

## 基于网格化拉马克学习机制的差分进化算法

王丛佼<sup>a</sup>, 王锡淮<sup>a</sup>, 肖健梅<sup>a</sup>, 吴华锋<sup>b</sup>

(上海海事大学 a. 物流工程学院, b. 商船学院, 上海 201306)

**摘要:** 引入拉马克进化理念, 提出一种基于网格化拉马克学习机制的差分进化算法. 该算法在网格划分机制建立起的分布式搜索框架下, 采用单元格最优解保护机制、学习步长机制、解空间同仁机制和定矢变异机制组成拉马克学习模式. 仿真结果表明, 所提算法可以充分发挥拉马克学习的局部搜索能力, 又可有效避免早熟收敛, 其求解精度明显优于其他比较算法. 将所提算法应用于电力系统最优潮流计算问题, 获得了良好的优化效果.

**关键词:** 拉马克主义; 达尔文进化; 差分进化算法; 获得性遗传; 网格化拉马克学习

中图分类号: TP18

文献标志码: A

## Differential evolution algorithm based on gridded Lamarckian learning

WANG Cong-jiao<sup>a</sup>, WANG Xi-huai<sup>a</sup>, XIAO Jian-mei<sup>a</sup>, WU Hua-feng<sup>b</sup>

(a. Logistics Engineering College, b. Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China. Correspondent: WANG Cong-jiao, E-mail: qinxiaojiayi@hotmail.com)

**Abstract:** By introducing Lamarckian evolutionism, an improved differential evolution algorithm based on the gridded Lamarckian learning mechanism (DE-GLam) is proposed. Under a distributed search framework set by mesh generation mechanism, this algorithm integrates the cell optimum protection mechanism, learning step mechanism, solution space mechanism and directive variation mechanism to form the Lamarck learning mode. The simulation results show that the DE-GLam algorithm not only fully exerts the local search ability of Lamarckian learning mechanism, but also effectively avoids premature convergence, and the solving precision is superior to other comparison algorithms. The validity of the proposed method is illustrated by the optimal power flow calculation.

**Keywords:** Lamarckism; Darwinism; differential evolution; acquired genetic; gridded Lamarckian learning

### 0 引言

差分进化算法 (DE) 是一种通过引入独特差分变异算子进行迭代搜索的启发式算法<sup>[1]</sup>, 其结构简单、受控参数少、鲁棒性好, 已被成功应用于各类科学研究与工程领域中<sup>[2-3]</sup>. 但与其他进化算法类似, DE 算法也存在进化后期收敛速度慢、局部搜索能力不足和易陷入局部最优等问题. 为此, 学者们提出了许多改进方法, 主要集中于设计参数的自适应控制<sup>[4-5]</sup>、变异算子的改进<sup>[6]</sup>、引入其他优化算法<sup>[7-8]</sup>以及多种群进化机制<sup>[9]</sup>. 这些方法都在一定程度上改善了 DE 的性能, 但目前针对算法局部寻优机制的研究较少, 并且大多数改进往往只能解决搜索中某一方面的不足, 如何在提高算法局部搜索能力的同时不影响其逃出局部最优的能力还有待解决.

目前, 进化算法领域多数秉承了生物界普遍认同

的达尔文进化论思想, 参照其遗传进化模式进行算法设计. 与达尔文进化论同期存在的拉马克进化论, 认为生物在环境的直接影响下产生的学习和适应性行为, 可一定程度地遗传给子代, 在生物进化学说中亦占有一席之地<sup>[10]</sup>, 其思维模式对优化算法界也有相当的参考价值. 文献[11]设计了一种拉马克学习规则, 建立了拉马克遗传算法框架, 并在工程理论上证明了其具有较好的收敛性能和局部搜索能力; 文献[12]基于“学习潜能”的新概念提出了拉马克个体学习机制, 使局部学习的优势得到充分发挥; 文献[13]分别分析了基于拉马克主义和基于达尔文主义的进化学习策略的异同, 同时利用马尔可夫链理论证明了此类算法的收敛性, 但指出后者具有更强的局部逃逸能力. 综上所述, 基于拉马克进化论的学习机制具有加快、加强局部搜索的优势, 但其局部逃逸能力相对较弱.

收稿日期: 2014-03-10; 修回日期: 2014-09-17.

基金项目: 国家自然科学基金项目(51279099); 上海市教委科研创新重点项目(12ZZ158).

作者简介: 王丛佼(1984-), 女, 博士生, 从事进化计算、智能信息处理的研究; 王锡淮(1961-), 男, 教授, 博士生导师, 从事智能控制、智能信息处理、智能图像处理等研究.

针对以上问题, 本文通过将两种进化方式的思维模式有机融合, 提出一种基于网格化拉马克学习机制的差分进化算法 (DE-GLam), 在避免陷入局部最优的网格化搜索机制下, 借助拉马克学习模式提高对最优解的搜索能力. 实验结果表明, 本文提出的算法具有优良的全局优化性能.

## 1 拉马克主义简介

拉马克主义与达尔文主义主要的分歧在于对获得性遗传的看法. 虽然过去学界普遍认同后者, 但新兴的表观遗传学通过实验证实了前者在某些物种遗传现象中的合理性. 著名的达尔文主义者斯宾塞等认为只有把两种学说结合起来才能正确地解释进化<sup>[14]</sup>.

如图1所示, 颈长不同的长颈鹿觅食不同水平高度的植物茎叶 (为简化讨论, 假定长颈鹿探颈角度十分有限). 依照达尔文优胜劣汰的观点, 身高与茎叶最茂盛的水平高度齐平的品种B最适宜存活并繁殖; 其他颈长的鹿将不同程度因食物短缺而消亡, 若干代后, 品种A和C在种群中所占比例显著减少, B则显著增多. 当环境植物特征发生变化时, 各品种占比又会在演化过程中发生自然选择的调节. 而依照拉马克用进废退的观点, 同样是品种B最适宜存活; 但其他颈长的鹿将不同程度因食物短缺而作出努力适应环境的行为, A不断练习缩颈, 而C不断练习伸颈. 不论A和C的努力是否成功, 其为适应环境而付出的努力将以表观基因 (不改变基因组序的附加遗传物质) 的方式遗传给后代. 若干代后, A与C的后代中有一部分通过后天获得性遗传的方式, 在颈长属性中已贴近B.

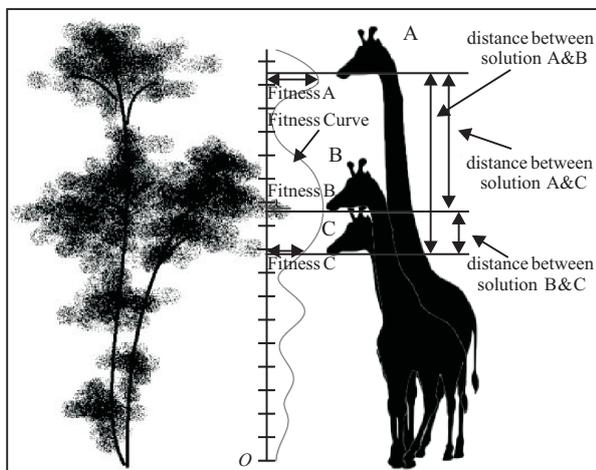


图1 获得性遗传的拉马克进化示意图

综上所述, 将两种主义联合起来的观点可表述为: 物种通过先天遗传及变异、后天适应并获得性遗传等多种方式, 将先天、后天所有的影响一起作用于生命演化. 对进化算法而言, 达尔文主义的变异模式通过扩大搜索空间以期找到更优的解, 而拉马克的获得性遗传模式可以赋予解集中具备一定条件的解通

过主动学习的方式, 依照较大的概率快速获得更高的适应性.

## 2 基本DE算法

DE算法是一种基于实数编码在连续空间进行随机搜索的进化算法, 其基本操作如下.

1) 种群初始化. 设定种群规模NP, 个体维数D, 进化代数  $G = 0, 1, \dots, G_{\max}$ , 则初始种群 (即  $G = 0$ ) 中的每个个体  $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}]$  在其自定义域范围内随机均匀产生, 即

$$x_{j,i,0} = x_{j,\min} + \text{rand}_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min}). \quad (1)$$

其中:  $x_{j,\max}$  和  $x_{j,\min}$  分别是个体变量  $X_{i,G}$  第  $j$  维的上界和下界,  $j = 1, 2, \dots, D$ ;  $\text{rand}_{i,j}[0, 1]$  是  $[0, 1]$  区间内均匀分布的随机数.

2) 变异操作. 利用种群中不同个体间的差分向量对每个个体进行扰动, 生成对应的变异个体  $V_{i,G}$ . 常用的DE/rand/1变异方式为

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}). \quad (2)$$

其中:  $r_1, r_2, r_3 \in [1, 2, \dots, NP]$  是不等于  $i$  且互不相同的随机整数, 代表不同个体的索引号;  $F \in (0, 2)$  是缩放因子, 控制差分向量的缩放程度.

3) 交叉操作. 本文采用二项式交叉操作. 目标个体  $X_{i,G}$  与变异个体  $V_{i,G}$  进行离散交叉, 从而产生试验个体  $U_{i,G} = [u_{1,i,G}, u_{2,i,G}, \dots, u_{D,i,G}]$ , 有

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{rand}(j) \leq \text{CR} \text{ or } j = \text{rand } n(i); \\ x_{j,i,G}, & \text{rand}(j) > \text{CR} \text{ and } j \neq \text{rand } n(i). \end{cases} \quad (3)$$

其中:  $\text{rand } n(i) \in [1, 2, \dots, D]$  是随机选择的维数变量索引, 以保证  $U_{i,G}$  至少有一位由  $V_{i,G}$  贡献, 而对于其他位, 由交叉概率因子CR决定,  $\text{CR} \in [0, 1]$ .

4) 选择操作. 只有适应度更优的个体进入到下一代种群中. 对于求解最小化问题, 该操作表示为

$$X_{i,G+1} = \begin{cases} U_{i,G}, & f(U_{i,G}) < f(X_{i,G}); \\ X_{i,G}, & f(U_{i,G}) \geq f(X_{i,G}). \end{cases} \quad (4)$$

其中:  $X_{i,G+1}$  是下一代的第  $i$  个个体,  $f$  是适应度函数.

## 3 基于网格化拉马克学习机制的DE算法

### 3.1 基本思想

拉马克学习是个体在表现型空间进行局部搜索的有力工具, 从某种程度而言, 直接在表现型空间进行局部搜索要好于在基因型空间进行交换和变异操作, 学习可以通过改变搜索空间的形状而提供更加优良的进化途径<sup>[15]</sup>. 本文在DE算法中融入了该学习环节, 以期加速对全局最优解的搜索. 在学习过程中, 学习目标太过分散, 不利于算法的收敛, 而过于集中地

向现有最优解靠拢, 则会大大牺牲种群多样性, 增加早熟的风险. 因此, 本文提出解空间网格化, 使拉马克学习在一定的局部区域内进行, 既避免学习的盲目性, 有效利用解空间的分布信息指导种群进化, 又保证多样化的学习目标, 从而预防种群陷入局部最优. 同时, 考虑到失败的学习会导致搜索资源的浪费, 为使学习效果充分发挥, 加快收敛, 需要考虑以下2个因素:

1) 地缘因素. 如图1所示, 鹿的品种C距离B较近, 通过拉马克学习, C较易以大概率搜索到更优的位置. 而A需要穿越较远的距离向B靠拢, 途中将经历变化丰富的适应度曲面, 学习过程会停留于什么样的适应度, 几乎失控. 此外, B是否为全局最优位置亦未可知, 则无法保证A进行学习后获得的适应度更优. 因此, 学习机制的发挥有必要进行地缘限制.

2) 适应度落差因素. 如图1所示, C与B的适应度差别显著, C通过向B学习可以大幅度提升其适应度. 而A与B的适应度差别不明显, 学习动力不足, 学习过程中也无法依概率确定获得更高适应度的几率. 可见学习机制的发挥与适应度的落差高度耦合.

### 3.2 学习机制

本文设计了网格划分机制、单元格最优解保护机制、学习步长机制、解空间同仁机制和定矢变异机制, 以组成网格化拉马克学习机制, 配合并加强差分进化算法的搜索、平衡算法的开采能力和勘探能力.

#### 3.2.1 网格划分机制

解空间需进行恰当的网格划分. 划分过细, 会增加冗余计算量, 影响算法收敛速度; 划分过粗, 群体的多样性无法保证, 使算法容易陷入局部最优. 综合考虑计算时间和寻优需求, 对于低维空间(10维以下), 以平均每维被均分为2个区间为宜; 对于高维空间(10维以上), 鉴于单元格属性值的计算成本, 可随机选取一部分维度进行划分, 即

$$\text{GridDimNum} = 10 \cdot [2 - 10/\text{DimNum}]. \quad (5)$$

其中: GridDimNum为高维空间随机选取的维度数目, DimNum  $\geq 10$ 为总维度数目. 式(5)在其定义域内为存在上限、增速递减的单增函数, 这样既保证了随着维数DimNum的升高而增加一定量的网格数, 保持划分精度, 又可避免因GridDimNum值过大而导致计算量剧增并影响后续学习机制的运行效果.

#### 3.2.2 单元格最优解保护机制

变异操作可以有效打散种群的局部集中, 但同时也容易丢失已经搜索到的优良解, 使搜索效率难以提升. 为了在一定程度上巩固单元格中拉马克学习的成果, 采取优势保护机制: 单元格中的历史最优解, 依照较大概率不参与差分变异环节, 而直接落入子代, 保

留概率 $p_{s_{m,n}}$ 为

$$p_{s_{m,n}} = \lambda + (1 - \lambda) \cdot \text{rand}(0, 1), \lambda \in (0, 1). \quad (6)$$

其中: rand(0, 1)为(0, 1)间的随机数;  $\lambda$ 为保留基数, 可根据实际求解问题在[0, 1]区间内自行设定. 当落入单元格的当代若干解中没有达到此单元格历史最优解水平时, 则该单元格的历史最优解以概率 $p_{r_{m,n}}$ 替换当代最劣解, 以期稳定各单元格的历史最优水平.

$$p_{r_{m,n}} = 1 - \exp\left(-\alpha \cdot \frac{f_{\text{worst}}(\text{cell}_{m,n}^G) - f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})}{f_{\text{worst}}(\text{cell}_{m,n}^{1\sim G}) - f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})}\right). \quad (7)$$

其中:  $f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})$ 为第 $m$ 行第 $n$ 列单元格(为方便说明, 以下均以2维为例说明)内历史最优解的适应度;  $f_{\text{worst}}(\text{cell}_{m,n}^G)$ 为该单元格内当前最劣解的适应度;  $f_{\text{worst}}(\text{cell}_{m,n}^{1\sim G})$ 为该单元格内历史最劣解的适应度;  $\alpha$ 为概率调节因子.

式(6)使得单元格历史最优解以区间 $(\lambda, 1)$ 内随机产生的概率值被保留. 显然, 当 $\lambda$ 较大时, 更利于优势保护; 而当 $\lambda$ 较小时, 更利于维持种群多样性. 本文为平衡两者, 设定 $\lambda = 0.5$ , 则保证单元格内的历史最优解以50%以上的概率被保留. 式(7)表明, 单元格内当前最劣解与历史最优解的适应度落差越大, 被历史最优解替换的概率 $p_{r_{m,n}}$ 越大, 从而在防止单元格内发生进化倒退的同时, 引导算法向趋于有利的方向搜索. 另外, 式(7)主项 $\exp[-x]$ 为负指数函数, 整体架构 $1 - \exp[-x]$ 在有意义区间内从0 ~ 1单调递增. 当 $x \in [0, 2]$ 时, 函数值在[0, 1]内显著变化; 当 $x > 2$ 时, 函数值无限趋近于1, 变化缓慢. 因此, 为了最大效用发挥函数的区分度, 以更好地自适应调节替换概率, 设定 $\alpha = 2$ . 为了消除量纲影响, 对适应度落差进行标准化处理, 则有 $\frac{f_{\text{worst}}(\text{cell}_{m,n}^G) - f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})}{f_{\text{worst}}(\text{cell}_{m,n}^{1\sim G}) - f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})} \in (0, 1)$ .

#### 3.2.3 学习步长机制

各单元格内实施当代解向最优解学习的机制, 学习过后的个体将与进行差分进化后的个体进行适应度比较.

如图2所示, 个体根据前述的地缘与适应度落差两个因素进行学习, 加快对局部空间的精细搜索.

个体的学习步长为

$$\text{Step}_{i+C_1,G}^{m,n} = \frac{d(x_{i+C_0,G}, x_{i+C_1,G})}{\text{diag} \frac{|f(x_{i+C_1,G}) - f(x_{i+C_0,G})|}{f_{\text{worst}}(\text{cell}_{m,n}^{1\sim G}) - f_{\text{best}}(\text{cell}_{m,n}^{1\sim G})} (x_{i+C_0,G} - x_{i+C_1,G})}. \quad (8)$$

其中:  $\text{Step}_{i+C_1,G}^{m,n}$ 表示第 $m$ 行第 $n$ 列单元格内编号 $i+C_1$ 个体 $x_{i+C_1,G}$ 的学习步长;  $d(x_{i+C_0,G}, x_{i+C_1,G})$ 表

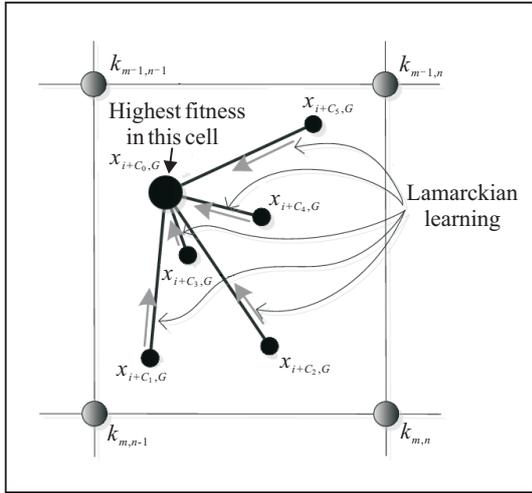


图 2 网格化拉马克学习机制示意图

示该单元格内当前最优解  $x_{i+C_0,G}$  与  $x_{i+C_1,G}$  之间的欧式距离,  $diag$  表示该单元格空间对角线欧式距离, 则  $d(x_{i+C_0,G}, x_{i+C_1,G})/diag$  表示地缘学习指数, 取值范围为  $[0, 1]$ ;  $|f(x_{i+C_1,G}) - f(x_{i+C_0,G})|$  表示  $x_{i+C_0,G}$  与  $x_{i+C_1,G}$  之间适应度落差的绝对值, 则  $|f(x_{i+C_1,G}) - f(x_{i+C_0,G})|/[f_{worst}(\text{cell}_{m,n}^{1\sim G}) - f_{best}(\text{cell}_{m,n}^{1\sim G})]$  表示适应度落差因素, 其取值范围为  $[0, 1]$ ;  $(x_{i+C_0,G} - x_{i+C_1,G})$  表示学习向量, 即个体  $x_{i+C_1,G}$  向当前最优解  $x_{i+C_0,G}$  进发的学习矢量。

### 3.2.4 解空间同仁机制与定矢变异机制

为了防止随机搜索造成的局部偏颇, 借助网格优势, 引入解空间同仁机制, 即解在冷僻单元格(当代落入解数目相对较少)加以保留, 同时在热门单元格(落入解数目相对较多)实施疏散的双项机制: 计算每个单元格的解落入频率, 落入频率越低的单元格中的解依照越大的概率不参与差分环节, 而直接落入子代; 落入频率越高的单元格中的解(除去其中的历史最优解)依照越大的概率进入差分环节, 有

$$\begin{cases} \text{Prob.save}_{m,n}^G = 1 - \text{Cong}_{m,n}^G, \\ \text{Cong}_{m,n}^G = \exp[-3 \cdot \text{Num}_{m,n}^G / \text{Num}_{m,n}^{G,\max}]. \end{cases} \quad (9)$$

其中:  $\text{Prob.save}_{m,n}^G(i+C_1)$  表示第  $G$  代第  $m$  行第  $n$  列单元格内个体直接落入子代的保留概率;  $\text{Num}_{m,n}^G$  表示落入该单元格的解的数量;  $\text{Num}_{m,n}^{G,\max}$  表示各单元格中落入解数量的最大值;  $\text{Num}_{m,n}^G / \text{Num}_{m,n}^{G,\max}$  表示相对拥挤程度的标准化指数;  $\text{Cong}_{m,n}^G$  表示第  $G$  代该单元格拥挤程度归一化指数。则根据式(9)可得个体进入差分环节的概率值。

在进行差分环节后, 引入定矢变异机制: 对所有单元格定义未开垦度(VL), 其值依据单元格在迭代历史中出现过的解的数目而定, 即

$$\text{VL}(\text{cell}_{m,n}^{1\sim G}) = \exp[-\text{Num}_{m,n}^{1\sim G}]. \quad (10)$$

其中:  $\text{VL}(\text{cell}_{m,n}^{1\sim G})$  表示第  $m$  行第  $n$  列单元格的未开垦度;  $\text{Num}_{m,n}^{1\sim G}$  表示迭代历史中落入该单元格的解的总数量。定矢变异操作即将当代所有解中的最劣解直接变异至未开垦度指数最高的单元格几何中心位置, 如图3所示。

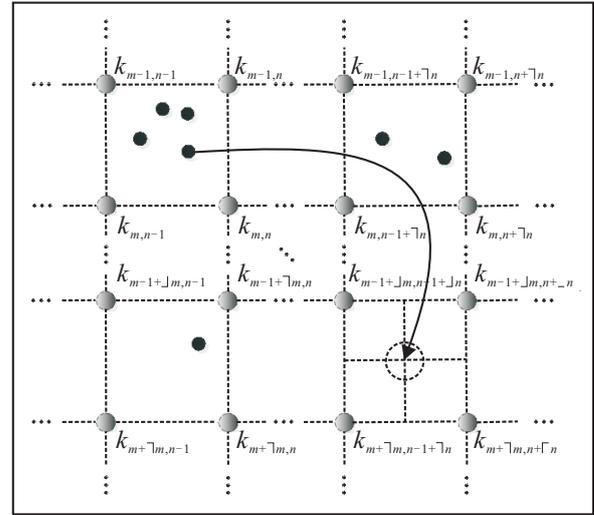


图 3 定矢变异操作

该机制利用定向变异最劣解来提高算法对解空间搜索的遍历性, 以提升差分进化算法的变异效能。另外, 对于差分操作后超出取值范围的不可行解, 定矢变异操作同样可依据式(10)将其定向变异至未开垦度指数最高的单元格几何中心, 成为一种简便高效的不可行解修复方法。

### 3.3 算法实现

DE-GLam 算法流程如下。

Step 1: 依据式(1)随机产生初始种群。

Step 2: 依据式(5)进行恰当的网格划分。

Step 3: 计算所有个体的适应度函数值, 更新各单元格的历史最优解和历史落入解数。

Step 4: 依据式(6)~(8)在网格内进行拉马克学习, 加强局部寻优与局部优势保留。

Step 5: 通过式(9)计算个体进入差分环节的概率, 再依据式(2)~(4)对这些个体进行差分变异、交叉操作。

Step 6: 选择操作。单元格内历史最优解以及其他未进入差分环节的个体, 依照拉马克学习机制实施概率性保留。对进入差分环节的个体, 计算其新的适应度值, 并与进行拉马克学习后的个体适应度值进行比较, 按照贪婪准则保留适应度值更优的个体。

Step 7: 依据式(10)将当代最劣解定向变异至未开垦度指数最高的单元格几何中心。

Step 8: 检验是否满足停止条件。若满足, 则停止迭代, 并输出当前最优解, 否则返回 Step 3。

## 4 实验与分析

本文对文献[16]中的10个经典Benchmark函数进行仿真实验, 所测函数的特性值如表1所示。

表1 测试函数的名称、维数、搜索空间和最优值

| 函数       | 名称            | 最大评价次数  | 维数 | 搜索空间             | 最优值 |
|----------|---------------|---------|----|------------------|-----|
| $f_1$    | Sphere        | 200 000 | 30 | $[-100,100]^D$   | 0   |
| $f_2$    | Schwefel 2.22 | 200 000 | 30 | $[-10,10]^D$     | 0   |
| $f_3$    | Schwefel 1.2  | 300 000 | 30 | $[-100,100]^D$   | 0   |
| $f_4$    | Rosenbrock    | 500 000 | 30 | $[-30,30]^D$     | 0   |
| $f_5$    | Noisy Quartic | 500 000 | 30 | $[-1.28,1.28]^D$ | 0   |
| $f_6$    | Rastrigin     | 200 000 | 30 | $[-5.12,5.12]^D$ | 0   |
| $f_7$    | Ackley        | 200 000 | 30 | $[-32,32]^D$     | 0   |
| $f_8$    | Griewank      | 200 000 | 30 | $[-600,600]^D$   | 0   |
| $f_9$    | Penalized 1   | 200 000 | 30 | $[-50,50]^D$     | 0   |
| $f_{10}$ | Penalized 2   | 200 000 | 30 | $[-50,50]^D$     | 0   |

表1中,  $f_1 \sim f_4$  为单模态函数, 主要用以测试算法的寻优精度, 检验算法的执行性能;  $f_5 \sim f_{10}$  为多

模态函数, 具有多个局部极小点, 可考察算法的全局搜索性能以及避免早熟的能力. 为了使DE-GLam算法的测试结果更具对比性, 将其与标准DE算法<sup>[1]</sup>、嵌入交叉爬山的局部搜索机制的DEahcSPX算法<sup>[17]</sup>、基于领域选择模型的DEGL算法<sup>[18]</sup>和基于小生境技术的NCDE算法<sup>[19]</sup>进行比较. 标准DE算法采用DE/rand/1变异策略, 设置 $F = 0.5$ ,  $CR = 0.9$ ; 其他比较算法均参照原文献的推荐设置, 其中文献[18]提出了4种DEGL变种, 本文取其表现性能最好的DEGL/SAW作为对比. 本文算法与标准DE采用同样的变异策略, 统一缩放因子和交叉因子, 设定种群规模 $NP = 200$ . 所有算法针对每个函数独立运算30次, 统计30次运行中函数评价次数达到最大值FEs(见表1)时的全局最优解精度. 表2为各算法在不同函数上的统计结果, 包括最优解平均值和标准差.

表2 DE-GLam和其他DE算法在30维Benchmark函数上的测试结果比较

| 函数       | DE/rand/1 Mean (Std Dev) | DEahcSPX Mean (Std Dev) | DEGL/SAW Mean (Std Dev) | NCDE Mean (Std Dev)   | DE-GLam Mean (Std Dev) |
|----------|--------------------------|-------------------------|-------------------------|-----------------------|------------------------|
| $f_1$    | 6.98 e-14 (2.10 e-14)    | 2.26 e-17 (2.31 e-17)   | 4.95 e-23 (3.83 e-21)   | 2.98 e-28 (2.15 e-27) | 6.73 e-51 (1.18 e-52)  |
| $f_2$    | 8.12 e-07 (6.01 e-07)    | 5.54 e-10 (3.29 e-10)   | 3.24 e-18 (2.53 e-16)   | 7.92 e-24 (1.99 e-24) | 5.38 e-37 (2.04 e-37)  |
| $f_3$    | 1.31 e+01 (3.29 e+00)    | 2.15 e+00 (1.72 e+00)   | 4.53 e-04 (2.90 e-04)   | 9.38 e-07 (6.12 e-07) | 2.14 e-07 (1.01 e-07)  |
| $f_4$    | 1.24 e+00 (1.31 e+00)    | 3.44 e-01 (3.84 e-01)   | 2.39 e-02 (2.64 e-01)   | 9.95 e-06 (2.34 e-07) | 4.36 e-26 (3.66 e-27)  |
| $f_5$    | 9.82 e-03 (4.05 e-04)    | 1.72 e-03 (2.53 e-04)   | 1.96 e-04 (2.15 e-04)   | 1.14 e-04 (1.54 e-04) | 4.24 e-07 (2.43 e-07)  |
| $f_6$    | 1.43 e+01 (3.42 e+00)    | 2.68 e+00 (2.91 e-01)   | 7.65 e-04 (1.03 e-04)   | 4.98 e-09 (4.53 e-08) | 5.84 e-14 (1.61 e-14)  |
| $f_7$    | 4.19 e-08 (3.26 e-08)    | 8.47 e-10 (2.17 e-10)   | 5.24 e-12 (2.13 e-12)   | 3.68 e-15 (5.97 e-16) | 1.29 e-17 (3.92 e-18)  |
| $f_8$    | 6.38 e-07 (3.84 e-08)    | 7.95 e-06 (8.26 e-07)   | 6.94 e-21 (2.23 e-21)   | 4.59 e-25 (7.03 e-26) | 0.00 e+00 (0.00 e+00)  |
| $f_9$    | 2.57 e-07 (6.03 e-06)    | 1.92 e-07 (2.45 e-07)   | 7.55 e-14 (6.39 e-13)   | 2.34 e-15 (2.87 e-15) | 6.27 e-21 (8.14 e-21)  |
| $f_{10}$ | 5.65 e-08 (6.32 e-10)    | 2.47 e-16 (4.21 e-16)   | 5.31 e-10 (6.17 e-10)   | 8.97 e-22 (7.51 e-23) | 4.65 e-23 (1.05 e-25)  |

由表2可知, DE-GLam算法在单模态函数和多模态函数上都取得了最好的优化结果, 其求得的最优解质量和收敛精度均优于其他4种比较算法, 且在Rosenbrock和Griewank函数上表现得尤为明显. Rosenbrock是公认难以求解的复杂病态函数, 其全局最优点位于一个平滑、狭长的抛物线形山谷内, 由于山谷内的值变化不大, 缺乏有效的指导信息, 使算法难以辨别搜索方向. 图4(a)给出了各算法求解该函数时的收敛曲线.

从图4(a)可以看出, 诸如标准DE、DEahcSPX算法在离其全局最优点很远的地方便停止了搜索. DEahcSPX虽然通过引入局部搜索在一定程度上提高了搜索能力, 但也使算法更易陷入局部最优, 因此无法搜索到其全局最优点. 同样的, DEGL算法和NCDE算法均表现不佳. 而DE-GLam算法通过拉马克学习机制实现了对局部区域的精细搜索, 同时引入网格划分机制建立起分布式搜索框架, 配合了有利于种群分布更均匀化的解空间同仁机制和提高对解空间遍历性的定矢变异机制, 使算法最大可能地避免陷入局部最优. 因此, 本文算法在一段时间的进化后能够

实现对最优解区域的准确定位, 并通过局部快速搜索提高收敛速度. 为了进一步对比各算法的收敛速度, 本文也从多模态函数中选择了难以求解的典型函数 $f_6$ 、 $f_7$ 和 $f_8$ , 绘制出如图4(b)~图4(d)所示的收敛性能曲线. 其中:  $f_6$ 有非常多的局部最小点包围在其全局最小点附近, 且峰形呈高低起伏跳跃性的出现, 很容易使算法陷入局部最优;  $f_7$ 是由指数函数叠加上适度放大的余弦函数得到的, 其全局最小点距离局部最小点较远, 一般的算法很难越过干扰山谷, 而是在离最优区域较远的位置就停止搜索;  $f_8$ 是由旋转抛物曲面叠加上余弦成分得到的, 函数曲面密集的峰谷阵列同样带来了较大的搜索难度. 由图4(b)~图4(d)可知, DE-GLam算法在进化初期的收敛速度不具有优势, 但在中后期, 收敛曲线的下降速度明显加快. 究其原因, 主要是由于随着进化过程, 个体在网格中的聚集程度增加, 使学习机制更有效地发挥了局部寻优的作用, 并因其采用概率性保留和保持群体多样性的其他机制, 在一定程度上抑制了变异交叉操作在进化后期可能带来的退化现象以及早熟收敛现象, 进而加速了整个收敛进程. 显然, DE-GLam算法对

于 Griewank 函数能够很快就找到其全局最优位置, 对于其他多模态函数的收敛效果也优于其他 4 种比较算法, 显示出了该算法具有良好的持续优化性能和逃逸局部极值点的能力。

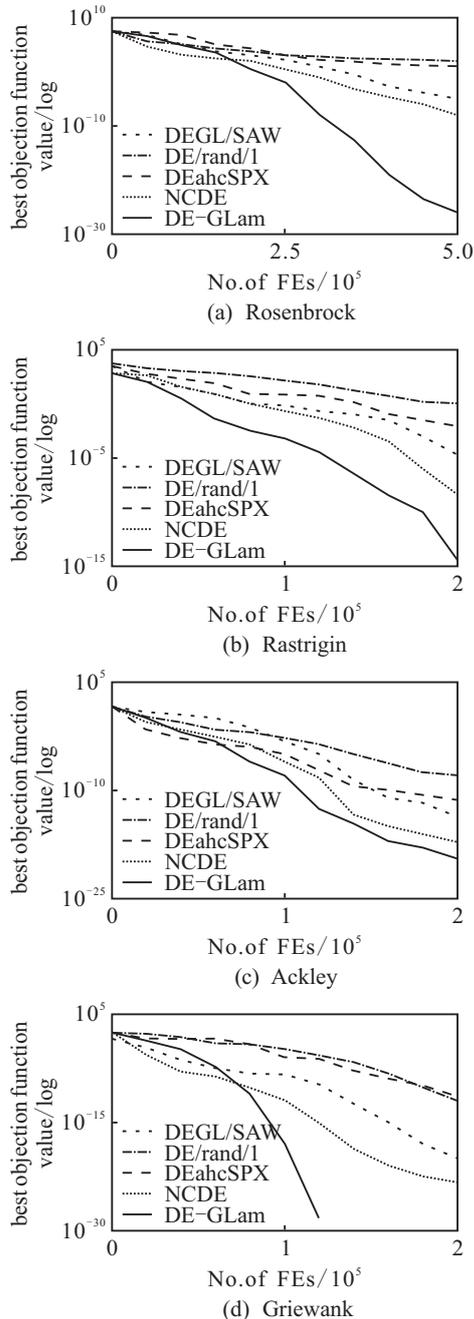


图 4 DE-GLam 与其他 DE 算法的收敛性能比较

## 5 应用实例

本文将 DE-GLam 算法应用于求解电力系统的最优潮流问题<sup>[20]</sup>。最优潮流是电力系统安全优化运行的核心问题, 其本质是一个复杂的非线性规划问题。一般采用发电费用最小为目标函数, 数学模型如下:

$$\begin{aligned} & \min f(\mathbf{x}, \mathbf{u}); \\ & \text{s.t. } g(\mathbf{x}, \mathbf{u}) = 0, \\ & \quad h(\mathbf{x}, \mathbf{u}) \leq 0. \end{aligned} \quad (11)$$

其中:  $\mathbf{u}$  表示控制变量向量, 包括发电机有功出力  $P_G$ 、发电机的机端电压  $V_G$  和变压器变比  $T$ ;  $\mathbf{x}$  表示状态变量向量, 包括平衡节点有功出力  $P_{LG}$ 、负荷母线电压  $V_L$  和发电机无功出力  $Q_G$ ;  $f(\mathbf{x}, \mathbf{u})$  表示发电费用; 等式约束  $g(\mathbf{x}, \mathbf{u})$  表示节点潮流方程; 不等式约束  $h(\mathbf{x}, \mathbf{u})$  包括发电机有功出力上下限约束、无功出力上下限约束和节点电压上下限约束等。

应用本文算法对 IEEE 30 节点系统进行多次最优潮流计算, 系统的详细参数和变量均参见文献 [21]。为了验证改进算法的有效性, 将其测试结果与标准 DE 和文献 [20] 中 MDE 算法的优化结果进行比较。MDE 算法的设置见原文献, 本文与其同样采用牛顿-拉夫逊法进行潮流计算, 并且采用罚函数处理式 (11) 中的约束问题, 通过增加对不可行解的惩罚项, 将该约束优化问题转换为无约束的优化问题, 具体可见文献 [20]。通过 100 次运算, 本文对比了 3 种方法发电费用的最大值、最小值和平均值, 结果见表 3。

表 3 IEEE 30 节点系统 100 次实验的结果比较

| 算法                  | 最大值/(USD/h) | 最小值/(USD/h) | 平均值/(USD/h) |
|---------------------|-------------|-------------|-------------|
| DE                  | 800.529     | 800.293     | 800.426     |
| MDE <sup>[20]</sup> | 800.378     | 799.449     | 799.768     |
| DE-GLam             | 798.795     | 798.732     | 798.749     |

从表 3 可以看出, 使用本文 DE-GLam 算法可以最大程度地节约发电成本, 其优化结果明显优于 DE 算法和 MDE 算法。DE-GLam 算法取得了最好的最优解质量, 并且能够稳定地收敛到最优解, 在电力系统最优潮流计算上的应用效果良好。

## 6 结 论

本文提出了基于网格化拉马克学习机制的差分进化算法, 设计了多种机制配合并加强差分进化环节对解空间的全局搜索。网格划分机制建立起分布式最优搜索框架, 使算法最大可能地避免陷入局部寻优; 单元格最优解保护机制可巩固局部搜索成果, 防止退化现象; 学习步长机制可将拉马克学习控制在最佳情态, 避免学习过度导致的过早收敛以及学习不足导致的局部搜索能力低下; 解空间同仁机制与定矢变异机制进一步使种群分布更均匀化, 增强算法对解空间的遍历性。对 Benchmark 函数的仿真结果表明, 本文算法可充分发挥拉马克学习的局部搜索能力, 并能有效防止种群被学习机制带入局部最优的困境。应用该算法对电力系统最优潮流问题进行求解, 获得了良好的优化效果, 具有一定的应用价值。

## 参考文献(References)

- [1] Kenneth P, Rainer S. Differential evolution—a practical approach to global optimization[M]. Berlin: Springer-Verlag, 2006: 133-152.

- [2] Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm[J]. *IEEE Trans on Systems, Man, and Cybernetics*, 2008, 38(1): 218-236.
- [3] 刘继栋, 王锡淮, 肖健梅. 基于十进制编码的差分进化算法在配电网重构中的应用[J]. *电力自动化设备*, 2012, 32(5): 94-97.  
(Liu J D, Wang X H, Xiao J M. Distribution network reconfiguration based on differential evolution algorithm using decimal encoding[J]. *Electric Power Automation Equipment*, 2012, 32(5): 94-97.)
- [4] Brest J, Grener S, Boskovic B. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems[J]. *IEEE Trans on Evolutionary Computation*, 2006, 10(6): 646-657.
- [5] 张春美, 陈杰, 辛斌. 参数适应性分布式差分进化算法[J]. *控制与决策*, 2014, 29(4): 701-706.  
(Zhang C M, Chen J, Xin B. Distributed differential evolution algorithm with adaptive parameters[J]. *Control and Decision*, 2014, 29(4): 701-706.)
- [6] 毕晓君, 刘国安, 肖婧. 基于新变异策略的动态自适应差分进化算法[J]. *计算机研究与发展*, 2012, 49(6): 1288-1297.  
(Bi X J, Liu G A, Xiao J. Dynamic adaptive differential evolution based on novel mutation strategy[J]. *J of Computer Research and Development*, 2012, 49(6): 1288-1297.)
- [7] Das S, Abraham A, Konar A. Particle swarm optimization and differential evolution algorithms: Technical analysis, applications and hybridization perspectives[J]. *Studies in Computational Intelligence*, 2008, 116: 1-38.
- [8] 熊伟丽, 徐迈, 徐保国. 基于差分蜂群算法的电力系统经济负荷分配[J]. *控制与决策*, 2011, 26(12): 1813-1817.  
(Xiong W L, Xu M, Xu B G. Differential bee colony algorithm for non-convex economic load dispatch[J]. *Control and Decision*, 2011, 26(12): 1813-1817.)
- [9] 王凌, 黄付卓, 李灵坡. 基于混合双种群差分进化的电力系统经济负荷分配[J]. *控制与决策*, 2009, 24(8): 1157-1166.  
(Wang L, Huang F Z, Li L P. Economic distribution of power systems based on hybrid differential evolution with double populations[J]. *Control and Decision*, 2009, 24(8): 1157-1166.)
- [10] Ong Y S, Keane A J. Lamarckian learning in memetic algorithm[J]. *IEEE Trans on Evolutionary Computation*, 2008, 8(2): 99-100.
- [11] 刘峰, 左晓静, 张坤. 一种新型遗传算法[J]. *计算机工程与应用*, 2010, 46(12): 46-50.  
(Liu F, Zuo X J, Zhang K. Novel genetic algorithm[J]. *Computer Engineering and Applications*, 2010, 46(12): 46-50.)
- [12] 栾志博, 黄其涛, 姜洪洲, 等. 遗传算法中两种学习机制的混合应用[J]. *系统工程与电子技术*, 2009, 31(8): 1985-1988.  
(Luan Z B, Huang Q T, Jiang H Z, et al. Mixed application of two learning mechanisms in genetic algorithm[J]. *Systems Engineering and Electronics*, 2009, 31(8): 1985-1988.)
- [13] 阎岭, 蒋静坪. 进化学习策略收敛性和逃逸能力的研究[J]. *自动化学报*, 2005, 31(6): 873-877.  
(Yan L, Jiang J P. Convergence and escape capacity research of evolution learning strategies[J]. *Acta Automatica Sinica*, 2005, 31(6): 873-877.)
- [14] Bird A. Perceptions of epigenetics[J]. *Nature*, 2007, 447(7143): 396-398.
- [15] Hinton G E, Nowlan S J. How learning can guide evolution[J]. *Complex System*, 1987, 1(3): 495-502.
- [16] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. *IEEE Trans on Evolutionary Computation*, 1999, 3(2): 82-102.
- [17] Noman N, Iba H. Accelerating differential evolution using an adaptive local search[J]. *IEEE Trans on Evolutionary Computation*, 2008, 12(1): 107-125.
- [18] Das S, Abraham A. Differential evolution using a neighborhood-based mutation operator[J]. *IEEE Trans on Evolutionary Computation*, 2009, 13(3): 526-553.
- [19] Qu B Y, Suganthan P N. Differential evolution with neighborhood mutation for multimodal optimization[J]. *IEEE Trans on Evolutionary Computation*, 2012, 16(5): 601-614.
- [20] 赵树本, 张伏生. 基于改进差分进化算法的电力系统最优潮流计算[J]. *电网技术*, 2010, 34(8): 123-128.  
(Zhao S B, Zhang F S. Solution of optimal power flow based on differential evolution and its modified algorithm[J]. *Power System Technology*, 2010, 34(8): 123-128.)
- [21] Zimmerman R, Gan D. MATPOWER: A Matlab power system[M]. New York: Cornell University, 1997: 34-36.

(责任编辑: 齐 霖)