

一种遗传算法与粒子群优化的多子群分层混合算法

金敏[†], 鲁华祥

(中国科学院半导体研究所 神经网络实验室, 北京 100083)

摘要: 针对遗传算法全局搜索能力强和粒子群优化收敛速度快的特点, 本文从种群个体组织结构上着手, 进行优势互补, 提出了一种遗传算法和粒子群优化的多子群分层混合算法(multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization, HGA-PSO). 算法采用分层结构, 底层由一系列的遗传算法子群组成, 贡献算法的全局搜索能力; 上层是由每个子群的最优个体组成的精英群, 采用钳制了初始速度的粒子群算法进行精确局部搜索. 文中分析论证了HGA-PSO算法具有全局收敛性, 并采用7个典型高维Benchmark函数进行测试, 实验结果显示该算法的优化性能显著优于其他测试算法.

关键词: 遗传算法; 粒子群优化; 分层混合算法; 多子群

中图分类号: TP18 文献标识码: A

A multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization

JIN Min[†], LU Hua-xiang

(Artificial Neural Networks Laboratory, Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China)

Abstract: To make use of the strong global search ability of the genetic algorithm and the high convergence rate of the particle swarm optimization, we combine these two algorithms and propose a multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization (HGA-PSO). This hybrid algorithm adopts a hierarchical structure; the base level is composed of a series of subgroups of Genetic algorithms, which provides the global search ability of the entire algorithm. The top level comprises all elite subgroups consisting of the best individual of each subgroup, which performs the accurate local search by using the particle swarm algorithm with cramped initial velocity. The global convergence analysis of HGA-PSO is given in this paper, and the performances of HGA-PSO have been evaluated through seven Benchmark functions. The experimental results show that the proposed method is superior to other related methods.

Key words: genetic algorithm; particle swarm optimization; hierarchical hybrid; multi-subgroup

1 引言(Introduction)

优化问题普遍存在于科学研究、工程技术和经济管理等诸多领域, 鉴于传统优化方法苛求问题的动力学信息(如连续、可微等)且不具备全局优化能力, 近年来以遗传算法和粒子群优化为代表的智能优化算法得到了迅速发展和广泛应用.

遗传算法(genetic algorithm, GA)是一类借鉴生物界的自然进化规律演化而来的随机搜索方法^[1], 由美国Holland教授于1975年首次提出. 算法采用“适者生存、优胜劣汰”的进化原则, 对包含可能解的种群反复使用遗传操作(包括选择、交叉和变异), 使种群不断进化, 以求得满足要求的最优解. 目前遗传算法已

成功应用到许多领域, 如优化设计^[2]、神经网络训练^[3]、模式识别^[4]、时序预测^[5]等.

粒子群优化(particle swarm optimization, PSO)是Kennedy和Eberhart于1995年提出的一种新型群体智能优化算法^[6-7], 源于对鸟群觅食行为的研究. 在PSO算法中, 每个粒子的位置代表着搜索空间的一个候选解, 粒子有位置和速度两个特征. 粒子位置坐标对应的目标函数值作为粒子的适应度. 算法首先初始化一群随机粒子, 然后通过迭代找到最优解. 在每次迭代中, 粒子通过跟踪两个“极值”来更新自己: 一个是粒子本身所找到的当前最优位置, 即个体极值 $p_{ib}(t)$; 另一个是整个种群所找到的当前最优位置, 即全局极

收稿日期: 2012-12-02; 收修改稿日期: 2013-03-31.

[†]通信作者. E-mail: jinmin08@semi.ac.cn; Tel.: +86 10-82304555.

基金项目: 中国科学院战略性先导科技专项基金资助项目(XDA06020700); 国家自然科学基金资助项目(61076014); 江苏省高校自然科学基金研究基金资助项目(10KJA50042).

值 $\mathbf{p}_{gb}(t)$. 标准PSO算法速度和位置的更新方程如下:

$$\mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 r_1 (\mathbf{p}_{ib}(t) - \mathbf{x}_i(t)) + c_2 r_2 (\mathbf{p}_{gb}(t) - \mathbf{x}_i(t)), \quad (1)$$

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

其中: \mathbf{v}_i 和 \mathbf{x}_i 分别表示第 i 个粒子的速度向量和位置向量; ω 为惯性权重(inertia weight); c_1 和 c_2 为加速常数(acceleration constant); r_1 和 r_2 为两个在 $[0, 1]$ 范围内服从均匀分布的随机变量.

粒子群优化简单易实现、收敛速度快, 在工程优化^[8-9]、图像处理^[10]等诸多领域得到了广泛的应用.

2 研究现状(Present situation of research)

遗传算法和粒子群优化都是一种基于群体的演化计算技术, 它们有自身的特点和优势, 同样也都存在某些缺陷和不足. 遗传算法有比较强的全局搜索能力, 但是局部搜索能力较差, 导致单纯的遗传算法比较费时, 进化后期搜索效率较低; 而粒子群优化实现简单, 收敛速度快, 但是也正因为收敛速度快, 导致种群多样性下降过快, 全局搜索能力欠佳, 易发生早熟收敛.

在面对小规模优化问题时, GA和PSO算法都有很成功的表现. 然而, 随着实际优化问题日趋复杂, 它们的缺陷也日益凸显, 在优化效率(时间)和求解质量上都显得“力不从心”, 因此改进它们的优化性能迫在眉睫. 鉴于遗传算法和粒子群优化有着几乎互补的优势, 人们想到将两者结合起来, 取长补短, 研究出性能更优的算法.

目前两者结合的方式主要有两种. 一种是将遗传算法的基本操作引入PSO算法, 来改善PSO算法的性能. Angeline引入选择机制改进PSO算法^[11], 能够加速收敛速度. 但从本质上看, 选择机制会加剧种群多样性的缺失, 对搜索能力的改善有害无益. Arumugam等人在PSO算法中引入了变异操作^[12], 变异操作的引入可以增加种群的多样性, 一定程度上提高了算法的收敛性能, 但不足以改变粒子群优化的本质特征.

另一种方式则是将两种算法以同等地位进行混合^[13-17]. 目前从已提出的一些混合算法中可以总结出串联^[15-16]和并联^[13-14, 17]两种主要的混合方式: 串联是指在每一代的进化过程中对所有个体都进行遗传操作和粒子群优化; 并联则是在每一代的进化过程中将种群一分为二, 遗传算法和粒子群优化各进化一半的个体. 如果算法采用相同的种群大小, 串联混合会比并联混合多一倍的计算量. 在搜索能力方面, 这两种方式混合的算法搜索能力都比单一的遗传算法或者粒子群优化要强. 但是这两种混合方式面临同样的问题, 即遗传算法和粒子群优化处于同等的地位, 两者之间分工不明确, 各自的优势没能得到充分发挥. 因此在遗传算法和粒子群优化的融合上还有很多值得挖掘的空间.

众所周知, 自然界物种进化由于受到地理条件的限制, 不可能在全球范围内全局进化, 物种只能分成若干个子群体分别进化, 偶尔有个别优秀的个体能够突破地域限制迁移到其他子群; 而在我国的中学教育中, 常采用“实验班”模式来达到因材施教的目的. 从每个班级中选择最优的几个学生组成实验班, 该班学生或在某方面非常有天赋、或对某些知识极度热爱, 他们需要不同于其他学生的教学方法. 采用“实验班”模式, 即可以最大程度地挖掘他们的长处, 也不会影响到其他学生的教育. 本文受到这些思想的启发, 提出了一种新的遗传算法和粒子群优化的混合算法. 算法采用分层结构, 底层由一系列的子群组成, 采用遗传算法进化, 贡献算法的全局搜索能力; 上层则是由每个底层子群的最优个体组成的精英群, 类似于学校中的实验班, 采用PSO算法进行精确局部搜索, 加快收敛速度. 本文算法从种群个体的组织结构上着手, 将全局搜索与局部搜索进行了分离, 即可以加快收敛速度, 又可以避免收敛过快引起的多样性下降削弱全局搜索能力, 有效克服了全局搜索能力与收敛速度的矛盾.

3 遗传算法和粒子群优化的多子群分层混合算法(Multi-subgroup hierarchical hybrid of genetic algorithm and particle swarm optimization)

HGA-PSO算法中个体的组织方式如图1所示, 底层由 N 个独立子群组成, 采用遗传算法进化, 主要负责全局搜索, 是整个算法的基石; 底层所有子群的最优个体构成了上层的精英群, 采用PSO算法进化, 主要负责精英个体的局部搜索, 加快算法收敛速度.

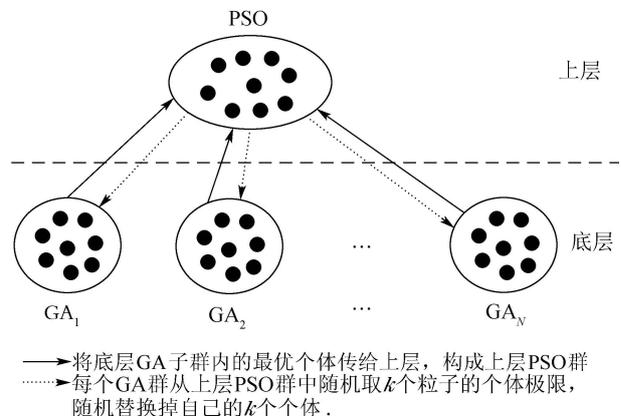


图1 HGA-PSO算法的种群个体组织方式

Fig. 1 Organization structure of individuals of HGA-PSO

算法首先随机初始化 N 个子群, 记它们为 GA_i , $i = 1, 2, \dots, N$. 每个子群独立运行各自的遗传算法, 一定代数之后, 将各自的最优个体取出构成上层的精英群, 记为PSO. 采用PSO算法进化精英群, 一定代数之后, 判断是否满足停止准则, 若满足, 则输出结果,

算法停止; 否则, 每个GA子群从上层的精英群中随机获得 k 个粒子的个体极值, 随机替换掉自己的 k 个个体. 至此, HGA-PSO算法的第1轮运行结束. N 个GA子群重新开始遗传算法操作, 不断循环, 直到满足停止准则为止. 图2为HGA-PSO算法的流程图.

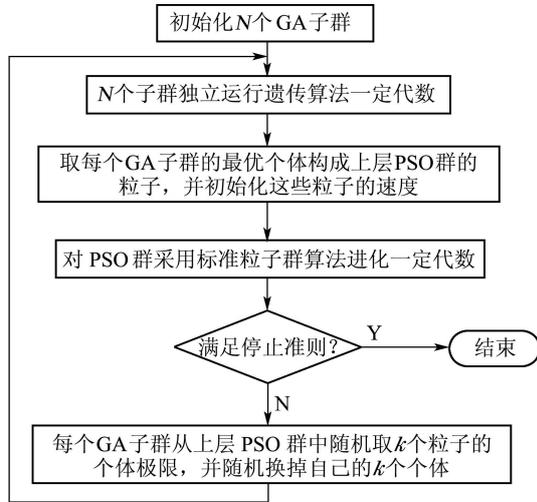


图 2 HGA-PSO 算法的流程图
Fig. 2 Flow chart of HGA-PSO

与其他遗传算法和粒子群优化的混合算法相比, HGA-PSO算法以遗传算法为主体, 保证了算法的全局搜索能力; 采用多子群结构可以更好地维持种群多样性, 对全局搜索有益, 而且还十分适合粗粒度并行计算; 采用粒子群算法对精英个体进行局部搜索, 收敛速度快; 分层结构的使用, 将底层负责全局搜索的个体和上层进行局部搜索的个体有效隔开, 即可以加快寻优速度, 又可以避免收敛过快引起的多样性下降削弱全局搜索能力.

3.1 HGA-PSO算法的收敛性分析(Convergence analysis of HGA-PSO)

Gunter Rudolph证明, 经典遗传算法只要在每代种群中都包含着历史最优解, 无论在选择算子前或后, 就可以收敛到全局最优解^[18], 称之为最优保留策略. 本文算法中进化底层子群的遗传算法采取最优保留策略, 在选择算子之前找出历史最优解, 如果它不在当前种群中时, 则用它随机取代当前种群中的任意一个个体. 因此底层子群使用的遗传算法具有全局收敛性.

理论已经证明^[19]: 在粒子群优化中, 假设 $\mathbf{p}_{ib}(t)$, $\mathbf{p}_{gb}(t)$ 在进化中保持不变, 则当 ω, c_1, c_2 满足:

$$\sqrt{2(1 + \omega - \varphi)^2 - 4\omega} < 2, \quad (3)$$

其中: $\varphi = \varphi_1 + \varphi_2$, $\varphi_1 = c_1 r_1$, $\varphi_2 = c_2 r_2$, 则PSO算法的 $\mathbf{x}_i(t)$ 收敛于 $\mathbf{p}_{ib}(t)$, $\mathbf{p}_{gb}(t)$ 的加权中心, 即

$$\mathbf{x}_i(t) \rightarrow \frac{\varphi_1 \mathbf{p}_{ib}(t) + \varphi_2 \mathbf{p}_{gb}(t)}{\varphi}. \quad (4)$$

考虑HGA-PSO算法, 设全局最优位置为 \mathbf{p}_{gbest} ,

因为底层子群的遗传算法具有全局收敛性, 当底层子群都收敛到全局最优解时, 上层的精英群中所有粒子都将处于全局最优位置, 粒子的个体极值和全局极值相同且在进化过程中保持不变, 都为 \mathbf{p}_{gbest} , 此时式(4)可化为

$$\mathbf{x}_i(t) \rightarrow \frac{\varphi_1 \mathbf{p}_{ib}(t) + \varphi_2 \mathbf{p}_{gb}(t)}{\varphi} = \frac{(\varphi_1 + \varphi_2) \mathbf{p}_{gbest}}{\varphi} = \mathbf{p}_{gbest}, \quad (5)$$

即上层粒子群算法也将收敛于全局最优.

因此当底层子群采用具有最优保留策略的遗传算法进化时, 上层的粒子群优化只要选择满足式(3)条件的 ω, c_1, c_2 , 则HGA-PSO算法具有全局收敛性.

3.2 底层子群的遗传算法流程(Process of genetic algorithm for underlying subgroups)

本文算法主要针对于连续量优化, 采用实数编码. 设子群已经初始化, 个体数目为 n , 个体维数为 d , 记个体为 $\mathbf{x}_i (i = 0, 1, \dots, n - 1)$, 中间代个体为 $\mathbf{x}'_i (i = 0, 1, \dots, n - 1)$, 个体的某一维分量为 $x_j (j = 0, 1, \dots, d - 1)$, 个体适应度值记为 $f_i (i = 0, 1, \dots, n - 1)$, 历史最优个体记为 \mathbf{I}_{best} , 其适应度值为 F_{best} , P_m 为变异概率. 设迭代代数 $t = 0$, 迭代 T 代停止. 具体步骤如下:

步骤 1 如果迭代代数 $t = T$, 则算法停止; 否则继续执行步骤2.

步骤 2 采用无偏联赛选择方法^[20], 选择出中间代的 n 个个体, 方程如下所示:

$$\mathbf{x}'_i = \begin{cases} \mathbf{x}_i, & f_i \leq f_k, \\ \mathbf{x}_k, & \text{其他,} \end{cases} \quad (6)$$

其中: $i = 0, 1, \dots, n - 1$, k 是随机产生的属于 $[0, n - 1]$ 的整数, 且 $k \neq i$.

步骤 3 对中间代个体进行完全交叉(交叉概率为100%)生成新一代个体, 方程如下:

$$\begin{cases} \mathbf{x}_i = \text{rand1} \cdot \mathbf{x}'_i + (1 - \text{rand2}) \cdot \mathbf{x}'_{i+1}, \\ \quad i = 0, 1, \dots, n - 2, \\ \mathbf{x}_i = \text{rand1} \cdot \mathbf{x}'_i + (1 - \text{rand2}) \cdot \mathbf{x}'_0, \\ \quad i = n - 1, \end{cases} \quad (7)$$

其中: $\text{rand1}, \text{rand2}$ 是 $[0, 1]$ 范围内服从均匀分布的随机变量.

步骤 4 采用非一致性变异, 按概率 P_m 对所有个体的每一维分量进行变异, 变异方程如下:

$$x_j = \begin{cases} x_j + \Delta(t, U_{\max}^j - x_j), & \text{rand} \leq 0.5, \\ x_j - \Delta(t, x_j - U_{\min}^j), & \text{其他,} \end{cases} \quad (8)$$

其中: $\text{rand} \sim \text{Uniform}(0, 1)$, $\Delta(t, y) = y \times (1 - r^{(1-t/T)^b})$, $r \sim \text{Uniform}(0, 1)$, $x_j \in [U_{\min}^j, U_{\max}^j]$, 常量 b 是变异时确定非一致性程度的一个参数, 它起着

调整局部搜索区域的作用. 一般 b 的取值范围为 $[2, 5]$, 取值越小, 变异的幅度越大, 即局部搜索区域相对大一些. 本文算法依靠遗传算法的全局搜索能力, 因此遗传算法的变异幅度就取得相对大些, 所以将常量 b 设为2.

步骤 5 计算个体的适应度值.

步骤 6 找出当前代中的最优个体, 如果优于历史最优个体, 则更新历史最优个体 I_{best} 及其适应度值 F_{best} ; 否则用历史最优个体随机替换掉当前代中的任意一个个体.

步骤 7 $t = t + 1$, 转到步骤1.

3.3 精英群粒子速度初始化(Velocity initialization of particles in elite subgroup)

在标准PSO算法里, 粒子速度向量 v_i 的每一维分量 $v_j (j = 0, 1, \dots, d - 1)$, 都会有一个速度区间 $[-V_{max}^j, V_{max}^j]$, 其中 V_{max}^j 是根据搜索空间设定的经验值. 初始化时对于所有粒子, 令 $v_j = r \cdot V_{max}^j$, 其中 r 是 $[-1, 1]$ 范围内服从均匀分布的随机变量.

本文算法采用标准PSO算法优化上层精英群, 但对粒子速度初始化做了些改变. 能够在精英个体附近进行精确的局部搜索, 提高寻优速度, 是本文算法对粒子群优化的定位. 如果仍然采用速度区间 $[-V_{max}^j, V_{max}^j]$ 初始化粒子, 则粒子速度方向各异, 呈发散状而达不到聚焦当前优解进行局部探索的目的. 因此本文在初始化精英群粒子速度时, 钳制粒子每一维速度的方向同时为正、或同时为负, 即同时采用速度区间 $[0, V_{max}^j]$ 或 $[-V_{max}^j, 0]$. 这时所有粒子初速度将集中在 $1/2^d$ 的空间, 粒子的自由度大大降低, 运动更聚集, 也更易相互合作进行精确局部搜索.

4 实验结果与分析(Simulation and analysis)

4.1 算法参数设置及Benchmark函数(Parameter settings and Benchmark functions)

为了验证本文提出的分层混合算法在全局搜索能力、寻优速度和稳定性方面的优势, 本文选用了遗传算法、粒子群算法(PSO)、一种采用并行混合方式的遗传算法粒子群优化混合算法(GA-PSO)^[14]和并行遗传算法中常用的环形孤岛模型遗传算法(Island-GA)^[21]进行对比实验. 文中引入了7个Benchmark函数问题(3个单模态、4个多模态)进行分析, 表1列出了这些Benchmark函数的定义、取值范围、全局最优值以及实验时的维数.

表2列出了各算法的参数, 算法中遗传算子(选择、交叉、变异)实现方式相同. 其中GA, PSO, GA-PSO算法采用单种群, 个体(粒子)数目设为512, 迭代1000代停止(函数评价次数 5.12×10^5). IslandGA采用32个子群, 每个子群16个个体(个体总数为512), 迭代1000代停止(函数评价次数 5.12×10^5), 每20代进行一次子群间的个体迁移. 迁移个体数目 k 设为2, 即迁移时每个子群从左邻居子群中获得2个最优个体, 并替换掉自己最差的2个个体. HGA-PSO采用30个子群, 每个子群16个个体, 每个子群的最优粒子组成了粒子数目为30的上层精英子群(个体总数为510). 迭代50轮停止, 每轮中遗传算法和粒子群算法各运行20代(函数评价次数 5.1×10^5), 每次替换底层子群中个体的数目 k 设为2. 优化时算法在搜索空间中随机初始化个体, 为了排除随机性, 实验以50次独立试验的统计结果进行分析. 实验基于Intel Core i5-2450 M 2.50 G, 4 GB, Windows 7, Visual Studio 2010环境, 算法采用C++语言编程实现.

表 1 Benchmark函数, $f_1 \sim f_3$ 为单模态函数; $f_4 \sim f_7$ 为多模态函数

Table 1 Benchmark functions, $f_1 \sim f_3$ are unimodal functions; $f_4 \sim f_7$ are multimodal functions

函数	表达式	搜索空间	维数	最小值/最优位置
Tablet	$f_1 = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	$(-100, 100)^n$	100	$0(0, \dots, 0)$
Quadric	$f_2 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$(-100, 100)^n$	100	$0(0, \dots, 0)$
Rosenbrock	$f_3 = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$(-2.048, 2.048)^n$	100	$0(1, \dots, 1)$
Griewank	$f_4 = \sum_i \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$(-600, 600)^n$	100	$0(0, \dots, 0)$
Rastrigin	$f_5 = 10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i))$	$(-5.12, 5.12)^n$	100	$0(0, \dots, 0)$
Ackley	$f_6 = -a \cdot e^{-b \cdot \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}} - e^{\frac{\sum_{i=1}^n \cos(c \cdot x_i)}{n}} + a + e^1$	$[-32.768, 32.768]$ $a = 20, b = 0.2$	100	$0(0, \dots, 0)$
Schaffer's	$f_7 = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \cdot [\sin(50 \cdot (x_i^2 + x_{i+1}^2)^{0.1}) + 1.0]$	$(-100, 100)^n$	100	$0(0, \dots, 0)$

表 2 算法的参数设置

Table 2 Parameter settings

算法	P_m	ω	c_1	c_2	k	种群数目	种群大小	迭代代数
GA	0.01	—	—	—	—	1	512	1000
PSO	—	[0.95,0.4]	2.0	2.0	—	1	512	1000
GA-PSO	0.01	[0.95,0.4]	2.0	2.0	—	1	512	1000
IslandGA	0.01	—	—	—	2	32	16	50/20
HGA-PSO	0.01	[0.95,0.4]	2.0	2.0	2	30	16	50/20

4.2 单模态函数优化对比实验(Performance comparison of unimodal functions)

表3是单模态Benchmark函数问题的优化性能数据表. 对于100维的Tablet函数, 在给定的迭代代数里, GA-PSO和HGA-PSO算法都100%找到了全局最优点; GA和IslandGA两个算法性能相似, 50次的寻优中都没有找到全局最优点, 但从均值上来看, 要优于PSO算法; 方差很小, 说明算法比较稳定. 反观PSO算法, 虽然有58%都找到了全局最优点, 但

50次运行结果的方差很大, 可以看出PSO算法非常不稳定, 对种群初始状态比较敏感. 对于100维的Quadric函数, HGA-PSO算法依旧100%找到了全局最优点, GA-PSO算法次之, 达到了96%, 方差很小, 说明算法稳定. PSO算法依旧表现出了不稳定的特点. 从表3中还可以看出HGA-PSO算法在Rosenbrock函数的寻优上有着非常优秀的表现, 当其他算法的结果还偏离最优值非常远的情况下, 它的结果已经非常接近全局最优值.

表 3 单模态函数的优化性能数据表

Table 3 Performance of optimization on unimodal functions

函数	算法	最小值	最大值	平均值	偏差
Tablet	GA	7.45e - 6	8.84e - 3	2.65e - 3	2.40e - 3
	PSO	0(58%)	1.19e + 2	5.64	1.80e + 1
	GA-PSO	0(100%)	0	0	0
	IslandGA	7.17e - 5	1.65e - 2	3.48e - 3	3.43e - 3
	HGA-PSO	0(100%)	0	0	0
Quadric	GA	2.18e - 4	7.52e - 2	1.97e - 2	1.87e - 2
	PSO	0(6%)	2.73e + 4	5.44e + 3	5.41e + 3
	GA-PSO	0(96%)	4.31e - 3	1.44e - 4	7.21e - 4
	IslandGA	9.13e - 5	8.15e - 2	1.90e - 2	1.93e - 2
	HGA-PSO	0(100%)	0	0	0
Rosenbrock	GA	9.87e + 1	9.89e + 1	9.88e + 1	4.93e - 2
	PSO	9.69e + 1	2.11e + 2	1.00e + 2	1.59e + 1
	GA-PSO	9.85e + 1	9.88e + 1	9.87e + 1	6.21e - 2
	IslandGA	9.85e + 1	9.89e + 1	9.88e + 1	7.07e - 2
	HGA-PSO	1.40e - 8	3.85e - 2	4.07e - 3	7.15e - 3

图3给出了3个单模态函数随着迭代代数的寻优情况, 图中使用的数据是算法50次独立优化的平均值, 横坐标是进化代数, 纵坐标是最优值的对数. 由于某些算法获得的最优值是0, 而0的对数是负无穷, 所以在纵轴上用“-Inf”表示, 为了图形清晰美观, 纵轴没有采用等比例. 图3中HGA-PSO算法的曲线出现了较多的跳变点, 这些点都出现在每轮迭代次数20的倍数上. 其出现的原因与HGA-PSO算法的流程相关. HGA-PSO算法每一轮中, 先是底层GA子群进化一定的代数, 完成后将其最优个体送到上

层组成精英群, 精英群再进化相同的代数. 而笔者取每一代的最优值是从精英群中取出的, 所以每一轮第一代的最优值往往与上一轮最后一代的最优值之间相差很大(中间GA子群进化了m代), 因此就会出现跳变点.

从图3(a)和图3(b)中可以看出HGA-PSO算法的寻优速度最快, GA-PSO算法次之. GA和IslandGA由于算法相同, 所以寻优速度接近, 但仔细观察能发现GA的寻优速度要稍快于IslandGA. 这是由于IslandGA采用多子群结构, 更利于种群中个体多样

性的维持,所以在收敛速度上就略逊.图中显示 PSO算法表现出来的平均寻优速度最慢,这是由于 PSO算法虽然收敛速度快,但全局搜索能力较弱,易发生早熟收敛,且算法不稳定,对初始状态很敏感,所以表现出来的平均寻优速度就很慢.从图3(c)中可以看出, HGA-PSO算法能够很好地解决Rosenbrock函数的优化问题. Rosenbrock函数是一个经典

复杂优化问题,它的全局最优点位于一个平滑、狭长的抛物线形山谷内.由于函数仅仅为优化提供了少量信息,使算法很难辨别搜索方向,找到全局最优点的机会微乎其微.从图3(c)中也可以看到, GA, PSO, GA-PSO以及IslandGA算法都陷入了局部极值出现了早熟收敛现象,但HGA-PSO算法却可以非常稳健地向着全局最优的方向收敛.

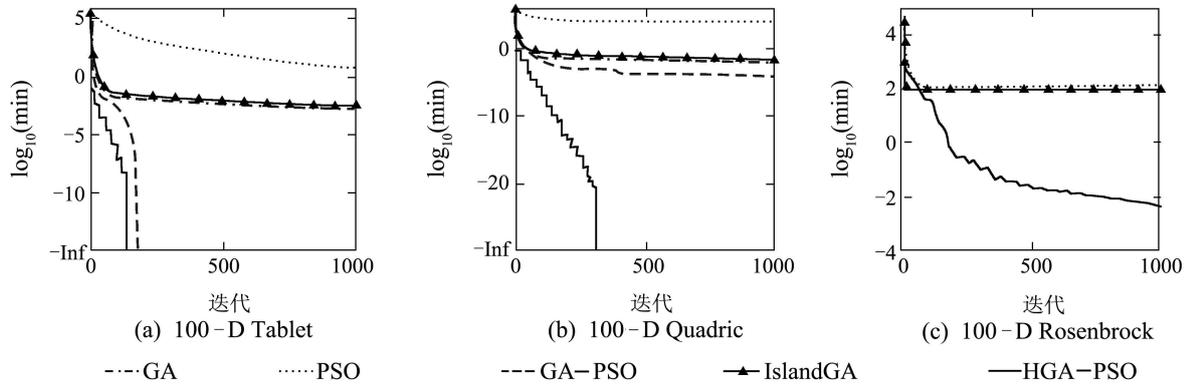


图3 单模态函数寻优情况对比图

Fig. 3 Comparison of performance of optimization on unimodal functions

4.3 多模态函数优化对比实验(Performance comparison of multimodal functions)

表4是4个多模态函数优化情况的数据表.从表4中可以看出性能最优的是HGA-PSO算法,它在优化4个函数时都100%地找到了全局最优点.而混

合了遗传算法和粒子群优化的GA-PSO算法性能优于GA, PSO以及IslandGA算法.

从均值上来看,性能最差的是PSO算法,其50次寻优的方差非常大,算法不稳定,对种群初始状态非常敏感.

表4 多模态函数的优化性能数据表

Table 4 Performance of optimization on multimodal functions

函数	算法	最小值	最大值	平均值	偏差
Griewank	GA	3.60e-6	1.88e-2	1.50e-3	2.72e-3
	PSO	0(78%)	9.10e+1	1.09e+1	2.94e+1
	GA-PSO	0(98%)	1.15e-4	2.29e-6	1.60e-5
	IslandGA	4.29e-5	1.05e-2	1.68e-3	2.11e-3
	HGA-PSO	0(100%)	0	0	0
Rastrigin	GA	6.1e-5	1.40e-2	3.47e-3	2.96e-3
	PSO	0(10%)	1.98e+2	5.87e+1	4.56e+1
	GA-PSO	0(98%)	5.49e-4	1.09e-5	7.69e-5
	IslandGA	7.35e-5	1.05e-2	3.58e-3	3.05e-3
	HGA-PSO	0(100%)	0	0	0
Ackley	GA	6.08e-4	1.77e-2	6.73e-3	3.87e-3
	PSO	0(70%)	1.21e+1	1.98	3.97
	GA-PSO	0(96%)	3.01e-3	9.81e-5	4.93e-4
	IslandGA	8.49e-3	3.36e-1	1.34e-1	8.02e-2
	HGA-PSO	0(100%)	0	0	0
Schaffer's f_7	GA	3.27e-1	5.12	3.43	1.05
	PSO	7.76e-3	7.46e+1	2.16e+1	2.13e+1
	GA-PSO	0(62%)	3.70	4.45e-1	9.25e-1
	IslandGA	1.43	4.88	2.99	8.27e-1
	HGA-PSO	0(100%)	0	0	0

图4给出了4个多模态函数随着迭代代数的平均寻优情况, 图中使用的数据是算法50次独立优化的平均值, 横坐标是进化代数, 纵坐标是最优值的对数. 由于某些算法获得的最优值是0, 而0的对数是负无穷, 所以在纵轴上用“-Inf”表示, 为了图形清

晰美观, 纵轴没有采用等比例. 图4中HGA-PSO算法曲线上也出现了跳变点, 其原因与图3一样. 从图4中可以看出HGA-PSO算法的寻优速度明显快于其他算法, GA-PSO算法次之, 最慢的依旧是PSO算法, 整体情况与单模态函数上的寻优情况基本一致.

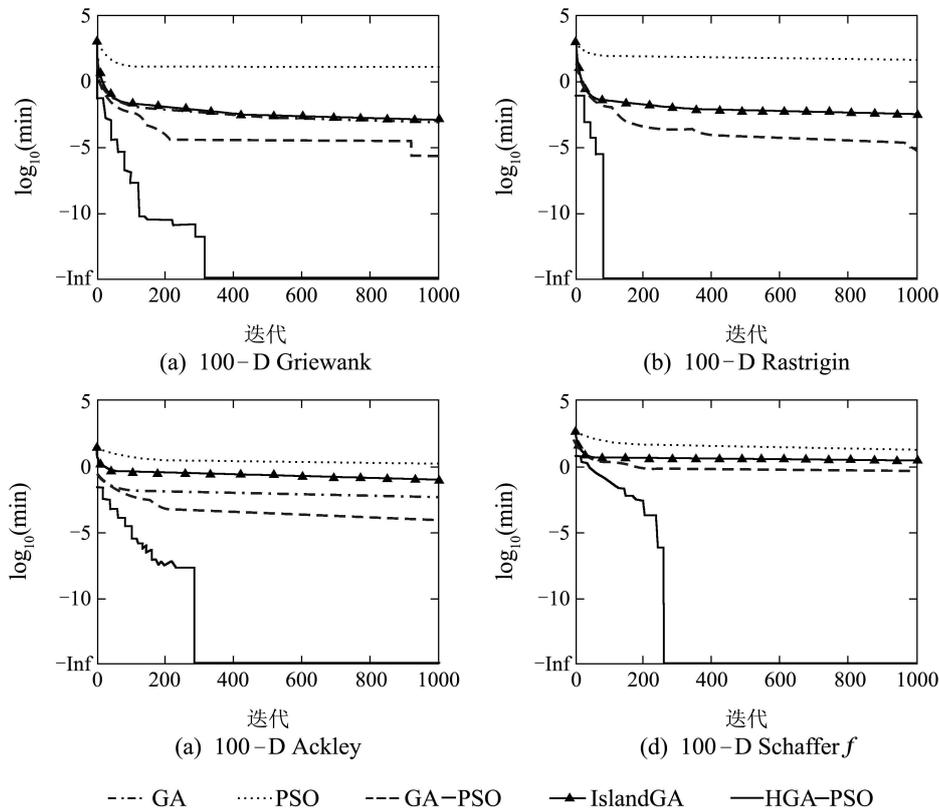


图 4 多模态函数寻优情况对比图

Fig. 4 Comparison of performance of optimization on multimodal functions

总体来说, 采用多子群分层混合的HGA-PSO算法具有很强的全局搜索能力, 在单模态和多模态问题上表现出的优化性能都显著优于其他算法, 不易陷入局部最优, 还能够解决Rosenbrock函数这种难以辨别搜索方向的复杂优化问题, 而且寻优速度快, 算法稳定性高.

5 结论(Conclusions)

本文提出了一种遗传算法与粒子群优化的多子群分层混合算法(HGA-PSO). 算法采用分层结构, 引入了具有全局收敛性的遗传算法进化底层子群, 贡献全局搜索能力; 上层是由每个底层子群的最优个体组成的精英群, 采用收敛速度快的粒子群算法进化, 并通过钳制精英群粒子的初始化速度, 使粒子群算法能够在精英个体附件进行精确局部搜索. 算法从种群个体的组织方式上着手, 将负责全局搜索的底层子群与负责局部搜索的精英群隔离, 既能够提高算法的收敛速度, 又可以避免收敛过快引起

的多样性下降削弱全局搜索能力.

文中分析论证了HGA-PSO算法具有全局收敛性, 并通过测试函数验证了所提出算法在全局搜索能力、寻优速度、稳定性等方面的优越性能. 下一步的主要工作将集中在HGA-PSO算法在复杂实际问题中的应用研究.

参考文献(References):

- [1] HOLLAND J H. *Adaptation in Natural and Artificial Systems* [M]. Cambridge: MIT Press, 1992.
- [2] 张袅娜, 张德江, 冯勇. 基于混沌遗传算法的柔性机械手滑模控制器优化设计 [J]. 控制理论与应用, 2008, 25(3): 451 - 455. (ZHANG Niaona, ZHANG Dejiang, FENG Yong. The optimal design of terminal sliding controller for flexible manipulators based on chaotic genetic algorithm [J]. *Control Theory & Applications*, 2008, 25(3): 451 - 455.)
- [3] ZOU X, HAN W, SHENG S Q. Application of genetic algorithm-neural network for the correction of bad data in power system [C] //2011 International Conference on Electronics, Communications and Control. New York: IEEE, 2011: 1894 - 1897.

- [4] KIM H D, PARK C H, YANG H C. Genetic algorithm based feature selection method development for pattern recognition [C] //2006 SICE-ICASE International Joint Conference. New Jersey: IEEE, 2006: 5382 – 5387.
- [5] LI P X, TAN Z X, YAN L L, et al. Time series prediction of mining subsidence based on genetic algorithm neural network [C] //Proceedings of the 2011 International Symposium on Computer Science and Society. Los Alamitos: IEEE, 2011: 83 – 86.
- [6] KENNEDY J, EBERHART R. *Particle Swarm Optimization* [M]. New York: IEEE, 1995.
- [7] SHI Y H, EBERHART R. *A Modified Particle Swarm Optimizer* [M]. New York: IEEE, 1998.
- [8] TAO Q, CHANG H Y, YANG Y. A rotary chaotic PSO algorithm for trustworthy scheduling of a grid workflow [J]. *Computers & Operations Research*, 2011, 38(5): 824 – 836.
- [9] WANG G L, ZHAO G Q, LI H P. Research on optimization design of the heating/cooling channels for rapid heat cycle molding based on response surface methodology and constrained particle swarm optimization [J]. *Expert Systems with Applications*, 2011, 38(6): 6705 – 6719.
- [10] GORAI A, GHOSH A. Hue-preserving color image enhancement using particle swarm optimization [C] //2011 IEEE Recent Advances in Intelligent Computational Systems. Piscataway: IEEE, 2011: 563 – 568.
- [11] ANGELINE P J. Using selection to improve particle swarm optimization [C] //1998 IEEE International Conference on Evolutionary Computation—Proceedings. New York: IEEE, 1998: 84 – 89.
- [12] ARUMUGAM M S, CHANDRAMOHAN A, RAO MVC. Competitive approaches to PSO algorithms via new acceleration co-efficient variant with mutation operators [C] //The 6th International Conference on Computational Intelligence and Multimedia Applications, Proceedings. Los Alamitos: IEEE, 2005: 225 – 230.
- [13] CHIAFENG J. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design [J]. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 2004, 34(2): 997 – 1006.
- [14] KAO Y T, ZAHARA E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions [J]. *Applied Soft Computing*, 2008, 8(2): 849 – 857.
- [15] LI S T, WU X X, TAN M K. Gene selection using hybrid particle swarm optimization and genetic algorithm [J]. *Soft Computing*, 2008, 12(11): 1039 – 1048.
- [16] MARINAKIS Y, MARINAKI M. A hybrid genetic—particle swarm optimization algorithm for the vehicle routing problem [J]. *Expert Systems with Applications*, 2010, 37(2): 1446 – 1455.
- [17] KUO R J, SYU Y J, CHEN Z Y. Integration of particle swarm optimization and genetic algorithm for dynamic clustering [J]. *Information Sciences*, 2012, 195: 124 – 140.
- [18] RUDOLPH G. Convergence analysis of canonical genetic algorithms [J]. *IEEE Transactions on Neural Networks*, 1994, 5(1): 96 – 101.
- [19] CLERC M, KENNEDY J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1): 58 – 73.
- [20] SOKOLOV A, WHITLEY D. Unbiased tournament selection [C] //Proceedings of the 2005 Conference on Genetic and Evolutionary Computation. New York: ACM, 2005: 1131– 1138.
- [21] CANTUPAZ E. *Efficient and Accurate Parallel Genetic Algorithms* [M]. Norwell: Kluwer Academic Publishers, 2000.

作者简介:

金敏 (1985–), 女, 博士研究生, 主要研究高性能计算、计算智能, E-mail: jinmin08@semi.ac.cn;

鲁华祥 (1965–), 男, 研究员, 博士生导师, 主要研究计算智能、信息处理, E-mail: luhx@semi.ac.cn.