第 51 卷　第 1 期　中山大学学报（自然科学版）　Vol. 51　No. 1
2012 年　1 月　ACTA　SCIENTIARUM　NATURALIUM　UNIVERSITATIS　SUNYATSENI　Jan.　2012

一种改进的微种群遗传算法*

燕乐纬[1]，陈洋洋[2]，周　云[1]
（1. 广州大学土木工程学院，广东 广州 510006；
2. 广州大学工程抗震研究中心，广东 广州 510006）

摘　要： 采用种群隔离机制、算术交叉、杰出者保留策略等对微种群遗传算法进行了改进。减少了重启动次数，增强了两次重启动之间遗传优化过程的全局和局部搜索能力，使算法在尽可能保有模式识别信息的前提下进行智能搜索；采用了实数编码，减少了编码和解码过程中的计算开销；引入了自适应随机变异算子，使之在不增加循环次数的前提下，增加了利用现有种群已经获得的遗传信息进行有效搜索的次数；引入了异种机制，有效提高了微种群遗传算法收敛于全局最优解的概率，加快了收敛速度。最后，标准测试函数的测试结果证明了这一改进的微种群遗传算法能够用远低于标准遗传算法的计算代价获得更佳的优化效果。

关键词： 微种群遗传算法；异种机制；自适应非均匀变异；算数交叉；实数编码

中图分类号： TU318；O224　　**文献标志码：** A　　**文章编号：** 0529－6579（2012）01－0050－05

An Improved Micro-genetic Algorithm

YAN Lewei[1], *CHEN Yangyang*[2], *ZHOU Yun*[1]
（1. School of Civil Engineering, Guangzhou University, Guangzhou 510006, China;
2. Earthquake Engineering Research and Test Center; Guangzhou University, Guangzhou 510006, China）

Abstract: Population isolation, arithmetic crossover and optimum reserved strategy are used to improve micro-genetic algorithm (mGA). Reset frequency is decreased while the global and local searching capabilities of mGA between two resets are enhanced, which makes mGA searching the parameter space intelligently as the mode recognition information is preserved as much as possible. Real-code is used to decrease the computing cost in encoding and decoding. Adaptive random mutation with existing genetic information of the current groups is used to increase efficient search. Heterogeneous strategy is used to improve the probability of convergence to global optimal solution and quicken up the convergence. Finally, standard functions testing demonstrate that the improved mGA can find better optimum solutions with less computing cost than standard genetic algorithm (SGA).

Key words: micro genetic algorithm; heterogeneous strategy; adaptive random mutation; arithmetic cross; real code

标准遗传算法（Standard Genetic Algorithm，SGA）在工程应用中有两个瓶颈问题[1-2]，一是控制参数过多，包括种群规模、位串长度、选择压力、交叉概率、变异概率等。这些控制参数对遗传算法的优化效果意义重大，但它们的设定需要对遗传算法和优化问题本身都有相当水平的专业知识。二是与基于梯度的寻优策略相比，作为一类随机优化方法，遗传算法的收敛速度相对较慢[3-4]。究其原因，是因为遗传算法在遗传进化过程中仅使用了目标函数值而未使用包含有更多优化信息的连续、

* 收稿日期：2011－07－15
基金项目：国家自然科学基金资助项目（11102045）；广东省自然科学基金博士启动资助项目（S2011040004039）；广东省高校优秀青年创新资助项目（LYM10108）
作者简介：燕乐纬（1978 年生），男，讲师，博士；E-mail：ylw21@139.com

梯度等条件。基于以上两个事实，SGA 的应用大多局限于设计空间非凸、离散、不连续，目标函数与设计变量之间的关系不明确、多峰、不可导等传统优化方法难以求解的复杂问题，且计算量十分庞大。

为提高遗传算法的优化效率，众多研究者从宏观策略和微观策略的不同角度提出了很多改进方法[5-7]。1989 年，Krishnakumar[8]在 SGA 的基础上提出了微种群遗传算法（micro Genetic Algorithm, mGA）[9]。mGA 的基本思想是尽可能地缩小遗传算法的种群规模 N，以提高遗传算法的优化效率。由于几乎所有遗传操作都与种群规模 N 有着直接的关系，减小种群规模显然可以减少遗传算法在编码、选择、交叉、计算个体的适应度等遗传操作中的计算开销。对于主要计算量集中在目标函数值计算过程的复杂工程优化问题来说，mGA 减小种群规模的做法意义非同凡响。

由于 mGA 的种群规模极小，种群多样性较差，很容易出现早熟收敛。mGA 采用重启动策略来解决早熟收敛的问题。此外，基于 SGA 中变异算子仅仅是以极小概率发生的辅助算子的思想，Krishnakumar 在 mGA 中摒弃了变异算子。种群规模极小且不含变异算子，这是 mGA 区别于其他遗传算法的两个重要特征。

但是，这一 mGA 存在着一定的缺陷[10]。首先，mGA 的每一次重启动，都仅仅保留了一个最优个体而抛弃了遗传优化过程得到的模式识别和分析信息，这一做法显然与遗传算法的基本思想背道而驰。其次，从适应度函数的一般特征来看，最优个体的邻域是需要重点搜索的区域。mGA 取消了变异算子，实际上是剥夺了遗传算法的局部搜索能力。

本文将在以往工作的基础上[11-13]，对 mGA 进行改进。基本的思路是减少重启动次数，增强两次重启动之间遗传优化过程的全局和局部搜索能力，使算法在尽可能保有模式识别信息的前提下进行智能搜索；加入自适应变异算子，使之参与 mGA 的全局搜索和局部搜索，充分发挥算法的广度和深度搜索能力。

1　宏观策略

1.1　实数编码

实数编码的批评者认为，实数编码不具备基因的外在表现形式，在交叉和变异过程中未能体现基因交换和基因突变的细节特征，与遗传算法建立在生物进化论思想上这一基础相背离。实数编码的支持者对这这一点的解释是：生物进化论只是遗传算法的思想基础而非必须遵守的准则；工程优化问题考虑的是优化问题本身，只要优化问题能够得到求解，遗传算法在形式上与生物进化论有所背离是可以接受的[14]。

mGA 种群中的个体只有 5 ~ 10 个，从生物学的角度来看，如此小的种群无法维持种群生存的基本要求。在进行交叉运算时，这一微小种群中所含的模式数目过少，模式识别和组合能力有限。因而在基因的重组和进化方面，二进制编码并不比实数编码更具优势。此外，Krishnakumar 提出 mGA 的根本目的就在于提高遗传算法的计算效率，避免过多的计算开销。二进制编码的频繁编码和解码过程显然与这一目的相悖。基于以上两个事实，可知 mGA 比 SGA 更有理由采用实数编码。

1.2　渐进阶段和骤变阶段

将进化过程分为渐进和突变两个不同的阶段，并采用自适应的交叉算子和变异算子实现从突变阶段到渐进阶段的逐渐转化，是遗传算法提高搜索效率的有效手段。

对于 mGA 来说，由于采用了重启动策略，渐进阶段和骤变阶段的含义与以往有所不同。从 mGA 的全局进化过程来看，由于每一次重启动都只保留了一个最优个体而抛弃了其余的所有进化信息，进化阶段的划分没有实际意义。而两次重启动之间则是一个完整进化过程，本文提出的改进 mGA 的渐进阶段和骤变阶段，就是针对这一进化过程而言的。

1.3　种群隔离机制

算法采用了被大量实践证明能够有效提高收敛速度、抑制早熟收敛的种群隔离机制。种群隔离机制实际上是一种人为干预遗传优化过程，增强遗传算法的并行性的方法。对于种群规模极小的 mGA 来说，种群隔离机制提高种群多样性、防止早熟收敛的意义更加明显。

2　微观策略

2.1　相同个体的处理

由于 mGA 的种群规模很小，即使种群中只有一对相同个体，也会大大削弱种群的多样性，所以绝对不允许出现相同个体。mGA 的所有遗传算子都带有摒弃相同个体的设置，在进行每一次遗传操作时，都要根据新产生的个体与现有种群中个体的距离大小考虑是可以保留还是需要重新生成。这是

mGA 必不可少的步骤之一。

2.2 选择算子

采用允许父代和子代相竞争，联赛规模为 2 的不放回锦标赛选择算子，选择过程持续到种群内的所有个体全部被选中或者淘汰为止。在这样的选择策略下，最优个体总是能够存活下来，从实际效果上实现了杰出者选择策略。

2.3 交叉算子

交叉算子采用算术交叉。设 X_i^T 和 X_j^T 分别是第 T 代种群中两个参与杂交的个体，实数编码的遗传算法得到的杂交子代是：

$$\begin{cases} X_i^{T+1} = X_i^T + \tau_1(X_i^T - X_j^T) \\ X_j^{T+1} = X_j^T + \tau_2(X_i^T - X_j^T) \end{cases} \tag{1}$$

式中 τ_1 和 τ_2 是在［-1，1］上均匀分布的随机数。

算术交叉的思想来源于凸集理论中向量的线性组合。它可以保证搜索区域覆盖 X_i^T 和 X_j^T 的所有邻域，并以较大概率落在二者之间。当用种群隔离机制剖分的子域中只包含一个局部最优解或某个局部最优解的适应度明显大于其他局部最优解时，算术交叉算子会使子域内的个体逐渐向这一最优解靠拢。这样，就使得遗传算法算法从渐进阶段逐渐过渡到了骤变阶段。算术交叉算子的这一特点使它具备了很高的搜索效率。

2.4 变异算子

囿于 SGA 中变异算子是以很小概率发生的辅助算子这一思想，Krishnakumar 提出的 mGA 中摒弃了变异算子。实际上，mGA 的种群规模小，搜索能力有限，加入变异算子能够在不增加循环次数的前提下，增加利用现有种群已经获得的遗传信息进行有效搜索的次数。此外，交叉算子全局搜索能力较强而局部搜索能力较弱，变异算子是交叉算子的有益补充。在本文所提出的改进 mGA 中，将采用自适应随机变异算子，并令变异概率为 100%，即所有个体全部执行变异操作。

设变异个体 X_i^T 的编码为

$$X_i^T = (x_{i1}^T \quad x_{i2}^T \quad \cdots \quad x_{ik}^T \quad \cdots \quad x_{in}^T) \tag{2}$$

其中 x_{ik}^T 为选中进行变异的元素。其变异结果是：

$$x_{ik}^{T+1} = x_{ik}^T + \tau b \tag{3}$$

式中的 τ 是在［-1，1］上取值的随机数，b 是变异算子的取值半径。这一变异算子能够保证 x_{ik}^{T+1} 在 x_{ik}^T 的邻域 $U(x_{ik}^T, b)$ 内取到。取值半径 b 是一个随繁殖代数变化的量：

$$b = \begin{cases} \dfrac{1}{q_m^T} b_u & \text{when} \quad \dfrac{1}{q_m^T} b_u > b_l \\ b_l & \text{else} \end{cases} \tag{4}$$

式中 T 是当前繁殖代数，b_l 和 b_u 分别是取值半径 b 的下界和上界。q_m 是这一非均匀变异算子引入的一个用于调整变异半径 b 的收缩速度的参数，其取值范围在 1.1～2.0 之间。

（4）式使这一变异算子具备了自适应变焦的能力。随着循环次数的增长，变异半径逐渐收缩。在优化的渐进阶段，变异范围较大，变异算子参与全局搜索，目的是快速锁定最优解所在区域；在优化的骤变阶段，变异范围缩小，变异算子用于执行局部精细搜索，目的是准确地确定最优解。式中设定变异下界 b_l 的目的是防止骤变阶段变异算子因变异范围缩得太小而失去意义。

非均匀变异算子和算术交叉算子的自适应特性使得遗传进化过程从渐进阶段逐渐转换到了骤变阶段，实现了遗传算法全局搜索和局部搜索的协调与配合。

2.5 异种机制

异种机制是指在遗传优化过程中，以较小的概率在当前种群的搜索范围之外继续选择种子进入循环进程，增加种群多样性的方法。异种机制模拟的是人工干预物种进化，获取最优物种的过程。遗传算法的过程性使这一干预得以实现。

异种机制的作用机理是：

1）对优化过程进行监测，如果发现当前种群发生了近亲繁殖，则启动异种机制；

2）在当前种群的搜索范围之外选择数个异种，用以代替现有种群中适应度较低的个体；

3）用带有异种的种群生成子代，进入下一次循环。

由于 mGA 的种群规模较小，发生近亲繁殖造成的后果更加严重，有必要严格控制相似个体的出现。在本文提出的改进 mGA 中，近亲繁殖判别式直接考虑任意两个个体之间距离是否小于预先设定的临界值 d_r，即

$$d(X_i^T, X_j^T) < d_r \tag{5}$$

（5）式即本文提出的改进 mGA 的近亲繁殖判别式。其中临界值 d_r 按下式自适应变焦：

$$d_r = \begin{cases} \dfrac{1}{q_c^{T_r}} d_{ru} & \text{when} \quad \dfrac{1}{q_c^{T_r}} d_{ru} > d_{rl} \\ d_{rl} & \text{else} \end{cases} \tag{6}$$

式中 T_r 是从最近一次重启动开始计数的繁殖代数。对于改进的 mGA 来说，两次重启动之间是一个完

整的遗传算法循环，临界值 d_r 只有随着 T_r 的变化自适应变焦才有意义。

此外，在确定 d_r 的上界 d_{ru} 和下界 d_{rl} 时，需要计算个体之间的"需要距离" d_t 。在本文提出的改进 mGA 中，由于种群规模过小，采用计算初始种群个体之间平均距离的方法计算"需要距离" d_t 会产生较大的误差。因此，需要在优化开始前先运行一个专门计算 d_t 的子程序。具体做法是在搜索空间中生产足够多的个体，然后在其中随机选择 N（种群规模）个个体计算平均距离，重复多次求其平均值，将这一平均值记做 d_t ，再取 $d_{ru} = (1/4)d_t$ 即可。根据设计空间的大小和优化问题求解精度 δ_L 的不同，d_r 的下界 d_{rl} 的取值可以扩大到 20 ~ 30 倍的 δ_L ，这样就可以使种群在骤变阶段加速收敛时还能保证一定的多样性。

基于同样的原因，本文提出的改进 mGA 在选择异种时仅采用随机选取异种的方法。异种机制的最大取值半径 b_h 也是 T_r 的函数

$$b_h = \begin{cases} \dfrac{1}{q_h^{T_r}} b_{hu} & \text{when } \dfrac{1}{q_h^{T_r}} b_{hu} > b_{hl} \\ b_{hl} & \text{else} \end{cases} \tag{7}$$

实际应用中，可以取 b_h 的上界 $b_{hu} = 4d_t$ ，下界 b_{hl} 为 50 ~ 100 倍的 δ_L 。

2.6　算法流程

在本章的最后，给出改进的 mGA 流程如下：

1）用实数编码对设计空间中的可行解编码；

2）随机生成 8 个个体的值作为初始种群；

3）分别计算这 8 个个体的适应度，用不放回锦标赛选择选出 4 个适应度较高的个体；

4）将这 4 个个体作为父代个体随机配对，实施算术交叉，生成 4 个子代个体；

5）将算术交叉得到的 4 个子代个体和原有的 4 个父代个体合并，形成中间种群；

6）用不放回锦标赛选择在中间种群中选出 4 个适应度较高的个体；

7）对这 4 个个体分别进行变异操作，生成 4 个变异个体；

8）将这 8 个个体合并得到新一代种群；

9）判断种群是否发生了近亲繁殖，若是，则启动异种机制；

10）判断是否满足终止条件，若是，进入第 13）步，否则进入第 11）步；

11）判断当前种群是否基因型收敛，若是，进入下一步，若否，转 3）；

12）启动重启动策略，采用不放回锦标赛选择在中间种群中选出 4 个适应度较高的个体，再随机产生 4 个个体，组成新的种群，并转至 3）；

13）将当前种群中适应度最大的个体作为改进 mGA 得到的最优解输出，结束算法的循环过程。

3　性能测试

为验证本文所提出算法的有效性，采用标准测试函数 Schaffer1 function（F6 函数）、Rana's function（F9 函数）和 Himmelbau's function（F14 函数）进行测试，并与带有杰出者保留策略的 SGA 进行了对比。所用 SGA 的种群规模 $N = 100$ ，交叉概率 $p_c = 0.8$ ，变异概率 $p_m = 0.05$ ，终止代数 $T = 300$ 。

Schaffer1 function（F6 函数）：

$$\begin{cases} \min f(x) = (x_1^2 + x_2^2)^{0.25} \\ [\sin^2(50 \times (x_1^2 + x_2^2))^{0.1} + 1.0] \\ x_i \in [-10, 10] \\ (x_1 *, x_2 *) = (0, 0) \\ f(x_1 *, x_2 *) = 0 \end{cases}$$

Rana's function（F9 函数）：

$$\begin{cases} \min f(x,y) = x \times \sin(\sqrt{|y + 1 - x|}) \times \\ \cos(\sqrt{|y + 1 - x|}) + (y + 1) \times \\ \cos(\sqrt{|y + 1 - x|}) \times \sin(\sqrt{|y + 1 - x|}) \\ x_i \in [-512, 512] \\ (x *, y *) = (-512, -512) \\ f(x *, y *) \approx -511.7 \end{cases}$$

Himmelbau's function（F14 函数）

$$\begin{cases} \min f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \\ x_i \in [-6, 6] \\ (x *, y *) \in \{(3.0, 2.0), (3.584\,4, -1.848\,2) \\ (-2.805\,1, 3.131\,3), (-3.773\,9, -3.283\,2)\} \\ f(x *, y *) = 0 \end{cases}$$

分别用 SGA 和本文所提改进后的 mGA 对以上三个测试函数进行了寻优计算。运行 50 次的对比测试结果如表 1 所示。表中"最优值"是 50 次运算得到的最优结果；"平均值"是指 50 次优化结果的平均；"运算次数"算法收敛所需的平均适应值计算次数。

从表中可以看出，改进后的 mGA 可以用远低于 SGA 的运算次数获得比 SGA 更佳的优化结果。对于工程优化问题来说，主要的运算开销集中于适应值的计算上，因而减少适应值的计算次数，往往意味着优化时间的缩减和计算效率的提高。

表1 SGA 和改进后的 mGA 测试结果对比
Table1 Comparison of the test result by SGA and improved mGA

算法		F6 函数	F9 函数	F14 函数
理论最优值		0	0	0
SGA	平均值	0.162 708	−508.857 2	1.70×10^{-2}
	运算次数	18 576	17 140	17 530
	最优值	0.067	−511.73	3.29×10^{-6}
改进的 mGA	平均值	0.026 948	−510.508 2	2.99×10^{-6}
	运算次数	4 702.76	13 186.12	4 113.94
	最优值	0.012 5	−511.77	2.57×10^{-8}

需要指出的是，计算结果显示，相比于 SGA，改进后的 mGA 对初始种群的平均适应度较为敏感。以 F6 函数的优化过程为例，mGA 收敛时适应值函数运算次数的最小值仅为 260 次，最大值则达到 30 122 次，平均值为 4 702.76。而 SGA 对应的这 3 个值分别是 5 900、29 900 和 18 576。这显然是因为 SGA 的种群规模大，并行性和鲁棒性都较好造成的结果。考虑到最终种群适应值的改善和平均计算次数的显著减小，总体而言，本文提出的改进 mGA 仍然是值得采用的方法。

4 结 论

本文采用种群隔离机制、算术交叉、杰出者保留策略等对 mGA 进行了改进。减少了重启动次数，增强了两次重启动之间遗传优化过程的全局和局部搜索能力，使算法在尽可能保有模式识别信息的前提下进行智能搜索。加入了自适应变异算子，使之参与 mGA 的全局搜索和局部搜索，充分发挥算法的广度和深度搜索能力。引入了异种机制，加快了遗传算法的收敛效率，提高了遗传算法收敛于全局最优解的概率。标准测试函数测试结果表明，这一改进的 mGA 能够用远低于 SGA 的适应值计算代价获得更佳的优化结果，具备较好的工程实用意义。

参考文献：

[1] HOLLAND J H. Adaptation in naturaland artificial systems[M]. Ann Arbor, Michigan: University of Michigan Press, 1975.

[2] HOLLAND J H. Induction: processes of inference, learning, and discovery [M]. Cambridge: MIT Press, 1987.

[3] GOLDBERG D. E. Genetic algorithm in search, optimization and machine learning [M]. MA: Addison-Wesley Publishing Company, 1989.

[4] MARINSKIS Y, MARINSKIS, M. A hybrid genetic-particle swarm optimization algorithm for the vehicle routing problem [J]. Expert Systems with Applications, 2010, 37(2): 1446 − 1455.

[5] AMIN J, MOHAMMAD A S, REZA T M. A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem [J]. International Journal of Advanced Manufacturing Technology, 2010, 9:1 − 14.

[6] VAGELIS P, MANOLIS P. A hybrid particle swarm-gradient algorithm for global structural optimization [J]. Computer-Aided Civil and Infrastructure Engineering, 2011, 26(1): 48 − 68.

[7] MOHAMED J A H, SIVAKUMAR R. A survey: hybrid evolutionary algorithms for cluster analysis[C]. Artificial Intelligence Review, 2011: 1 − 26.

[8] KRISHNAKUMAR K. Micro-genetic algorithm for stationary and non-stationary function optimization [M]. SPIE Proceedings: Intelligent Control and Adaptive Systems, 1989: 289 − 296.

[9] HAMZEH A, RAHMANI A. Approximating arbitrary reinforcement signal by learning classifier systems using micro genetic algorithm [J]. Fundamenta Informaticae, 2008, 86(1/2): 93 − 111.

[10] ROY S, GHOSH S, SHIVPURI R. A new approach to optimal design of multi-stage metal forming processes with micro genetic algorithms [J]. International Journal of Machine Tools and Manufacture, 1997, 37(1): 29 − 44.

[11] 燕乐纬，陈树辉. 基于改进遗传算法的非线性方程组求解 [J]. 中山大学学报：自然科学版，2011，50(1)：9 − 13.

[12] SU R K L, YAN L W. Provision of reinforcement in concrete solids using the generalized genetic algorithm [J]. Journal of Computing in Civil Engineering, ASCE, 2011, 25(3): 211 − 217.

[13] 燕乐纬，陈树辉. 一种改进的广义遗传算法及其在结构动力优化问题中的应用 [J]. 工程力学，2010，27(5)：21 − 26.

[14] 王正志，薄涛. 进化计算 [M]. 长沙：国防科技大学出版社，2000.