

基于最速方向搜索的混合遗传算法

涂井先, 刘 伟

(广东工业大学 应用数学学院, 广东 广州 510006)

摘要: 针对目标函数复杂求导困难或目标函数不可导的优化问题, 给出了最速方向搜索方法, 它在搜索过程不需要导数信息, 克服了一些局部搜索算法需要求出导数的缺点. 在此基础上提出了基于最速方向搜索的混合遗传算法, 将最速方向搜索算法与遗传算法有效结合, 增强了遗传算法的搜索能力. 数值实验证明, 改进后的遗传算法性能优于当前一些较好的优化算法.

关键词: 遗传算法; 最速方向; 局部搜索; 适应值共享

中图分类号: TP183

文献标志码: A

文章编号: 1007-7162(2011)04-0034-04

遗传算法是一种模拟生物进化的智能优化算法. 在解决实际优化问题上, 遗传算法是最有效的优化算法之一^[1-3]. 传统的优化算法具有局部搜索能力很强但全局搜索能力弱的特点, 而遗传算法具有全局搜索能力很强但局部搜索能力弱的特点. 所以, 如何将传统优化算法和遗传算法有效结合以改善遗传算法的搜索性能一直是研究的热点^[4-8]. 局部搜索算法分为两类, 第一类是需要求偏导数的, 如最速下降法、共轭梯度法和牛顿法等; 第二类是不需要求偏导数的, 如单纯形法、模式搜索法和 Powell 法等^[9]. 在很多优化问题中, 目标函数复杂偏导数求解困难, 或者目标函数不可导, 此时第一类局部搜索算法往往求解无效, 而笔者给出的最速方向搜索算法就很好地解决了此类问题, 在个体周围产生足够多的方向, 并找出使目标函数下降最快的方向, 沿着该方向进行一维搜索, 使个体逐步靠近局部最优. 最速方向搜索算法属于第二类算法, 具有强大的局部搜索能力, 与全局搜索能力强的遗传算法结合产生的新算法同时继承了两个算法的优点, 具有强全局搜索能力和强局部搜索能力. 本文拟将最速方向搜索算法与基于适应值共享的遗传算法^[10-11]有效结合, 提出基于最速方向搜索的混合遗传算法, 并通过数值实验验证新遗传算法的有效性.

1 最速方向搜索算法

函数优化模型描述如下:

$$\min f(x),$$

其中 $x \in \mathbf{R}^n$, $\mathbf{R}$ 为实数域, $x = (x_1, x_2, \dots, x_n)$, $x_i \in$

$[a_i, b_i]$, $f(x)$ 为连续函数.

在下面的描述中, n 均指自变量的维数. 最速方向搜索算法的思想是在给定 x^* 和 $4n$ 个单位方向向量中, 找出一个使 $f(x)$ 在 x^* 处下降最快的方向 v^* , $f(x)$ 在 x^* 处沿着 v^* 进行一维搜索得到迭代点 x' , 然后用 x' 替代 x^* 并重新产生 $4n$ 个单位方向向量, 重复以上操作, 直至找到 $f(x)$ 的近似局部最优.

1) 随机单位向量 $d = (d_1, d_2, \dots, d_n)$ 的产生算法.

步骤 1: $i = 1$.

步骤 2: 随机产生 $[-1, 1]$ 之间的随机数, 并赋给 d_i .

步骤 3: 如果 $i < n$, $i = i + 1$, 返回步骤 2, 否则, 将 d 单位化, 结束.

2) 单位方向向量组 $v_1, v_2, \dots, v_{4n}$ 的产生算法.

步骤 1: 将 $e_1, e_2, \dots, e_n$ 和 $-e_1, -e_2, \dots, -e_n$ 分别赋给 $v_1, v_2, \dots, v_{2n}$, 其中, e_i 为第 i 个分量为 1、其它分量为 0 的 n 维行向量.

步骤 2: 用算法 1) 的方法产生 $2n$ 个随机单位向量, 并赋给 $v_{2n+1}, v_{2n+2}, \dots, v_{4n}$.

3) 已知点 x^* 和一组单位方向向量组 $v_1, v_2, \dots, v_{4n}$, $f(x)$ 在点 x^* 处的最速方向 v^* 的产生算法.

步骤 1: 初始化. 给定自变量增量 Δx , $\Delta x > 0$, $i = 1$, $\text{flag} = 0$.

步骤 2: 若 $\text{flag} = 0$ 且 $f(x^* + \Delta x v_i) < f(x^*)$, 则 $v^* = v_i$, $\text{flag} = 1$; 若 $\text{flag} = 1$ 且 $f(x^* + \Delta x v_i) < f(x^* + \Delta x v^*)$, 则 $v^* = v_i$.

步骤 3: 如果 $i < 4n$, 则 $i = i + 1$, 返回步骤 2, 否

收稿日期: 2011-05-03

作者简介: 涂井先(1987-), 男, 硕士研究生, 主要研究方向为智能计算.

则结束.

在给定单位方向向量组 $v_1, v_2, \dots, v_{4n}$ 和点 x^* 下, 当 $\text{flag} = 1$ 时, 说明 v^* 为 $f(x)$ 在 x^* 处的最速方向, 当 $\text{flag} = 0$ 时, 说明 $f(x)$ 在 x^* 处无最速方向.

4) 最速方向搜索算法的步骤.

步骤 1: 初始化. 给定精度 ε , 初始点 x^* .

步骤 2: 利用 2) 中的算法产生一组单位方向向量组 $v_1, v_2, \dots, v_{4n}$.

步骤 3: 利用 3) 的算法判断 $f(x)$ 在 x^* 处以及给定单位方向向量组 $v_1, v_2, \dots, v_{4n}$ 下是否有最速下降方向 v^* , 若无最速下降方向, 则结束.

步骤 4: 对 $f(x)$ 在 x^* 处沿着 v^* 进行一维搜索, 得到新的点 x' ; 若 $|f(x') - f(x^*)| > \varepsilon$, 则 $x^* = x'$, 返回步骤 2, 否则结束.

最速方向搜索算法不要求函数偏导数, 不仅能求解目标函数偏导数存在的优化问题, 而且能求解目标函数偏导数不存在的优化问题, 求解的精度很高. 该算法通过在个体周围产生足够多的方向, 计算个体沿着每个方向函数值的递减值, 把递减值最大的方向估计个体的最速下降方向, 沿着这个方向进行一维搜索, 从而逐步靠近局部最优解.

2 基于最速方向搜索的混合遗传算法

2.1 编码方式

算法采用的是实数编码.

2.2 杂交

杂交算子采用的是算术杂交, 两个父代个体分别为 X 和 Y , r 为杂交系数, 是 $(0, 1)$ 区间上的随机数, 杂交后的两个子代分别是 X' 和 Y' . 其中:

$$X' = rX + (1 - r)Y, \quad (1)$$

$$Y' = (1 - r)X + rY. \quad (2)$$

2.3 变异

变异算子采用的是非均匀变异, 它是由 Kusum Deep 和 Manoj Thakur^[12] 提出的.

$X = (x_1, x_2, \dots, x_n)$ 为一个个体, 假设需要变异的基因是 $x_i, i = 1, 2, \dots, n$, 变异后的基因为

$$x'_i = \begin{cases} x_i + \Delta(t, x_i^u - x_i), & \text{如果 } r < 0.5, \\ x_i - \Delta(t, x_i - x_i^l), & \text{其他.} \end{cases} \quad (3)$$

其中, t 为当前种群代数, r 是 $[0, 1]$ 区间上的随机数, x_i^u 和 x_i^l 分别为 x_i 的上、下界, 函数 $\Delta(t, y)$ 的值是在 $[0, y]$ 区间上, 具体定义如下:

$$\Delta(t, y) = y(1 - u^{(1-t/T)})^b, \quad (4)$$

其中, u 是 $(0, 1)$ 区间上的随机数, T 是最大进化代数, b 为常数.

2.4 适应值共享

适应值共享是用来维持种群多样性的, 它可以降低聚集在一起的个体的适应值, 它首先由 Holland^[10] 提出, 然后被 Goldberg 和 Richardson^[11] 改进.

给定一个小生境半径 r , 设当前种群代数为 t , 种群规模为 M , 父代种群个体表示为 $X_i, i = 1, 2, \dots, M$, a 为共享参数, 是大于零的. d_{ij} 为个体 X_i 到 X_j 的欧式距离, f_i 为个体在未进行适应值共享时的适应值, 它总是大于零的, 且适应度越高说明个体越好, 个体 X_i 和 X_j 的共享值为

$$s_{ij} = \begin{cases} 1 - (t^2 d_{ij}/r)^a, & \text{如果 } d_{ij} < r/t^2, \\ 0, & \text{其他.} \end{cases} \quad (5)$$

个体 X_i 的总共享值为 $s_i = \sum_{j=1}^M s_{ij}$, 适应值共享后的适应度为 $f'_i = f_i/s_i$.

2.5 混合遗传算法的步骤

步骤 1: 参数的初始化. 设定种群规模 M , 最大进化的代数 T , 杂交概率 p_c , 变异概率 p_m , 自变量增量 Δx , 它是大于零的, 局部搜索概率 p_f 定义为

$$p_f = \alpha + (\beta - \alpha)t/T, \quad (6)$$

其中, α 和 β 都是大于零的常数.

步骤 2: 初始化种群.

步骤 3: 对种群以式 (6) 中的概率 p_f 进行最速方向搜索, 若对个体 x^* 进行最速方向搜索得到新的点 x' , 用 x' 替代 x^* .

步骤 4: 对种群进行算术杂交和非均匀变异.

步骤 5: 计算种群每个个体的适应值, 然后用适应值共享来降低比较密集的个体的适应值.

步骤 6: 保留最优的个体, 用转轮法从父代种群及其后代中选择 $M - 1$ 个个体到下一代父代种群.

步骤 7: 如果 $t = T$ 或者搜索到满意的解, 迭代停止; 否则 $t = t + 1$, 返回步骤 3.

3 数值实验

数值实验是在 Matlab 7.0 中完成, 本文提出的算法被记为 ASDFSGA. ASDFSGA 的参数初始化为

$$M = 100, T = 20, p_c = 0.8, p_m = 0.1, \Delta x = 10^{-10}, \alpha = 0.01, \beta = 0.1.$$

实验结果表明: 当限定 $\Delta x \in [10^{-15}, 10^{-10}]$, $\alpha \in [0.001, 0.01]$, $\beta \in [0.05, 0.1]$ 时, 算法效率较好.

3.1 ASDFSGA、StGA^[13] 和 FEP^[14] 的算法比较

为了比较 ASDFSGA、StGA 和 FEP 的效果, 本文选用了文献 [13] 和文献 [14] 中的测试函数 f_1 、 f_2 和 f_3 , 前两个函数是单峰函数, 最后一个是多峰函数, 3 个函数的求导都很困难, 需要求偏导数的混合遗传

算法不适合解决此类优化问题,本文提出的算法可以解决此类优化问题.

$f_1(x) = \max(|x_i|, 1 \leq x_i \leq n)$, 其中 $n = 30$, $-100 \leq x_i \leq 100, f_{\min} = 0$.

$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$, 其中 $n = 30$, $-10 \leq x_i \leq 10, f_{\min} = 0$.

$$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \exp(1),$$

其中 $n = 30$, $-32 \leq x_i \leq 32, f_{\min} = 0$.

每次实验重复了 30 次,本文统计了 30 次实验每个测试函数最好解的平均值、标准差以及 30 次实验每个测试函数的平均适应值函数评价次数(见表 1), StGA 和 FEP 的实验结果分别来自文献 [13] 和 [14].

表 1 ASDFSGA、StGA 和 FEP 的比较

测试函数	算法	平均值	方差	适应值函数评价次数
f_1	ASDFSGA	8.65E-13	6.54E-14	422 300
	StGA	2.01E-08	3.42E-09	32 000
	FEP	0.3	0.50	500 000
f_2	ASDFSGA	4.21E-11	3.98E-11	98 652
	StGA	2.03E-07	2.95E-08	17 600
	FEP	8.1E-02	7.7E-04	200 000
f_3	ASDFSGA	9.23E-12	2.46E-13	99 185
	StGA	3.52E-08	3.51E-09	10 000
	FEP	1.8E-02	2.1E-03	150 000

从表 1 可以看出,在优化 f_1, f_2 和 f_3 时,在 30 次实验最好解的平均值及其方差的比较中,ASDFSGA 要优于 StGA 和 FEP; 在平均适应值函数评价次数的比较上,ASDFSGA 要优于 FEP.

3.2 ASDFSGA、HSOGA^[8] 和 LEA^[15] 的算法比较

为了比较 ASDFSGA、HSOGA 和 LEA 的效果,本文选用了文献 [8] 和文献 [15] 中的测试函数 f_4 和 f_5 , 它们都是多峰函数,求导都很困难.

在 f_4 和 f_5 中,

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$$

其中 $y_i = 1 + \frac{1}{4}(x_i + 1)$, a, k 和 m 是常数.

$$f_4(x) = \frac{1}{10} \left\{ 10 \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4),$$

$$\left. \sin^2(3\pi x_{i+1}) \right] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \Big\} +$$

$$\sum_{i=1}^n u(x_i, 5, 100, 4),$$

其中 $n = 30$, $-50 \leq x_i \leq 50, f_{\min} = 0$.

$$f_5(x) = \frac{\pi}{30} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})] (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4).$$

其中 $n = 30$, $-50 \leq x_i \leq 50, f_{\min} = 0$.

每次实验重复了 30 次,本文统计了 30 次实验每个测试函数每次实验最好解的平均值、标准差以及 30 次实验每个测试函数平均适应值函数评价次数(见表 2), HSOGA 和 LEA 的实验结果分别来自文献 [8] 和 [15].

表 2 ASDFSGA、HSOGA 和 LEA 的比较

测试函数	算法	平均值	方差	适应值函数评价次数
f_4	ASDFSGA	9.147 5E-08	4.124 9E-09	92 567
	HSOGA	4.131 6E-05	3.098 7E-08	105 518
	LEA	1.734 0E-04	1.205 0E-04	130 213
f_5	ASDFSGA	8.238 9E-12	3.843 5E-13	50 968
	HSOGA	2.080 8E-11	1.254 4E-10	98 745
	LEA	2.482 0E-06	2.274 0E-04	132 642

从表 2 可以看出: 在优化 f_4 和 f_5 时,在 30 次实验最好解的平均值及其方差的比较中,ASDFSGA 要优于 HSOGA 和 LEA; 在平均适应值函数评价次数的比较上,ASDFSGA 要优于 HSOGA 和 LEA.

总的来说,在 30 次实验最好解的平均值及其方差和平均适应值函数评价次数的比较中,ASDFSGA 优于其它 4 个算法,新算法具有高效稳健的特点.

4 小结

本文提出了基于最速方向搜索的混合遗传算法,继承了传统的遗传算法全局搜索能力强和最速方向法局部搜索能力的优点. 该算法既适用于求解目标函数偏导数存在的函数优化问题,也适用于目标函数偏导数不存在的复杂函数优化问题. 数值实验显示改进的遗传算法是一种便于实现、通用性强和高效稳健的全局优化算法.

参考文献:

- [1] Tu Chengyuan, Zeng Yanjun. A new genetic algorithm based upon globally-optimal choosing and its practices [J]. Engi-

- neering Science, 2003, 5(2): 28-29.
- [2] Ming Zhou, Sun Shudong. Principle of Genetic Algorithm and Application [M]. Beijing: National Defence Industry Press, 1999.
- [3] Zhao Chuanxin, Ji Yimu. Particle swarm optimization for 0/1 knapsack problem [J]. Microcomputer Development, 2005(10): 23-25.
- [4] 周明, 孙树栋. 遗传算法原理及其应用 [M]. 北京: 国防工业出版社, 1999.
- [5] 刘伟, 刘海林. 基于外点法的混合遗传算法求解约束优化问题 [J]. 计算机应用, 2007, 27(1): 238-240.
- [6] 白向军, 彭国华, 陈晓. 基于遗传算法和最速下降法的Bézier曲线拟合 [J]. 计算机工程与设计, 2009, 30(1): 194-196.
- [7] 张晓伟, 邢志栋, 董建民. 求解一类无约束优化的混合遗传算法 [J]. 西北大学学报: 自然科学版, 2005, 35(2): 130-132.
- [8] 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 [J]. Journal of Software, 2010, 21(6): 1296-1307.
- [9] 唐焕文, 秦学志. 实用最优化方法 [M]. 3版. 大连: 大连理工大学出版社, 1999.
- [10] Holland J H. Adaptation in Natural and Artificial Systems [M]. Ann Arbor: The MIT Press, 1992.
- [11] Goldberg D E, Richardson. Genetic algorithms with sharing for multimodal function optimization [C] // Grefenstette J ed. Proc 2nd Int Conf Genetic Algorithms and Their Applications. Hillsdale, NJ: Lawrence Erlbaum, 1987, 41-49.
- [12] Kusum Deep, Manoj Thakur. A New Crossover Operator for Real Coded Genetic Algorithms [M]. India: Elsevier Inc, 2007.
- [13] Tu Zhenguo, Lu Yong. A Robust stochastic genetic algorithm for global numerical optimization [J]. Evolutionary Computation, IEEE Transactions on, 2004, 8(5): 456-470.
- [14] Yao Xin, Liu Yong, Lin Guang-min. Evolutionary programming made faster [J]. Evolutionary Computation, IEEE Transactions on, 1999, 3(2): 82-102.
- [15] Wang Y P, Dang C Y. An evolutionary algorithm for global optimization based on level-set evolution and Latin squares [J]. Evolutionary Computation, IEEE Transactions on, 2007, 11(5): 579-595.

A Hybrid Genetic Algorithm Based on the Search Algorithm of the Steepest Direction

Tu Jing-xian, Liu Wei

(Faculty of Applied Mathematics, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: The search algorithm of the steepest direction is proposed to solve the optimization problem that the objective function is complex and the derivative is hard or impossible to obtain. The algorithm doesn't need to obtain the derivative in its searching process, overcoming the weakness of needing to acquire the derivative during their searching process, which some other local search algorithms have. A hybrid genetic algorithm, based on the search algorithm of the steepest direction, is proposed. The algorithm combines the search algorithm of the steepest direction with the genetic algorithm effectively, enhancing the searching ability of the genetic algorithm. The experimental results show that the improved genetic algorithm is more effective than some current optimization algorithms.

Key words: genetic algorithm; steepest direction; local search; fitness sharing