed swarm intelligence algorithms							
测试函数	指标	SFLA	APSO	DCPSO	GABC	MEABC	GC-SFLA
f_1	Mean	5.41e-1	1.45e-150	7.31e-230	4.70e-16	1.55e-56	1.99e-277
	Std.Dev	6.40e-1	5.73e-150	0	2.57e-15	3.67e-55	0
f_2	Mean	7.84e0	5.15e-84	1.94e-85	1.26e-15	1.08e-29	1.21e-111
	Std.Dev	2.37e1	1.44e-83	3.07e-84	6.90e-15	2.21e-28	1.24e-110
f_3	Mean	1.00e2	1.0e-10	2.22e-22	5.49e3	8.31e3	5.81e-86
	Std.Dev	1.03e2	2.13e-10	6.50e-21	3.00e4	7.54e3	7.62e-85
f_4	Mean	2.17e3	4.66e-3	3.34e-4	5.00e-2	2.18e-2	2.20e-4
	Std.Dev	4.19e-3	1.7e-3	1.01e-3	2.74e-1	3.77e-3	2.15e-4
f_5	Mean	5.12e0	5.8e-15	0	0	0	0
	Std.Dev	1.25e-14	1.01e-14	0	0	0	0
f_6	Mean	7.80e-3	1.11e-14	2.96e-15	2.91e-14	4.13e-14	5.88e-16
	Std.Dev	3.81e-2	3.55e-15	0	1.59e-13	2.17e-15	0
f_7	Mean	1.45e-1	1.67e-2	0	3.70e-17	0	0
	Std.Dev	5.67e-1	2.41e-2	0	2.08e-16	0	0
f_8	Mean	3.03e-2	3.76e-31	1.42e-1	4.75e-16	3.02e-17	1.31e-2
	Std.Dev	2.77e-1	1.2e-30	1.88e-1	2.60e-15	0	2.15e-2

表 4 GC-SFLA 算法与其他 5 种算法的 t 检验结果
Table 1 The results of t test of GC-SFLA algorithm and the other 5 algorithms

测试函数	SFLA	APSO	DCPSO	GABC	MEABC
f_1	+	=	+	=	=
f_2	+	+	=	=	=
f_3	+	+	=	=	+
f_4	+	+	=	=	+
f_5	+	+	+	+	+
f_6	=	+	+	=	+
f_7	=	+	+	=	+
f_8	=	-	+	-	-
$w/t/l$	5/3/0	6/1/1	5/3/0	6/1/1	5/2/1

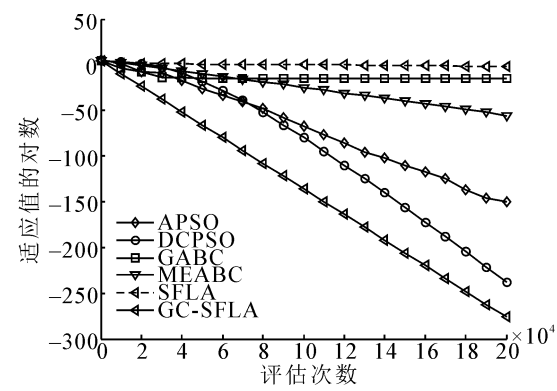
为进一步在统计意义上比较 6 种算法的性能,采用 Friendman 检验对结果进行分析。表 5 给出 SFLA、APSO、DCPSO、GABC、MEABC 和 GC-SFLA 6 种算法在 8 个测试函数上总体性能的平均排名。算法秩均值越小,性能越好,排名越高(排名最高的算法秩均值用粗体显示)。从表 5 中数据可知,GC-SFLA 明显高于其他 5 种算法。

表 5 6 种优化算法的 Friendman 检验结果
Table 5 The results of Friendman test of 6 algorithms

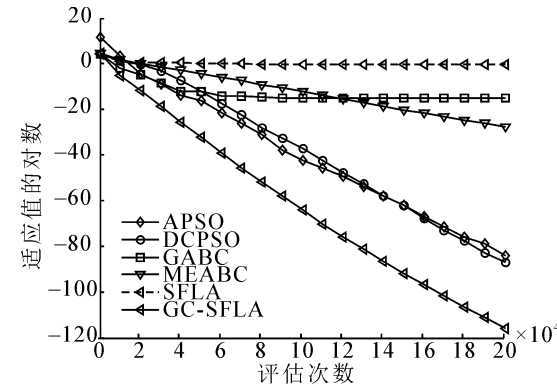
算法	秩均值
GC-SFLA	2.44
DCPSO	2.81
APSO	3.13
GABC	3.44
MEABC	3.56
SFLA	5.63

为清晰地描述 6 种优化算法在进化过程中的收敛速率。本文给出了 SFLA、APSO、DCPSO、GABC、MEABC 和 GC-SFLA 在 8 个测试函数 30 维上的收敛性能曲线图,如图 3 所示。由图 3 可知,GC-SFLA 算法能很好地增强算法逃离局部最优的能力,以及加速算法后期的收敛速度。GC-SFLA 算法在处理单模态函数时,优势相当明显,其中 f_1 、 f_2 、 f_3 3 个函数的进化曲线几乎呈直线下降。GC-SFLA 算法在处理复杂的多模态函数时,收敛速度也具有非常大的优势,特别是 f_5 和 f_7 2 个函数,在评估次数在 2 万次左

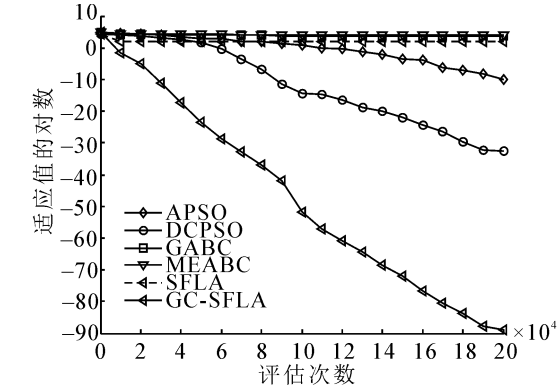
右就可以寻找到最优位置,其他 5 种算法却很容易陷入局部最优,造成收敛速度变慢,甚至停滞不前。



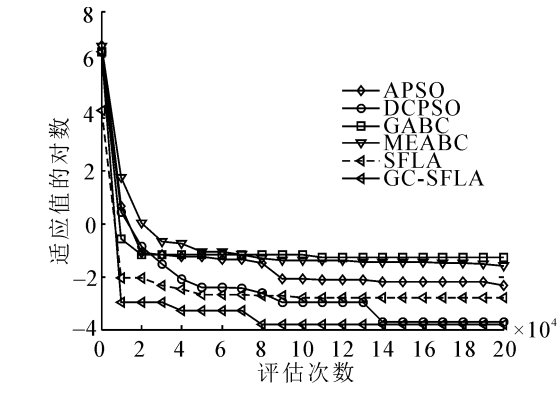
(a) f_1 函数



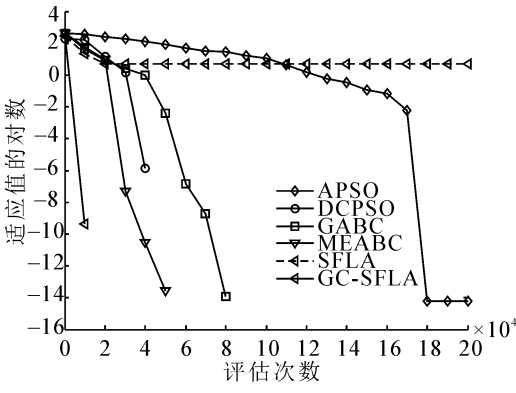
(b) f_2 函数



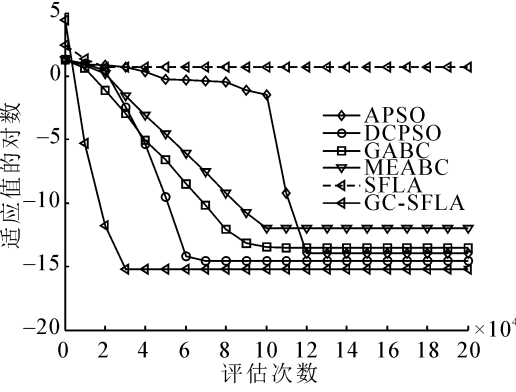
(c) f_3 函数



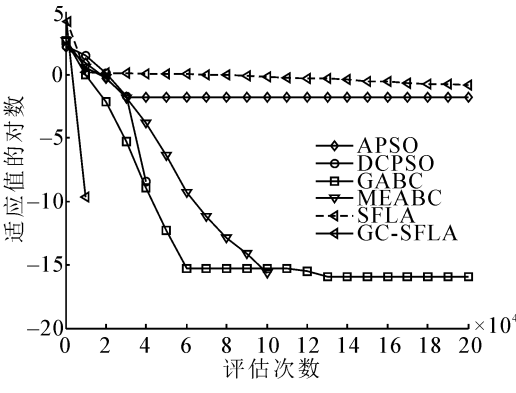
(d) f_4 函数



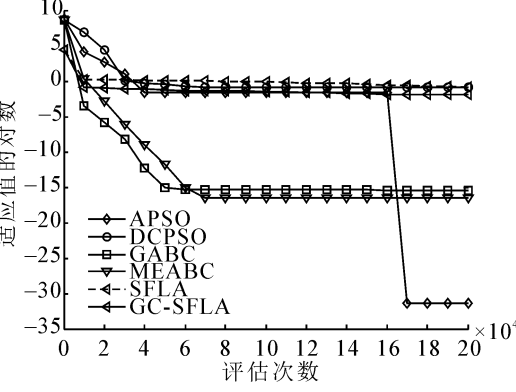
(e) f_5 函数



(f) f_6 函数



(g) f_7 函数



(h) f_8 函数

图 3 8 种测试函数的进化曲线

Fig. 3 The evolutionary curve of eight benchmark functions

4 结束语

本文在传统混合蛙跳算法的基础上,提出广义中心混合蛙跳算法。该算法分析传统混合蛙跳算法存在的族群之间学习能力不强的问题,引入广义中心青蛙的概念,设计广义中心策略以改进族群进化规则,该方法极大改善了族群之间的信息共享能力,增强了族群的多样性以及加快了算法的收敛速度。后续将加强算法在各类实际问题中的应用研究。

参考文献:

- [1] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [2] MOSCATO P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, Technical report C3P 826[R]. Pasadena, USA: California Institute of Technology, 1989.
- [3] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proceedings of the IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948.
- [4] RAHIMI-VAHED A, MIRZAEI A H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem[J]. Computers & Industrial Engineering, 2007, 53(4): 642-666.
- [5] 崔文华, 刘晓冰, 王伟, 等. 混合蛙跳算法研究综述[J]. 控制与决策, 2012, 27(4): 481-486, 493.
CUI Wenhua, LIU Xiaobing, WANG Wei, et al. Survey on shuffled frog leaping algorithm[J]. Control and Decision, 2012, 27(4): 481-486, 193.
- [6] FAN Tanghuai, LU Li, ZHAO Jia. Improved shuffled frog leaping algorithm and its application in node localization of wireless sensor networks[J]. Intelligent Automation & Soft Computing, 2012, 18(7): 807-818.
- [7] XIA Li, LUO Jianping, CHEN Minrong, et al. An improved shuffled frog-leaping algorithm with extremal optimisation for continuous optimisation[J]. Information Sciences, 2012, 192: 143-151.
- [8] ROY P, ROY P, CHAKRABARTI A. Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect[J]. Applied Soft Computing, 2013, 13(11): 4244-4252.
- [9] VAISAKH K, REDDY A S. MSFLA/GHS/SFLA-GHS/SDE algorithms for economic dispatch problem considering multiple fuels and valve point loadings[J]. Applied Soft Computing, 2013, 13(11): 4281-4291.
- [10] 罗雪晖, 杨烨, 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.
LUO Xuehui, YANG Ye, LI Xia. Modified shuffled frog-leaping algorithm to solve traveling salesman problem[J]. Journal on Communications, 2009, 30(7): 130-135.
- [11] NIKNAM T, FIROUZI B B, MOJARRAD H D. A new evolutionary algorithm for non-linear economic dispatch[J]. Expert Systems with Applications, 2011, 38(10): 13301-13309.
- [12] 张潇丹, 胡峰, 赵力, 等. 基于分子动力学模拟的改进混合蛙跳算法[J]. 数据采集与处理, 2012, 27(3): 327-332.
ZHANG Xiaodan, HU Feng, ZHAO Li, et al. Improved

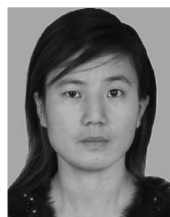
shuffled frog leaping algorithm based on molecular dynamics simulations[J]. Journal of Data Acquisition & Processing, 2012, 27(3): 327-332.

- [13] SUN Hui, ZHAO Jia. Application of particle sharing based particle swarm frog leaping hybrid optimization algorithm in wireless sensor network coverage optimization[J]. Journal of Information and Computational Science, 2011, 8(14): 3181-3188.
- [14] 汤可宗. 遗传算法与粒子群优化算法的改进及应用研究[D]. 南京: 南京理工大学, 2011: 1-100.
TANG Kezong. Modifications and application research on genetic algorithm and particle swarm optimization algorithm [D]. Nanjing, China: Nanjing University of Science & Technology, 2011: 1-100.
- [15] LIU Yu, QIN Zheng, SHI Zhewen, et al. Center particle swarm optimization[J]. Neurocomputing, 2007, 70(4/5/6): 672-679.
- [16] 汤可宗, 柳炳祥, 杨静宇, 等. 双中心粒子群优化算法[J]. 计算机研究与发展, 2012, 49(5): 1086-1094.
TANG Kezong, LIU Bingxiang, YANG Jingyu, et al. Double center particle swarm optimization algorithm[J]. Journal of Computer Research and Development, 2012, 49(5): 1086-1094.
- [17] YAO Xin, LIU Yong, LIN Guangming. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [18] ZHAN Zhihui, ZHANG Jun, LI Yun, et al. Adaptive particle swarm optimization[J]. IEEE Transactions on Systems Man, and Cybernetics—Part B: Cybernetics, 2009, 39(6): 1362-1381.
- [19] ZHU Guopu, KWONG S. Gbest-guided artificial bee colony algorithm for numerical function optimization[J]. Applied Mathematics and Computation, 2010, 217(7): 3166-3173.
- [20] WANG Hui, WU Zhijian, RAHNAMAYAN S, et al. Multi-strategy ensemble artificial bee colony algorithm[J]. Information Sciences, 2014, 279: 587-603.

作者简介:



赵嘉,男,1981年生,副教授,主要研究方向为计算智能、群体智能、智能信息处理。



吕莉,女,1982年生,副教授,主要研究方向为计算智能、目标跟踪。



樊棠怀,男,1962年生,教授,博士,主要研究方向为无线传感器网络、数据采集与处理、信息融合。