

一种高效混合遗传算法

刘 杰,王 媛

(河海大学土木工程学院,江苏 南京 210098)

摘要 针对基本遗传算法在优化应用中遇到的诸如局部搜索能力差、计算量大、对较大搜索空间适应能力差和早熟收敛等问题,将传统的单纯形搜索算法与遗传算法相结合,提出了一种混合遗传算法,并在算法中增加了小生境淘汰运算和加速循环操作.典型测试函数数值算例验证表明,本文方法不但可以有效地克服基本遗传算法的上述缺陷,而且计算速度和计算精度都有显著提高.

关键词 混合遗传算法;单纯形算法;小生境运算;加速循环

中图分类号 O221 **文献标识码** A **文章编号** 1000-198X(2002)02-0049-05

遗传算法是基于生物进化仿生学算法的一种,它建立于达尔文生物进化的“物竞天择,适者生存”的基本原理之上,是一种自适应概率性全局优化搜索算法,可处理设计变量离散、目标函数多峰值且导数不存在、可行域狭小且为凹形等优化问题.遗传算法作为一种智能化的全局搜索算法,自 20 世纪 80 年代问世以来便在数值优化、系数控制、结构优化设计、参数辨识等诸多领域的应用中展现出其特有的魅力,同时也暴露出许多不足和缺陷.大量研究表明,遗传算法的实际应用中存在以下几方面的缺陷:(a)全局搜索能力极强而局部寻优能力较差.具体表现为遗传算法可以用极快的速度搜索到最优解附近,但要进一步达到最优解则速度极慢.(b)对搜索空间变化的适应能力差.当搜索空间增大时,基本遗传算法需急剧扩大种群规模以实现全局搜索,从而极大地增加了计算量,不仅影响了算法效率,更阻碍了遗传算法在工程实践中的应用.(c)易出现早熟收敛现象.当种群规模较小时,如果在进化初期出现适应度较高的个体,群体的多样性往往会受到破坏,从而出现早熟收敛现象.以上缺陷是遗传算法在解决优化问题时普遍存在的.本文针对基本遗传算法的上述缺陷,对算法加以改进,将传统的单纯形搜索算法与遗传算法相结合,提出一种混合遗传算法,并在算法中增加了小生境淘汰运算和加速循环操作,力求克服以上缺陷,增加遗传算法在求解优化问题特别是工程应用中的有效性.

1 基本遗传算法(GA)及算子的选择

1.1 编码

遗传算法中用以表示参数向量结构的常用编码方式有 3 种:二进制编码、格雷编码、浮点编码.二进制码编码、解码操作简单易行,不仅便于实现交叉、变异等进化操作,而且便于利用模式定理对算法进行理论分析.格雷码是二进制码的变形,具有与二进制码相似的特点,其编码、解码操作相对复杂,但基于格雷编码的算法局部搜索能力有较大提高.浮点编码方式编码长度等于参数向量的维数,达到同等精度要求的情况下,编码长度远小于二进制码和格雷码.此外,浮点编码使用的决策变量的真实值,不需反复进行数据转换.因此,本文采用浮点编码方式.

1.2 适应度函数

适应度函数是遗传进化操作的基础,它的构造是遗传算法的关键.合理的适应度函数能引导搜索朝最优化方向进行.本文构造基于序的适应度函数.它的特点是个体被选择的概率与目标函数的具体值无关.将种群中的所有个体按其目标函数值的大小降序排列,设参数 $\alpha \in (0, 1)$,定义基于序的适应度函数

$$e_{\text{val}}(X_i) = \alpha(1 - \alpha)^{i-1} \quad i = 1, 2, \dots, p_{\text{size}} \quad (1)$$

式中: X_i ——种群排序后第 i 个个体; p_{size} ——种群中个体总数.

1.3 选择算子

本文采用比例选择算子. 该算子是一种回放式随机采样的方法, 以旋转赌轮 p_{size} 次为基础, 每次旋转都可选择一个体进入子代种群. 父代个体 X_i 被选择的概率 p_{s_i} 为

$$p_{s_i} = e_{\text{val}}(X_i) / \sum_{i=1}^{p_{\text{size}}} e_{\text{val}}(X_i) \quad (2)$$

比例选择算子的具体操作过程见文献 [1].

1.4 交叉算子

交叉算子是遗传算法区别于其他进化算法的重要特征, 在算法中起着关键作用, 是产生新个体的主要方法. 交叉算子的设计和实现与所研究问题密切相关, 一般要求既不能过多地破坏种群中的优良个体模式, 又要能够有效地产生一些较好的新个体模式. 本文采用算术交叉算子. 该算子由两个个体的线性组合产生两个新的个体. 算术交叉算子专门针对采用浮点编码的种群进行操作. 首先依据交叉概率通过轮盘赌随机确定进行交叉操作的父代, 并两两配对. 对其中任一对 (X_i, X_j) 按下式产生两个新的个体:

$$X_i^{t+1} = \beta X_j^t + (1 - \beta) X_i^t \quad (3)$$

$$X_j^{t+1} = \beta X_i^t + (1 - \beta) X_j^t \quad (4)$$

式中: β ——进化代数决定的变量, $\beta \in (0, 1)$; t ——进化代数.

1.5 变异算子

变异操作的主要目的是改善算法的局部搜索能力, 并维持群体的多样性, 防止出现早熟现象. 本文采用非均匀变异算子. 类似于交叉算子, 设定变异概率 P_m 产生变异操作父代, 并依次指定个体编码中每个基因座为变异点, 对每个变异点进行以下操作. 设某一父代个体为 $X_i^t = (x_1^t, x_2^t, \dots, x_k^t, \dots, x_L^t)$, 由 X_i^t 向 $X_i^{t+1} = (x_1^t, x_2^t, \dots, x_k^{t+1}, \dots, x_L^t)$ 进行变异操作时, 若变异点参数 x_k^t 的变化范围是 $[U_{\min}^k, U_{\max}^k]$, 则新的基因值 x_k^{t+1} 由下式确定:

$$x_k^{t+1} = \begin{cases} x_k^t + \Delta(t, U_{\max}^k - v_k) & \text{if } \text{random}(0, 1) = 0 \\ x_k^t + \Delta(t, v_k - U_{\min}^k) & \text{if } \text{random}(0, 1) = 1 \end{cases} \quad (5)$$

式中: $\text{random}(0, 1)$ ——以均等的概率从 $[0, 1]$ 中取一个值; v_k ——微小扰动量; $\Delta(t, y)$ —— $[0, y]$ 范围内符合非均匀分布的随机数, 可按下式定义:

$$\Delta(t, y) = y(1 - r^{(1-t/T)^b}) \quad (6)$$

式中: r —— $(0, 1)$ 内的随机数; b ——系统参数, 本文取 2.0; T ——允许最大进化代数.

2 混合遗传算法(HGA)及改进策略

2.1 单纯形局部探索操作

遗传算法使用有导向性的概率搜索技术, 从由很多可行解组成的一个初始种群开始进行最优解的搜索过程, 其间同时使用多个可行解的搜索信息进行交叉、变异等随机操作产生新个体, 从而使搜索范围可达到整个解空间. 遗传算法这种具有隐含并行性的多点搜索, 具有极强的全局搜索能力, 而这种大范围的随机搜索也使得算法的局部搜索能力较差. 正如前文所述, 局部搜索能力差在很大程度上限制了遗传算法的应用. 目前, 常采用扩大种群规模等参数调节手段以及一些优秀的遗传算子加强算法的局部搜索能力, 但这些措施往往会带来诸如计算量增大、收敛减慢等负面效应. 特别地, 由于遗传算法具有较强的鲁棒性, 其搜索的最终结果受参数影响的程度并不显著. 对于解空间很大的优化问题, 即使无限制地增加种群规模仍无法搜索到问题的全局最优解, 搜索往往在最优解附近反复进行. 因此, 上述常用的措施并不能彻底改变基本遗传算法局部搜索能力差的特点. 本文考虑到传统的单纯形算法局部搜索能力极强的特点, 将基本遗传算法与传统的单纯形搜索算法结合形成了一种混合遗传算法, 从而在继承基本遗传算法极强的全局搜索能力的基础上, 克服了其局部搜索能力差的缺陷. 大量的研究和实践证明, 在给定合适的初值条件下, 单纯形算法能以很高的精度收敛于初值附近的最优解. 因此, 本文在遗传算法的进化操作之后, 分别以子代群体的前 NS 个优秀个体

为初始点形成初始单纯形,进行单纯形局部搜索,从而搜索到各优秀个体在目前环境下所对应的局部最优解,并以此最优解替换子群体中相应的个体,形成新的子代种群以改善种群总体性能,从而可以达到提高算法局部搜索能力的目的。

2.2 小生境淘汰操作

前文指出遗传算法在应用过程中普遍存在早熟收敛现象,虽然通过扩大种群规模,增大交叉、变异概率等措施,可在一定程度上改善群体的多样性,但是扩大种群规模将带来计算量的急剧增加,过大的交叉、变异概率将使遗传算法的性能近似于随机搜索。而且,笔者大量试算表明,存在大量的优化问题,此类参数调整措施无法抑制早熟收敛现象。本文在遗传算法变异操作之后加入小生境淘汰操作,在不破坏指定距离内优良个体的条件下,在只牺牲少量计算量的情况下可最大限度地保持群体的多样性,从而能有效抑制早熟收敛现象的出现。小生境运算的基本思想是:首先两两比较种群中各个体的海明距离,若这个距离在预先指定的距离内,再比较两个个体之间的适应度大小,并对基本适应度较小的个体施加一个较强的罚函数,极大地降低其适应度。这样,受到惩罚的个体在后面的进化过程中被淘汰掉的概率极大,因此,在指定距离内将只存在一个优良的个体,从而可有效地维护群体的多样性。显然,小生境运算使算法运行过程中群体的多样性得以保持,从而可以消除早熟收敛现象。

2.3 加速循环操作

遗传算法在运行过程中,每一代进化都需要重新计算种群的适应度函数,因此如果进化代数较大,将带来很大的计算量。本文引入加速循环操作,利用前两次进化操作产生的 NA 个优秀个体的最大可能变化区间重新生成遗传算法的初始种群,重新进行进化操作。如此循环迭代,使产生初始种群的区间逐步收缩,从而可加快进化迭代进程,减少迭代次数,降低算法的计算量。显然循环操作加强了对进化迭代产生的优秀个体的利用,有利于提高算法的整体运行效率。

3 混合遗传算法的实现

3.1 算法运行步骤

a. 编码:对待优化参数按上述方法进行编码。

b. 种群初始化:设置进化代数计数器 $IT = 1$,依据初始区间随机产生 p_{size} 个符合约束条件的初始染色体,形成初始种群。

c. 目标函数计算:依据上一代种群解集计算相应目标函数,并对染色体由好到坏进行降序排列,记忆前 NP 个个体。

d. 适应度评价:依据上述排序结果计算个体适应度 $F(i = 1, 2, \dots, p_{size})$ 。

e. 遗传进化操作:依据前述方法对父代种群进行选择、交叉、变异操作产生子代种群。

f. 小生境淘汰操作:将子代种群的 p_{size} 个个体和 c 步记忆的 NP 个个体合并在一起,得到一个含有 $M = p_{size} + NP$ 个个体的新群体,对这 M 个个体按照式(7)计算每两个个体 X_i, X_j 的海明距离

$$\|X_i - X_j\| = \sqrt{\sum_{k=1}^{p_{size}} (x_{ik} - x_{jk})^2} \quad i = 1, 2, \dots, M-1; j = i+1, \dots, M \quad (7)$$

当 $\|X_i - X_j\| < L$ 时,比较个体的适应度大小,并对其中适应度较低的个体处以罚函数进一步降低其适应度

$$F_{\min}(X_i, X_j) = P \quad (8)$$

依据个体新的适应度重新排序,取前 p_{size} 个个体组成新的子代种群,记忆前 NP 个个体。

g. 单纯形局部搜索:分别以小生境淘汰运算得到的子代群体中前 NS 个优秀个体作为初始点形成初始单纯形,进行单纯形局部搜索,并以搜索到的最优解替换子代群体中相应的个体,形成新的子代种群。

h. 进化迭代:返回步 c,迭代 2 次。

i. 加速循环:上述步骤构成进化迭代过程,用 2 次迭代产生的群体的前 NS 个优秀个体最大可能的变化区间作为产生初始种群的初始区间,返回步 b 加速 1 次。

j. 收敛准则:以种群最优目标函数值小于 EPS 为判据并以允许最大循环进化代数 $IT_{\max} = 15$ 为辅助判据。若不收敛,更新计数器 $IT = IT + 1$,返回步 b,否则,转步 k。

k. 输出结果 输出终止进化迭代代数 IT 、最优个体各基因值及优化函数值.

3.2 算法运行参数

种群中个体总数 p_{size} 、交叉概率 P_c 、变异概率 P_m 等这些参数对算法的运行性能也有影响,需结合具体问题认真选择.本文依据一般建议及计算经验,取交叉概率 P_c 为 0.7,取变异概率 P_m 为 0.1.加速循环的最优个体数目 NA 取为 p_{size} 的 10%,单纯形局部搜索的个体数目 NS 取为 p_{size} 的 10%,小生境优秀个体数目 NP 取为 p_{size} 的 15%,小生境个体适应度罚函数 P 取为 10^{-30} .

4 算例分析

基于以上算法,笔者编制了相应的程序,并利用评价遗传算法性能的 2 个典型测试函数:De Jong 函数 $F2$ 和 Schaffer 函数 $F6$,进行混合遗传算法(HGA)和基本遗传算法(GA)性能的对比研究.

4.1 De Jong 函数 $F2$

De Jong 函数 $F2$ 是一个二维函数,具有一个全局极小点 $f_2(1.0,1.0)=0.0$.该函数虽然是单峰值函数,但它却是病态的,在函数曲面上沿着曲线 $x_2 = x_1^2$ 有一条较为狭窄的山谷,传统的梯度优化方法搜索到山谷边缘时,往往会发生振荡,难以进行全局优化. De Jong 函数 $F2$ 的具体形式为

$$f_2(x_1,x_2) = 10(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad -2.048 \leq x_i \leq 2.048 \quad (i = 1,2) \tag{9}$$

表 1 所示为 p_{size} 分别取 100,500 时,De Jong 函数 $F2$ 的基本遗传算法(GA)收敛过程及 p_{size} 分别取 50,150 时,De Jong 函数 $F2$ 的混合遗传算法(HGA)收敛过程.由表 1 可看出,当 p_{size} 取 100 时,GA 在第三代就停留在某一局部最优值上不再进化,有明显的早熟收敛现象,而采用 HGA 搜索,当 p_{size} 取 50 时,已消除了早熟收敛现象,且收敛精度明显提高.当基本遗传算法(GA)的种群规模增大到 500 时,收敛结果以较高的精度逼近解析解,而采用混合遗传算法(HGA)搜索,当种群规模仅增大到 150 时,收敛结果就以很高的精度逼近解析解.由以上分析可见,混合遗传算法(HGA)即使在种群规模较小的(远小于 GA 的规模)情况下,也能有效地抑制早熟收敛现象,而且收敛精度比 GA 有显著提高,或者说 HGA 在耗费较小计算量(远少于 GA 的计算量)的情况下,可以得到精度明显高于 GA 的寻优结果.表 1 的结果也显示,当种群规模增加时,GA 的早熟收敛现象在一定程度上受到了抑制,GA 和 HGA 的计算精度都有所提高,而 HGA 的提高速度更快.

表 1 De Jong 函数 $F2$ 的 GA 和 HGA 收敛过程

Table 1 Convergence processes of GA and HGA for De Jong $F2$ function

进化 代数	GA $p_{size} = 100$			GA $p_{size} = 500$			HGA $p_{size} = 50$			HGA $p_{size} = 150$		
	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2
1	0.108 042	1.016 9	1.066 9	0.066 392	0.813 6	0.679 7	0.161 288	0.619 5	0.370 9	0.017 192	1.114 4	1.248 2
2	0.007 913	0.963 5	0.936 5	0.002 059	1.030 7	1.058 9	0.064 203	0.756 1	0.578 6	0.002 277	0.974 2	0.953 1
3	0.002 716	0.950 6	0.905 4	0.000 200	0.998 1	0.995 7	0.015 670	1.044 1	1.101 8	0.000 089	0.990 6	0.981 2
4	0.002 716	0.950 6	0.905 4	0.000 014	0.998 7	0.997 0	0.003 123	1.026 7	1.059 0	0.000 026	1.000 3	1.001 1
5	0.002 716	0.950 6	0.905 4	0.000 000	1.000 3	1.000 6	0.003 123	1.026 7	1.059 0	0.000 001	1.001 0	1.001 8
6	0.002 716	0.950 6	0.905 4	0.000 000	1.000 3	1.000 6	0.001 618	0.970 4	0.939 0	0.000 000	0.999 9	0.999 8
7	0.002 716	0.950 6	0.905 4	0.000 000	1.000 1	1.000 1	0.001 358	0.965 5	0.930 9	0.000 000	1.000 0	1.000 0
8	0.002 716	0.950 6	0.905 4	0.000 000	1.000 0	1.000 1	0.001 358	0.965 5	0.930 9	0.000 000	1.000 00	0.999 9

4.2 Schaffer 函数 $F6$

Schaffer 函数 $F6$ 的具体形式为

$$f_6(x_1,x_2) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2} \quad -100 \leq x_i \leq 100 \quad i = 1,2 \tag{10}$$

虽然该函数在其定义域内只有一个全局极小点 $f_6(0.0,0.0)=0.0$,但由于变量的取值范围大,采用传统的直接搜索法求解时,因搜索空间太大而无法求得全局最优解,采用 GA 搜索时,由于算法局部搜索能力差,因而需要设置相当大的种群规模,需耗费巨大的计算量以得到全局最优解.如何有效地求解这类搜索空间巨大的全局优化问题一直是人们关注的一个焦点.本文采用加强了局部搜索能力的混合遗传算法有效地解决了这一问题.表 2 所示为分别用 GA 和 HGA 求解该问题的收敛过程.由表 2 可以看出,当 p_{size} 取 100 时,与 De Jong 函数 $F2$ 相似,GA 在第三代就停留在某一局部最优值上不再进化,有明显的早熟收敛现象,不同的是,当 GA

种群规模增大到 1500 时 ,GA 的收敛结果仍远离全局最优解 .而采用 HGA 搜索 ,当 p_{size} 取 100 时 ,不仅消除了早熟收敛现象 ,且收敛精度高于种群规模为 1500 时 GA 的精度 ;当种群规模仅增大到 300 时 ,HGA 就用较快的速度 ,以 10^{-5} 的精度收敛于解析解 .可见 ,相对于 GA 而言 ,对于搜索空间巨大的优化问题 ,HGA 在节省计算量、提高算法效率、提高计算精度等方面的优越性更加显著 .表 2 的结果同时表明 ,由于搜索空间极大 ,增加种群规模对 GA 运算性能的影响不如前一算例明显 ,而 HGA 的运算性能随种群的增大仍然有大幅度提高 .可见 HGA 对大搜索空间的适应性明显优于 GA .

表 2 Schaffer 函数 $F6$ 的 GA 和 HGA 收敛过程

Table 1 Convergence processes of GA and HGA for Schaffer $F6$ function

进化 代数	GA $p_{size} = 100$			GA $p_{size} = 500$			HGA $p_{size} = 50$			HGA $p_{size} = 150$		
	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2	$F(x)$	x_1	x_2
1	0.271 275 12	0.569 01	4.413 3	0.012 062	0.484 7	3.051 3	0.188 723	3.039 3	5.529 0	0.126 991	12.059 5	-3.494 2
2	0.178 326 14	0.061 1	6.945 0	0.012 062	0.484 7	3.051 3	0.078 999	7.669 4	-5.409 4	0.037 224	-5.814 7	-2.364 6
3	0.178 222 14	0.051 3	6.993 9	0.009 972	3.004 8	0.848 4	0.009 783	-0.852 3	-3.012 0	0.015 466	-2.349 9	-2.194 4
4	0.178 222 14	0.051 3	6.993 9	0.009 731	3.011 8	0.868 7	0.009 749	3.020 0	-0.832 6	0.009 729	-2.915 2	-1.172 3
5	0.178 222 14	0.051 3	6.993 9	0.009 716	0.589 1	3.083 0	0.009 727	-1.417 3	-2.796 5	0.009 716	-0.800 9	-3.034 5
6	0.178 222 14	0.051 3	6.993 9	0.009 716	0.560 2	3.088 1	0.009 718	2.838 7	1.342 3	0.009 716	-2.975 4	-0.998 5
7	0.178 222 14	0.051 3	6.993 9	0.009 716	0.883 1	3.010 8	0.005 835	-0.011 1	0.756 0	0.008 609	0.004 1	-0.092 8
8	0.178 222 14	0.051 3	6.993 9	0.009 716	0.565 1	3.087 2	0.003 790	0.002 2	-0.061 5	0.000 000	0.000 0	0.000 0
9	0.178 222 14	0.051 3	6.993 9	0.009 716	0.549 8	3.090 0	0.003 790	0.002 2	-0.061 5	0.000 000	0.000 0	0.000 0

5 结 语

- a. 遗传算法具有极好的全局收敛性能 ,单纯形算法具有极强的局部搜索能力 ,本文将两种方法有机地结合起来形成的混合遗传算法有两者优点 ,提高了算法的运行效率和求解质量 .
- b. 本文引入的小生境淘汰运算有效地维持了群体的多样性 ,抑制了早熟收敛现象 ,从而可减小遗传算法的种群规模 ,降低运算量 ,为遗传算法用于大规模工程优化实践提供了可能 .

参考文献 :

[1] 周明 ,孙树栋 .遗传算法原理及应用[M].北京 :国防工业出版 ,2000.23 ~ 25 .

An Efficient Hybrid Genetic Alogrithm

LIU Jie , WANG Yuan

(College of Civil Engineering , Hohai Univ . , Nanjing 210098 , China)

Abstract : The limitations of the primary genetic algorithm applied to practice are discussed , including poor local search ability , premature convergence , large amount of calculation , and bad adaptability to large search space . A hybrid genetic algorithm is developed by combination of the simplex algorithm with the genetic algorithm , and niche selection and accelerated operation are used and combined with the present algorithm to solve the above mentioned problems . Case studies on typical test functions show that the hybrid genetic algorithm is efficient and reliable .

Key words : hybrid genetic method ; simplex algorithm ; niche selection ; accelerated operation