

多元优化算法的渐近性分析

李宝磊¹, 吕丹桔¹, 张钦虎¹, 施心陵^{1†}, 安镇宙²

(1. 云南大学 信息学院, 云南 昆明 650091; 2. 玉溪师范学院 信息技术工程学院, 云南 玉溪 653100)

摘要: 本文提出了一种多元化智能个体分工明确、协同合作的超启发式智能优化算法——多元优化算法。多元优化算法通过交替的全局、局部搜索迭代对解空间搜索以逐渐逼近全局最优解。搜索个体按照分工不同可以分为全局搜索个体(全局元)和局部搜索个体(局部元)。全局元负责对整个解空间进行全局搜索以快速找到较优潜在解区域, 局部搜索元负责对各个潜在解区域进行局部搜索以提高解的质量。该算法具有两个特点: 分工明确的搜索策略不需要考虑均衡全局搜索和局部搜索, 能够保证局部搜索能力的同时加强全局搜索以避免陷入局部最优解; 全局、局部交替搜索保证了算法对全局最优解的渐近性。本文从理论上证明了算法的渐近性并且基于复杂多模态测试函数比较了几个优秀的进化算法。实验结果表明多元优化算法在渐近性方面优于其他几个比较的算法。

关键词: 多元优化算法; 渐近性分析; 全局元; 局部元; 多模态函数优化; 进化算法; 优化

中图分类号: TP301.6 文献标识码: A

On asymptotic property of multivariate optimization algorithm

LI Bao-lei¹, LÜ Dan-jü¹, ZHANG Qin-hu¹, SHI Xin-ling^{1†}, AN Zhen-zhou²

(1. School of Information Engineering, Yunnan University, Kunming Yunnan 650091, China;

2. School of Information Technology and Engineering, Yuxi Normal University, Yuxi Yunnan 653100, China)

Abstract: We propose a meta-heuristic intelligent optimization algorithm named as multivariate optimization algorithm, in which intelligent searchers have specific and defined roles in cooperation. To obtain the global optimal solution gradually, we search the solution space through alternate iterations of global exploration and local exploitation. According to different responsibilities, the searchers (atoms) can be divided into two kinds: the global atoms and the local ones. The global atoms explore the whole solution space to locate the potential areas rapidly. The local atoms exploit each potential area to improve the quality of the solution found by the global atom. The algorithm has two characters: on the one hand, the search strategy with clear division of responsibilities eliminates the need of balancing global exploration and local exploitation, which helps the global exploration to escape from local traps while ensuring the local exploitation. On the other hand, the alternate iterations of global exploration and local exploitation guarantee the asymptotic behavior of algorithm. The asymptotic property of multivariate optimization algorithm is proved theoretically. Extensive comparisons with some outstanding evolutionary algorithms are carried out based on eight complex multi-modal benchmark functions. Results show that this multivariate optimization algorithm is superior to the compared algorithms in asymptotic property.

Key words: multivariate optimization algorithm; asymptotic analysis; global atom; local atom; multimodal optimization; evolutionary algorithms; optimization

1 引言(Introduction)

启发式群智能优化算法以其操作简单, 收敛速度快而被应用在路径规划、无线传感网络等实际问题中^[1-2]。其中精英保留策略的遗传算法(genetic algorithm with one elitist, EGA)^[3], 自适应惯性权重的粒子群优化算法(particle swarm optimization algorithm with inertia weight, PSO-w)^[4]被普遍认为是经典有效的方法。然而实际优化问题的适应度面往往是复杂多峰的^[5], 这些算法容易陷入局部最优解而早熟收敛^[6]。

为了解决该问题, 许多学者通过引入新策略以提高算法对最优解的渐近性从而逃出局部最优解, 逐步逼近全局最优解^[7]。Liang教授等提出的全面学习粒子群优化算法(comprehensive learning particle swarm optimizer, CLPSO)^[8]利用所有粒子历史最优信息更新粒子的位置。CLPSO算法中, 粒子的不同维度向不同的粒子学习, 该策略加强了全局搜索能力有助于逃出局部最优解。Yang教授提出来的萤火虫算法(firefly algorithm, FA)^[9]是受启发于萤火虫之间的相互吸引

收稿日期: 2014-06-16; 录用日期: 2014-10-23.

†通信作者. E-mail: xlshi@ynu.edu.cn; Tel.: +86 15288439908.

国家自然科学基金项目(61261007, 61361010), 云南省自然科学基金重点项目(2013FA008)资助。

Supported by National Natural Science Foundation of China (61261007, 61361010) and Yunnan Provincial Natural Science Fund (2013FA008).

力与它们各自的亮度成正比与它们的距离成反比. 萤火虫算法中萤火虫之间的交流仅限于与其相近的个体, 这有利于局部搜索, 消除了由于存在多个最优解给搜索群带来的迷惑. 虽然这些算法在大部分优化问题中表现出了良好的性能, 但是它们都是从搜索策略出发要么加强全局搜索能力避免陷入局部最优解; 要么集中在各个局部的搜索避免搜索个体在多峰之间振荡而不收敛. 由于所有搜索个体职责相同导致算法的全局和局部搜索能力相互制约. 这些算法需要均衡全局和局部搜索才能获得较好的结果.

为了同时加强全局和局部搜索能力, 本文提出了一种多元化搜索个体分工明确、协同合作的超启发式智能优化算法. 由于搜索个体(元)分工不同具有多元化的特点, 因此本文把该算法命名为多元优化算法(multivariate optimization algorithm, MOA). 在本文提出的算法中, 多个搜索元基于高效的信息共享机制按照各自的分工对解空间进行搜索以逼近全局最优解. 搜索元按照分工不同可以分为全局搜索元和局部搜索元, 全局搜索元在解空间中随机生成负责对全局进行搜索并对发现的潜在解区域进行记忆, 局部搜索元在潜在解区域内进行局部搜索以找到该局部区域内的更优的解. 搜索元利用结构表完成高效的信息筛选、记忆和共享以进行分工合作.

本文第2节介绍了多元优化算法及其复杂度. 第3节对该算法的渐近性进行了理论证明. 第4节通过模拟实验从适应度值渐近曲线和最优解渐近精度两个方面比较了多元优化算法, 精英保留策略的遗传算法, 自适应惯性权重的粒子群优化算法, 全面学习粒子群优化算法, 萤火虫算法的性能. 实验结果表明多元优化算法在渐近性方面优于其他几个参与比较的算法.

2 多元优化算法(Multivariate optimization algorithm)

2.1 多元优化算法的实现(Implementation of multivariate optimization algorithm)

多元优化算法通过交替的全局-局部搜索对解空间进行迭代搜索并逐渐逼近全局最优解. 在迭代过程中, 多元优化算法的搜索个体被命名为元(atom), 具有职责不同, 分工明确的多元化特点. 按照其职责可以分为全局元和局部元, 全局元负责全局探索, 局部元负责局部开发.

全局元 G_a 根据式(1)在整个搜索空间中随机生成

$$\begin{cases} G_a = [h_1 \cdots h_d], \\ h_i = \text{unifrnd}(\min_i, \max_i), \end{cases} \quad (1)$$

式中: d 是问题的维度; \min_i 和 \max_i 分别为解空间第 i 维的下届和上届; $\text{unifrnd}(\min_i, \max_i)$ 函数返回一个均匀分布在 \min_i 和 \max_i 之间的随机数. 由随机

数 $h_i(i = 1, \cdots, d)$ 构成的全局元是一个随机向量, 其为解空间中的任意解的概率相等且大于零, 因此全局元具有全局搜索的能力. 可以看出, 全局元的生成与遗传算法(GA)和PSO算法中种群的初始化相似, 都是整个解空间中随机生成搜索个体实现全局搜索. 不同的是全局元只是种群的一部分且在每次迭代中生成, 迭代次数越多, 全局搜索覆盖率就越大. 因此全局元的全局搜索没有以消弱局部搜索为代价, 也不会随着种群多样性的减少而衰弱. 同时, 多元优化算法的全局搜索与为避免陷入局部最优解而提出的重启策略唯一不同的是: 全局元在每次迭代中都进行全局搜索, 而不是像重启策略算法陷入局部最优解后才进行再次全局搜索. 全局元仅对相对具有开发潜力的区域进行记忆, 而重启策略无法保证每次重启后种群都具有开发潜力的区间开始进行搜索. 因此, 全局元一旦发现比历史信息更优的区域, 立即对其进行局部开采, 反应速度快且保证了局部开采仅在相对有潜力的区域内进行.

笔者认为适应度值越好的全局元所在的区域, 越有可能包含最优解, 越具有开发潜力, 越值得进行高粒度的局部搜索. 全局探索的目的是对整个解空间进行大致预览以找到多个具有开发潜力的区域. 适应度较好的全局元被作为具有开发潜力的区域中心记录在结构体的全局链表中.

局部元根据公式(2)在以全局元 G_a 为中心 r 为半径的局部邻域内随机生成, 实现对该具有开发潜力区域的局部搜索.

$$L_a = G_a + r \times [l_1 \cdots l_d], \quad (2)$$

式中: $l_i(i = 1, \cdots, d)$ 是-1到1之间均匀分布的随机数. 可见, L_a 为该局部邻域内的任意解的概率相等且大于零. 当 G_a 不为该区域内的最优解时, 则 L_a 优于 G_a 的概率大于零. 局部开发的目的一方面为了找到该局部区域内较全局元更优的解并取代全局元作为新的潜在解区域中心实现向最优解逼近; 另一方面为了通过不同局部元组对不同局部区域独立搜索的方法来尽量避免搜索群在多个局部最优解之间震荡而无法收敛于全局最优解. 新局部元与历史局部元竞争, 较优的元作为历史信息进入局部链表保留下来. 如果局部元优于该区域的全局元, 则该局部元将取代全局元作为新的潜在解区域中心进入全局链表.

算法基于如图1所示的上三角型结构体完成对历史信息的记忆和共享来指导多元化搜索元分工合作对解空间进行全局探索和局部开发.

结构体由一个全局有序链表和 n 个局部有序链表构成. 图1中, n 为全局链表长度, 第 i 个局部链表长度为 $n - i + 1$.

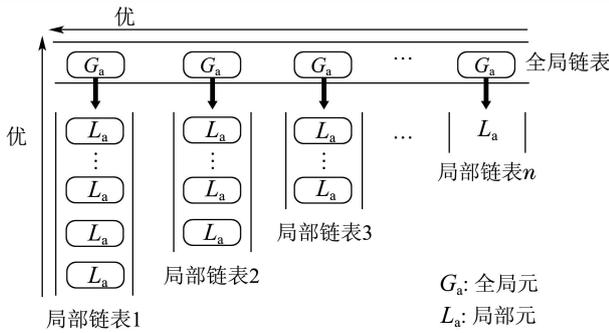


图 1 多元优化算法结构体

Fig. 1 Structure table of multivariate optimization algorithm

结构表中每个链表实际是按照搜索元适应度值优劣顺序记录搜索元信息的双向链表. 全局有序链表根据全局元的适应度值按照适应度从右到左越来越优的顺序记录全局元. 全局元位置越靠左, 其对应的局部队列的节点就越多, 其周围局部元就越多, 其所在局部区域内搜索粒度越高. 全局链表通过对全局元的管理实现对全局潜在解区域信息的记忆, 更新与共享. 每个局部有序链表根据全局元的适应度值按照适应度从上到下越来越优的顺序记录局部元. 局部链表负责其所在局部区域内历史信息的记忆与更新. 基于 C++ 的链表的实现以及插入、删除、排序操作的详细描述可以参考文献 [10].

多元优化算法的程序流程如图 2 所示.

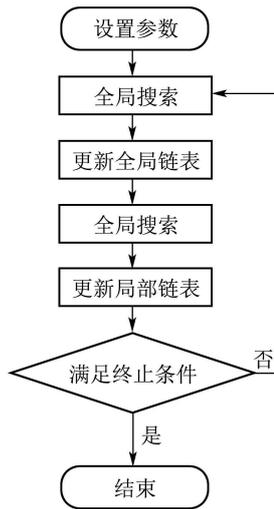


图 2 多元优化算法程序流程

Fig. 2 Program flow of multivariate optimization algorithm

由图 2 可知, 算法通过以下 6 个步骤实现对解空间的搜索:

步骤 1 设置每次迭代中全局元个数 m , 第 i 个局部区域内分配的局部元个数 m_i , 局部半径 r , 最大迭代次数 I_{max} , 问题的维度 d , 解空间第 i 维的上、下界 \min_i 和 \max_i .

步骤 2 根据式 (1), 在整个解空间随机生成全局元实现全局搜索, 以期快速找到多个具有潜力的解

区域.

步骤 3 更新全局链表的目的是有两个: 第一是对比历史全局元更优秀的新全局元进行记忆与共享; 第二是通过按照适应度值排序使得全局元越优秀, 其所在区域就分配越多的局部搜索元. 因此更新全局链表需要首先对新全局元进行评价, 然后根据历史和新全局元的适应度值对其排序, 最后把适应度值较好的全局元按照顺序放入全局链表. 为了保证全局链表的长度固定不变, 较差的全局元被丢弃. 适应度值较好的全局元作为潜在解区域的中心被记录在结构体的全局链表中.

步骤 4 在每个以全局元为中心的潜在解区域随机生成不同数量的局部元实现不同粒度的局部搜索. 新生成局部元个数与其对应的全局元下的局部链表节点个数相同. 全局元越优, 其在结构表中对应的局部链表节点越多则其所在区域生成的局部元就越多, 搜索粒度越高.

步骤 5 更新堆栈. 新产生的局部元与其对应堆栈中历史局部元竞争, 越优的局部元越能向上移动. 当局部元优于该局部范围内的全局元时, 其将取代全局元进入全局链表成为新潜在区域的中心. 这样局部元就能逐渐向更优的区域移动并逐渐向最优解逼近. 为了保证局部链表节点个数不变, 较差的局部元在竞争中被挤出结构体.

步骤 6 检查终止条件, 如果满足则终止算法运行. 如果条件不满足则转到步骤 2 开始新一轮的迭代, 直到最大迭代次数到达为止.

2.2 多元优化算法计算复杂度分析 (Complexity analysis of multivariate optimization algorithm)

PSO 和 GA 算法的优点之一就是计算复杂度低, 本文分析比较了 MOA 算法与 PSO 和 GA 算法计算复杂度. 当全局元数目为 n , 第 i 个潜在解内生成局部元数目为 $n - i + 1$, 优化问题的维度为 D 时, MOA 算法一次迭代中每个步骤的计算复杂度如表 1 所示. 全局搜索操作生成 n 个 D 维元的计算复杂度为 $O(nD)$. 更新队列操作的计算复杂度包括评价 n 个 D 维元的计算复杂度 $O(nD)$ 和 $2n$ 个全局元排序的计算复杂度 $O(\frac{2n(2n-1)}{2})$. 局部搜索操作生成 $\frac{n(n+1)}{2}$ 个 D 维元的计算复杂度为 $O(\frac{n(n+1)}{2}D)$. 更新第 i 个堆栈操作的计算复杂度包括评价 $n - i + 1$ 个 D 维元的计算复杂度 $O((n - i + 1)D)$ 和最多 $O(n - i + 1)^2$ 次比较, 则更新所有队列的总计算复杂度为 $O(\frac{n(n+1)}{2}D + \frac{1}{6}(2n^3 + 3n^2 + n))$. MOA 算法一次迭代的总复杂度为 $O((n^2 + 3n)D + \frac{1}{6}(2n^3 + 15n^2 - 5n))$. 为了比较 MOA, GA 和 PSO 的复杂度, 定义 3 种算法的种群大小

均为 $M = 0.5n^2 + 1.5n$. 因为 $2M$ 大于 n^2 , 所以 MOA 算法一次迭代的总复杂度低于 $O(2MD + (2(2M)^{\frac{3}{2}} + 30M - 5(2M)^{\frac{1}{2}})/6) \approx O(2MD + 0.5M^{\frac{3}{2}} + 5M)$.

表1中同时列出了, 当种群大小也为 M , GA和PSO的复杂度. 详细的GA和PSO算法复杂度分析可以参考文献[11-12].

表 1 MOA, GA和PSO计算复杂度
Table 1 Complexity of MOA, GA and PSO

算法	步骤	计算复杂度
MOA	全局搜索	$O(nD)$
	更新全局链表	$O(nD + \frac{2n \times (2n - 1)}{2})$
	局部搜索	$O(\frac{(n^2 + n)D}{2})$
	更新局部链表	$O(\frac{(n^2 + n)D}{2} + \frac{1}{6}(2n^3 + 3n^2 + n))$
	总复杂度	$O(2MD + 0.5M^{\frac{3}{2}} + 5M)$
GA	轮盘赌法选择 M 个基因	$O(M^2)$
	单点交叉	$O(MD)$
	单点变异	$O(M)$
	评价 M 个基因	$O(MD)$
	总复杂度	$O(2MD + M^2 + M)$
PSO	更新速度	$O(6MD)$
	更新位置	$O(MD)$
	评价 M 个粒子	$O(MD)$
	更新全局和个体历史最优解	$O(2M)$
	总复杂度	$O(8MD + 2M)$

由表1可知: 当种群大小为常数时, MOA, GA和PSO算法的复杂度主要取决于问题的维度 D , 它们的复杂度都是 D 的1次方; 当问题的维度 D 为常数, 且种群大小相等时, 算法复杂度主要取决于种群的大小, MOA, GA 和 PSO 算法的复杂度分别为 M 的1.5, 2和1次方. 根据大O的定义^[13]可得, 当问题的维度以及种群的大小增加时, MOA算法的计算复杂度增长速度与GA和PSO属于同一个数量级.

MOA算法中用于管理多元化个体的结构表增加了算法的空间复杂度, 但是目前计算机内存足够大, MOA算法通过充分利用了计算机大内存的优势来实现高效的信息记忆与共享.

3 多元优化算法渐近性(Asymptotic property of multivariate optimization algorithm)

本文基于最小化优化问题证明多元优化算法的渐近性. 给定有限非空集合 S 中存在最优解集合 $\Omega = \{x^* | x^* \in S, f(x^*) < \varepsilon\}$, 其中 $f: S \rightarrow \mathbb{R}$ 为目标函数, ε 是可接受的目标函数值. 最小化优化问题亦即在解空间 S 中找到至少一个使目标函数最小化的点 x^* .

定义 1 适应度渐近性. 令 k 时刻算法找到的最优解为 x_k^* , $\{d_k | d_k = f(x_k^*) - f(x^*), 1 \leq k \leq N\}$ 为优化算法所生成的非负随机过程. 如果当 $f(x_k^*) > f(x^*)$ 时, 存在一个正常数 τ , 使得 $E(d_{k+1}) \leq E(d_k)$

$-\tau$ 则称算法具有适应度渐近性.

定理 1 多元优化算法具有适应度渐近性. 对于多元优化算法所生成的非负随机过程 $\{d_k | d_k = f(x_k^*) - f(x^*), 1 \leq k \leq N\}$, 当 $f(x_k^*) > f(x^*)$ 时, 存在一个正数 τ , 使得 $E(d_{k+1}) \leq E(d_k) - \tau$.

证 由于多元优化算法通过优胜劣汰的机制决定搜索元进入或者退出结构体, 所以 $P(f(x_{k+1}^*) - f(x_k^*) > 0) = 0$. 又由于多元优化算法全局和局部元在搜索空间内随机生成且 $f(x_k^*) > f(x^*)$, 则 $P(f(x_{k+1}^*) = f(x_k^*)) \neq 1$. 由此可得 $P(f(x_{k+1}^*) - f(x_k^*) < 0) > 0$.

令 $E(f(x_{k+1}^*) - f(x_k^*)) = -\tau_{k+1}$. 显然 $\tau_{k+1} > 0$, 则

$$E(d_{k+1} - d_k) = E((f(x_{k+1}^*) - f(x^*)) - (f(x_k^*) - f(x^*))) = E(f(x_{k+1}^*) - f(x_k^*)) = -\tau_{k+1}.$$

即 $E(d_{k+1}) = E(d_k) - \tau_{k+1}$.

令 $\tau = \min\{\tau_N, \dots, \tau_1\}$ 则 $E(d_{t+1}) \leq E(d_t) - \tau$.

定理得证. 证毕.

定义 2 全局最优解渐近性. 令 k 时刻算法找到的离全局最优解最近的解为 x_k' . 对于优化算法找到的所有解与全局最优值之间最短距离 $D_k = |x_k' - x^*|$ 所生成的非负随机过程 $\{D_k, 1 \leq k \leq N\}$. 如果

当 $x'_k \neq x^*$ 时, 存在正数 ε , 满足 $E(D_{k+1}) \leq E(D_k) - \varepsilon$, 则称算法具有全局最优解渐近性.

定理2 多元优化算法具有全局最优解渐近性. 令 x'_k 为 k 时刻多元优化算法访问过的与全局最优值距离最近的解. 多元优化算法所生成的非负随机过程 $\{D_k, 1 \leq k \leq N\}$, 当 $x'_k \neq x^*$ 时, 存在一个正常数 ε , 使得 $E(D_{k+1}) \leq E(D_k) - \varepsilon$.

证 对于 k 时刻搜索元访问过的解集合 S_k , 经过全局和局部搜索运算 g 后生成新集合 S'_k , 可得到 $k+1$ 时刻搜索元访问过的集合 $S_{k+1} = S_k \cup S'_k$. 虽然搜索过程中适应度值最优的解不一定离全局最优解最近, 但是多元优化算法利用结构表记录了搜索历史信息. 所以访问过解离全局最优解的最短距离不可能增加即 $P(D_{k+1} - D_k > 0) = 0$. 当 $x'_k \neq x^*$ 时, 在第 $k+1$ 次搜索中, 全局搜索元以概率访问到比 x'_k 离全局元更近的解, 局部元以概率访问到离某一部最优解或全局最优解更近的解. 因此 $P(D_{k+1} - D_k = 0) \neq 1$, $P(D_{k+1} - D_k < 0) > 0$. 令 $E(D_{k+1} - D_k) = -\varepsilon_{k+1}$, 易证 $\varepsilon_{k+1} > 0$. 则 $E(D_{k+1}) = E(D_k) - \varepsilon_{k+1}$, 令 $\varepsilon = \min\{\varepsilon_N, \dots, \varepsilon_1\}$, 则 $E(D_{k+1}) \leq E(D_k) - \varepsilon$. 定理得证. 证毕.

4 数值仿真(Numerical experiments)

4.1 标准测试函数(Benchmark functions)

本文从参考文献中选择了8个具有多峰性和欺骗性的二维复杂测试函数用以比较5种算法的渐近性. 尽管这些函数维度低, 但是大部分智能优化算法容易陷入局部最优解而无法找到这些函数的全局最优解^[14-18]. 因此这些函数具备一定的复杂性来挑战新方法. 为了以分贝(dB)为单位比较各个方法的渐近性, 本文对部分测试函数增加了一个偏移量使其最优值为0. 各个测试函数表达式如下:

1) Michalewicz^[14].

$$f_1(X) = \sum_{i=1}^d \left(\sin\left(\frac{i \times x_i^2}{\pi}\right) \right)^{20} \sin x_i + 1.81.$$

搜索空间为 $x_1 \in [0, 5]$, $x_2 \in [0, 5]$, 全局最优解 $x_1 = 2.2$, $x_2 = 1.57$.

2) Langermann^[15].

$$f_2(x, y) = - \sum_{i=1}^5 c_i \cdot \cos(z_i \pi) \cdot e^{-\frac{z_i}{\pi}} + 5.163,$$

$$z_i = (x - a_i)^2 + (y - b_i)^2,$$

$$a = [3, 5, 2, 1, 7],$$

$$b = [5, 2, 1, 4, 9],$$

$$c = [1, 2, 5, 2, 3].$$

搜索空间为 $x \in [0, 10]$, $y \in [0, 10]$. 全局最优解为 $x = 2$, $y = 1$.

3) YangStandingWave^[17].

$$f_4(X) = e^{-\sum_{i=1}^d (x_i/15)^6} - 2e^{-\sum_{i=1}^d x_i^2} \prod_{i=1}^d \cos^2 x_i.$$

搜索空间为 $x_1 \in [-20, 20]$, $x_2 \in [-20, 20]$. 全局最优解为 $x_1 = 0$, $x_2 = 0$.

4) Damavandi^[16].

$$f_3(X) = [1 - \left| \frac{\sin[\pi(x_1 - 2)] \sin[\pi(x_2 - 2)]}{\pi^2(x_1 - 2)(x_2 - 2)} \right|^5] \times (2 + (x_1 - 7)^2 + (x_2 - 7)^2) + 1.$$

搜索空间为 $x_1 \in [0, 14]$, $x_2 \in [0, 14]$. 全局最优解为 $x_1 = 2$, $x_2 = 2$.

函数5)–8)是文献[18]中提出的新型合成多峰复杂测试函数. Suganthan教授在其个人主页<http://www.ntu.edu.sg/home/EPNSugan/>上提供了这些函数的代码及其所使用的正交矩阵、偏移全局最优解、控制参数. 这4个函数搜索空间为 $x_i \in [-5, 5]$ ($i = 1, \dots, 10$), 最小值均为0.

5) CF1.

$cf_1 - cf_{10}$: Sphere Function.

$$[\delta_1 \dots \delta_{10}] = [1 \dots 1],$$

$$[\lambda_1 \dots \lambda_{10}] = [0.05 \dots 0.05].$$

全局最优解: $x_1 = -1.9402$, $x_2 = -3.7831$.

6) CF3.

$cf_1 - cf_{10}$: Griewank's Function.

$$[\delta_1 \dots \delta_{10}] = [1 \dots 1],$$

$$[\lambda_1 \dots \lambda_{10}] = [1 \dots 1].$$

全局最优解: $x_1 = 1.0600$, $x_2 = -3.7617$.

7) CF4.

$cf_1 - cf_2$: Ackley's Function,

$cf_3 - cf_4$: Rastrigin's Function,

$cf_5 - cf_6$: Weierstrass Function,

$cf_7 - cf_8$: Griewank's Function,

$cf_9 - cf_{10}$: Sphere Function.

$$[\delta_1 \dots \delta_{10}] = [1 \dots 1],$$

$$[\lambda_1 \dots \lambda_5] = [5/32 \ 5/32 \ 1 \ 1 \ 10],$$

$$[\lambda_6 \dots \lambda_{10}] = [10 \ 0.05 \ 0.05 \ 0.05 \ 0.05].$$

全局最优解: $x_1 = 3.3253$, $x_2 = -1.2835$.

8) CF5.

$cf_1 - cf_2$: Rastrigin's Function,

$cf_3 - cf_4$: Weierstrass Function,

$cf_5 - cf_6$: Griewank's Function,

$cf_7 - cf_8$: Ackley's Function,

$cf_9 - cf_{10}$: Sphere Function.

$$\begin{aligned} [\delta_1, \dots, \delta_{10}] &= [1, \dots, 1], \\ [\lambda_1, \dots, \lambda_5] &= [1/5, 1/5, 10, 10, 0.05], \\ [\lambda_6, \dots, \lambda_{10}] &= [0.05, 5/32, 5/32, 0.05, 0.05]. \end{aligned}$$

全局最优解: $x_1 = 1.5953, x_2 = 2.6440$.

4.2 算法参数设置(Parameter settings for algorithms)

为了通过统计分析的方法公平合理地比较5种不同算法的渐近性, 本文对每个测试函数进行了100次寻优测试. 所有参与比较的算法在每次测试中具有相同的种群大小、迭代次数和初始种群: 种群大小为20, 最大迭代次数为200, 初始种群是在整个解空间中均匀采样生成的. 本文实验是在为双核1.6 GHz CPU, 2 GB内存的个人笔记本电脑上使用MATLAB 7.8 (The Mathworks, Inc., Natick, MA, USA)编程实现的.

参与比较的算法以及文献中给出的它们最优参数如下:

- 精英保留策略的遗传算法: 每次迭代种群更新率(generation gap)为0.8; 以轮盘赌的方法对染色体进行选择; 均匀交叉和单点变异的概率分别设置为0.2和0.01^[3].
- 自适应惯性权重的粒子群优化算法: 惯性权重随着迭代次数的增加从1.4按照指数递减到0.5; 加速常数 c_1 和 c_2 均为1.49445^[4].
- 全面学习粒子群优化算法: 惯性权重随着迭代次数的增加从0.9按照指数递减到0.4; 加速常数 c 为1.49445; 更新间隔(refreshing gap)为7^[8].
- 萤火虫算法: 随机搜索因子 α 为0.2; 吸引力参数 β_0 为1; 亮度吸收参数 γ 为1^[9].
- 多元优化算法: 结构体为三角形结构; 队列长度分别为5; 堆栈深度从左到右依次递减1, 左边第一个堆栈栈深为5; 全局搜索元个数与队列长度相等; 局部搜索元组种群大小与其对应的堆栈深度相等; 局部搜索半径 r 为0.1.

4.3 结果与讨论(Results and discussion)

4.3.1 局部搜索区域半径(Radius of local search area)

r 决定了局部搜索的范围, r 过大无法保证渐近精度, 然而 r 过小一方面不宜与搜索速度, 另一方面也不利于局部元通过自身的搜索跳出局部最优解. 为了研究多元优化算法的控制参数 r 取不同值时对优化性能的影响, 本文基于CF1, CF3, CF4与CF5等4个具有挑战性的测试函数分别取解空间上下界跨度的0.01, 0.05, 0.1, 0.15, 0.2倍作为 r 值进行了测试. 对于每个 r 值都进行了20次随机测试. 图3展示了 r

取不同值时20次试验获得的平均适应度值随迭代次数的变化曲线.

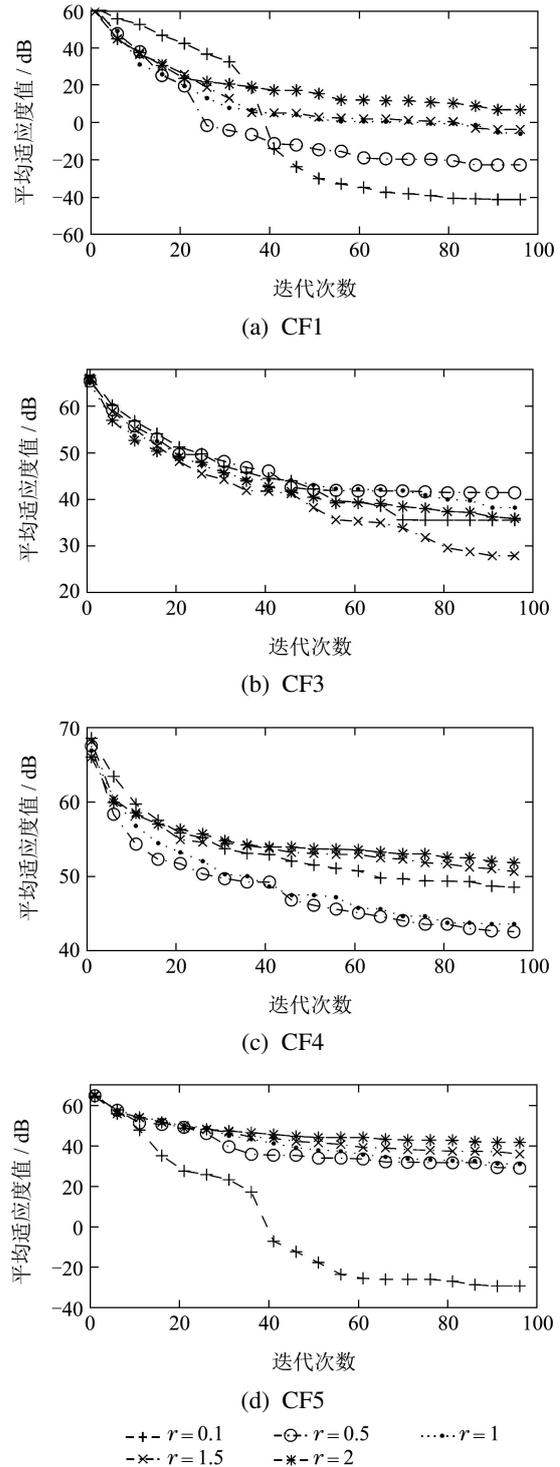


图3 平均适应度值曲线

Fig. 3 Curve of mean fitness value

由图3可以得出如下结论:

- 1) r 取值分别为0.1和2时获得的结果之间的比较可以看出: $r = 2$ 时, 适应度值一开始下降较快, 然而搜索后期适应度值下降较慢且最终适应度值较差; $r = 0.1$ 时, 适应度值一开始下降较慢, 然而最终

适应度值较低. 可见取较大的 r 值时, 算法搜索速度快, 渐近精度低; 取较小的 r 值时, 算法搜索速度慢, 渐近精度高.

2) 对函数CF3和CF4的寻优结果可以看出当 $r = 0.1$ 或者 $r = 2$ 时算法都没能获得最低的适应度值. $r = 0.1$ 时算法没有获得最低的适应度值的原因是CF3和CF4函数的适应度值面上存在许多距离较近的局部陷阱^[18], r 值过小, 算法容易陷入局部最优解而无法到达全局最优解. $r = 2$ 时算法没有获得最低的适应度值的原因是CF3和CF4函数的全局最优解在一个非常狭小的范围内, r 值过大, 算法很难保证渐近精度. 综上所述, 当 r 取值为0.5, 1或1.5时算法获得CF3和CF4函数较优适应度值也是合理的.

如实说, 很难找到一个对所有优化问题都能同时保证渐近进度以及搜索速度的最佳 r 值. 值得庆幸的是 r 值的选取可以参考小生境算法中物种半径(niching radius)的设置方法, 因为它们都是用来确定局部搜索范围. 为了获得较快搜索速度的同时保证渐近精度, r 的设置应采用文献^[19]中物种半径的设置方法既把 r 设置为解空间上下界跨度的0.05 ~ 0.1倍之间. 该方法被广泛认可^[20-21]并且仿真结果也显示了该方法在多元优化算法局部搜索半径设置中的普遍有效性. 运算时间充裕的情况下可以把 r 值设的小一点以获得较高的渐近精度. 渐近精度要求不高的情况下可把 r 值设的大一点以获得较快的搜索速度. 为在200次迭代后获得较高的渐近精度, 本文把 r 值设为解空间上下界跨度的0.01倍既0.1.

4.3.2 对比结果(Comparison results)

本文从渐近曲线和渐近误差两个方面比较算法的渐近性. 图4显示了5种优化算法在200次测试中找到的最优适应度均值随着迭代次数变化的渐近曲线.

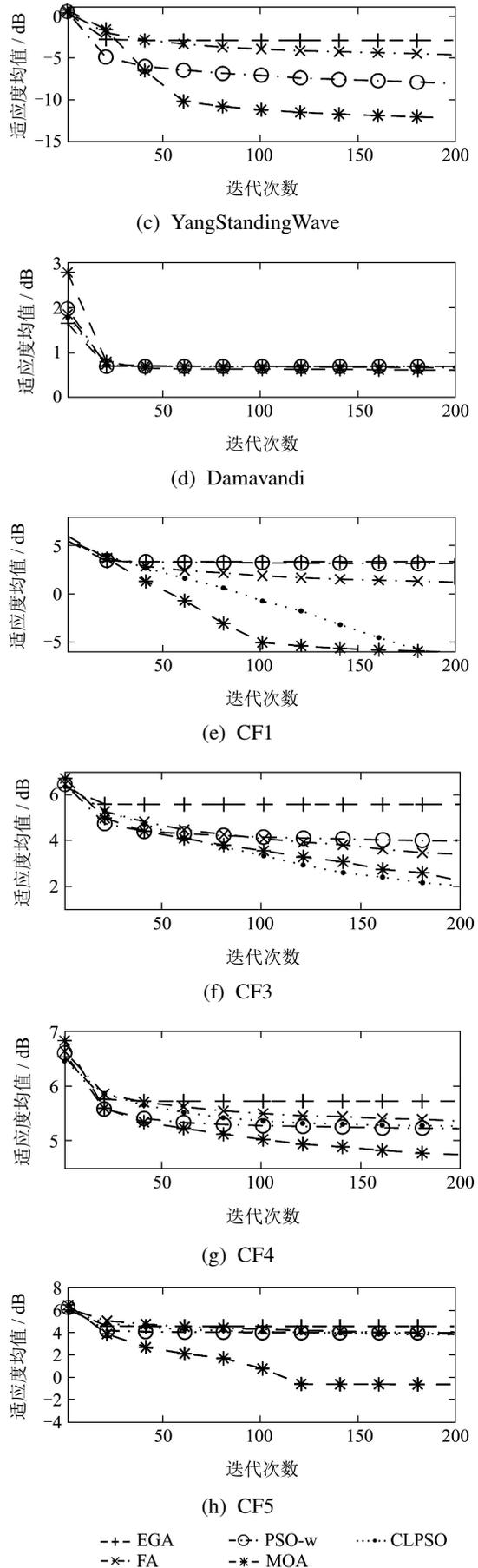
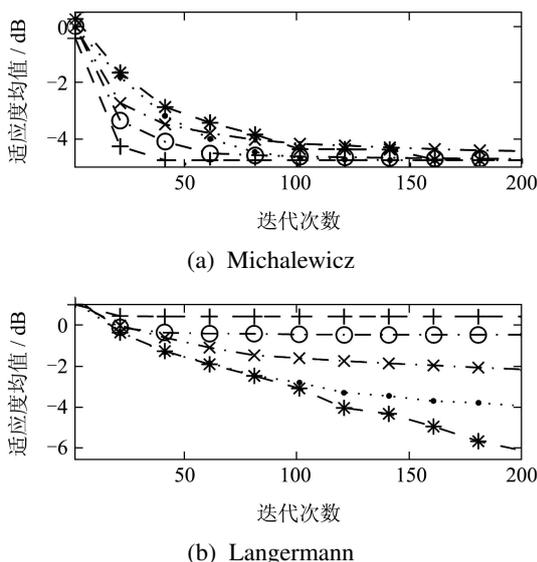


图 4 适应度均值渐近曲线
 Fig. 4 Asymptotic curve of mean fitness value

由图4本文可以得出如下结论:

1) 除了CF3函数的测试以外,多元优化算法的最优适应度均值随着迭代次数的增加不断降低并低于其他4个参与比较的算法.这说明多元优化算法在大部分优化问题中逐渐逼近全局最优解,具有全局最优解适应度值渐近性.这与定理1的结论相吻合.CF3函数中多元优化算法表现较差,其原因正如图3所描述的那样,由于CF3函数的适应度面上存在许多距离较近的局部陷阱, r 取0.1时其值较小,算法以一定概率陷入CF3函数局部陷阱而无法到达全局最优解.尽管如此,MOA算法获得的适应度均值较最低适应度均值相差不大.

2) 精英保留策略的遗传算法、自适应惯性权重的粒子群优化算法和萤火虫算法的最优适应度均值达到一定值之后进入停滞状态,不随着迭代次数的

增加而减少.这一方面说明这些方法容易陷入局部最优解,早熟收敛;另一方面也说明这些测试函数具有一定的欺骗性和复杂性.使得这些算法不能逐渐逼近全局最优解,测试算法具有一定挑战性.

3) 在大部分测试函数中,全面学习粒子群优化算法的最优适应度均值随着迭代次数的增加不断降低并在函数CF3的测试中低于多元化优化算法.这说明与精英保留策略的遗传算法、自适应惯性权重的粒子群优化算法和萤火虫算法相比,全面学习粒子群优化算法的跳出局部最优解的能力更强,全局最优解渐近性更好.

为了比较几种算法的渐近精度,本文定义算法找到的最优解与全局最优解之间的欧式距离为渐近误差.表2中列出了5种优化算法在100次测试中渐近误差的均值及标准差.

表2 渐近误差统计

Table 2 Statistic of asymptotic error

测试函数	EGA	PSO-w	CLPSO	FA	MOA
Michalewicz	0.0031 ± 0.00	0.003 ± 0.00	0.003 ± 0.00	0.003 ± 0.00	0.0053 ± 0.00
Langermann	0.0081 ± 0.02	2.0974 ± 3.79	2.4774 ± 3.28	0.0070 ± 0.00	0.0431 ± 0.02
Damavandi	17.3039 ± 16.52	4.9763 ± 10.76	28.0014 ± 2.82	22.5267 ± 11.37	22.4049 ± 11.43
YangStandingWave	6.5056 ± 1.92	7.0711 ± 0.00	7.0711 ± 0.00	7.0711 ± 0.00	5.0683 ± 3.18
CF1	0.0010 ± 0.00	0.7722 ± 1.46	0.7672 ± 1.65	0.0173 ± 0.00	0.0122 ± 0.01
CF3	0.3079 ± 1.01	1.0501 ± 1.56	2.0622 ± 1.55	0.0749 ± 0.52	0.0231 ± 0.01
CF4	3.038 ± 2.92	3.6777 ± 2.31	4.2574 ± 2.38	3.1623 ± 1.40	1.9254 ± 2.50
CF5	0.0215 ± 0.29	2.1141 ± 2.08	3.0268 ± 2.13	0.524 ± 1.37	0.0201 ± 0.01

由表2可以看出:在大多数优化问题中,多元优化算法找到的最优解与全局最优解误差均值较其他方法小.这说明多元优化算法具有全局最优解渐近性且逼近全局最优解的精度较高.不同函数的适应度面差异导致同一个方法对不同测试函数的渐近精度迥异.对于Michalewicz, Langermann和CF1函数,各个方法都能达到较高的渐近精度.尽管MOA的渐近精度不是所有参与比较的算法中最高的,但是其误差均值与最小误差均值相差不大.对于Damavandi函数由于其适应度面具有较强的欺骗性,所有的方法的渐近精度都较低.对于YangStandingWave, CF3, CF4和CF5等4个测试函数,大部分方法的渐近误差都较大,但是MOA获得最小的渐近误差.这说明本文提出的方法在大部分测试函数中具有较高的渐近精度.

5 结论(Conclusions)

本文提出了一种搜索个体分工多元化的启发式优化算法并从理论上证明了该方法的渐近性.多元优化算法中,多元化搜索元分工协作分别对解空间

进行全局和局部搜索,而不必考虑均衡全局搜索和局部搜索的问题,这保证了算法能够以一定的渐近精度逐渐逼近全局最优解.仿真结果也显示了该方法在渐近性和渐近精度方面优于其他几个经典的启发式群智能优化算法.该算法在复杂优化问题中具有较好的渐近性,对于解决容易导致算法早熟收敛的全局优化问题具有良好的表现.

本文主要集中在从理论和仿真两个方面讨论多元优化算法的渐近性.进一步的工作将集中在通过利用结构体中的历史搜索信息预测搜索结果^[22]以保证渐近精度的前提下提高渐近速度.

参考文献(References):

- [1] LIANG J J, SONG H, QU B Y, et al. Comparison of three different curves used in path planning problems based on particle swarm optimizer [J]. *Mathematical Problems in Engineering*, 2014, 2014(1): 1 - 15.
- [2] 夏克文, 高峰, 武睿, 等. 云粒子群优化算法在无线传感器网络中的应用 [J]. *控制理论与应用*, 2011, 28(9): 1175 - 1178.
(XIA Kewen, GAO Feng, WU Rui, et al. Optimal wireless sensor network using cloud adaptive particle-swarm-optimization algorithm [J].

- Control Theory & Applications*, 2011, 28(9): 1175 – 1178.)
- [3] DJURIŠIĆ A B. Elite genetic algorithms with adaptive mutations for solving continuous optimization problems-application to modeling of the optical constants of solids [J]. *Optics Communications*, 1998, 151(1): 147 – 159.
- [4] SHI Y, EBERHART R C. A modified particle swarm optimizer [C] // *Proceedings of IEEE World Congress on Computational Intelligence*. New York: IEEE, 1998: 69 – 73.
- [5] LI L, TANG K. History-based topological speciation for multimodal optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2014, DOI: 10.1109/TEVC.2014.2306677.
- [6] 陈国初, 俞金寿. 两群微粒群优化算法及其应用 [J]. *控制理论与应用*, 2007, 24(2): 294 – 298.
(CHEN Guochu, YU Jinshou. Two sub-swarms particle swarm optimization algorithm and its application [J]. *Control Theory & Applications*, 2007, 24(2): 294 – 298.)
- [7] 巩敦卫, 张勇, 张建华, 等. 新型粒子群优化算法 [J]. *控制理论与应用*, 2008, 25(1): 111 – 114, 119.
(GONG Dunwei, ZHANG Yong, ZHANG Jianhua, et al. Novel particle swarm optimization algorithm [J]. *Control Theory & Applications*, 2008, 25(1): 111 – 114, 119.)
- [8] LIANG J J, QIN A K, SUGANTHAN P N. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(3): 281 – 295.
- [9] YANG X S. Firefly algorithm, stochastic test functions and design optimisation [J]. *International Journal of Bio-Inspired Computation*, 2010, 2(2): 78 – 84.
- [10] 苟常兴. 多元优化算法的C++实现及其对多模态优化问题研究 [D]. 昆明: 云南大学, 2014.
(GOU Changxing. *On the study of the implementation of multivariate optimization algorithm based on C++ and its application to multimodal optimization problems* [D]. Kunming: Yunnan University, 2014.)
- [11] SUDHOLT D, WITT C. Runtime analysis of binary PSO [C] // *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York: ACM, 2008: 135 – 142.
- [12] YAO J, KHARMA N, GROGONO P. Bi-objective multipopulation genetic algorithm for multimodal function optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(1): 80 – 102.
- [13] DASGUPTA S, PAPADIMITRIOU C H, VAZIRANI U V. *Algorithms* [M]. New York: McGraw-Hill, 2006: 13 – 15.
- [14] SURJANOVIC S, BINGHAM D. Virtual library of simulation experiments: text functions and datasets [EB/OL]. [2014-06-10]. <http://www.sfu.ca/~ssurjano/michal.html>.
- [15] BERSINI H, DORIGO M, LANGERMAN S. Results of the first international contest on evolutionary optimization [C] // *IEEE International Conference on Evolutionary Computation*. New York: IEEE, 1996: 611 – 615.
- [16] DAMAVANDI N, SAFAVI-NAEINI S. A hybrid evolutionary programming method for circuit optimization [J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2005, 52(5): 902 – 910.
- [17] YANG X S. *Engineering Optimization: an Introduction with Metaheuristic Applications* [M]. New Jersey: John Wiley & Sons, 2010: 265 – 265.
- [18] LIANG J J, SUGANTHAN P N, DEB K. Novel composition test functions for numerical global optimization [C] // *Proceedings of IEEE Swarm Intelligence Symposium*. New York: IEEE, 2005: 68 – 75.
- [19] DEB K, GOLDBERG D E. An investigation of niche and species formation in genetic function optimization [C] // *Proceedings of the 3rd International Conference on Genetic Algorithms*. Las Vegas, Nevada: Morgan Kaufmann Publishers Inc, 1989: 42 – 50.
- [20] WANG H, MOON I, YANG S, et al. A memetic particle swarm optimization algorithm for multimodal optimization problems [J]. *Information Sciences*, 2012, 197(2012): 38 – 52.
- [21] KAMYAB S, EFTEKHARI M. Using a self-adaptive neighborhood scheme with crowding replacement memory in genetic algorithm for multimodal optimization [J]. *Swarm and Evolutionary Computation*, 2013, 12(2013): 1 – 17.
- [22] LIU B, ZHANG Q, GIELEN G. A Gaussianprocess surrogate model assisted evolutionary algorithm for medium scale expensive black box optimization problems [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(2): 180 – 192.

作者简介:

李宝磊 (1987–), 男, 博士研究生, 主要研究方向为智能优化算法、信号处理, E-mail: bl.li@qq.com;

吕丹桔 (1977–), 女, 博士研究生, 主要研究方向为智能优化算法、信号检测, E-mail: lvdanjv@gmail.com;

张钦虎 (1988–), 男, 硕士研究生, 主要研究方向为智能优化算法、模式识别, E-mail: 15887152994@163.com;

施心陵 (1956–), 男, 教授, 主要研究方向为人工智能、生物医学工程、信号处理, E-mail: xlshi@ynu.edu.cn;

安镇宙 (1977–), 男, 副教授, 博士, 主要研究方向为智能优化算法、智能控制, E-mail: anzhenzhou@sina.com.