第 17 卷 第 1 期 控 制 与 决 策 2002 年 1 月
Vol. 17 No. 1 Control and Decision Jan. 2002

文章编号: 1001-0920(2002) 01-0019-05

一种快速收敛的混合遗传算法

向 丽, 顾培亮
(天津大学 系统工程研究所, 天津 300072)

摘 要: 利用遗传算法早熟的特点, 构造出一种快速收敛的混合算法来求解优化问题, 并分析了它的收敛性。它是使用遗传算法来生成搜索方向, 从而保证了算法的收敛性。该算法利用遗传算法的全局搜索能力, 并采用 Nelder-Mead 单纯形法来加强算法的局部搜索能力, 加快了算法的收敛速率。模拟实验表明, 该方法具有高效性和鲁棒性。

关键词: 遗传算法; 全局优化; 收敛性; Nelder-Mead 单纯形法
中图分类号: TP 18 文献标识码: A

Hybrid genetic algorithm with quick convergence

XIANG Li, *GU Pei-liang*
(Institute of Systems Engineering, Tianjin University, Tianjin 300072, China)

Abstract: Premature convergence and low converging speed are the distinct weaknesses of genetic algorithms. A hybrid algorithm that can quickly converge to the optimal set is proposed and its convergence is analyzed. Some hybrid genetic algorithms use the genetic algorithms as the main body and directly act on the solution space of the problem. They are different from the hybrid algorithm, because the hybrid algorithm implements indirect search, that is, the search direction is generated by using GAs. On the one hand, the global search capability of GAs is utilized to guarantee the convergence of the hybrid algorithm. On the other hand, Nelder-Mead Simplex is used to strong the local search and fast convergence of the hybrid algorithm. Computed results and theory analysis indicate that the method is a robust and efficient algorithm with global optimization.

Key words: genetic algorithms; global optimization; convergence; Nelder-Mead simplex

1 引 言

以遗传算法(GA) 为代表的进化计算已广泛应用于各个领域。然而, 遗传算法在应用中存在着收敛速度慢和早熟等问题[1,2], 其根本原因是该算法既从群体的全局多样性出发, 又采用定向性遗传算子。前者保证了 GA 有较大的搜索范围, 使群体不易陷入局部最小区域; 后者强调了将群体引入一个特定的子空间。显然, 这二者是矛盾的。为此, 许多学者做了很多相关的改进工作[3], 主要包括: 1) 调整遗传算法中的各种参数; 2) 维持群体的多样性; 3) 引入子群体。然而这些方法并未从根本上解决上述问题。因此, 本文构造了一种新的混合算法。

收稿日期: 2000-04-18; 修回日期: 2001-07-20
基金项目: 国家自然科学基金项目(79770060)
作者简介: 向丽(1972—), 女, 陕西山阳人, 博士生, 从事决策理论及方法、可持续发展等研究; 顾培亮(1935—), 男, 上海人, 教授, 博士生导师, 从事大系统、系统决策和可持续发展等研究。

2 算法描述

令优化问题 $\min f$ 的目标函数 $f: R^n \to R$ 为一非连续或多峰实数函数。

定义 1 令 $F = \{M^k, 1 \le k \le p\}$ 为 p 个二进制编码矩阵 M 的集合, 即

$$M^k = (m_{ij}^k) \tag{1}$$

其中, $m_{ij} = 0$ 或 1, $1 \le i \le n$, $1 \le j \le n_m$, n_m 为一任意正整数。本文取 $n_m = 16$。

定义 2 令 $\Gamma: F \to R^n$ 为一单值映射, 即

$$d^k = \Gamma(M^k) \tag{2}$$

$$d_i^k = (-1)^{m_{i1}^k} \sum_{j=2}^{n_m} 2^{j-2} m_{ij}^k \tag{3}$$

显然, 可通过矩阵 M 来生成 R^n 中的每个矢量。

混合遗传算法的具体步骤如下:

Step1 设置算法的相关参数

p: 随机搜索的方向个数; ϵ: 算法收敛的精度阈值; L: 算法搜索方向上的步长; t: 算法第 t 次迭代次数; max-iteration: 每次迭代算法执行的内循环次数。

Step2 算法初始化

$z_1 = x_t$

for $k = 1$ **to** p **do**

  for $i = 1$ **to** n **do**

    for $j = 1$ **to** n_m **do**

      随机生成 m_{ij}^k 的第 j 位

      基因 = random(0, 1)

    endfor

  endfor

  由式(3) 计算 d^k; 使用 Nelder-Mead 法求解问题

$$\min_{L \in R} f(z_h + L^k d^k) \tag{4}$$

endfor

Step3 基于遗传算法的搜索方向生成

Label: for $k = p + 1$ **to** $2p$ **do**

  使用遗传算法生成新的矩阵 M^k, 并由式(2) 计算 d^k; 使用 Nelder-Mead 单纯形法求解问题(4)

endfor

从 $2p$ 个目标值 $f(z_h + L^k d^k)$ 中选出 p 个较小目标值所对应的矩阵 $M^1, M^2, \cdots, M^p$; 令 $f(z_h + L^* d^*) = \min_{1 \le k \le 2p} f(z_h + L^k d^k)$, 即 L^* 和 d^* 为最优的步长和方向; $z_{h+1} = z_h + L^* d^*$

if $h <$ max-iteration **then**

  goto **Label**; $h = h + 1$

else

  goto **Step4**

endif

Step4 线性模式搜索

$d = z_{h+1} - z_1$; 使用 Nelder-Mead 法求解问题 $\min_{L \in R} f(z_h + \hat{L} d)$; $x_{t+1} = z_{h+1} + \hat{L} d$

Step5 终止条件检验

if $\| x_{t+1} - x_t \| \ge \epsilon$ **then**

  $h = 1$; $z_1 = x_{t+1}$; $t = t + 1$; goto **Step3**

else

  Stop

endif

应注意到算法中的两个重要参数: 参数 p 影响算法的搜索密度, 其值越大, 算法基于当前点搜索到的点就越多; 参数 max-iteration 影响搜索的多样性, 其值越大, 算法搜索不同的区域就越多。该参数与 p 的比值表示算法搜索多样性与强度之间的比值。

对于一般的优化问题, Nelder-Mead 单纯形法能较快地逼近最优解, 具有较强的局部搜索能力, 但它对初始解具有较强的依赖性[4]。

3 遗传算法

由定义 1 知, 矩阵 M 决定着搜索方向。对于任意的 k, 矩阵 M^k 对应一个搜索方向, 它相当于一个染色体, M^k 的个数 p 即为群体大小。因此, 可采用标准遗传算法[5] 来生成新的搜索方向。适应度的计算由式(4) 完成。

4 收敛性分析

定义 3 令 X 和 Y 分别为空间 R^p 和 R^q 上的非空闭集, $A: X \to Y$ 为点到集合的映射。若 $x_t \in X$, $x_t \to x$; $y_t \in Y$, $y_t \to y$, 则称映射 A 在 $x \in X$ 处封闭。

定义 4 若 $Z \subset X$ 且映射 A 在 Z 上的每个点都封闭, 则称映射 A 在 $Z \subset X$ 上封闭。

定理 1 令 X 为 R^n 空间上的非空集合, $\Omega \subset X$ 为非空解集, $A: X \to X$ 为点到集合的映射。给定 $x_1 \in X$, 假设序列 $\{x_t\}$ 生成方式如下: 若 $x_t \in \Omega$, 则算法停止; 否则, 令 $x_{t+1} \in A(x_t)$, $t = t + 1$, 继续。

假设算法生成的序列 $x_1, x_2, \cdots$ 包含在 X 的一个紧子集中; 存在一个连续下降函数 f, 当 $x \notin \Omega$, $y \in A(x)$ 时, 有 $f(y) < f(x)$。

若映射 $A: X \to X$ 在 Ω 的补集上封闭, 则算法或者在有限步内收敛于 Ω 中的一个点, 或者算法生成的无限序列 $\{x_t\}$ 满足:

1) 序列 $\{x_t\}$ 的每个收敛的子序列在 Ω 上都有极限, 即 $\{x_t\}$ 的所有聚点都属于 Ω;

2) 对于某些 $x \in \Omega$, 有 $f(x_t) \to f(x)$。

证明　若 $x_t \in \Omega$, 则算法停止。现在假设算法生成无限序列 $\{x_t\}$。令 $\{x_t\}_\zeta$ 为一收敛的子序列, 其极限为 x 且 $x \in M$。由函数 f 是连续的, 则当 $t \in \zeta$ 时, 有 $\lim_{t\to\infty} f(x_t) = f(x)$。对于任意给定的 $\epsilon > 0$, 存在一个 $T \in \zeta$ 使得当 $t \geq T$ 时, 有 $f(x_t) - f(x) < \epsilon$。而当 $t = T$ 时, 有 $f(x_T) - f(x) < \epsilon$; 当 $t > T$ 时, 有 $f(x_t) < f(x_T)$。因此 $f(x_t) - f(x) < \epsilon$。显然, 对于任意的 $t > T$ 和 $\epsilon > 0$, 都有 $\lim_{t\to\infty} f(x_t) = f(x)$。

使用反证法来证明 $x \in \Omega$。假设 $x \notin \Omega$。考虑序列 $\{x_{t+1}\}_\zeta$, 因为序列 $x_1, x_2, \cdots$ 包含在 X 的一个紧子集中, 且序列 $\{x_t\}$ 的每个收敛子序列在 Ω 上都有极限, 因此有 $f(\bar{x}) = f(x)$。

由映射 A 封闭, 则当 $t \in \zeta$ $x_t \to x$, $x_{t+1} \in A(x_k)$, $x_{t+1} \to \bar{x}$ 时, 有 $\bar{x} \in x(A)$, $f(\bar{x}) < f(x)$。这与命题相矛盾, 故 $x \in \Omega$, 第 1 部分证毕。由极限定理, 第 2 部分也成立。□

定理 2　令目标函数 $\min f(x)$ $(x \in R^n)$ 为可微函数, 矢量 $y \in \arg\min f(x, d_n)$, 其中 d_n 为以 x 为初始点的 n 个归一化搜索方向。假设下列条件为真:

1) 若存在一个 $\epsilon > 0$, 使得 $\det[D(x)] \geq \epsilon$, 其中 $D(x)$ 为 $n \times n$ 矩阵, 每列为由算法生成的搜索方向;

2) 沿任何方向的搜索, $y \in \arg\min f(x, d_n)$ 都是唯一的。

给定初始点 x_1, 假定算法生成序列 $\{x_t\}$ 如下: 若 $\nabla f(x_k) = 0$, 则算法停止; 否则, $x_{t+1} \in \arg\min f(x_t, d_n)$, $k = k + 1$。若序列 $\{x_t\}$ 包含在 R^n 的一个紧子集中, 则 $\{x_t\}$ 的聚点 x 必满足 $\nabla f(x) = 0$。

证明　若序列有限, 则上述结论立即成立。令 ζ 为一正整数的无限序列, 并假设算法生成无限序列 $\{x_t\}_\zeta$ 且收敛于 x。

使用反证法来证明 $\nabla f(x) = 0$。设 $\nabla f(x) \neq 0$。考虑序列 $\{x_{t+1}\}_\zeta$, 由假设知, 序列 $\{x_{t+1}\}_\zeta$ 包含于 R^n 中的一个紧子集, 因此存在 $\zeta' \subset \zeta$ 使得 $\{x_{t+1}\}_{\zeta'}$ 收敛于 x'。

下面证明由点 x 出发, 沿 n 个线性独立的方向求解目标函数便可获得 x'。令 $D(x)_t$ 为 $n \times n$ 矩阵, 它的每列 $d_{1t}, d_{2t}, \cdots, d_{nt}$ 为算法第 t 次迭代生成的搜索方向。因此, $x_{t+1} = x_t + D_t L_t = x_t + \sum_{j=1}^{n} L_{jt} d_{jt}$, 其中 L_{jt} 为沿方向 d_{jt} 移动的距离, 即步长。

令 $y_{1t} = x_t$, $y_{j+1,t} = y_{jt} + D_{jt} L_{jt}$, $j = 1, 2, \cdots, n$。显然, 有 $x_{t+1} = y_{n+1,t}$, $f(y_{j+1,t}) \leq f(y_{jt} + L d_{jt})$。

因为 $\det[D(x)_t] \geq \epsilon > 0$ 且 $D(x)_t$ 可逆, 所以 $L_t = D_t^{-1}(x_{t+1} - x_t)$。又因为 $D(x)_t$ 的每列为归一化列, 所以存在 $\zeta'' \subset \zeta'$, 使得 $D(x)_t \to D(x)$ 且 $D(x)$ 可逆。

对于 $t \in \zeta''$, 有 $x_{t+1} \to x'$, $x_t \to x$, $D(x)_t \to D(x)$, $L_t \to L$, 其中 $L = D^{-1}(x' - x)$。显然, $x' = x + DL = x + \sum_{j=1}^{n} L_j d_j$。令 $y_1 = x$, $y_{j+1} = y_j + L_j D_j$, 则 $x' = y_{n+1}$。

注意: 当 $t \in \zeta''$ 且 $t \to \infty$ 时, $L_{jt} \to L_j$, $x_t \to x$, $d_{jt} \to d_j$, $x_{t+1} \to x'$, $y_{jt} \to y_j$。由函数 f 的连续性, 得 $f(y_{j+1}) \leq f(y_j + L d_j)$。

至此, 我们证明了由点 x 出发, 沿 n 个线性独立的方向求解目标函数可获得 x'。

显然, $f(x') \leq f(x)$。首先考虑 $f(x') < f(x)$。因为序列 $\{f(x_t)\}$ 为非增加序列, 又因为当 $t \in \zeta''$ 且 $t \to \infty$ 时, $f(x_t) \to f(x)$, 即 $\lim_{t\to\infty} f(x_t) = f(x)$。显然, 这是不可能的, 因为当 $t \in \zeta''$ 且 $t \to \infty$ 时, $x_{t+1} \to x'$, 这与命题 $f(x') < f(x)$ 相矛盾。

其次考虑 $f(x') = f(x)$。由定理的条件 2) 知, $\nabla f(x)^t d_j = 0$。因为 $d_1, d_2, \cdots, d_n$ 线性独立, 故有 $\nabla f(x) = 0$。□

定理 3　令函数 $f(x)$ 为连续函数, X 为 R^n 空间上的非空集合, $\Omega \subset X$ 为非空解集。考虑点到集合的映射 $C: X \to X$ 满足条件: 给定 $x \in X$, $y \in C(x)$, 则 $f(y) \leq f(x)$。若点到集合的映射 $B: X \to X$ 在 Ω 的补集上封闭, 并满足条件: 给定 $x \notin \Omega$, $y \in B(x)$, 则 $f(y) < f(x)$。

现在考虑由复合映射 $A = CB$ 所刻画的算法。给定 $x_1 \in X$, 假设序列 $\{x_t\}$ 生成方式如下: 若 $x_t \in \Omega$, 则算法停止; 否则, 令 $x_{t+1} \in A(x_t)$, $t = t + 1$, 继续。

若集合 $\Lambda = \{x: f(x) \leq f(x_1)\}$ 为紧的, 则算法或者在有限步内收敛于 Ω 中的一个点, 或者序列 $\{x_t\}$ 的所有聚点属于 Ω。

证明　若 $x_t \in \Omega$, 则算法停止。现在假设算法生成无限序列$\{x_t\}$。令$\{x_t\}_\zeta$为一收敛的子序列, 其极限为 x。当 $t \in \zeta$时, 由定理 1 知$\lim_{t\to\infty} f(x_t) = f(x)$。

设 $x \notin \Omega$, 考虑序列$\{x_{t+1}\}_\zeta$。根据复合映射 A 的定义, 以及 $y_t \in B(x_t)$, $x_{t+1} \in C(y_t)$, $y_t, x_{t+1} \in \Lambda$, 则存在一个指标集 $\zeta' \subset \zeta$ 使得当 $t \in \zeta'$时, 有 $y_t \to y$, $x_{t+1} \to x'$成立。由映射 B 封闭, 得 $f(y) < f(x)$。

因为 $x_{t+1} \in C(y_t)$, 由假设知当 $t \in \zeta'$时, 有 $f(x_{t+1}) \le f(y_t)$ 成立。两边取极限, 得 $f(x') \le f(y)$。

因为 $f(y) < f(x)$, 所以 $f(x') < f(x)$。显然有 $x \in \Omega$。□

定理 4　混合遗传算法收敛于全局最优解, 如果: 1) 函数 f 可微, 最优解集 $x^* = \{x : \nabla f(x^*) = 0\}$; 2) max-iteration $\ge n$; 3) 最优方向 d^* 是线性独立的。

证明　算法每次迭代都有 max-iteration $\ge n$ 个线性独立的搜索方向和一个模式搜索方向组成。令 N 表示 max-iteration $\ge n$ 个搜索方向的集合, C 表示一个模式搜索方向的集合。由定理 2 知 N 是紧的。若沿任何方向搜索, 目标函数 f 的最小值都唯一, 则 $f(x_{t+1}) < f(x_t)$, $\forall x_t \notin x^*$, 且目标函数 f 拟凸。根据映射 C 的定义, 有 $f(z) \le f(x_{t+1})$, $\forall z \in C(x_{t+1})$。显然, 集合$\{x : f(x) \le f(x_1)\}$是紧的, 其中 x_1 为算法的任一初始点。由定理 3 及目标函数 f 的性质知, 算法收敛于全局最小点。□

5　模拟实验与结果分析

5.1　性能比较参数

衡量一种算法的性能参数包括:

1) 最终解的优劣度或精度: 最终解的优劣度可通过误差值来度量。

2) 获取最优解的概率: 可通过多次运行中成功得到最优解的次数作为估计值。当群体中最好解达到一定精度时, 即认为算法得到最优解。本文取 $\epsilon < 10^{-5}$。

3) 计算时间: 在保证解一定精度的条件下, 计算时间越少, 采样点越少, 算法性能就越好。

表 1　求解 2 维 **Rosenbrock** 函数时的性能比较

起始点	采样次数				最小值			
	ARS	SA	SGA	HA	ARS	SA	SGA	HA
1 001, 1 001	3 411	500 001		2 789	1 586.4	1.8E-10		1.3E-12
1 000, -999	131 841	508 001		5 642	8.6E-9	2.6E-9		2.6E-10
-999, -999	15 141	524 001		6 246	1.2E-9	1.2E-9		4.4E-11
-999, 1 001	3 802	484 001	10 000	3 201	583.2	4.2E-8	3.3E-8	2.5E-10
1 443, 1	181 280	492 001		3 897	4.7E-10	1.5E-8		3.5E-10
1, 1 443	2 629	512 001		5 866	1 468.9	1.6E-9		2.2E-10
1.2, 1	6 630	488 001		1 532	5.5E-7	2.0E-8		1.8E-16

表 2　求解 4 维 **Rosenbrock** 函数时的性能比较

起始点	采样次数				最小值			
	ARS	SA	SGA	HA	ARS	SA	SGA	HA
101, 101, 101, 101	519 632	1 288 001		228 534	1.9E-6	5.0E-7		4.8E-9
101, 101, 101, -99	194 720	1 328 001		214 532	1.7E-6	1.8E-7		2.1E-9
101, 101, -99, -99	183 608	1 264 001		268 547	3.8E-6	5.9E-7		2.2E-8
101, -99, -99, -99	195 902	1 296 001		301 102	2.3E-6	7.4E-8		2.8E-9
-99, -99, -99, -99	190 737	1 304 001	400 000	44 526	2.7E-6	3.3E-7	3.7E-8	3.4E-9
-99, 101, -99, 101	4 172 290	1 280 001		225 412	2.6E-6	2.8E-7		3.9E-8
101, -99, 101, -99	53 878	1 272 001		220 103	3.7	2.3E-7		4.2E-8
201, 0, 0, 0	209 415	1 288 001		162 489	1.1E-6	7.5E-7		3.0E-8
1, 201, 1, 1	215 116	1 304 001		267 849	1.2E-6	4.6E-7		5.8E-8
1, 1, 1, 201	29 069 006	1 272 001		261 345	2.2E-6	5.2E-7		4.3E-8

本文采用函数评价的次数(采样次数) 和解的精度来评价算法的好坏。

5.2 模拟实验

我们比较自适应随机搜索(ARS)、模拟退火(SA)[6]、标准遗传算法(SGA) 和本文混合算法(HA) 的性能。选择 Rosenbrock 函数: $f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$, 它的全局最小点为(1, 1, …, 1), 最小值为 0。

实验结果列于表 1 和表 2。在 Rosenbrock 函数曲面山谷中的点, 其最速下降方向几乎与到函数最小值的最佳方向垂直, 因此传统算法(如最速下降法) 较难求解此问题。实验中发现, 在高维情况下, 标准遗传算法也难以求解该问题。

需要说明的是, SGA 为群体算法, 其结果为 10 次的平均值。在 4 维问题的求解中, 有 3 次没有找到最优解。无论从解的质量还是从采样次数看, HA 算法均优于其它算法。

6 结　　论

本文将遗传算法与 Nelder-Mead 单纯形法相结合, 构造出一种快速收敛的混合算法来求解优化问题。它不象其它混合遗传算法以遗传算法为主体并直接作用于问题的解空间, 而是实施间接搜索, 即使用遗传算法来生成搜索方向。因此, 算法能搜索整个解空间, 保证了算法的收敛性。该算法利用遗传算法的全局搜索能力, 并采用 Nelder-Mead 单纯形法来加强算法的局部搜索能力, 从而加快了算法的收敛速率。理论分析和模拟实验表明, 该算法是一种具有高效性和鲁棒性的算法。

我们下一步要做的工作是分析算法的收敛速率。

参考文献(References):

[1] Goldberg D E. Genetic algorithms in search, optimization and machine learning[M]. Reading: Addison-Wesley, 1989.

[2] Rudolph G. Convergence analysis of canonical genetic algorithms[J]. IEEE Trans on Neural Networks, 1994, 5(1): 96-101.

[3] 曹先彬, 高竣, 王煦法. 基于生态竞争模型的遗传强化学习[J]. 软件学报(J of Software), 1999, 10(6): 658-662.

[4] 陈开明. 非线性规划[M]. 上海: 复旦大学出版社, 1991.

[5] 陈国良. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1996.

[6] Corana A. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm [J]. ACM Trans on Math Software, 1987, 13(3): 262-280.

(上接第 18 页)

[3] Ding S X, Ding E L, Jeinsch T. An approach to analysis and design of observer and parity relation based FDI systems[A]. Proc of the 14th IFAC World Congress [C]. Beijing, 1999. 37-42.

[4] Ding X, Guo L, Frank P M. Parameterization of linear observers and its application to observer design[J]. IEEE Trans on Automatic Control, 1994, 39(8): 1648-1652.

[5] Frank P M, Ding X. Survey of robust residual generation and evaluation methods in observer-based fault detection systems[J]. J Process Control, 1997, 7(6): 403-424.

[6] Frank P M. Enhancement of robustness in observer-based fault detection[J]. Int J Control, 1994, 59(4): 955-981.

[7] Wang H, Lam J. An optimization approach to design robust fault detection observers[A]. The 3rd Asia Control Conf[C]. Shanghai, 2000. 3052-3056.

[8] Patton R J, Hou M. On sensitivity of robust fault detection observers[A]. Proc 14th IFAC World Congress Conf[C]. Beijing, 1999. 67-72.

[9] Niemann H, Saberi A, Stoorvogel A A, et al. Exact, almost and delayed fault detection: An observer based approach[J]. Int J Robust and Nonlinear Control, 1999, 9(3): 215-238.

[10] Kim J H, Park H B. H_∞ state feedback control for generalized continuous/discrete time-delay system [J]. Automatica, 1999, 35(6): 1443-1451.

[11] Mahmoud M S, Zribi M. H_∞-controllers for time-delay systems using linear matrix inequalities[J]. J of Optim Theory and Appl, 1999, 100(1): 89-122.