

一类非线性超定方程组数值解法的研究

向欢

(长沙理工大学数学与统计学院,长沙,410114)

摘 要 本文介绍了求解非线性超定方程组的 4 种数值方法,改进了穷举法和蒙特卡洛算法,提出了蒙特卡洛—穷举混合算法,应用这些数值方法求解太阳影子定位技术中提出的非线性超定方程组,根据数值试验结果分析了各算法的优缺点;最后通过数值实例,比较了各算法的求解时间和精度,验证了各算法的有效性和蒙特卡洛—穷举混合算法的高效性.

关键词 非线性超定方程组 穷举法 遗传算法 蒙特卡洛算法 MATLAB

The Study of Numerical Solutions for a Class of Overdetermined System of Nonlinear Equations

Huan Xiang

(School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha 410114, China)

Abstract In this paper Four numerical methods for solving the overdetermined system of nonlinear equations are introduced. The Enumeration method and the Monte Carlo method are improved. The Monte Carlo—Enumeration method is put forward. These numerical methods are applied to solve the overdetermined system of nonlinear equations which established at the base of the technology for positioning with the shadow of sun, advantages and disadvantages of each algorithm is analyzed according to the results of numerical experiments. At last, The computing time and accuracy of each algorithm are compared with numerical examples, the efficiency of every method and the high efficiency of Monte Carlo—Enumeration method are verified.

Key words Overdetermined system of nonlinear equations Enumeration method Genetic algorithm Monte Carlo method Matlab

1 引言

在科学与工程计算等很多领域中,很多实际问题都可转化为非线性超定方程组的求解问题,例如太阳影子定位技术等,而超定方程组一般是矛盾方程组^[1,8],需要用数值方法求近似解,因此研究数值方法具有重要的实际意义.特别是随着电脑技术以及各种编程软件的快速发

展,数值计算方法备受关注,长期以来国内外学者一直在试图寻找非线性超定方程组的高效数值解法.

含 m 个方程的 n 个未知量的非线性超定方程组的一般形式为

$$F(X) = \begin{bmatrix} f_1(X) \\ f_2(X) \\ \vdots \\ f_m(X) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (1)$$

其中 $X = (x_1, x_2, \dots, x_n) \in R^n$, $m > n$; $x_i \in [a_i, b_i]$, $i = 1, 2, \dots, n$, a_i, b_i 为常数; $f_i(X)$ ($i = 1, 2, \dots, m$) 为 n 元线性或非线性函数. 记

$$U(X) = \sum_{i=1}^m f_i^2(x_1, x_2, \dots, x_n). \quad (2)$$

若给定精度 ϵ , 那么求方程组(1)的近似解 \hat{X} , 等价于求满足 $U(\hat{X}) \leq \epsilon$ 条件的 \hat{X} [2].

2 几种数值解法

2.1 改进的穷举法

穷举法的基本原理是将定义域 R^n 用网格进行划分,再将每个格点代入方程组一一进行检验,从而求解的方法.经过改进的穷举法就是先用粗网格不断缩小解的存在范围,再用细网格在小范围中搜寻使得方程组误差达到最小的近似解.

2.2 遗传算法

遗传算法(Genetic Algorithm)是基于自然群体演化机制的一种高效全局优化搜索算法[3-4],它将一个复杂问题的解看作是群体中的个体,通过随机生成初始群体,根据目标适应度函数对每个个体进行评价,然后根据评价的好坏反复对个体进行选择、交叉和变异,使之不断进化成为更优的群体,直至其满足终止条件,最终群体中的个体即为求得的近似解.其求解非线性超定方程组的基本步骤为:

编码:将非线性超定方程组的可能解看做是群体的个体或染色体,选择合适的个体/染色体编码形式,一般采用二进制编码,对于定义域为 R^n 的方程组(1),我们可采用十进制编码.

初始化:随机生成编码字符串,即在定义域中随机取多个点,得到指定规模的初始群体:

$$\{(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}), (x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}), \dots, (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})\}.$$

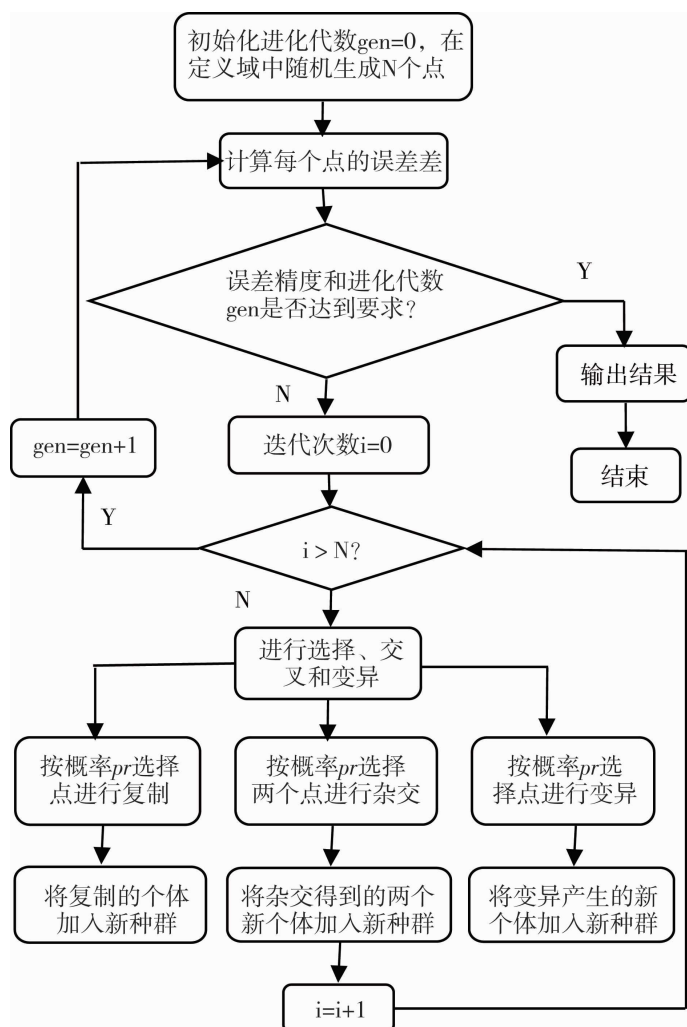
评价:将方程组的误差(2)式作为目标适应度函数 $U(X)$, 按 $U(X)$ 计算群体中每个个体 $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$, $i = 1, 2, \dots, k$ 的适应度 $U(X^{(i)})$.

选择:根据适应度,按概率 $p_r \in [0, 1]$ 从群体中选择个体 $(x_1^{(p)}, x_2^{(p)}, \dots, x_i^{(p)}, \dots, x_n^{(p)})$, 将个体复制后加到新群体中.

交叉:根据适应度,按概率 $p_c \in [0, 1]$ 从群体中选择多对个体 $(x_1^{(p)}, x_2^{(p)}, \dots, x_i^{(p)}, \dots, x_j^{(p)}, \dots, x_n^{(p)})$ 和 $(x_1^{(q)}, x_2^{(q)}, \dots, x_i^{(q)}, \dots, x_j^{(q)}, \dots, x_n^{(q)})$, 将交换部分染色体后得到的新个体 $(x_1^{(p)}, \dots, x_i^{(q)}, \dots, x_j^{(p)}, \dots, x_n^{(p)})$ 和 $(x_1^{(q)}, \dots, x_i^{(p)}, \dots, x_j^{(q)}, \dots, x_n^{(q)})$ 加入新群体中.

变异:依据适应度,按概率 $p_m \in [0, 1]$ 选择个体 $(x_1^{(p)}, x_2^{(p)}, \dots, x_i^{(p)}, \dots, x_j^{(p)}, \dots, x_n^{(p)})$, 改变部分染色体编码得到新个体 $(x_1^{(p)}, x_2^{(p)}, \dots, y_i^{(p)}, \dots, y_j^{(p)}, \dots, x_n^{(p)})$, 将其加入新群体中.

判断:若新群体满足终止条件(误差精度或进化代数达到要求),则停止进化,否则不断进行选择、交叉和变异,直至进化的群体满足终止条件,那么最后得到的群体中的个体即为非线性超定方程组的近似解.相应的技术流程图如下图.



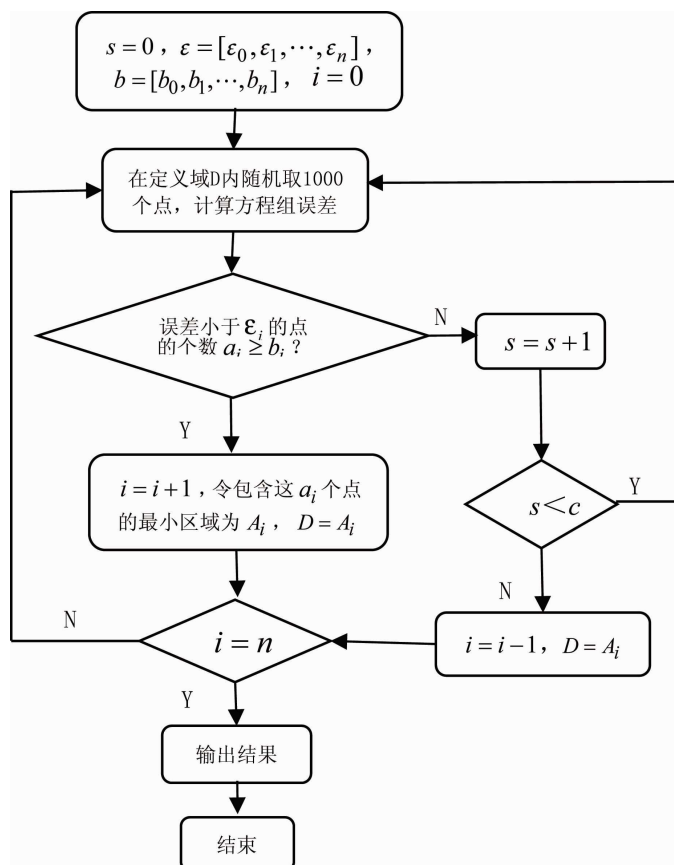
2.3 改进的蒙特卡洛算法

蒙特卡洛算法(Monte Carlo Method)是一种以概率统计理论为指导的数值计算方法,它是一种通过使用随机数(随机取点)来解决很多计算问题的统计模拟方法^[5],主要是通过统计抽样实验为各种各样的数学问题提供近似解.对于非线性超定方程组(1)的求解,蒙特卡洛算法就是通过随机取点代入方程组验证来搜索方程组的解,也就是通过在方程组的定义域 $D = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n] \subset R^n$ 中生成随机数,得到多个随机点 $(x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)})$ 代入(2)式,寻找在给定精度 ϵ 下,满足 $U(\hat{X}) \leq \epsilon$ 的近似解 \hat{X} .

改进的蒙特卡洛算法就是在控制误差精度 ϵ 越来越小的条件下分层进行多次随机取点,将定义域作为第一层区域,在第 i 层区域所取点中挑出至少 b_i 个误差精度达到 ϵ_i 的点(b_i 和 ϵ_i 的值可根据实际情况或根据数值模拟结果适当进行选取),将包含这些点的最小区域作为第 $i+1$ 层区域(若在一定迭代次数 c 的限制下,无法找到 b_i 个点,则返回第 $i-1$ 层区域随机取点重新进行搜索,得到新的第 i 层区域),从而逐步缩小解的存在范围,得到

$$[a_1, b_1] \times \cdots \times [a_n, b_n] = [a_1^{(1)}, b_1^{(1)}] \times \cdots \times [a_n^{(1)}, b_n^{(1)}] \supseteq \cdots \supseteq [a_1^{(k)}, b_1^{(k)}] \times \cdots \times [a_n^{(k)}, b_n^{(k)}],$$

那么最终区域 $[a_1^{(k)}, b_1^{(k)}] \times [a_2^{(k)}, b_2^{(k)}] \times \cdots \times [a_n^{(k)}, b_n^{(k)}]$ 中的最优点就作为方程组的近似解.其技术流程图如下图.



蒙特卡洛—穷举混合算法是将蒙特卡洛算法和穷举法结合的数值计算方法,蒙特卡洛算法收敛速度快,而穷举法计算精度高,因此将两者结合,先使用蒙特卡洛算法将解的存在范围缩小到一定程度,再利用穷举法在小范围内对方程组的解进行搜索,寻找更精确的解.

3 数值实例

太阳影子定位技术中的一些问题可转化为一类非线性超定方程组的求解问题,如 2015 年全国大学生数学建模竞赛的 A 题《太阳影子定位技术》中的问题 1 最终可转化为求解含 21 个方程、4 个未知量的非线性超定方程组(3)的问题.

$$\begin{cases} f(x_1, x_2, x_3, x_4, x_5^{(1)}, x_6^{(1)}) = 0, \\ f(x_1, x_2, x_3, x_4, x_5^{(2)}, x_6^{(2)}) = 0, \\ \vdots \\ f(x_1, x_2, x_3, x_4, x_5^{(21)}, x_6^{(21)}) = 0, \end{cases} \quad (m = 21 > n = 4) \quad (3)$$

其中, x_1 和 x_2 为直杆所处经、纬度, x_3 和 x_6 为直杆杆长和影长, x_4 和 x_5 为测量直杆影长的日期和时间^[6], 它们之间的关系由下式决定:

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = \frac{x_3}{\tan(\arcsin(\sin x_2 \sin(\sigma(x_4))) + \cos x_2 \cos(\sigma(x_4)) \cos(\gamma(x_1, x_5))))} - x_6.$$

而上式是由(4)式中太阳赤纬 σ 、时角 γ 与太阳高度角 θ 之间的关系确定得到的:

$$\begin{cases} \sigma(x_4) = 23.45 \times \sin\left(\frac{2\pi \times (284 + x_4)}{365}\right), \\ \gamma(x_1, x_5) = 15 \times (x_5 - 12 + \frac{(x_1 - 120)}{15}) = 15 \times x_5 + x_1 - 300, \\ \sin\theta = \sin x_2 \sin\sigma + \cos x_2 \cos\sigma \cos\gamma, \\ \tan(\theta) = \frac{x_3}{x_6}. \end{cases} \quad (4)$$

在方程组(3)中, $(x_5^{(i)}, x_6^{(i)}), (i = 1, 2, \dots, 21)$ 为 21 组测量数据(见表 1).

进行多次数值实验发现,穷举法、遗传算法、蒙特卡洛算法以及蒙特卡洛—穷举算法在选用不同的参数的情况下求解非线性超定方程组(3)可以得到多个不同的近似解,见表 2 至表 5.

表 1 直杆影长测量数据

序号	北京时间(x_5)	x 坐标(米)	y 坐标(米)	影子长度(x_6)
1	14:42	1.0365	0.4973	1.149626
2	14:45	1.0699	0.5029	1.182199
3	14:48	1.1038	0.5085	1.215297
4	14:51	1.1383	0.5142	1.249051
5	14:54	1.1732	0.5198	1.283195
6	14:57	1.2087	0.5255	1.317993
7	15:00	1.2448	0.5311	1.353364
8	15:03	1.2815	0.5368	1.389387
9	15:06	1.3189	0.5426	1.426153
10	15:09	1.3568	0.5483	1.4634
11	15:12	1.3955	0.5541	1.501482
12	15:15	1.4349	0.5598	1.540232
13	15:18	1.4751	0.5657	1.579853
14	15:21	1.516	0.5715	1.620145
15	15:24	1.5577	0.5774	1.661271
16	15:27	1.6003	0.5833	1.703291
17	15:30	1.6438	0.5892	1.746206
18	15:33	1.6882	0.5952	1.790051
19	15:36	1.7337	0.6013	1.835014
20	15:39	1.7801	0.6074	1.880875
21	15:42	1.8277	0.6135	1.927918

表 2 改进的穷举法求解方程组(3)的结果

区域	结果	误差	计算时间	参数
$[0,180] \times$	(109.1,19.4,2.02,110)	3.3419×10^{-7}	102.430792s	见源程序
$[0,90] \times$	(109.1000,19.4000,2.0200,110)	3.3419×10^{-7}	5.593912s	见源程序
$[0,4] \times$	(108.7500,19.2000,2.0300,108)	5.9491×10^{-7}	13.951687s	见源程序
$[0,365]$	(111.1000,14.7000,1.9500,137)	1.0772×10^{-6}	31.12834s	见源程序
	(106.8000,18.6000,2.1100,101)	3.1974×10^{-6}	28.363471s	见源程序

上表为求解含 4 个未知数的方程组(3)的结果,将方程组未知数增至 5 个或将定义域成倍

增大,实验发现穷举法的求解时间就会大大增加甚至难以求得结果.因此,改进的穷举法虽原理简单,便于编程实现,但只有当方程组未知数个数较少、定义域较小时,求解结果才稳定且较精确.

表 3 遗传算法求解方程组(3)的结果

区域	结果	误差	迭代次数	参数
[1,180]×	(109.107,19.098,2.013,108.787)	1.8175×10^{-6}	10	N=400,n1=20, n2=10,e=10 ⁻¹⁰
	(109.225,19.156,2.01,109.482)	2.0245×10^{-6}	7	
[1,90]×	(105.207,17.131,2.167,251.141)	8.9035×10^{-6}	13	N=200,n1=20, n2=10,e=10 ⁻¹⁰
[1,4]×	(108.776,10.982,2.036,206.515)	1.3914×10^{-6}	11	
[1,365]	(108.315,19.05,2.046,106.126)	1.7973×10^{-6}	7	
	(109.123,18.728,2.049,104.505)	3.9836×10^{-6}	6	

注:N 为初始种群规模,n1 为遗传代数,n2 为停滞代数,e 为要求的误差精度,x0 为迭代初始点,没有列出来的参数为遗传算法工具箱中设定的默认值.

遗传算法迭代收敛速度极快,能很快得到结果,因为它采用群体搜索,具有良好的并行性和较强的搜索能力;且遗传算法对复杂方程组的求解具有较强的鲁棒性,适用于大规模、高度非线性及无精确解的方程组的求解.但遗传算法的求解结果不稳定,参数的选取对结果有较大的影响,常常需要经过多次数值实验,根据实验结果才能选择到合适的参数.

表 4 改进的蒙特卡洛算法求解方程组(3)的结果

区域	结果	误差	计算时间	参数
[0,180]×	(109.2300,19.2900,2.0120,110.0000)	3.1564×10^{-7}	533.35638s	见源程序
[0,90]×	(109.1052,19.3666,2.0187,109.8383)	4.8209×10^{-7}	5.507489s	
	(109.8180,19.6107,1.9977,113.8901)	7.2791×10^{-7}	7.973128s	
[0,4]×	(108.8216,19.2582,2.0278,108.4089)	7.3852×10^{-7}	99.765679s	
	(109.8857,19.6295,1.9958,114.3120)	9.2205×10^{-7}	19.707082s	

单纯的蒙特卡洛算法不太稳定,可能难以求得方程组的近似解,但改进的蒙特卡洛算法的求解结果精确且较为稳定,具有较强的鲁棒性,对更高阶方程组仍具较高的求解效率,但由于蒙特卡洛算法的随机性,改进的蒙特卡洛算法可能需经过多次反复搜索才能将解的存在域逐步缩小,从而会比单纯的蒙特卡洛算法耗费更长的时间.

表 5 改进的蒙特卡洛算法求解方程组(3)的结果

区域	结果	误差	计算时间	参数
$[0,180] \times$	(109.2300,19.2900,2.0120,110.0000)	3.1564×10^{-7}	533.35638s	见源程序
$[0,90] \times$	(109.1052,19.3666,2.0187,109.8383)	4.8209×10^{-7}	5.507489s	
$[0,4] \times$	(109.8180,19.6107,1.9977,113.8901)	7.2791×10^{-7}	7.973128s	
	(108.8216,19.2582,2.0278,108.4089)	7.3852×10^{-7}	99.765679s	
$[0,365]$	(109.8857,19.6295,1.9958,114.3120)	9.2205×10^{-7}	19.707082s	

蒙特卡洛一穷举混合算法高效且稳定,它结合了蒙特卡洛算法和穷举法的优点,在牺牲一些速度的条件下,提高了求解精度,适用范围也更广.达到了最优的效果.但蒙特卡洛算法的随机性使得它也有一定的不稳定性,因此适当调参往往能更快地得到更理想的结果.

4 结论

随着电脑技术的快速发展,计算机的运行速度越来越快,因此相比于提高求解速度,提高求解精度才是当务之急.经数值模拟实验调参后,以精度为第一衡量标准,各算法的最优结果见表 6:

表 6 非线性超定方程组(3)的求解结果

方法	区域	结果	误差	计算时间	参数
改进的 穷举法	$[0,180] \times [0,90] \times$ $[0,4] \times [0,365]$	(109.1,19.4,2.02,110)	3.3419×10^{-7}	102.430792s	见源程序
遗传算法	$[1,180] \times [1,90] \times$	(109.107,19.098,2.013,108.787)	1.8175×10^{-6}	8.848377s	N=400, n1=20,
	$[1,4] \times [1,365]$	(105.207,17.131,2.167,251.141)	8.9035×10^{-6}	11.737675s	n2=10, e=10 ⁻¹⁰
改进的蒙特 卡洛算法	$[0,180] \times [0,90] \times$	(109.1052,19.3666,2.0187,109.8383)	4.8209×10^{-7}	5.507489s	见源程序
蒙特卡洛— 穷举混合算法	$[0,4] \times [0,365]$	(109.6337,19.5176,2.0022,112.6514)	2.9698×10^{-7}	18.950221s	见源程序

注:N 为初始种群规模,n1 为遗传代数,n2 为停滞代数,e 为要求的误差精度,x0 为迭代初始点,没有列出来的参数为遗传算法工具箱中设定的默认值.

非线性超定方程组(3)的真实解为(109.5, 18.3, 2, 108),由上表可知,从求解精度上来看,遗传算法求得的结果(109.107, 19.098, 2.013, 108.787)是最接近真实解的近似解,而蒙特卡洛-穷举混合算法的误差最小,即该方法求得的近似解最接近解析解;从求解速度上来看,改进后的蒙特卡洛算法最快,改进后的穷举法最慢.

现今,随着计算机的发展和大数据时代的到来,由人工智能思想做支撑的数值计算方法逐渐显示出优越性,加之蒙特卡洛算法等统计方法的渗透,超定方程组的解法趋于多样化,方兴未艾的数值计算方法也朝着高效率、高精度的方向发展^[7].其中,改进的穷举法虽然能保持计算结果的高精度和高稳定性,但随着方程组阶数的增加,求解速度大大下降,不适合求解高阶方程组.遗传算法虽鲁棒性强、速度快,但结果过于依赖参数的选取,十分不稳定,常常需要经过多次实验,才能得到较为精确的结果.改进的蒙特卡洛算法比较优秀,它具有很强的鲁棒性,求解结果精度较高,计算速度也较快.蒙特卡洛-穷举混合算法是最优的算法,它结合了两种方法的优点,在提高精度的同时,又保证了一定的求解速度,适用范围广,且较为稳定.

参考文献

- [1] 陈明达,凌永祥.实用计算机数值方法[M].西安:西安交通大学出版社,1992:1—200.
- [2] 李庆扬等编.数值分析[M].华中科技大学出版社,2000:20—280.
- [3] 易昆南.基于数学建模的数学实验[M].北京:中国铁道出版社,2014:1—156.
- [4] 苏三买.遗传算法及其在航空发动机非线性数学模型中的应用研究[D].西安:西北工业大学,2002.
- [5] 朱陆陆.蒙特卡洛方法及应用[D].武汉:华中师范大学,2014.
- [6] 武琳.基于太阳阴影轨迹的经纬度估计技术研究[J].天津大学计算机科学与技术学院,2010:1—27.
- [7] Zhou Fu—Zhao, Zhang Lei and Hu Xi—Yan. Least—square solutions for inverse problems of centrosymmetric matrices[J]. Computers and Math. with Appl.45(2003): 1581—1589.
- [8] Kendall E. Atkinson, An introduction to Numerical Analysis[M]. New York.1978:20—300.