

双精英协同进化遗传算法^{*}

刘 全^{1,2+}, 王晓燕¹, 傅启明¹, 张永刚², 章晓芳^{1,3}

¹(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

²(符号计算与知识工程教育部重点实验室(吉林大学), 吉林 长春 130012)

³(计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210093)

Double Elite Coevolutionary Genetic Algorithm

LIU Quan^{1,2+}, WANG Xiao-Yan¹, FU Qi-Ming¹, ZHANG Yong-Gang², ZHANG Xiao-Fang^{1,3}

¹(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University), Changchun 130012, China)

³(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

+ Corresponding author: E-mail: quanliu@suda.edu.cn

Liu Q, Wang XY, Fu QM, Zhang YG, Zhang XF. Double elite coevolutionary genetic algorithm. *Journal of Software*, 2012, 23(4): 765–775. <http://www.jos.org.cn/1000-9825/4040.htm>

Abstract: A new double elite coevolutionary genetic algorithm (DECGA) is proposed to avoid the premature convergence and low speed of convergence, based on the elite strategy and the concept of coevolutionary. In the DECGA, the two different and high fitness individuals (elite individuals) are selected as the core of the evolutionary operation, and the team members are selected by the different evaluation functions to form two teams through these two elite individuals. The two sub-populations can balance the capability of exploration and exploitation by the different evolutionary strategies. Theoretical analysis proves that the algorithm converges to the global optimization solution. Tests on the functions show that the algorithm can find the global optimal solution for most test functions, and it can also maintain the population diversity to a certain range. Compared with the existing algorithms, DECGA has a higher performance in precision of convergence and search efficiency.

Key words: genetic algorithm; evolutionary algorithm; elitist strategy; coevolution; population diversity

摘 要: 针对传统遗传算法早熟收敛和收敛速度慢的问题,提出一种双精英协同进化遗传算法(double elite coevolutionary genetic algorithm,简称 DECGA)。该算法借鉴了精英策略和协同进化的思想,选择两个相异的、高适应度的个体(精英个体)作为进化操作的核心,两个精英个体分别按照不同的评价函数来选择个体,组成各自的进化子种群,两个子种群分别采用不同的进化策略,以平衡算法的勘探和搜索能力。理论分析证明,该算法具有全局收敛性。通过对测试函数的实验,其结果表明,该算法能搜索到几乎所有测试函数的最优解,同时能够有效地保持种群的多样性。与已有算法相比,该算法在收敛速度和搜索全局最优解上都有了较大的改进和提高。

* 基金项目: 国家自然科学基金(60873116, 61070223, 61103045, 61170020); 江苏省自然科学基金(BK2008161, BK2009116); 江苏省高校自然科学研究项目(09KJA520002, 09KJB520012); 吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172012K04)

收稿时间: 2010-06-23; 修改时间: 2011-01-31; 定稿时间: 2011-04-28

关键词: 遗传算法;进化算法;精英策略;协同进化;种群多样性

中图法分类号: TP181

文献标识码: A

遗传算法(GA)是进化算法中产生最早、影响最大、应用较为广泛的一种分支算法.它采用达尔文进化论的适者生存、优胜劣汰的进化思想.作为一种全局搜索方法,它具有简单性、鲁棒性和通用性等特点.自提出以来,在工程设计、机器学习、模式识别、图像处理等方面都得到了广泛应用.尽管如此,遗传算法在理论和应用方法上还有很多亟待完善之处,如早熟收敛、收敛速度慢等问题.

精英个体通常是指适应度较高的个体,文献[1]指出,群体中的精英个体和全局最优解之间的亲和度要大于群体中其他个体和全局最优解之间的亲和度,并且与精英个体有较大亲和度的个体也应有较高的适应度.因此,精英个体对种群的进化起着重要的推动作用,充分利用精英个体的特征信息是保证算法性能的关键.

协同进化是近年来计算智能研究的一个热点,最早由 Ehrlich 和 Raven 提出^[2].它是指多个种群之间通过适应度的关联同时进化,从而提高各自的性能.它被广泛定义为一种适应度基于种群密度、种群自身及相互作用种群的遗传成分的进化.与传统遗传算法相比,它具有强搜索能力和渐进学习能力,能够克服传统遗传算法的早熟收敛现象.

协同进化有两种进化方式:竞争协同进化和协作共同进化.协同进化个体在自身进化过程中常受以下 3 个因素的影响:个体适应度、所处生存环境和个体之间的相互竞争,通过协调这 3 个因素,能够有效地解决遗传算法中进化模式单一性的问题,从而较好地保持种群个体的多样性,避免了未成熟收敛和收敛速度慢等问题. Potter 等人^[3]较早提出了一种合作性协同进化遗传算法(CCGA),其基本思想是,将待求问题的解向量分裂成多个子群体独自进化,子群体组合成完整的解,并通过适应度函数来评价完整解的优劣.

近年来,遗传算法的研究仍然被众多学者所关注,并取得了显著的成果.岳崑等人^[4]通过对多维解析函数的多次重复计算并对计算结果进行统计分析证明,遗传算法计算结果具有一定的稳定性,也可以通过采用多次重复计算方法提高遗传算法的可靠性和可信度.陈皓等人^[5]提出了族群进化算法,该算法采用族群机制,设计了针对二进制编码方式的族群分类方法,同时,基于该族群结构形成具有双规协同进化特征的族群进化算法以及相应的族群算子,能够显著提高群体的抗早熟能力和搜索效率.周昌乐等人^[6]提出了一种基于遗传算法的宋词生成计算模型,该模型采用精英主义和轮盘赌算法的选择策略,同时采用部分映射和启发式交叉算子和启发式变异算子,从而实现了机器自动生成宋词的目标,较好地将遗传算法应用于实际项目中.应伟勤等人^[7]提出了一种集合热力学选择.该选择首先定义角度熵,通过扇形采样来度量种群逼近方向的多样性,然后利用距离精英定义距离来度量种群的逼近程度,从而有效地提高了热力学遗传算法的运行效率和解集分布均匀性.江中央等人^[8]提出了一种基于正交实验设计的混合自适应正交遗传算法(GSOGA),该算法利用正交实验设计方法设计交叉算子,并提出一种自适应正交交叉算子,同时引入局部搜索策略提高算法局部搜索能力和收敛速度.

与此同时,很多学者结合精英、协同进化思想提出了不同的改进方案.文献[9]提出了一种解决多目标优化问题的分布式协同进化算法(CCGA),该算法采用了与文献[3]相同的进化思想,将待求问题的解向量分裂成多个子种群协同进化;文献[10]提出了一种混合竞争协同进化遗传算法(hlCGA),其基本思想是,将局部搜索方法引入到协同进化遗传算法中,提高了算法局部搜索的能力.但上述两种算法都采用了解空间进行分割的策略,因此比较容易陷入局部极小值的问题.文献[1]提出了一种蜜蜂进化型遗传算法,该算法充分利用了精英个体的信息.但该算法中随机种群的比例参数需要通过人为经验来确定,一定程度上增加了算法的随机性.文献[11]提出了一种 M-精英协同进化数值优化算法,算法结合精英策略和协同进化思想,以精英个体作为进化的主体,各精英个体团队协同进化.但算法中的种群划分需要人为控制,一定程度上影响了算法的自适应调节能力.文献[12]提出了一种解决多维背包问题的新型并行遗传算法,该算法将整个种群分成 3 个子种群分别进化,但算法中各子种群采用相同的进化策略,并且进化过程相对独立,一定程度上降低了算法的开采和勘探能力.同时,算法中需要人工设定的参数较多,一定程度上增加了算法的随机性.

在上述算法的基础上,本文提出了一种双精英协同进化遗传算法(double elite coevolutionary genetic

algorithm,简称 DECGA).该算法首先选择两个精英个体,这两个精英个体按照不同的选择策略来完成各自进化子种群中个体成员的选择.两个子种群有不同的进化目标:子种群 1 的进化主要是加快算法的收敛速度,通过引入水平集概念自动地对种群进行分割,进而通过协作交叉操作避免相似个体间的无效交叉;子种群 2 的主要进化目标是提高种群的多样性,并加入随机种群,通过分阶段调节随机种群的比例参数,更好地平衡算法的勘探和开采操作,避免算法早熟收敛或收敛速度慢等问题的产生.两个子种群协同进化,最终协同完成整个种群的进化操作.

本文算法在一定程度上减少了参数对算法的影响,提高了算法的自学习能力.理论分析证明,DECGA 收敛于全局最优解.实验数据表明,算法计算代价低、寻优能力较强.

1 双精英协同进化遗传算法 DECGA

1.1 个体评价策略

协同进化遗传算法的难点是如何恰当地确定子群体之间的合作关系,本文采用两种策略对群体中的个体进行适应度评价.

定义 1. 任意优化问题的二进制编码空间为 $\{0,1\}^L$,群体规模设计为 n ,群体所包含的个体集合 $P=\{a_1, a_2, \dots, a_n\}$,个体 a_i 和 a_j 之间的差异度为

$$D(i, j) = \frac{1}{L} \sum_{l=1}^L (a_{il} - a_{jl}), \quad i, j = 1, 2, \dots, n \quad (1)$$

其中, $a_{il} - a_{jl} = \begin{cases} 0, & a_{il} = a_{jl} \\ 1, & a_{il} \neq a_{jl} \end{cases}$, a_{il}, a_{jl} 分别表示 i 和 j 个体第 l 位的值.不难推算,若个体 i, j 的每一位相异,则差异度

$D(i, j)=1$;若个体 i, j 每一位相同,则差异度 $D(i, j)=0$.

本文的第 1 种个体评价策略采用常规的方法来评价个体的好坏,即适应度函数完全依赖于目标函数本身,将其适应度函数表示为 $F(x_i)$, i 表示种群中的个体.但是常规的个体评价策略会使算法的多样性急剧下降,从而导致算法早熟收敛.针对上述问题,本文设计了第 2 种个体评价策略,其适应度函数表示如公式(2)所示:

$$F'(x_j) = D(i, j) \times F(x_j) \quad (2)$$

其中, i 表示除去 j 以外的其他种群个体.第 2 种评价策略在常规评价策略的基础上引入了差异度作为其权系数,该策略能够有效地保持种群的多样性,从而避免因有效模式缺失而导致的早熟收敛问题.

1.2 协作交叉操作

传统的遗传算法采用统一的交叉、变异策略,很大程度上增加了算法的随机性,降低了算法的收敛速度.本文受 MECA 算法^[11]的启发,设计了一种新的进化操作:协作交叉操作.

定义 2. 设群体所包含个体集合 $P(0)=\{a_1, a_2, \dots, a_n\}$,群体规模设计为 n ,适应度函数为 $f(x)$,令 $\bar{f}_0 = \sum_{i=1}^n \frac{f(x_i)}{n}$,做集合 $H_{\bar{f}_0} = \{X_i \in P(0) | f(x_i) \geq \bar{f}_0, i=1, 2, \dots, n\}$, $H_{\bar{f}_0}$ 称为 f 关于 $P(0)$ 的水平集.

定义 3(种群分割策略). 设进化第 t 代的种群为 $P(t)$,将其按适应度函数降序排列,采用常规选择策略,并利用定义 2 求得其水平集 $H_{\bar{f}_t} = \{X_i \in P(0) | f(x_i) \geq \bar{f}_t, i=1, 2, \dots, n\}$,记下 $H_{\bar{f}_t}$ 中大于等于 \bar{f}_t 的最小个体的位置 l_t .这时,整个种群将被划分为两个子种群,位于 l_t 之前的子种群记为 $POPe$,后一部分子种群记为 $POPc$.

利用定义 3 提出的种群分割思想,将种群分割成两部分,对这两部分个体采用不同的进化策略.种群分割思想能够充分利用种群信息,使算法的进化具有方向性和适应性.

协作交叉操作的选择成员来自于 $POPe$,本文为协作交叉操作定义了两种交叉算子:单点交叉算子和翻转交叉算子.当个体 i, j 的差异度 $D(i, j) > u$ 时,采用单点交叉算子产生新个体 $x=(x_1, x_2, \dots, x_L)$ 和 $y=(y_1, y_2, \dots, y_L)$.

当个体 i, j 较为相似,即 $D(i, j) \leq u$ 时,普通的交叉操作很难产生新的个体,此时使用如公式(3)所示的翻转交叉算子:

$$\begin{cases} x' = (x_1, x_2, \dots, y_{k+t}, \dots, y_{k+1}, y_k, \dots, x_L) \\ y' = (y_1, y_2, \dots, x_{k+t}, \dots, x_{k+1}, x_k, \dots, y_L) \end{cases} \quad (3)$$

1.3 双精英进化机制

精英个体对于种群的进化起着至关重要的作用,采用精英策略的遗传算法能够较快地收敛到全局最优解. DECGA 算法采用双精英进化策略,以精英个体作为进化操作的核心,充分发挥精英个体在进化中的推动作用,以有方向地引导个体朝着好的方向进化.精英进化抽象模型如图 1 所示.

初始种群生成时,首先选择种群中前 M 个适应度最高的相异个体组成精英库种群,每一代中的两个精英个体都是从精英库中选择的,精英库中的个体也随着种群的进化而不断更新.将两个精英个体分别表示为 *EliteA* 和 *EliteB*,两个精英个体协同进化,最终完成算法的整个进化操作.

(1) 以 *EliteA* 为中心的子种群进化

以 *EliteA* 为中心的子种群进化目标是提高算法的收敛速度.*EliteA* 是每一代精英库中适应度值最高的个体,在以 *EliteA* 为中心的进化团队 $TeamA(i), i=1, 2, \dots, \frac{n}{4}$ 中,成员个体的选择采用常规选择策略,即第 1 种个体评价策略.首先通过种群分割策略对待选种群进行种群分割,将种群分割成两部分:优秀种群(*POPe*)和普通种群(*POPc*),并根据选择个体所属种群的不同采用不同的交叉策略.这一策略能够有效避免因种群个体过于相似而导致的无效交叉,从而提高算法的进化效率.以 *EliteA* 为中心的子种群进化模型如图 2 所示.

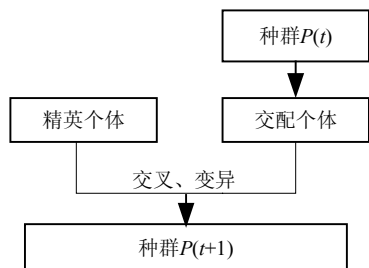


Fig.1 Abstract model of elite evolution

图 1 精英进化抽象模型

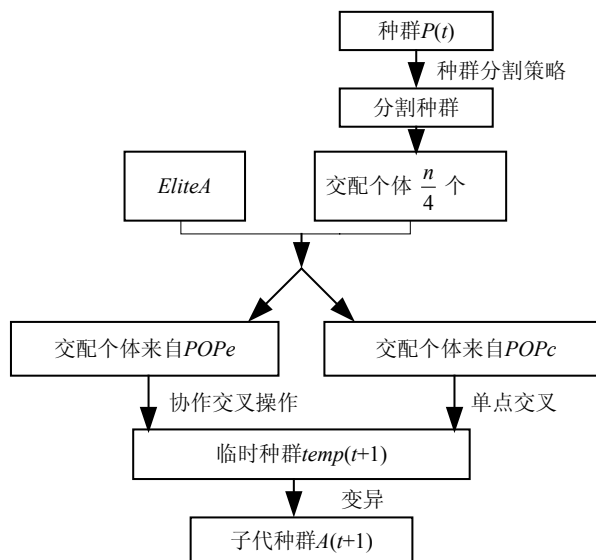


Fig.2 Evolutionary model of sub-population taking *EliteA* as the center

图 2 *EliteA* 进化模型图

(2) 以 *EliteB* 为中心的子种群进化

以 *EliteB* 为中心的子种群的进化目标是,避免出现算法由于选择压力而造成的过早收敛现象.通过比例选择在精英库中选择与 *EliteA* 相异的个体,记为 *EliteB*.在以 *EliteB* 为中心的进化团队 $TeamB(j), j=1, 2, \dots, \frac{n}{4}$ 中,成员个体的选择策略采用第 2 种个体评价策略,其函数如公式(3)所示.在此选择策略中引入了个体差异度的概念,避免因完全依赖目标值函数值的大小而造成的局部最优个体过早地占领整个种群这一现象的发生,从而能够有效地保持种群的多样性.随着进化代数的增加,种群多样性逐渐降低,此时,仅依赖于选择策略很难较好地维持种群多样性.基于此,文中引入随机种群 *randPop* 这一概念.以 *EliteB* 为中心的子种群进化模型如图 3 所示.

(3) 种群规模参数的调整

保持种群的多样性是保证遗传算法搜索到全局最优解的基本条件,同时它也对理解遗传算法的进化状态以及其运行机理有重要意义.

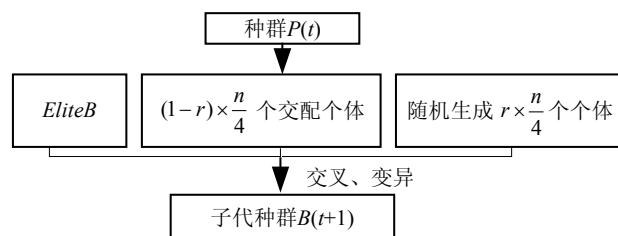


Fig.3 Evolutionary model of sub-population taking *EliteB* as the center

图3 *EliteB* 进化模型图

定义 4. 任意优化问题的二进制编码空间为 $\{0,1\}^L$, 群体规模设计为 n , 群体所包含的个体集合 $P=\{a_1, a_2, \dots, a_n\}$, 其中, $a_j=\{a_{1j}, a_{2j}, \dots, a_{Lj}\}$, $j=1, 2, \dots, n$, 其多样性测度如公式(4)所示:

$$div(P)=1-\frac{1}{L \times n} \sum_{l=1}^L \left(\max \left\{ \sum_{j=1}^n a_{lj}, \sum_{j=1}^n (1-a_{lj}) \right\} - \min \left\{ \sum_{j=1}^n a_{lj}, \sum_{j=1}^n (1-a_{lj}) \right\} \right) \quad (4)$$

其中, $\sum_{j=1}^n a_{lj}$ 表示种群所有个体第 l 位的 1 的个数; $\sum_{j=1}^n (1-a_{lj})$ 表示种群中所有个体第 l 位的 0 的个数;

$\max \left\{ \sum_{j=1}^n a_{lj}, \sum_{j=1}^n (1-a_{lj}) \right\} - \min \left\{ \sum_{j=1}^n a_{lj}, \sum_{j=1}^n (1-a_{lj}) \right\}$ 表示种群中所有个体第 l 位 0 和 1 的分布情况, 其值越大, 表示 0 和 1 的分布越不平衡, 即表示种群多样性的降低, 对于 t 代种群, 其种群多样性表示为 $div(P(t))$.

种群多样性过小而导致的有效基因缺失是造成算法早熟收敛的根本原因; 而种群多样性过大也会影响算法的收敛速度. 所以, 需要将种群的多样性保持在合适的范围内, 从而使算法达到对勘探和开采操作平衡的目的.

双精英进化策略中, 以 *EliteB* 为中心的子种群的进化能够有效地保持种群的多样性. 通过控制以 *EliteB* 为中心的子种群中随机种群的规模参数 r , 以显著地改善种群的多样性. 本文采用分阶段调控方法来调整参数 r , 其具体调整方法如下所示:

$$r=r_0+\lambda \times d \quad (5)$$

其中, $\lambda = \begin{cases} 0, & t \in [0, \max Gen/4) \\ 1, & t \in [\max Gen/4, \max Gen/2) \\ 2, & t \in [\max Gen/2, 3 \cdot \max Gen/4) \\ 3, & t \in [3 \cdot \max Gen/4, \max Gen] \end{cases}$, t 表示进化代数, $\max Gen$ 表示最大进化代数; d 表示参数调整步长.

1.4 DECGA算法描述

算法 1. 双精英协同进化遗传算法(DECGA).

Step 1. 初始化种群 $P(0)$, 并计算种群的适应度值, 将种群 $P(0)$ 按适应度值降序排列;

Step 2. 选择前 M 个适应度值相异的优秀个体作为精英库 $Best(0)$ 成员, 并令 $t=0$;

Step 3. 选择 $Best(t)$ 中的最优个体作为 *EliteA*, 并利用种群分割策略得到分割种群 *POPe* 和 *POPc*. 选择交配个体组成进化团队 $TeamA(i)$, $i=1, 2, \dots, \frac{n}{4}$. *EliteA* 个体与进化团队 $TeamA$ 中的个体完成进化操作:

(1) 若选择个体来自于 *POPe*, 则进行协作交叉操作. 计算 *EliteA* 个体与该个体的差异度 $D(EliteA, i)$, 若 $D(EliteA, i) \leq u$, 则进行翻转交叉; 否则, 进行单点交叉;

(2) 若选择个体来自于 *POPc*, 则进行单点交叉. 最终完成进化操作, 得到子种群 $A(t+1)$;

Step 4. 按比例选择法选择与 *EliteA* 相异的个体 *EliteB*, 以公式(2)作为种群个体评价函数, 在种群 $P(t)$ 中选

择 $(1-r) \times \frac{n}{4}$ 个个体,并随机生成 $r \times \frac{n}{4}$ 个个体,EliteB 的交配个体组成进化团队 $TeamB(j), j=1,2,\dots,\frac{n}{4}$. 与 EliteB 完成进化操作,得到子种群 $B(t+1)$;

Step 5. 将 $A(t+1)$ 和 $B(t+1)$ 种群合并,得到种群 $Temp(t+1)$,并计算 $Temp(t+1)$ 的适应度值;

Step 6(竞争操作). 精英库 $Best(t)$ 与 $Temp(t+1)$ 中的个体竞争,若 $Temp(t+1)$ 中不存在此精英个体,则用此精英个体替代 $Temp(t+1)$ 中的最差个体;否则不进行任何操作.得到下一代种群 $P(t+1)$,并将种群按适应度降序排列;

Step 7(精英库更新操作). 若 $P(t+1)$ 中的最优个体 $bestIndividual$ 大于 $Best(t)$ 中的最优个体,则用最优个体 $bestIndividual$ 替代 $Best(t)$ 中的最差个体.得到更新后的精英库 $Best(t+1)$;

Step 8. 是否满足终止条件:若是,则结束;否则,转到 Step 3.

2 算法收敛性分析

定理 1. 双精英协同进化遗传算法的运行过程是有限齐次 Markov 链.

证明:由于 DECGA 采用二进制编码,故假设在以 $\{0,1\}^L$ 为个体空间, 2^L 为种群空间的遗传算法中, L 为个体长度, n 为种群规模,则种群空间中有 2^{nL} 个个体, 2^n 为种群序列的状态空间,并且它是有限的.从 DECGA 算法的进化过程可以看出, $t+1$ 时刻的种群 $P(t+1)$ 只与 t 时刻的种群 $P(t)$ 有关.所以, DECGA 的运行过程是有限齐次 Markov 链. \square

定义 5. 设 S 为种群规模为 n 的种群组合的集合,则 $|S| = \begin{bmatrix} 2^L + n - 1 \\ n \end{bmatrix}$.

定义 6. 设 $X_t = \{x_1, x_2, \dots, x_n\}$, t 表示进化代数, X_t 表示在第 t 代的一个种群序列,令

$$s^* = \{x \mid \max_{x \in X_t} f(x) = f^*\} \quad (6)$$

称 s^* 为最优解集,其中 f^* 为全局最优值.

定义 7. 设 $f_t = \max_{x_i \in X_t} \{f(x_i) : i=1,2,\dots,n\}$ 是一个随机变量序列,序列中的变量代表在时间步 t 状态中的最佳适应度值.若对于任意给定的 $\varepsilon > 0$,有

$$\lim_{t \rightarrow \infty} P\{|f_t - f^*| \leq \varepsilon\} = 1 \quad (7)$$

则称算法全局收敛.定义 3 表明,算法收敛是指,当算法迭代到足够多的代数后,种群中包含全局最优解的概率几乎为 1.故要证明 DECGA 算法的全局收敛性,只需证明公式(7)成立即可.

定理 2. DECGA 算法的种群最优值序列是单调不递减的,即对于任意 $t \geq 0, f(X_{t+1}) \geq f(X_t)$.

证明:在 DECGA 算法中,以精英个体作为进化操作的核心,在每代种群进化结束后,精英库中的精英种群会与进化后的临时种群进行竞争操作,即临时种群中若没有此精英个体,则精英个体替代种群中的最差个体;否则不进行任何操作,以上操作能够保证每一代的最佳个体都被遗传到下一代.因而对于任意 $t \geq 0, f(X_{t+1}) \geq f(X_t)$. \square

定理 3. DECGA 算法具有全局收敛性.

证明:设 $s_i \in S$ 表示 s_i 是 S 中的第 i 个状态,设 $p_i(t)$ 为种群 X_t 处在状态 s_i 的概率, $p_t = \sum_{i \in I} p_i(t)$, 则由 Markov 的性质可知:

$$p_{t+1} = \sum_{s_i \in S} \sum_{j \notin I} p_i(t) p_{ij}(t) = \sum_{i \in I} \sum_{j \notin I} p_i(t) p_{ij}(t) + \sum_{i \notin I} \sum_{j \notin I} p_i(t) p_{ij}(t) \quad (8)$$

由于

$$\sum_{i \notin I} \sum_{j \notin I} p_i(t) p_{ij}(t) + \sum_{i \notin I} \sum_{j \in I} p_i(t) p_{ij}(t) = \sum_{i \notin I} p_i(t) = p_t \quad (9)$$

将公式(9)变换得到:

$$\sum_{i \notin I} \sum_{j \notin I} p_i(t) p_{ij}(t) = p_t - \sum_{i \notin I} \sum_{j \in I} p_i(t) p_{ij}(t) \quad (10)$$

将公式(10)带入公式(8),得到:

$$p_{t+1} = \sum_{i \in I} \sum_{j \notin I} p_i(t) p_{ij}(t) + p_t - \sum_{i \notin I} \sum_{j \in I} p_i(t) p_{ij}(t) = p_t.$$

所以,

$$\lim_{t \rightarrow \infty} p_t = 0 \quad (11)$$

因为,

$$\lim_{t \rightarrow \infty} P\{|f_t - f^*| \leq \varepsilon\} = 1 - \lim_{t \rightarrow \infty} \sum_{i \notin I} p_i(t) = 1 - \lim_{t \rightarrow \infty} p_t \quad (12)$$

将公式(11)代入公式(12)中,得到:

$$\lim_{t \rightarrow \infty} P\{|f_t - f^*| \leq \varepsilon\} = 1$$

因而,DECGA 算法具有全局收敛性. \square

3 仿真实验

为了验证算法的可行性和有效性,将该算法应用于典型数值函数优化问题中,其相关参数设置见表 1.

Table 1 Experimental setup of parameters

表 1 实验参数的设置

精英库个体数目 M		5
差异度阈值 u		0.1
r_0		0.1
参数调整步长 d		0.3
EliteA 子种群	p_c	1.0
	p_m	0.02
EliteB 子种群	p'_c	0.85
	p'_m	0.08

考察如下 14 个仿真函数,其中 $f_{11} \sim f_{14}$ 是较为复杂的复合函数,受文章篇幅所限,故具体情况请参照文献[13]中的函数 CF1(f_{11}),CF2(f_{12}),CF4(f_{13}),CF6(f_{14}).

$$f_1 = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, x_i \in [-10, 10],$$

$$f_2 = \cos(5x_1) \cos(5x_2) e^{-0.0001(x_1^2 + x_2^2)}, x_i \in [-2.048, 2.048],$$

$$f_3 = \left[4 - 2.1x_1^2 + \frac{1}{3}x_1^4 \right] x_1^2 + x_1x_2 + (4 - 4x_2^2)x_2^2, x_i \in [-3, 3],$$

$$f_4 = \prod_{i=1}^2 \sum_{j=1}^5 \{j \cos[(j+1)x_i + j]\}, x_i \in [-10, 10],$$

$$f_5 = 500 - \left[0.002 + \sum_{j=0}^{24} (1 + j + (x_1 + a(j))^6 + (x_2 - b(j))^6)^{-1} \right]^{-1},$$

$$a(j) = 16((j \bmod 5) - 2), b(j) = 16(\lfloor j/5 \rfloor - 2), x_i \in [-65.536, 65.536],$$

$$f_6 = -\sum_{i=1}^{10} x_i \sin(\sqrt{|x_i|}), x_i \in [-512, 512]$$

$$f_7 = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, x_1, x_2 \in [-5, 5]$$

$$f_8 = |(1 - x)x^2 \sin(200x)|, x \in [0, 1]$$

$$f_9 = (-1) \times (x^2 + 2y^2 - 0.3 \cos(3x) - 0.4 \cos(3y)) + 4, x, y \in [-1, 1],$$

$$f_{10} = \frac{x_1^2 + x_2^2}{2} - \cos(2x_1) \cos(2x_2), x \in [-10, 10].$$

3.1 DECGA算法的实验结果

表 2 给出了对以上 14 个函数独立运行 100 次的实验结果,包括这 100 次实验中的最好值、最差值、平均最优值和平均收敛代数.

Table 2 Experimental results of DECGA on 14 testing functions over 100 trials
表 2 DECGA 算法对 14 个测试函数 100 次实验的实验结果

函数	全局最优值	平均最优值	最好值	最差值	平均收敛代数
f_1	1.0	0.999 99	1.0	0.999 98	28
f_2	1.0	0.999 99	1.0	0.999 84	35
f_3	-1.031 68	-1.031 625	-1.031 68	-1.031 589	27
f_4	-186.730 9	-186.730 7	-186.7309	-186.730 2	13
f_5	499.002	499.001 9	499.002	499.001 9	15
f_6	-4189.828 9	-4 189.711 1	-4 189.816 7	-4 189.697 2	205
f_7	0	6.148E-8	1.455E-9	1.166E-6	525
f_8	0.1481 48	0.1481 47	0.1481 48	0.1481 46	20
f_9	4.7	4.699 99	4.7	4.699 98	29
f_{10}	-1.0	-0.999 999	-1.0	-0.999 989	39
f_{11}	0	1.5924E-4	2.397E-6	4.7821E+1	682
f_{12}	0	1.204E+1	7.112E-1	8.820E+3	830
f_{13}	0	3.3016E+0	7.367E-2	2.3087E+3	1 739
f_{14}	0	5.3091E+2	3.2982E+0	1.322E+4	1 688

3.2 种群多样性变化图

由于优化函数较多,这里选择 4 个函数(f_1, f_8, f_9, f_{10})来比较算法改进前后种群多样性的变化,如图 4 所示.从图中不难看出,SGA 中种群的多样性在进化较少代数后就降为 0,从而使算法陷入早熟收敛.而 DECGA 能够有效地维持种群的多样性,从而有效地避免了算法早熟收敛,很大程度上提高了算法的搜索性能.

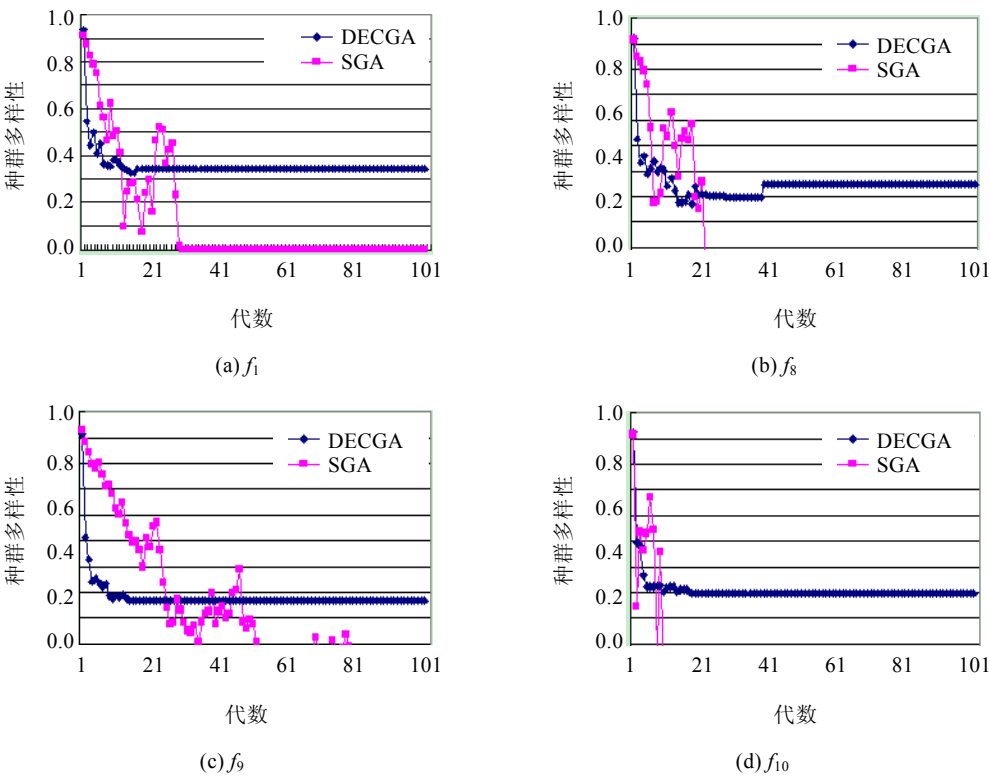


Fig.4 Changing maps of population diversity
图 4 种群多样性变化图

3.3 DECGA算法与DSGA算法、LGA算法的比较

为了验证算法的可行性和有效性,本文从效率和质量两方面来考虑算法的性能.算法的效率用平均收敛代数、收敛到最优解的次数来衡量;质量用满足停止准则时所需要的函数评估次数的均值、最小次数、最大次数和消耗的平均时间这4项指标衡量.实验参数、算法停止条件的设置均与文献[14]保持一致.

首先,选取函数 f_1, f_2, f_3 这3个函数来测试算法的效率.这3个函数都是多峰函数,在全局最优值附近有很多局部最小值点,因此算法很容易陷入局部最优状态.实验结果见表3.

Table 3 Average generations and times of reaching global optimum for three algorithms within 200 generations

表3 在200代以内3种算法收敛的平均代数和搜索到全局最优点的次数

函数	算法	平均收敛代数	搜索次数
f_1	DSGA	53	90
	LGA	32	98
	DECGA	28	99
f_2	DSGA	96	94
	LGA	58	100
	DECGA	35	99
f_3	DSGA	76	91
	LGA	43	99
	DECGA	27	100

接下来,选取函数 f_4, f_5, f_6 这3个函数来测试算法的质量.这3个函数是遗传算法常用的代表性测试函数,且都是非常复杂的多峰函数.实验结果见表4.

Table 4 Mean, min, max and time of satisfying termination rule for two algorithms

表4 满足停止准则时两种算法所需函数评估次数均值、最小次数、最大次数和所消耗的时间

函数	算法	均值	最小次数	最大次数	消耗时间(s)
f_4	DSGA	6 977	639	9 763	0.4
	LGA	4 382	498	7 396	0.28
	DECGA	1 920	430	6 640	0.25
f_5	DSGA	2 277	640	4 195	0.133
	LGA	1 879	453	3 809	0.102
	DECGA	1 258	430	3 650	0.098
f_6	DSGA	31 693	26 340	34 824	30.019
	LGA	15 694	13 240	32 509	18.328
	DECGA	15 260	12 620	32 850	17.843

表3、表4的实验结果表明,从算法的搜索效率来看,在收敛的平均代数上,DECGA比LGA缩短了近一半;在搜索到全局最优点的次数上,虽然对于 f_2 ,DECGA搜索到全局最优点的次数比LGA减少了1次,但对于函数 f_1, f_3 ,DECGA的结果优于LGA.总的来看,DECGA在全局收敛性和收敛代数上均比DSGA和LGA有了一定的改进.从算法质量来看,与DSGA,LGA相比,DECGA的收敛速度也有了明显的提高.

3.4 DECGA算法与SGA算法、EGA算法、BEGA算法的比较

选取文献[1]中的测试函数,实验参数的设置、适应度函数的相关设计、停止准则也均与文献[1]相同,因此,DECGA算法与SGA,EGA,BEGA算法运行一代时,函数的评价次数相同.统计搜索到的全局最优值及其平均次数,实验结果见表5.

Table 5 Mean number of evolutionary generation (MNEG) and success rate on five functions

表5 各函数的平均进化代数及其成功率

算法	f_7		f_8		f_9		f_2		f_{10}	
	平均进化代数	成功率(%)	平均进化代数	成功率(%)	平均进化代数	成功率(%)	平均进化代数	成功率(%)	平均进化代数	成功率(%)
SGA	—	1.6	6.4	90.8	30.7	99.9	24.5	94.2	204.0	37.5
EGA	—	2.1	6.1	94.2	19.5	100	20.6	99.6	173.3	50.1
BEGA	559.8	91.4	2.3	100	6.9	100	4.2	100	125.4	100
DECGA	442	94	1.6	100	2.9	100	3.3	100	20.2	100

表 5 中,平均进化代数和成功率分别表示 1 000 次独立实验时算法成功实验的平均进化代数和找到满意解的概率.从表 5 的实验结果可以看出,对于函数 f_7 ,SGA,EGA 算法在有限代内很难满足停止准则,BEGA 算法的平均进化代数为 559.8,而本文 DECGA 算法的平均进化代数只有 442 代;DECGA 算法对于函数 f_8, f_9, f_2, f_{10} 而言,其平均进化代数均优于前 3 种算法.对于函数 f_7 ,SGA,EGA 算法达到收敛准则的成功次数非常低,效果不够理想,BEGA 算法的成功率达到 91.4%,而 DECGA 算法的成功率已经达到 94%;对函数 f_8, f_9, f_2, f_{10} 而言,成功次数的统计结果均为 100%,已达到最佳效果,没有再进一步优化可能性.因此,DECGA 算法对各函数的平均进化代数均少于 SGA,EGA 和 BEGA.综上所述,同等条件下,DECGA 的优化结果明显优于 SGA,EGA 和 BEGA.实验结果见表 6.

Table 6 Mean optimal solution when the running time is 0.4s and the running times is 1 000

表 6 运行时间为 0.4s 且算法运行次数为 1 000 时,各函数的平均最大适应度

算法	f_7	f_8	f_9	f_2	f_{10}
SGA	0.737 557	0.146 702	4.698 624	0.992 972	0.847 256
EGA	0.769 190	0.147 868	4.699 996	0.999 852	0.885 756
BEGA	0.997 381	0.148 137	4.700 000	1.000 000	0.998 723
DECGA	0.998 144*	0.148 147*	4.700 000	1.000 000	0.999 999*

从表 6 的实验数据可以看出,在规定的、较短的时间内,DECGA 优化速度较快.DECGA 对函数 f_2, f_9 的优化结果与 BEGA 相同的原因是,它们都已经搜索到全局最优值,故已经没有再改进的可能性.而对于 f_7, f_8, f_{10} 这 3 个较为复杂的函数,DECGA 的优化结果优于前 3 种算法.

4 结 论

本文针对传统遗传算法因种群多样性的急剧下降而造成的早熟收敛问题,提出双精英协同进化遗传算法,该算法通过引入精英策略和协同进化思想,有效地保持了种群的多样性,避免了算法早熟收敛现象的发生.通过双精英协同进化机制,有效地平衡了传统遗传算法全局收敛和收敛速度慢之间的矛盾.理论分析证明,算法以概率 1 收敛于全局最优解.通过与已有算法的比较表明,DECGA 具有较强的寻优能力,其性能优于其他算法.

遗传算法自身的随机性使得算法需要通过多次实验来增加其可靠性和可信度.如何进一步结合其他策略完善算法的自我进化机制、增强算法的可靠性和自学习能力,是进一步需要研究的问题.

致谢 在此,我们向对本文的工作给予支持和建议的同行表示感谢.

References:

- [1] Meng W, Han XD, Hong BR. Bee evolutionary genetic algorithm. *Acta Electronica Sinica*, 2006,34(7):1294–1300 (in Chinese with English abstract).
- [2] Ehrlich PR, Raven PH. Butterflies and plants: A study in coevolution. *Evolution*, 1964,18(4):586–608. [doi: 10.2307/2406212]
- [3] Potter MA, De Jong KA. A cooperative coevolutionary approach to function optimization. In: Davidor Y, Schwefel HP, Männer R, eds. *Proc. of the Parallel Problem Solving from Nature—PPSN III, Int'l Conf. on Evolutionary Computation*. LNCS 866, Berlin: Springer-Verlag, 1994. 249–257. [doi: 10.1007/3-540-58484-6_269]
- [4] Yue Q, Feng S. The statistical analyses for computational performance of the genetic algorithms. *Chinese Journal of Computers*, 2009,32(12):2389–2392 (in Chinese with English abstract).
- [5] Chen H, Cui DW, Cui YA, Tao YQ, Liang K. Ethnic group evolution algorithm. *Journal of Software*, 2010,21(5):978–990 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3484.htm> [doi: 10.3724/SP.J.1001.2010.03484]
- [6] Zhou CL, You W, Ding XJ. Genetic algorithm and its implementation of automatic generation of Chinese songci. *Journal of Software*, 2010,21(3):427–437 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3596.htm> [doi: 10.3724/SP.J.1001.2010.03596]
- [7] Jiang ZY, Cai ZX, Wang Y. Hybrid self-adaptive orthogonal genetic algorithm for solving global optimization problems. *Journal of Software*, 2010,21(6):1296–1307 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3592.htm> [doi: 10.3724/SP.J.1001.2010.03592]

- [8] Ying WQ, Li YX, Shen Phillip CY, Wu H, Yu FH. Geometric thermodynamical selection for evolutionary multi-objective optimization. Chinese Journal of Computers, 2010,33(4):754-767 (in Chinese with English abstract).
- [9] Tan KC, Yang YJ, Goh CK. A distributed cooperative coevolutionary algorithm for multiobjective optimization. IEEE Trans. on Evolutionary Computation, 2006,10(5):527-549. [doi: 10.1109/TEVC.2005.860762]
- [10] Danoy G, Bouvry P, Martins T. Hlcga: A hybrid competitive coevolutionary genetic algorithm. In: Hoes L, ed. Proc. of the 6th Int'l Conf. on Hybrid Intelligent Systems. Computer Society Press, 2006. 48-51. [doi: 10.1109/HIS.2006.264931]
- [11] Mu CH, Jiao LC, Liu Y. M-Elite coevolutionary algorithm for numerical optimization. Journal of Software, 2009,20(11): 2925-2938 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3496.htm> [doi: 10.3724/SP.J.1001.2009.03496]
- [12] Zhou Q, Luo WJ. A novel multi-population genetic algorithm for multiple-choice multidimensional knapsack problems. In: Cai Z, ed. Proc. of the 5th Int'l Symp. on Advances in Computation and Intelligence. Berlin: Springer-Verlag, 2010. 148-157. [doi: 10.1007/978-3-642-16493-4_16]
- [13] Liang JJ, Suganthan PN, Deb K. Novel composition test functions for numerical global optimization. In: Proc. of the IEEE Swarm Intelligence Symp. Piscataway: IEEE Press, 2005. 68-75. [doi: 10.1109/SIS.2005.1501604]
- [14] Li QH, Yang SD, Ruan YL. Improving optimization for genetic algorithms based on level set. Journal of Computer Research and Development, 2006,43(9):1624-1629 (in Chinese with English abstract).

附中文参考文献:

- [1] 孟伟,韩学东,洪炳镕.蜜蜂进化型遗传算法.电子学报,2006,34(7):1294-1300.
- [4] 岳焱,冯珊.遗传算法的计算性能的统计分析.计算机学报,2009,32(12):2389-2392.
- [5] 陈皓,崔杜武,崔颖安,陶永芹,梁琨.族群进化算法.软件学报,2010,21(5):978-990. <http://www.jos.org.cn/1000-9825/3484.htm> [doi: 10.3724/SP.J.1001.2010.03484]
- [6] 周昌乐,游维,丁晓君.一种宋词自动生成的遗传算法及其机器实现.软件学报,2010,21(3):427-437. <http://www.jos.org.cn/1000-9825/3596.htm> [doi: 10.3724/SP.J.1001.2010.03596]
- [7] 江中央,蔡自兴,王勇.求解全局优化问题的混合自适应正交遗传算法.软件学报,2010,21(6):1296-1307. <http://www.jos.org.cn/1000-9825/3592.htm> [doi: 10.3724/SP.J.1001.2010.03592]
- [8] 应勤伟,李元香,Shen Phillip CY,吴昱,余法红.演化多目标优化中的几何热力学选择.计算机学报,2010,33(4):754-767.
- [11] 慕彩虹,焦李成,刘逸.M-精英协同进化数值优化算法.软件学报,2009,20(11):2925-2938. <http://www.jos.org.cn/1000-9825/3496.htm> [doi: 10.3724/SP.J.1001.2009.03496]
- [14] 李庆华,杨世达,阮幼林.基于水平集的遗传算法优化的改进.计算机研究与发展,2006,43(9):1624-1629.



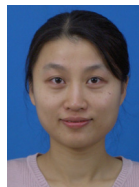
刘全(1969—),男,河北定兴人,博士,教授,博士生导师,CCF 会员,主要研究领域为强化学习,智能信息处理,自动推理.



张永刚(1975—),男,博士,CCF 会员,主要研究领域为约束求解,智能优化算法.



王晓燕(1985—),女,硕士生,主要研究领域为遗传算法,强化学习.



章晓芳(1980—),女,博士,CCF 会员,主要研究领域为程序分析与软件测试,智能优化算法.



傅启明(1985—),男,硕士生,CCF 学生会会员,主要研究领域为强化学习,图像检索.