

文章编号: 1001-0920(2011)02-0175-07

定向多尺度变异克隆选择优化算法

陶新民¹, 刘福荣², 刘玉¹, 付丹丹¹

(1. 哈尔滨工程大学信息与通信工程学院, 哈尔滨 150001;

2. 哈尔滨电力职业技术学院信息工程系, 哈尔滨 150030)

摘要: 提出一种定向多尺度变异克隆选择优化算法. 为了实现抗体间信息共享, 算法利用定向进化机制引导抗体向着抗体群最优解区域逼近. 采用多尺度高斯变异机制, 在算法初期利用大尺度振荡变异实现了全局最优解空间的快速定位. 随着适应值的提升, 小尺度变异会随之减低, 使得算法在进化后期通过小尺度变异完成局部精确解的搜索. 将算法应用到5个经典函数优化问题, 结果表明, 该算法不仅具有更快的收敛速度, 而且全局解搜索能力和稳定性均有显著提高.

关键词: 克隆选择; 多尺度变异; 定向进化; 勘探和开采

中图分类号: TP18

文献标识码: A

Clone selection optimization algorithm with directional multi-scale mutation

TAO Xin-min¹, LIU Fu-rong², LIU Yu¹, FU Dan-dan¹

(1. College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China;

2. Department of Information Engineering, Harbin Power Vocational Technology College, Harbin 150030, China.

Correspondent: TAO Xin-min, E-mail: taixinmin@hrbeu.edu.cn)

Abstract: To deal with the problem of single-scale mutation, premature convergence and slow search speed, a clone selection algorithm(CSA) with directional multi-scale gaussian mutation is proposed. To implement the share of information between antibodies, the directional evolution mechanism is utilized to induce the antibodies to evolve to the best solution region. The special multi-scale Gaussian mutation operators are introduced to make antibodies explore the search space more efficiently. The large-scale mutation operators can be utilized to quickly localize the global optimized space at the early evolution, while the small-scale mutation operators can implement local accurate minima solution search at the late evolution, which can make the algorithm explore the global and local minima thoroughly at the same time. The comparison of the performance of the proposed approach with other CSAs with different mutations is experimented. The experimental results show that the proposed method can not only effectively solve the premature convergence problem, but also significantly speed up the convergence and improve the stability.

Key words: clone selection; multi-scale mutation; direction evolution; exploration and exploitation

1 引言

生物免疫系统作为一种高度并行的自适应信息学习系统, 能够自适应地识别和排除机体的抗原性异物, 并且具有学习、记忆和自适应调节能力, 保护机体体内环境的稳定. 人工免疫算法便是一种受生物免疫系统启发而设计的新型智能优化算法, 它作为一个新的研究方向, 为当前智能信息处理领域提供了一种有效的工具. 由于其在信息处理方面具有较强的鲁

棒性, 并且在解决复杂问题时具有强大的信息处理能力, 近年来已广泛应用于智能优化、模式识别、信息安全、机器学习、故障诊断等^[1-4]诸多领域.

目前, 针对人工免疫算法的研究主要包括免疫遗传算法、克隆选择算法、阴性选择算法和免疫学习算法等^[5-7]. 其中克隆选择算法作为一种新的全局优化算法, 结合了先验知识和生物免疫系统的自适应能力两大特点, 具有较强的鲁棒性和信息处理能力. 并且

收稿日期: 2009-10-22; 修回日期: 2009-12-10.

基金项目: 国家自然科学基金项目(61074076); 中国博士后科学基金项目(20090450119); 中国博士点新教师基金项目(20092304120017); 黑龙江省博士后基金项目(LBH-Z08227).

作者简介: 陶新民(1973-), 男, 副教授, 博士, 从事智能信号处理、软计算等研究; 刘福荣(1970-), 女, 副教授, 博士, 从事自然计算、模式识别等研究.

在对问题进行求解时,不要求目标函数具有可导等高附加信息,在搜索过程中能更快地收敛到全局最优解,被认为是具有强大潜力的搜索算法,现已应用于模式识别、异常和故障诊断、机器人控制、网络入侵检测、函数优化等领域,表现出卓越的性能和效率^[8-10]. 针对传统克隆选择算法收敛速度慢等缺点,众多学者对其进行了深入研究并提出了很多新的改进算法,其中文献[11]提出了基于高斯变异的克隆选择算法. 高斯变异具有较强的局部搜索能力,但也存在一些不足,如易陷入局部极值、迭代后期出现停止不前等现象,为此,[12]提出了一种基于并行进化策略的克隆选择算法,[13]提出了带有量子交叉变异操作的克隆优化算法. 虽然上述算法部分解决了高斯变异克隆选择算法易陷入局部极值的不足,但这些算法在实现变异时均采用单一的变异尺度,使其在算法后期无法实现最优解的探索. 因此,如何增强算法的多样性,使其在快速定位到最优解区域的同时提高最优解精度值得进一步研究.

鉴于此,本文提出一种定向多尺度变异克隆选择优化算法(DMCSA). 该算法利用大小不同的变异尺度实现解空间的探索,这种或大或小的变异机制,使得克隆后的抗体能以尽量分散的变异尺度对解空间进行更加详尽的探索. 同时,高斯变异的范围随着适应值的变小也逐渐降低,有利于在后期提高最优解的精度. 在利用不同大小变异算子提高局部精确搜索能力的同时,该算法利用定向变异机制使算法向着最优解区域逼近,以加快算法收敛速度. 最后,利用实验对不同评测函数进行测试,测试结果均验证了新算法具有优良的优化性能.

2 传统克隆选择及其改进算法分析

克隆选择理论用来描述当一个非自体的抗原被B细胞识别时所引起的免疫应答过程,它包括抗体与抗原的匹配、抗体的克隆殖、抗体的进化变异、抗体记忆细胞的自然选择4个过程. 当一个B细胞受体与抗原之间的亲和力达到一定程度时,开始进行克隆增殖(自我复制),从而分泌新的抗体. 在自我复制过程中,B细胞的后裔经历了变异过程,从而使得B细胞能够更好地与所选择的抗原以更高的亲和力相匹配. 抗体与抗原相结合从而导致抗原最终被其他免疫细胞所消灭. 变异和选择过程类似物种的自然选择过程,激活的具有较高抗原亲和力的B细胞转化为具有较长寿命的记忆细胞.

文献[11-13]从不同角度模拟上述克隆选择机理,相继提出了不同的克隆选择算法. 不失一般性,克隆选择算法主要包括3种操作:克隆操作、免疫基因操作和克隆选择操作. 抗体群的状态转移情况可以表示

成如下的随机过程:

$$C_m : B(k) \xrightarrow{\text{clone}} B'(k) \xrightarrow{\text{mutation}} B''(k) \xrightarrow{\text{selection}} B(k+1).$$

克隆操作有效地实现了搜索空间的扩张;免疫基因操作实现了抗体亲和度的成熟和多样性的产生;克隆选择操作通过局部择优,有效地压缩了种群的大小. 免疫学认为,亲合度成熟和抗体多样性的产生主要依靠抗体的高频变异,因此基因变异操作是整个算法的核心,改善它对于提高算法性能具有十分重要的作用. 传统的基因变异操作主要依靠均匀变异算子来实现. 均匀变异算子通过将来自某区间的随机数加入克隆后的种群 $B'(k)$ 中某个 b'_i 抗体的第 l 个分量 $b'_{i(l)}$,进而产生变异,即

$$b''_{i(l)} = b'_{i(l)} + \Delta_l \text{Rand}(0, 1). \quad (1)$$

其中

$$\Delta_l = \begin{cases} b'_{i(l)}^{\min} - b'_{i(l)}, & \text{Rand}(0, 1) < 0.5; \\ b'_{i(l)}^{\max} - b'_{i(l)}, & \text{Rand}(0, 1) \geq 0.5. \end{cases} \quad (2)$$

$\text{Rand}(0, 1)$ 表示从0到1均匀分布的随机变量. 式(2)表示每次随机选择一个 Δ_l ,每个 $b'_{i(l)}$ 准确地位于一个对应区间 $[b'_{i(l)}^{\min}, b'_{i(l)}^{\max}]$ 内. 由于均匀变异能产生远离原始抗体的变异,使得算法在迭代过程中通过较大均匀变异算子保持了解空间的勘探能力,维护了种群的多样性.

为了提高局部搜索能力,文献[11]提出了高斯变异的克隆选择算法,即

$$b''_{i(l)} = b'_{i(l)} + \sigma_l \text{Randn}(0, 1). \quad (3)$$

其中: $\text{Randn}(0, 1)$ 表示均值为0,方差为1的高斯分布的随机变量; σ_l 表示初始方差.

为了进一步改善高斯变异的克隆选择算法易陷入局部极值、迭代后期出现停止不前等缺点,文献[12]提出了一种高斯变异和均匀变异相结合的自适应并行免疫进化策略. 在算法中,根据抗体抗原亲和度将初始抗体种群分为两个子群,相应地提出了精英克隆(高斯)算子和超变异(均匀)算子,通过高斯变异算子提高算法局部搜索能力,同时利用均匀变异算子维持种群多样性,通过这两个功能互补算子的并行操作实现种群进化.

3 定向多尺度克隆选择优化算法

上述改进算法虽然能够维护种群的多样性,提高局部极小点的逃逸能力,然而在进化前期,算法不能保证取到合适的变异算子,使种群从局部极值的邻域变异到全局最优解的邻域. 同时,算法对初始种群和高斯变异算子标准差的初始值 σ_l 敏感,一旦初始种群的性能不够优良(即初始种群中没有任何一个个体落

在全局最优解的邻域), 并使用了较小的高斯变异标准差, 则将导致算法以较高的概率收敛于局部极值. 这种缺陷只能依靠均匀变异算子来克服, 因此具有较强的随机性. 同时随着维度的增多, 这种逃逸能力显得微不足道, 最终导致算法需要多次迭代才可以收敛到全局最优解.

为此, 本文提出一种多尺度高斯变异机制, 该机制使得克隆后抗体群能在算法初期通过大尺度振荡高斯变异算子使抗体具有较好的局部解逃逸能力, 快速到达全局最优解的附近. 同时, 在后期逐渐递减的小尺度高斯变异算子能够保证抗体以足够高的精度逼近最优解, 最终实现在保证抗体多样性的同时使局部具有精确的解搜索能力.

3.1 多尺度变异操作

通过上述分析可知, 多尺度高斯变异机制可以保证算法具有全局解搜索能力的同时进一步提高最优解的精度. 具体算子描述如下: 设尺度个数为 M , 首先初始化多尺度高斯变异算子的初始方差为

$$\sigma^{(0)} = (\sigma_1^{(0)}, \sigma_2^{(0)}, \dots, \sigma_M^{(0)}). \quad (4)$$

初始时, 方差一般设定为优化变量的取值范围, 随着迭代次数的增加, 多尺度高斯变异算子的方差会随之进行调整, 具体调整方式为: 首先根据适应值的大小对种群中的抗体进行由小到大排序; 然后对其进行组合, 生成 M 个子群, 每一个子群的抗体个数为 $P = N/M$. N 是抗体数目, 计算每一个子群的适应度为

$$\text{Fit}X_m^{(k)} = \sum_{i=1}^P f(x_i^m) / P, \quad m = 1, 2, \dots, M. \quad (5)$$

其中: k 是当前迭代次数, f 是抗体的适应值计算函数. 不同变异尺度之间相互竞争, 根据适应能力的不同设置不同的变异能力, 因此第 m 个变异算子的标准差为

$$\sigma_m^{(k)} = \sigma_m^{(k-1)} \exp\left(\frac{M\text{Fit}X_m^{(k)} - \sum_{m=1}^M \text{Fit}X_m^{(k)}}{\text{Fit}X_{\max} - \text{Fit}X_{\min}}\right), \quad (6)$$

$$\text{Fit}X_{\max} = \max(\text{Fit}X_1^{(k)}, \text{Fit}X_2^{(k)}, \dots, \text{Fit}X_M^{(k)}), \quad (7)$$

$$\text{Fit}X_{\min} = \min(\text{Fit}X_1^{(k)}, \text{Fit}X_2^{(k)}, \dots, \text{Fit}X_M^{(k)}). \quad (8)$$

变异算子的进化是一个递归过程, 排在后面的变异算子可能很大, 因此对变异算子的标准差做如下规范: 若 $\sigma_m^{(k)} > W/4$, 则

$$\sigma_m^{(k)} = |W/4 - \sigma_m^{(k)}|, \quad (9)$$

其中 W 为待优化变量空间的宽度. 重复使用式 (9), 直到满足 $\sigma_m^{(k)} < W/4$. 通过本文的多尺度变异算子能实现整个搜索空间的覆盖, 其中大尺度变异算子具有振荡性质, 有利于实现解空间的粗搜索, 可以快速定位到最优解区域, 逐渐减小的小尺度变异算子能在进化

后期实现局部精确解的搜索.

3.2 定向进化操作

为了提高克隆选择算法的收敛速度, 使其在算法初期迅速定位到最优解区域, 本文在传统克隆选择算法中增加了定向进化操作, 使得算法的抗体之间实现信息的最大化共享, 这样有利于算法初期快速定位到最优解区域. 同时, 由于本文算法采用了多尺度变异机制, 保证算法不会因为定向进化的影响陷入局部极值点. 具体实现机制描述如下: 定义算法每次迭代后的抗体群中适应值最优的抗体为

$$\text{Best}^{(k)} = \{x^*, f(x^*) < f(x), x \in B^{(k)}\}.$$

定向进化操作定义为

$$x^{(k)} = x^{(k)} + rc(\text{Best}^{(k)} - x^{(k)}). \quad (10)$$

其中: r 为从 0 到 1 均匀分布的随机变量; c 为加速常数, 一般取值为 1.

3.3 算法流程

本文提出的定向多尺度克隆选择优化算法是将定向进化操作、克隆操作、多尺度变异操作和克隆选择操作合为一体的进化算法. 具体流程如下:

Step 1: 参数初始化. 设置环境容纳的抗体数目为 N , 尺度个数为 M , 其中 N 为 M 的整数倍, 循环次数 $k = 0$. 设置初始变异标准差 $\sigma_0 = [\sigma_1^0, \sigma_2^0, \dots, \sigma_M^0]$ 均为优化变量的取值范围.

Step 2: 定向进化操作. 选择当前抗体群中适应度最高的抗体 $\text{Best}^{(k)}$, 抗体群中每一个抗体按照式 (10) 进化, 生成新的抗体群.

Step 3: 抗体群的克隆操作. 克隆是指通过无性繁殖可连续传代并形成群体. 在本文算法中, 对抗体群 $B^{(k)} = \{b_1, b_2, \dots, b_N\}$ 中每一个抗体进行克隆操作, 产生新的 $b_j^i = b_i (i = 1, 2, \dots, N, j = 1, 2, \dots, q_i)$, 最终生成 $B^{(k)}$. 为了方便起见, q_i 定义为 $M + 1$ 大小.

Step 4: 多尺度变异操作. 对抗体群中的抗体使用多尺度变异算子来繁殖后代. 种群 $B^{(k)}$ 基因变异操作的基本内容是对抗体某些基因位置上的基因值进行变动, 本文设抗体群 $B^{(k)}$ 中抗体 b_i 的分量为 $b_{i(l)}$, 则更新方程为

$$b_{i(l)}'' = b_{i(l)}' + \Delta_i^{(k)}, \quad (11)$$

其中 $\Delta_i^{(k)}$ 为服从 $N(0, \sigma_m^{(k)2}) (m = 1, 2, \dots, M)$ 分布的高斯白噪声. 此外, 为了能最大范围地实现空间勘探能力, 在进行完多尺度高斯变异后, 对剩下的克隆抗体进行一次均匀变异操作, 其实现如式 (1) 所示. 比较后, 通过克隆选择操作取所有变异后适应值最好的抗体进入新的抗体群.

Step 5: 克隆选择操作. 克隆选择操作是从抗体各

自克隆增殖后的子代中选择优秀的个体,从而形成新的种群. 克隆选择操作是克隆操作的逆过程,同一个抗体 b_i 经过克隆后形成的子群体在经过基因变异操作编辑后,通过克隆选择操作可实现局部亲和力的提升. $\forall i = 1, 2, \dots, N, \exists j \in \{1, 2, \dots, M+1\}$, 若抗体 b_i^j 为子群体 $\{b_i^1, b_i^2, \dots, b_i^{M+1}\}$ 中亲和度最高的抗体,则在子群体 $\{b_i^1, b_i^2, \dots, b_i^{M+1}\}$ 中,抗体 b_i^j 被选择代替 b_i , 成为 $b_i' = b_i^j$, 生成新的 $B''(k+1)$.

4 算法的优化机理及收敛性能分析

4.1 算法的优化机理分析

为了说明方便, 设优化函数 $f(x)$ 只有一个自变量函数, 如图 1 所示. 多尺度变异算法中抗体群的某一个抗体经过多次迭代后到达 a 点, 经过小尺度变异后, 向下“下山”找到个体 a' ; 然后经过定向进化操作, 进化到 a'' ; 再经过大尺度变异后找到适应值更高的抗体 b ; 最后抗体 b 被选入下一代. 经过定向进化操作, 若干代进化后, 找到抗体 b' . 抗体 b' 经过大尺度变异操作找到微粒 c , 则 c 被选为进入下一个世代, 再经过定向进化和小尺度变异, 最终找到最优解 c' .

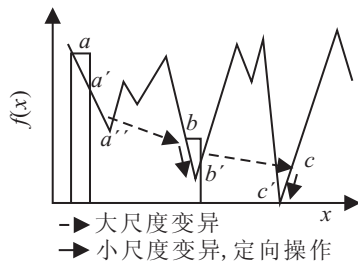


图 1 定向多尺度变异优化机理

由上述可知, 多尺度变异操作将整个解空间看作一个山谷, 经过大尺度变异操作进行“下山”以寻求适应值更高的区域 ($a'' \rightarrow b, b' \rightarrow c$). 大尺度变异操作用于搜索全局解空间, 是粗搜索. 如果大尺度变异操作找到了适应值更高的区域, 则定向进化操作和小尺度变异操作能通过小区域搜索在该区域进行局部搜索, 以寻求更高精度的解 ($a \rightarrow a', a' \rightarrow a'', b \rightarrow b', c \rightarrow c'$). 定向进化操作和小尺度变异操作用于搜索局部解空间. 因此, 本文算法是一个勘探和开采能力相互交替进行的搜索算法, 具有全局和局部两层邻域搜索机制, 从而保证了算法全局和精确的局部寻优性能.

4.2 算法的收敛性分析

与文献 [10] 相同, 在定向多尺度变异克隆选择算法中, 抗体种群序列 $B^{(K)}$ 仍是一个有限齐次可约马尔可夫链. 虽然在算法中增加了定向进化操作和多尺度变异操作, 但根据式 (10) 可知, 定向进化操作中上一代的 $\text{Best}^{(k)}$ 没有变化, 因此具有保优性质, 同时结合克隆选择操作的保优性可得

$$P\{\vartheta(B^{(K+1)}) = 0 | \vartheta(B^{(K)}) \neq 0\} = 0,$$

其中 $\vartheta(B^{(K)})$ 表示抗体种群 $B^{(K)}$ 中包含最优抗体的个数. 对于任意初始状态 $B^{(0)}$, 定向多尺度变异克隆选择优化算法仍以概率 1 收敛于最优抗体集合 B^* , 证明与文献 [10] 类似, 此略.

5 对比实验及结果分析

5.1 参数设置及 Benchmark 函数实验

为了分析本文提出算法的全局搜索性能、收敛速度和算法的稳定性, 选择标准 CSA 算法, 基于高斯变异的克隆选择算法 (GCSA)^[11], 并行进化策略的克隆选择算法 (PCSA)^[12] 以及基于量子交叉变异的克隆选择算法 (QCSA)^[13] 进行对比实验. 其中 CSA 算法的种群规模为 100, 克隆规模为 10, 变异概率为 0.5; QCSA 算法中的量子的变异概率为 0.5, 交叉概率为 0.8, 交叉种群比例为 0.5. 本文 DMCSA 算法 $M = 10$, 初始方差 σ_0 矢量均为优化变量的范围. 所有实验函数维度设置为 30, 每次运行 2 000 代, 为了排除算法内部随机操作对性能的影响, 对每一个函数独立运行 50 次并对其统计结果进行分析.

本文选择了克隆选择算法和遗传算法经常使用的 5 个 Benchmark 函数问题进行数值实验, 并根据函数性质分为具有单一极小点 (单模态) 和多个极小点 (多模态) 两大类.

单模态函数定义如下:

Quadric 函数为

$$f_1 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2. \quad (12)$$

取值范围为 $[-100, 100]$, 维度为 30, 最小值和位置为 $0(0, \dots, 0)$.

Rosenbrock 函数为

$$f_2 = \sum_{i=1}^n (100(x_{i+1} - x_i^2) + (x_i - 1)^2). \quad (13)$$

取值范围为 $[-50, 50]$, 维度为 30, 最小值和位置为 $0(1, \dots, 1)$.

多模态函数定义如下:

Griewank 函数为

$$f_3 = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1. \quad (14)$$

取值范围为 $[-300, 300]$, 维度为 30, 最小值和位置为 $0(0, \dots, 0)$.

Rastrigrin 函数为

$$f_4 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10). \quad (15)$$

取值范围为 $[-5.12, 5.12]$, 维度为 30, 最小值和位置为 $0(0, \dots, 0)$.

Schaffer F7 函数为

$$f_5 = \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} \times \left[\sin \left(50 \times \sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.1} \right)^2 + 1.0 \right]. \quad (16)$$

取值范围为 $[-100, 100]$, 维度为 30, 最小值和位置为 $0(0, \dots, 0)$.

5.2 单模态 Benchmark 函数对比实验

图 2 显示了 30 维 Quadric 函数的优化实验结果. 可以看出, 对于 Quadric 函数的优化问题, DMCSA 算法能在算法的初期阶段迅速定位到最优解区域, 体现了多尺度变异算子强劲的全局搜索能力. 而在算法进化后期, DMCSA 算法利用定向进化公式进化, 同时利用小尺度变异算子实现局部精确解的搜索, 大大提高了算法全局最优解的开采能力. 其他改进算法虽然也在进化, 并且有效地实现了局部最优解的逃逸, 但是单一变异机制的影响使得算法本身在初始阶段很难定位到全局最优解区域, 同时, 后期的变异操作带有随机性, 反而减缓了算法的收敛速度.

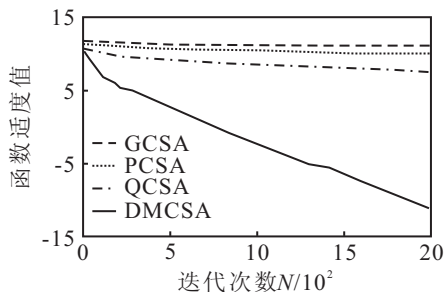


图 2 30 维 Quadric 函数收敛情况对比

如图 3 所示, 对于复杂单模态 30 维 Rosenbrock 函数优化问题, DMCSA 算法同样在算法初期具有较强的全局解定位能力. 由于 Rosenbrock 函数仅给算法提供了少量的信息, 使 GCSA 算法很难在短时间内辨别搜索方向, 极易陷入局部最优解. 本文算法采用了多尺度变异, 能够有效地逃出这些局部极值点, 实现最优解区域的准确定位, 使得算法能在短时间内找到正

确的搜索方向, 并具有较快的下降速度. 其他算法采用的变异机制无法在短时间内有效地定位到最优解区域, 因此下降较缓慢.

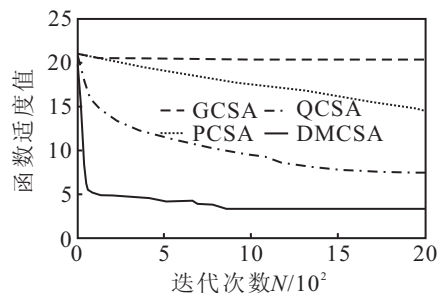


图 3 30 维 Rosenbrock 函数数值对比

表 1 为单模态 Benchmark 问题的对比实验结果. 从各个算法最优值和最差值的实验结果可以看出, DMCSA 算法在所有函数上均取得了较好的实验结果, 具有较快的局部精确解的搜索能力, 这是因为通过简单的高斯变异和量子变异使 CSA 算法丧失了全局解空间的勘探能力, 且仅通过单一的均匀变异无法弥补这一不足. DMCSA 算法能够实现勘探能力和开采能力的协调, 在增强全局解搜索性能的同时增加了算法对局部精确解的搜索能力. 同时, 实验中可以发现, 由于其他 CSA 算法在种群初始化和进化过程中均具有随机性, 算法的结果具有一定的随机性. 表 1 的实验结果显示, DMCSA 算法能够在算法的初始阶段快速定位到最优解区域, 显著提高了算法的搜索性能和收敛速度, 数据差异相对较小, 具有更好的稳定性和健壮性.

5.3 多模态 Benchmark 函数对比实验

多模态 Griewank 函数优化结果见图 4. DMCSA 算法能够在开始阶段寻找到全局最优解区域, 全局搜索能力明显优于其他算法, 能够有效逃出局部极小点找到全局最小点. 表 2 为多模态 Benchmark 问题的对比实验结果. 由表 2 可知, DMCSA 算法的中值、平均值和标准方差等各项数据均显著优于其他算法, 显示了新算法的稳定性和健壮性. 多模态 Rastrigrin 函数

表 1 DMCSA 与其他 CSA 算法在单模态 Benchmark 问题上的数值对比

| Function | Algorithm | Min | Median | Mean | Deviation | Max |
|------------|-----------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Quadric | CSA | 4.511 7e+004 | 4.681 1e+004 | 4.712 7e+004 | 3.607 9e+003 | 5.681 3e+004 |
| | GCSA | 4.461 9e+004 | 4.667 4e+004 | 4.667 4e+004 | 2.906 3e+003 | 4.872 9e+004 |
| | PCSA | 1.225 4e+002 | 1.709 4e+002 | 1.680 7e+002 | 1.581 1e+001 | 1.884 9e+002 |
| | QCSA | 1.419 9e+002 | 1.709 4e+002 | 1.695 1e+002 | 1.567 5e+001 | 1.977 6e+002 |
| | DMCSA | 1.227 8e-005 | 1.288 0e-005 | 1.413 4e-005 | 2.710 7e-006 | 1.724 5e-005 |
| Rosenbrock | CSA | 5.723 3e+008 | 7.680 2e+008 | 7.729 1e+008 | 1.488 1e+008 | 9.921 2e+008 |
| | GCSA | 5.005 4e+008 | 7.170 2e+008 | 7.238 4e+008 | 1.228 2e+008 | 9.099 9e+008 |
| | PCSA | 1.097 7e+006 | 1.832 2e+006 | 1.768 9e+006 | 3.224 7e+005 | 2.337 2e+006 |
| | QCSA | 1.371 9e+006 | 1.886 3e+006 | 1.826 6e+006 | 2.988 6e+005 | 2.361 4e+006 |
| | DMCSA | 23.647 3 | 27.311 1 | 26.623 9 | 2.699 4 | 28.913 3 |

表 2 DMCSA 与其他 CSA 算法在多模态 Benchmark 问题上的数值对比

| Function | Algorithm | Min | Median | Mean | Deviation | Max |
|------------|-----------|-----------------|-----------------|-----------------|----------------|-----------------|
| Griewank | CSA | 270.127 8 | 274.195 1 | 276.127 8 | 8.178 9 | 282.197 3 |
| | GCSA | 261.256 2 | 267.225 5 | 266.059 6 | 4.339 6 | 269.697 2 |
| | PCSA | 171.657 4 | 186.230 8 | 190.769 0 | 15.023 3 | 211.305 0 |
| | QCSA | 185.815 8 | 200.802 2 | 198.377 4 | 9.905 2 | 209.103 6 |
| | DMCSA | 58.643 5 | 59.083 1 | 59.569 0 | 1.241 9 | 60.980 3 |
| Rastrigrin | CSA | 301.185 9 | 347.075 7 | 343.581 0 | 24.173 5 | 359.541 5 |
| | GCSA | 294.889 3 | 336.679 4 | 332.718 0 | 17.506 8 | 353.301 5 |
| | PCSA | 116.055 4 | 143.791 7 | 141.438 8 | 14.670 6 | 158.292 3 |
| | QCSA | 116.818 8 | 157.776 3 | 153.173 0 | 28.347 1 | 180.320 5 |
| | DMCSA | 20.894 1 | 26.366 4 | 29.517 1 | 7.851 1 | 39.798 3 |
| Schaffer | CSA | 261.256 2 | 267.225 5 | 266.059 6 | 4.339 6 | 269.697 2 |
| | GCSA | 171.657 4 | 186.230 8 | 190.769 0 | 15.023 3 | 211.305 0 |
| | PCSA | 185.815 8 | 200.802 2 | 198.377 4 | 9.905 2 | 209.103 6 |
| | QCSA | 192.844 8 | 195.495 1 | 199.431 0 | 7.756 0 | 211.439 1 |
| | DMCSA | 48.201 8 | 55.082 3 | 55.288 8 | 6.807 5 | 62.788 7 |

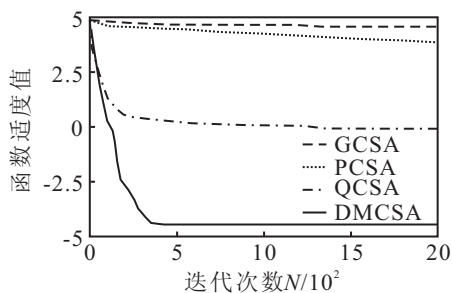


图 4 30 维 Griewank 函数的数值对比

是一个典型的用来测试算法全局搜索性能的函数,从表 2 可以看出,DMCSA 算法具有良好的全局搜索性能和较快的搜索速度。

图 5 为 30 维 Schaffer 函数的运行结果.从图 5 中可以看出,DMCSA 算法同样能在算法的初期迅速寻找到全局最优解区域,然后利用定向进化和小尺度变异算子实现局部精确解的搜索.由于 Schaffer 函数的特殊性质以及随机变异的引入,降低了其他改进 CSA 算法的搜索性能,使其陷入了局部极小点.从表 2 的结果可以看出,其他算法通过增加变异操作虽然能改进传统 CSA 算法早熟收敛的问题,但是 DMCSA 算法在各个方面均大大改善了算法全局解寻优能力.这是由于本文算法采用了多尺度变异算子,使得算法能在搜索空间内进行分散式搜索,加快了全局最优解的搜索速度.从表 2 中最佳和最差解实验结果之间的差

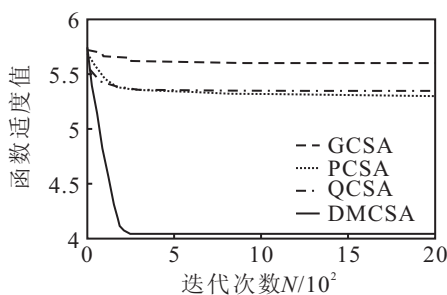


图 5 30 维 Schaffer 函数的数值对比

异可以看出,无论是 CSA 还是其他改进算法均不具备稳定性,而本文算法相对于其他算法稳定性大大提高,这种结果同大尺度振荡高斯变异算子的引入密切相关.即使初始种群的性能不好,振荡高斯算子也可以使种群的个体在较小的时间内获得更大的变异,从而脱离初始区域进入可行解的优化区域。

6 结 论

本文提出一种新的定向多尺度变异的克隆选择优化算法,利用不同大小方差的高斯变异机制实现解空间的探索.这种多个或大或小的变异机制,能促使整个种群以尽量分散的变异尺度来对解空间进行更加详尽的探索.同时,小尺度高斯变异的范围随着适应值的变小也逐渐降低,有利于提高最优解的精度.在利用不同大小变异算子提高全局及局部精确搜索能力的同时,该算法利用定向进化操作加快算法收敛速度.通过 5 个 benchmark 数据进行测试,实验结果表明,新算法在算法初期就能够快速定位到搜索空间的最优解区域,进而使得抗体在算法后期通过定向进化操作及小尺度变异算子向着最优精确解空间逼近,使其在进化过程中保持局部最优解空间开采的能力,加快了算法的收敛速度。

参考文献(References)

- [1] Paul K H, Paul D W, Gregg H G. An artificial immune system architecture for computer security applications[J]. IEEE Trans on Evolutionary Computation, 2002, 6(3): 252-280.
- [2] Guab C L, Wei C C. Immune model-based fault diagnosis[J]. Mathematics and Computers in Simulation, 2005, 67(6): 515-539.
- [3] Nareli C C, Carlos A C. Multiobjective optimization using the clonal selection principle[J]. Lecture Notes in Computer

- Science, 2003, 27(23): 158-170.
- [4] Yan J L, Tie J W. A novel immune algorithm doe complex optimization problems[C]. Proc of the 5th World Congress on Intelligent Control and Automation. Hangzhou: IEEE, 2004: 2279-2283.
- [5] 赵良辉, 杨海东. 小生境免疫算法解决作业车间调度问题[J]. 系统工程与电子技术, 2009, 31(7): 1642-1646.
(Zhao L H, Yang H D. Niche-based immune algorithm for job shop scheduling problems[J]. Systems Engineering and Elect ronics, 2009, 31(7): 1642-1646.)
- [6] 钱海, 马建辉, 王煦法. 一种新的基于克隆选择原理的人工免疫算法[J]. 中国科学技术大学学报, 2008, 38(12): 1397-1406.
(Qian H, Ma J H, Wang X F. New artificial immune algorithm based on the clonal selection principle[J]. J of University of Science and Technology of China, 2008, 38(12): 1397-1406.)
- [7] 王向军, 嵇斗. 一种多群竞争进化规划算法[J]. 电子学报, 2004, 32(11): 1824-1828.
(Wang X J, Ji D. A multi-subgroup competition evolutionary programming algorithm[J]. Acta Electronica Sinica, 2004, 32(11): 1824-1828.)
- [8] 杨孔雨, 王秀峰. 自适应多模态免疫进化算法的研究与实现[J]. 控制与决策, 2005, 20(6): 717-720.
(Yang K Y, Wang X F. Research and implement of adaptive multimodal immune evolution algorithm[J]. Control and Decision, 2005, 20(6): 717-720.)
- [9] 陶新民, 杜宝祥. 基于高阶统计特征实值阴性克隆选择算法的轴承故障检测[J]. 机械工程学报, 2008, 44(7): 230-236.
(Tao X M, Du B X. Bearing fault detection using real-valued negative clone selection[J]. Chinese J of Mechanical Engineering, 2008, 44(7): 230-236.)
- [10] Gong M G, Du H F, Jiao L C. Optimal approximation of linear systems by artificial immune response[J]. Science in China, 2006, 49(1): 63-79.
- [11] 刘若辰, 杜海峰, 焦李成. 一种免疫单克隆策略算法[J]. 电子学报, 2004, 32(10): 1880-1884.
(Liu R C, Du H F, Jiao L C. An immune mono clonal strategy algorithm[J]. Acta Electronica Sinica, 2004, 32(10): 1880-1884.)
- [12] 程博, 郭振宇. 一种并行免疫进化策略算法研究[J]. 控制与决策, 2007, 22(12): 1395-1420.
(Cheng B, Guo Z Y. A parallel immune evolutionary strategy[J]. Control and Decision, 2007, 22(12): 1395-1420.)
- [13] 马文萍, 焦李成. 基于量子克隆优化的SAR图像分类[J]. 电子学报, 2007, 35(12): 2241-2246.
(Ma W P, Jiao L C. SAR image classification based on quantum clonal optimization[J]. Acta Electronica Sinica, 2007, 35(12): 2241-2246.)

~~~~~

(上接第174页)

- [4] Petersen I R, McFarlane D C. Optimal guaranteed cost control and filtering for uncertain linear systems[J]. IEEE Trans on Automatic Control, 1994, 37(7): 1971-1977.
- [5] Petersen I R. Guaranteed cost LQG control of uncertainty linear systems[J]. IEE Proc of Control Theory Application, 1995, 142(2): 95-102.
- [6] Moheimani S O R, Petersen I R. Optimal guaranteed cost control of uncertainty system via static and dynamic output feedback[J]. Automatica, 1996, 32(4): 575-579.
- [7] 俞立, 陈国定, 潘海天. 不确定离散时间系统的  $H_2/H_\infty$  最优保性能控制[J]. 控制与决策, 2001, 16(2): 151-154.  
(Yu L, Chen G D, Pan H T.  $H_2/H_\infty$  optimal guaranteed cost control of uncertain discrete-time systems. Control and Decision, 2001, 16(2): 151-154.)
- [8] 高振斌, 钱富才, 刘丁. 基于LMI不确定随机系统输出反馈保性能控制[J]. 信息与控制, 2008, 37(4): 385-390.  
(Gao Z B, Qian F C, Liu D. Output feedback guaranteed cost control of uncertainty discrete stochastic system based on LMI[J]. Information and Control, 2008, 37(4): 385-390.)
- [9] 俞立. 鲁棒控制—线性矩阵不等式处理方法[M]. 北京: 清华大学出版社, 2002.  
(Yu L. Robust control – Linear matrix inequality method[M]. Beijing: Tsinghua University Press, 2002.)
- [10] 郭尚来. 随机控制[M]. 北京: 清华大学出版社, 2000.  
(Guo S L. Stochastic control[M]. Beijing: Tsinghua University Press, 2000.)