

求解一类凸优化问题的区间迭代算法

唐敏^{1,2} 邓国强²

(华东师范大学软件学院 上海 200062)¹ (桂林电子科技大学数学与计算科学学院 桂林 541004)²

摘要 研究了一类非线性带约束的凸优化问题的求解。利用 Kuhn-Tucker 条件将凸优化问题等价地转化为多元非线性方程组的求解问题。基于区间算术的包含原理及改进的 Krawczyk 区间迭代算法,提出一个求解凸优化问题的区间算法。对于目标函数和约束函数可微的凸优化,所提算法具有全局寻优的特性。在数值实验方面,与遗传算法、模式搜索法、模拟退火法及数学软件内置的求解器进行了比较,结果表明所提算法就此类凸优化问题能找到较多且误差较小的全局最优解。

关键词 区间迭代算法,凸优化,Krawczyk 算子,Kuhn-Tucker 条件

中图法分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.2.051

Interval Iterative Algorithm for a Class of Convex Optimization

TANG Min^{1,2} DENG Guo-qiang²

(Software Engineering Institute, East China Normal University, Shanghai 200062, China)¹

(School of Mathematics & Computing Science, Guilin University of Electrical Technology, Guilin 541004, China)²

Abstract This paper studied solving method for nonlinear constrained convex optimization and pointed out how to make use of Kuhn-Tucker condition to transfer convex programming to find roots of multivariate nonlinear systems equivalently. Our interval approach for finding optimal solutions can be combined with the principle of interval arithmetic and the modified iterative version given by R. Krawczyk. In the case of a convex programming with differentiable objective and constrained functions, the ability of global optimization can be improved. We illustrated the numerical experiments in contrast to the classical algorithms such as genetic algorithm, pattern search method, simulated annealing method and some built-in facilities of mathematical software packages. The implementation indicates that our method can provide a relative guaranteed bound on the error of the computed value and obtain more global optimal solutions.

Keywords Interval iterative method, Convex programming, Krawczyk operator, Kuhn-Tucker condition

1 引言

凸优化是一类特殊的非线性数学规划问题,包括最小二乘和线性规划,这两类问题已经有较完备的理论和较有效的数值求解方法。关注更一般的带约束的非线性凸优化问题(目标函数和约束条件都可以是非线性的函数)的求解具有理论价值和实际意义。凸优化在自动控制系统、估计、信号处理、通信和网络、电路设计、数据分析和建模、统计、金融、组合优化和全局优化问题上有广泛的应用^[1]。

非线性数学规划的求解要比线性规划问题困难得多,从理论上说,在非线性的情形下,通过计算机得到严格的最优解是不可能的。求解方法通常是迭代法,在达到精度要求时终止,因而得到的是近似的极小值。现代优化计算方法(演化和仿生算法)的兴起,使得优化问题的求解有了新的方向和进一步的发展,比如遗传算法^[2]、模拟退火算法^[3]、蚁群算法^[4]等,它们不需利用函数的解析性质,不考虑目标函数和约束条件的凹凸性,对于任意目标函数和约束,均能获得较优的结果。

有大量软件内置数学规划问题的求解器,比如 Matlab、Excel、Lindo、Lingo、R。当模型包含几百到几千个约束时,还可使用建模语言 AMPL、MPL、OPL、GAMS 对模型进行有效的设计和管理^[5]。传统数值法的精确性和稳定性是一个较难处理的问题,与此相比,区间算术利用包含原理确保数值程序的输出包含了原数学问题的准确解。

求极值问题实际上是在给定的集合上计算函数值域的上界和下界,国内外的许多学者已将区间算术运用于优化问题的求解^[8-13]。区间算术是定义在集合上的算术运算,Moore 在 1962 年的博士论文中给出了区间算术的基本原理和方法,将之应用于数值计算的误差分析中^[6]。区间算术可用于求解基本初等函数、定积分、线性方程(组)、非线性方程(组)、积分方程(组)、微分方程(组),可应用于计算机辅助证明、带约束的全局优化问题、参数估计、机器人应用、化学工程应用、水资源分配网络设计、结构工程应用、计算机图形学、物理常数的计算等^[7]。

本文提出的求解一类凸优化问题的区间迭代算法的基本

到稿日期:2014-04-09 返修日期:2014-06-25 本文受国家自然科学基金面上项目(NSFC11371143),华东师范大学科研创新基金重点项目资助。

唐敏(1980—),女,博士生,讲师,主要研究方向为高可信计算、浮点计算,E-mail:52121500011@ecnu.cn;邓国强(1979—),男,硕士,讲师,主要研究方向为数值计算,E-mail:dengtangmin@126.com(通信作者)。

思想是利用 Kuhn-Tucker 条件将凸优化等价地转化为多元非线性方程组的求解问题,再基于区间算术的包含原理及改进的 Krawczyk 区间迭代算法,判定给定区间内方程组根的存在性,并迭代地求出根的区间包含。算法对整个有界可行域进行划分,存在多个初始测试区间,因而不会陷入局部最优。

本文第 2 节介绍了区间算术的基本概念和常用的区间算术软件包;第 3 节给出如何将凸优化转化为方程组求解及其等价性;第 4 节利用区间算术及改进的 Krawczyk 算法设计方程组求根的迭代算法,并分析时间复杂度;第 5 节中给出了两个不同规模和复杂度下的数值实验结果。

2 区间算术

区间算术(Interval Arithmetic)是定义在集合上的算术运算,利用包含原理确保数值程序的输出包含了原数学问题的解。基于区间算术的算法在舍入误差、累积误差、截断误差、不确定性初值的引入和传播下,能得到真实解的严格的界。

2.1 区间

在区间算术中,通常用端点表示法表示一个区间(Interval),即 $X=[\underline{X}, \bar{X}]$,其中 $\underline{X}$ 是 X 的左端点(下界),而 $\bar{X}$ 是区间 X 的右端点(上界)。如果 $x \in [\underline{X}, \bar{X}]$,称 $[\underline{X}, \bar{X}]$ 为 x 的区间包含(interval enclosure)。

区间宽度定义为 $w(X) = \bar{X} - \underline{X}$,中点定义为 $m(X) = \frac{1}{2}(\underline{X} + \bar{X})$ 。基本区间算术运算定义为 $X \circ Y = \{x \circ y | x \in X \wedge y \in Y\}$,其中 $\circ \in \{+, -, \times, /\}$,且当 $\circ = /$ 时, $0 \notin Y$ 。

2.2 区间向量

区间向量(Interval Vector)指向量的每个元素都是区间,对于区间向量 $X = (X_1, \dots, X_n)^T$, X 的宽度定义为 $w(X) = \max_i w(X_i)$, X 的中点定义为 $m(X) = (m(X_1), \dots, m(X_n))$ 。

2.3 区间算术软件包

最常用的区间算术软件包是用 C++ 语言编写的 PROFIL/BIAS 和用 Fortran77 编写的 INTLAB^[14]。Fortran90 版本的 INTLAB 使用了运算符重载^[15]。GNU Fortran 编译器的区间扩展版本中将区间定义为基本内部数据类型^[16]。区间算术软件包支持实数或复数区间标量、向量和矩阵(稠密或者稀疏),包含诸如标量乘、矩阵-向量乘、外积、矩阵乘法等线性代数算法、自动微分、非线性方程组求解等。

3 凸优化问题的数学模型及库恩-塔克(Kuhn-Tucker, K-T)条件

3.1 凸优化问题的数学模型

数学规划问题可用

$$\begin{cases} \min f(X), X \in R^n \\ g_j(X) \geq 0, j=1, 2, \dots, l \end{cases} \quad (1)$$

表示,其中 $f(X)$ 为目标函数, $g_j(X) \geq 0, j=1, 2, \dots, l$ 为约束条件。如果 $f(X)$ 为凸函数, $g_j(X) \geq 0, j=1, 2, \dots, l$ 为凹函数,那么式(1)是凸优化问题。

3.2 K-T 条件

K-T 条件可描述为:设 $x^* = (x_1, \dots, x_n)^T$ 是式(1)的极小点,而且在 x^* 处各起作用约束的梯度线性无关,则存在向量 $\Gamma^* = (\gamma_1^*, \gamma_2^*, \dots, \gamma_l^*)^T$,使下述条件成立:

$$\begin{cases} \nabla f(x^*) - \sum_{j=1}^l \gamma_j^* \nabla g_j(x^*) = 0 \\ \gamma_j^* g_j(x^*) = 0, j=1, 2, \dots, l \\ \gamma_j^* \geq 0, j=1, 2, \dots, l \end{cases} \quad (2)$$

条件式(2)称为 K-T 条件,满足这个条件的点称为 K-T 条件点。K-T 条件是非线性规划领域中最重要理论成果之一,给出了判定某点为最优点的必要条件。根据 K-T 条件,优化问题式(1)的极小点满足式(2)。一般来说, K-T 条件并不是充分条件,因而满足这个条件的点不一定就是最优点,但是对于凸优化,它是最优点存在的充分必要条件^[17]。

易见,式(2)等价于方程组

$$\begin{cases} \frac{\partial f(x^*)}{\partial x_1} - \gamma_1^* \frac{\partial g_1(x^*)}{\partial x_1} - \dots - \gamma_l^* \frac{\partial g_l(x^*)}{\partial x_1} = 0 \\ \vdots \\ \frac{\partial f(x^*)}{\partial x_n} - \gamma_1^* \frac{\partial g_1(x^*)}{\partial x_n} - \dots - \gamma_l^* \frac{\partial g_l(x^*)}{\partial x_n} = 0 \\ \gamma_1^* g_1(x^*) = 0 \\ \vdots \\ \gamma_l^* g_l(x^*) = 0 \end{cases} \quad (3)$$

由式(2)和式(3)的等价性可知,凸优化问题的求解等价于式(3)的多变元非线性方程组的求根问题,并满足 $\gamma_j^* \geq 0, j=1, 2, \dots, l$ 。

在方程组(3)中,共有 $n+l$ 个未知数 $x_j (j=1, 2, \dots, n)$ 和 $\gamma_j^* (j=1, 2, \dots, l)$,方程个数也为 $n+l$,因此它是一个非线性(绝大多数情况下)多变元适定方程组。

4 求解凸优化的区间迭代法

在 4.1 节中,首先给出改进的 Krawczyk 区间迭代算法,它可用于验证给定区域内非线性方程组根的存在性;4.2 节给出求解给定定义域内非线性方程组的根的迭代算法;4.3 节给出凸优化问题求解的区间迭代法。

4.1 改进的 Krawczyk 区间迭代法

对于 n 个方程、 n 个未知数的非线性方程组

$$f(X) = 0, X = (x_1, x_2, \dots, x_n) \in R^n \quad (4)$$

Moore 提出一个改进的 Krawczyk 区间迭代算法^[18],其给出了方程组(4)的解的存在性的计算性测试方法。

定理 1 令 y 是属于区间向量 X 的任意实向量,定义

$$K(X) := y - Yf(y) + \{I - YF'(X)\}(X - y)$$

其中, Y 是非奇异实矩阵, $F'(X)$ 是 $f(X)$ 的区间 Jacobian 矩阵。如果 $K(X) \subseteq X$,则非线性方程组(4)在 X 中存在一个解。

定理 1 的证明基于均值定理和 Schauder 不动点定理。由定理 1,对于给定的一个初始区间向量 $X^{(0)}$,可根据 $K(X^{(0)})$ 和 $X^{(0)}$ 的包含关系判断 $X^{(0)}$ 区间内方程组(4)是否有解,如果存在唯一的解,通过 Krawczyk 区间迭代算法可得到解的区间包含。下面给出该算法的思想的形式化描述。

算法 1 改进的 Krawczyk 区间迭代法

输入:初始区间 $X^{(0)}$,精度 ϵ_1 。

输出:如果在 $X^{(0)}$ 内存在 $f(X) = 0$ 的唯一根,则返回 $f(X) = 0$ 的根的区间包含,并且满足区间包含(区间向量)的宽度小于 ϵ_1 ;否则,返回 -1。

计算步骤:

步骤 1 [计算 $K(X^{(0)})$]

$$k := 0, y^{(0)} := m(X^{(0)})$$

$Y^{(0)} := [m(F'(X^{(0)}))]^{-1}$ 的近似值

$K(X^{(0)}) := y^{(0)} - Y^{(0)} f(y^{(0)}) + \{I - Y^{(0)} F'(X^{(0)})\} (X^{(0)} - y^{(0)})$

步骤 2 [判定根的存在性]

如果 $K(X^{(0)}) \subseteq X^{(0)}$, 那么在 $X^{(0)}$ 内存在唯一的根, 进入步骤 3 的迭代求解过程; 否则, 返回 -1, 算法终止。

步骤 3 [计算 $X^{(0)}$ 内方程组 $f(X)=0$ 的根的区间包含]

如果 $w(X^{(k)}) < \epsilon_1$, 那么返回 $X^{(k)}$, 算法终止;

否则, $X^{(k+1)} := X^{(k)} \cap K(X^{(k)})$

$k := k+1$

$y^{(k)} := m(X^{(k)})$

$r_k := \|I - Y^{(k)} F'(X^{(k)})\|$

如果 $\|I - Y^{(k)} F'(X^{(k)})\| \leq r_{k-1}$,

$Y^{(k)} := Y$, 为 $[m(F'(X^{(k)}))]^{-1}$ 的近似值。

否则, $Y^{(k)} := Y^{(k-1)}$

$K(X^{(k)}) := y^{(k)} - Y^{(k)} f(y^{(k)}) + \{I - Y^{(k)} F'(X^{(k)})\} (X^{(k)} - y^{(k)})$

返回步骤 3

对算法 1 进行收敛性分析, 如果 $K(X^{(0)}) \subseteq X^{(0)}$, 并且 $r_0 < 1$, 那么 $f(X)=0$ 在 $X^{(0)}$ 中存在唯一解, 且满足:

(1) $x \in X^{(k)} \subseteq X^{(k-1)}, k=1, 2, \dots$;

(2) $w(X^{(k)}) \leq r_0^k w(X^{(0)})$ 。

由算法 1 获得的 $\{X^{(k)}\}$ 是嵌套序列, 至少线性收敛到 $X^{(0)}$ 中的唯一解 X , 如果 $\{r_k\}$ 收敛到 0, 那么 $\{X^{(k)}\}$ 的收敛速度是超线性的^[18]。

Rump 证明了, 如果 $K(X^{(0)}) \subset X^{(0)}$, 那么在区间 $X^{(0)}$ 内存在唯一解, 无需计算矩阵的范数^[19]。

4.2 求解非线性方程组的根的区间算法

在改进的 Krawczyk 区间迭代算法基础上, 提出一个求解给定定义域内非线性方程组的区间算法, 形式化的描述见算法 2。算法 2 通过将定义域进行划分, 并利用算法 1 进行根的存在性判定及求解, 从而完成非线性方程组的求根过程。

对于 n 个方程、 n 个未知数的非线性方程组

$$f(X)=0, X=(X_1, X_2, \dots, X_n) \in B \quad (5)$$

其中, $B=(B_1, B_2, \dots, B_n)$, $B_i=[\underline{B}_i, \overline{B}_i]$, $i=1, 2, \dots, n$, 它是一个区间向量, 称为盒子(Box)。

给定区间向量宽度容差 ϵ_1 , 利用算法 2 求解方程组(5)的根的区间包含, 保证与准确根之间的距离不超过 ϵ_1 。

算法 2 求解非线性方程的根的区间包含算法

输入: 精度 ϵ_1, ϵ_2 , 定义域 $B_i=[\underline{B}_i, \overline{B}_i], i=1, 2, \dots, n$ 。

输出: 方程组(5)的根的区间包含 $X_{i,j}=[\underline{X}_{i,j}, \overline{X}_{i,j}], i=1, 2, \dots, m, j=1, 2, \dots, n$ 。其中 m 是算法找到的根的个数, 保证在 $X_{i,j}$ 内存在方程组(5)的一个根, 且 $w(X_{i,j}) < \epsilon_1$ 。

计算步骤:

步骤 1 [初始化]

令 $List := \{B\}$, $OutputList := \emptyset, i := 0$ 。

步骤 2 [算法终止性判断]

如果 $List := \emptyset$ 或者 $List$ 的第一个元素 X (初始时第一个元素为 B) 的宽度 $w(X) < \epsilon_2$, 算法终止; 否则, 删除 X , 进入步骤 3。

步骤 3 [计算 $K(X)$]

如果 $K(X) \cap X = \emptyset$, 那么可以判定在 X 中不存在 $f(X)=0$ 的根;

否则, 如果 $K(X) \subset X$, 则可判定在 X 中存在唯一的一个 $f(X)=0$ 的根, 利用算法 1 进行求解, 满足精度 ϵ_1 时算法终止, 将计算结果保存到 $OutputList$ 中, 即

$OutputList := OutputList \cup \{X_{i,j}\}$

其中 $X_{i,j}=[\underline{X}_{i,j}, \overline{X}_{i,j}], j=1, 2, \dots, n, i := i+1$;

否则, 如果 $K(X) \cap X \neq \emptyset$, 那么 $X := K(X) \cap X$, 返回步骤 3;

否则, 如果 $K(X)$ 没有定义, 对 X 二分, 满足 $X=X^{(1)} \cup X^{(2)}$,

其中, $X^{(1)} := (X_1, \dots, X_{k-1}, X_k^{(1)}, X_{k+1}, \dots, X_n)$

$X^{(2)} := (X_1, \dots, X_{k-1}, X_k^{(2)}, X_{k+1}, \dots, X_n)$

$X_k^{(1)} := [\underline{X}_k, \frac{1}{2}(\underline{X}_k + \overline{X}_k)]$

$X_k^{(2)} := [\frac{1}{2}(\underline{X}_k + \overline{X}_k), \overline{X}_k]$

$w(X_k) = w(X) = \max_{1 \leq k \leq n} w(X_k)$

将 $X^{(1)}$ 和 $X^{(2)}$ 放到 $List$ 的末尾。

返回步骤 2。

步骤 4 [输出 $OutputList$]

$List$ 是由一系列的盒子(区间向量)组成的一个表(队列), 初始时只有一个盒子 B 。算法每次对 $List$ 中的第一个盒子进行根存在性判定, 如果 X 内不存在根, 则将 X 丢弃; 如果存在唯一的根, 使用算法 1 进行求解, 得到一个误差界为 ϵ_1 的根; 如果 $K(X^{(1)}) \cap X^{(1)} \neq \emptyset$, 那么令它们的交集为初始测试的盒子; 否则, 将 X 二分后得到的两个子盒子放在 $List$ 的末尾, 进行下一次测试。在算法的执行过程中, 可能测试出多个包含方程组根的小盒子, 每个小盒子都作为算法 1 的输入, 因此存在多个初始点。

4.3 求解凸优化问题的区间算法

由 K-T 条件, 可将凸优化的求解转化为式(2), 它是 $n+l$ 个方程 $n+l$ 个未知数的非线性方程组。如果目标函数和约束函数可微, 则凸优化问题都可转化为非线性方程组的求根问题, 使用算法 2 求出满足精度的一系列小盒子, 它们的并集是可行域中所有全局极小点的集合 x^* 的子集。对于区间宽度小于 ϵ_2 而又无法判定的小盒子, 算法 2 放弃对其的测试。算法 3 给出了基于算法 2 求解凸优化问题的过程。

算法 3 凸优化问题求解的区间算法

输入: 式(1)描述的凸优化问题。

输出: 一系列区间(向量), 每个区间(向量)均包含可行域中的一个全局极小点。

计算步骤:

步骤 1: 利用 K-T 条件将凸优化问题转化为非线性方程组的求根问题。

步骤 2: 调用算法 2 进行求解。

对算法 3 进行时间复杂度分析: 步骤 1 的转化过程仅需求导和赋值操作, 步骤 2 (即算法 2) 是凸优化区间算法效率的关键。在算法 2 的关键操作步骤 3 中, 执行的次数与问题本身包含的根的个数呈线性关系, 如果满足 $K(X) \subset X$, 则调用算法 1 求根。算法 1 的主要操作是矩阵求逆及乘积运算, 并与精度要求 ϵ_1 有关。算法 2 在最好情形下, 调用一次算法 1 得到方程组唯一的一个根, 时间复杂度为 $O(1)$; 在最坏情形下, 进行二分的次数是 $n^{\lceil \log_2 w(B)/\epsilon_2 \rceil}$, 时间复杂度是 $O(n^{\lceil \log_2 w(B)/\epsilon_2 \rceil})$ 。

4.4 关键代码

算法 2 的关键部分是 K 算子的求解, 程序 1 是基于 Matlab 环境, 并使用区间算术软件包 INTLAB 计算 K 算子的子程序, 参数 X 是测试区间, f 是要求解的方程组。

程序 1 计算 $K(X)$

function [result] = Krawczyk_step(X, f)

y = mid(X);

```

n=length(X);
iy=midrad(y,0);
fy=feval(f,iy);
Xg=gradientinit(X);
FXg=feval(f,Xg);
mid(FXg,dx);
Y=inv(mid(FXg,dx));
KX=y-Y*f+(eye(n,n)-Y*FXg,dx)*(X-y);
result=KX;

```

5 数值实验

程序运行的硬件环境为 Intel(R) Core(TM) 2 Quad 2.40GHz处理器和 4GB 内存,操作系统为 Windows7。算法 3 的实现基于 Matlab 7.6.0 (R2008a)环境,区间算术运算部分使用 Toolbox 中的区间算术软件包 INTLAB。

5.1 凸优化问题的一个实例

考虑式(6)的数学规划问题:

$$\begin{cases} \min f(x)=(x-2)^2 \\ g_1(x)=x \geq 0 \\ g_2(x)=6-x \geq 0 \end{cases} \quad (6)$$

目标函数 $f(x)$ 的二阶偏导数大于 0,由凸函数判定的二阶充要条件,可知 $f(x)$ 是凸函数,约束条件 $g_1(x)$ 和 $g_2(x)$ 都是线性的,可以看成是凹函数,所以式(6)为凸优化问题。

图 1 是可行域 $R=\{x|0 \leq x \leq 6\}$ 中的所有点对应的目标函数值 $f(x)$ 的图形,利用图解法或者定性分析可知,式(6)有唯一的全局最优值 $f^*=f(2)=0$,2 是全局最优值。

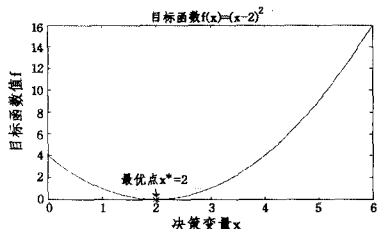


图 1 求解式(6)的图解法

利用 K-T 条件,式(6)转化为 3 个未知数 3 个方程的非线性方程组:

$$\begin{cases} 2(x^*-2)-\gamma_1^*+\gamma_2^*=0 \\ \gamma_1^* x^*=0 \\ \gamma_2^* (6-x^*)=0 \\ \gamma_1^*, \gamma_2^* \geq 0 \end{cases} \quad (7)$$

使用算法 3 对式(7)进行求解,得到解向量 $(x^*, \gamma_1^*, \gamma_2^*)$,其中

$$\begin{aligned} x^* &= [1.9999999999999999, 2.0000000000000001] \\ \gamma_1^* &= [-0.0000000000000001, 0.0000000000000001] \\ \gamma_2^* &= [-0.0000000000000001, 0.0000000000000001] \end{aligned}$$

x^* 的值即为包含凸优化问题(6)的全局极小点的区间包含,此结果表明算法 3 能收敛到理论最优值。

作为比较,本文选取了几个常用的优化算法,包括遗传算法(Genetic Algorithm,GA)、模式搜索法(Pattern Search Algorithm,PSA)、模拟退火算法(Simulated Annealing,SA)分别进行求解。表 1 选取了 GA 和 SA 算法的两个较优的运行结果,并给出了这些算法的执行时间。

表 1 使用算法 3 与 GA、PSA、SA 求解式(6)的结果

算法	x^*	$f(x^*)$	执行时间(s)
GA	1.9999948	2.662958×10^{-9}	0.152
	2.000043	1.870472×10^{-10}	0.155
PSA	2	0	0.245
SA	2.000296	8.789804×10^{-8}	0.575
	2.000133	1.780669×10^{-8}	0.456
算法 3	$[x^*, x^*]$	1×10^{-28}	0.746

其中 $[x^*, x^*] = [1.9999999999999999, 2.0000000000000001]$ 。

针对本节的实例,算法 3 得到的最优解与理论最优解之间的误差比 GA、SA 算法都要小,而且是误差可控的算法;在程序执行时间的比较上,GA 是最快的。凸优化区间算法的执行时间尽管比其它算法的略长,但仅以秒计。相比启发式算法,算法 3 保证具有至少线性的收敛速度,以找到更多的根为核心,以得到精度有保证的可验证解为终止条件,对于根较稀疏的情形特别有效;并且,算法的有效性应主要体现在处理大规模复杂问题时的能力上,5.2 节给出了一个较复杂的应用,算法 3 在此问题上体现出较好的全局寻优的能力。

5.2 机器人上的应用

很多全局优化问题都包含非线性方程组的求解,其中一个应用是机器人操作臂的角度的求解问题。机器人问题的目标是求包含 8 个未知数 8 个方程的非线性方程组 $f(X)=0$ 的根。方程组的解包含 4 对数 $\{(x_{2i-1}, x_{2i})\}_{i=1}^4$,它们表示机器人操作臂的角度(余弦值和正弦值)^[20]。

例: $f: R^8 \rightarrow R^8$

$$\begin{aligned} f_1 &= 0.004371x_1x_3 - 0.3578x_2x_3 - 0.1238x_1 - \\ &\quad 0.001637x_2 - 0.9338x_4 + 1.000x_7 - 0.3571 \\ f_2 &= 0.2238x_1x_3 + 0.7623x_2x_3 + 0.2638x_1 - \\ &\quad 0.07745x_2 - 0.6734x_4 - 0.6022 \\ f_3 &= 1.000x_6x_8 + 0.3578x_1 + 0.004731x_2 \\ f_4 &= -0.7623x_1 + 0.2238x_2 - 0.3461 \\ f_5 &= x_1^2 + x_2^2 - 1 \\ f_6 &= x_3^2 + x_4^2 - 1 \\ f_7 &= x_5^2 + x_6^2 - 1 \\ f_8 &= x_7^2 + x_8^2 - 1 \end{aligned} \quad (8)$$

其中, $x_i \in [-1, 1], 1 \leq i \leq 8$ 。式(8)用向量表示即为 $f(X)=0$,求出非线性方程组的所有根是一个难问题,因此解决有多个根的机器人问题非常有挑战性。

本文采用 3 种方法求解:分别用数学软件 Maple 和 Matlab 内置的非线性方程组求解指令 fsolve 及算法 3。

调用 Maple 内置的 fsolve 命令,给出的结果为:

$$x = (-0.217, -0.976, 0.108, -0.994, -0.994, -0.107, -0.637, -0.771)$$

在 Matlab 的 fsolve 命令中,需要给出初始点,数值实验在可行域内选取了 21 个初值 $X = (j, j, \dots, j)^T, j = 0.1k - 1, k = 0, 1, 2, \dots, 20$ 。Matlab 给出的结果为:

$$\begin{aligned} x^{(1)} &= (-0.217, -0.976, 0.108, -0.994, -0.994, \\ &\quad -0.107, -0.637, -0.771) \\ x^{(2)} &= (-0.971, -0.205, -0.155, -1.031, -0.685, \\ &\quad -0.726, -0.823, -0.503) \\ x^{(3)} &= (0.211, -0.978, 0.258, -0.966, -0.996, \\ &\quad -0.090, -0.611, 0.792) \\ x^{(4)} &= (0.910, 0.414, 0.985, 0.174, 0.855, -0.518, \\ &\quad 0.775, 0.632) \end{aligned}$$

算法 3 给出的结果为:

$$x^{(1)} = \begin{bmatrix} [-0.2174, -0.2173] \\ [-0.9762, -0.9761] \\ [0.1079, 0.1080] \\ [-0.9942, -0.9941] \\ [0.9942, 0.9943] \\ [-0.1069, -0.1068] \\ [-0.6374, -0.6373] \\ [-0.7706, -0.7705] \end{bmatrix}$$

$$x^{(2)} = \begin{bmatrix} [0.2110, 0.2111] \\ [-0.9775, -0.9774] \\ [0.2576, 0.2577] \\ [-0.9663, -0.9662] \\ [0.9959, 0.9960] \\ [-0.0896, -0.0895] \\ [-0.6111, -0.6110] \\ [0.7916, 0.7917] \end{bmatrix}$$

$$x^{(3)} = \begin{bmatrix} [0.9102, 0.9103] \\ [0.4140, 0.4141] \\ [0.9847, 0.9848] \\ [0.1741, 0.1742] \\ [0.8552, 0.8553] \\ [-0.5182, -0.5181] \\ [0.7747, 0.7748] \\ [0.6323, 0.6324] \end{bmatrix}$$

$$x^{(4)} = \begin{bmatrix} [-0.3399, -0.3398] \\ [-0.9405, -0.9404] \\ [-0.9738, -0.9737] \\ [0.2274, 0.2275] \\ [-0.9706, -0.9705] \\ [-0.2408, -0.2407] \\ [0.8520, 0.8521] \\ [-0.5236, -0.5235] \end{bmatrix}$$

$$x^{(5)} = \begin{bmatrix} [0.7525, 0.7526] \\ [0.6585, 0.6586] \\ [-0.2847, -0.2846] \\ [-0.9587, -0.9586] \\ [0.9485, 0.9486] \\ [-0.3167, -0.3166] \\ [-0.5099, -0.5098] \\ [0.8602, 0.8603] \end{bmatrix}$$

$$x^{(6)} = \begin{bmatrix} [0.4903, 0.4904] \\ [-0.8716, -0.8715] \\ [-0.9790, -0.9789] \\ [0.2043, 0.2044] \\ [-0.9058, -0.9057] \\ [-0.4240, -0.4239] \\ [0.9147, 0.9148] \\ [0.4041, 0.4042] \end{bmatrix}$$

从 3 种方法的计算结果可以看出,算法 3 找到的非线性方程组的解的个数是最多的。这表明对于复杂问题,本文提出的区间算法 3 在全球寻优的能力上有一定的优势,能保证不会陷入局部最优。

结束语 凸优化的应用领域非常广泛,因其目标函数和

约束条件的非线性性,对它的求解不像线性规划问题那样有完备的理论,所以设计能有效避免陷入局部最优的算法是解决凸优化问题的一大目标。本文提出的区间迭代算法 3 在整个有界可行域内进行搜索,如果判定给定的区域内包含全局极小点,给出其区间包含,算法与准确全局最小点之间的误差可控,亦不会陷入局部最优。数值实验与遗传算法、模式搜索算法、模拟退火算法和数学软件内置的非线性方程组求解器进行比较,表明区间迭代算法在求解一类凸优化问题上是有效的。在将来的工作中,基于区间算术的方法可推广到更一般的非线性规划问题,如不要求目标函数和约束条件可微及具有凹凸性。

参 考 文 献

- [1] Boyd S P, Vandenberghe L. Convex Optimization [M]. Cambridge university press, 2004
- [2] Goldberg D E. Genetic Algorithms in Search, Optimization, & Machine Learning [M]. Reading Menlo Park: Addison-wesley, 1989
- [3] Kirkpatrick S, Vecchi M P. Optimization by simulated annealing [J]. Science, 1983, 220(4598): 671-680
- [4] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [C]//Proceedings of the First European Conference on Artificial Life, 1991, 142: 134-142
- [5] Hillier F S, Lieberman G J. Introduction to Operations Research (Ninth Edition) [M]. 北京: 清华大学出版社, 2010
- [6] Moore R E. Interval Arithmetic and Automatic Error Analysis in Digital Computing [D]. Stanford: Department of Mathematics, Stanford University, 1962
- [7] Moore R E, Kearfott R B, Cloud M J. Introduction to Interval Analysis [M]. Society for Industrial & Applied Mathematics, 2009
- [8] 张建华. 基于区间数学的全局优化算法及其应用研究 [D]. 合肥: 合肥工业大学, 2012
- [9] Szlama A, Kalauz K, Heckl I, et al. Solving a separation-network synthesis problem by interval global optimization technique [J]. Computers & Chemical Engineering, 2013, 56: 142-154
- [10] Mazhoud I, Hadj-Hamou K, Bignon J, et al. Interval-based global optimization in engineering using model reformulation and constraint propagation [J]. Engineering Applications of Artificial Intelligence, 2012, 25(2): 404-417
- [11] Markót M C, Fernandez J, Casado L G, et al. New interval methods for constrained global optimization [J]. Mathematical Programming, 2006, 106(2): 287-318
- [12] Karmakar S, Mahato S K, Bhunia A K. Interval oriented multi-section techniques for global optimization [J]. Journal of Computational and Applied Mathematics, 2009, 224(2): 476-491
- [13] Byrne R P, Bogle I D L. Global optimisation of constrained non-convex programs using reformulation and interval analysis [J]. Computers & Chemical Engineering, 1999, 23(9): 1341-1350
- [14] Kearfott R B, Dawande M, Du K, et al. Algorithm 737: Intlib—a portable fortran 77 interval standard-function library [J]. ACM Transactions on Mathematical Software (TOMS), 1995, 20(4): 447-459
- [15] Kearfott R B. Algorithm 763: INTERVAL_ARITHMETIC: A Fortran 90 module for an interval data type [J]. ACM Transactions on Mathematical Software (TOMS), 1996, 22(4): 385-392
- [16] Schulte M J, Zelov V, Akkas A, et al. The interval-enhanced GNU Fortran compiler [J]. Reliable Computing, 1999, 5(3):

- [17] 《运筹学》教材编写组. 运筹学(第四版)[M]. 北京:清华大学出版社,2012
- [18] Moore R E. A test for existence of solutions to nonlinear systems[J]. SIAM Journal on Numerical Analysis, 1977, 14(4): 611-615
- (上接第 227 页)
- [6] Li Hui, Ding Shi-fei. Research of Individual Neural Network Generation and Ensemble Algorithm Based on Quotient Space Granularity Clustering[J]. Applied Mathematics & Information Sciences, 2013, 7(2): 701-708
- [7] Xu Xin-zheng, Ding Shi-fei, Jia Wei-kuan, et al. Research of assembling optimized classification algorithm by neural network based on Ordinary Least Squares (OLS) [J]. Neural Comput & Applic, 2013(22): 187-193
- [8] Schapire R E. The strength of weak learnability[J]. Machine Learning, 1990, 5(2): 197-227
- [9] Breiman L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140
- [10] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976
- [11] Liu Xiao-yong, Fu Hui. A fast affinity propagation clustering algorithm[J]. Journal of Shandong University (Engineering Science), 2011, 41(4): 20-28
- [12] Xia Ding-yin, Wu Fei, Zhang Xu-qing, et al. Local and global approaches of affinity propagation clustering for large scale data [J]. Journal of Zhejiang University-Science A, 2008, 9(10): 1373-1381
- [13] Lazic N, Givoni I, Aarabi P, et al. FLoSS: Facility location for subspace segmentation [C] // Proceedings of 12th International Conference on Computer Vision (ICCV). Kyoto: IEEE Press, 2009: 825-832
- [14] Lazic N, Frey B J, Aarabi P. Solving the uncapacitated facility location problem using message passing algorithms [C] // Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS). Sardinia: Microtome Publishing, 2010: 429-436
- [15] Tang Dong-ming, Zhu Qing-xin, Yang Fan, et al. Solving large-scale location problem using affinity propagation clustering[J]. Application Research of Computers, 2010, 27(3): 841-844
- [16] Dueck D, Frey B J. Nonmetric affinity propagation for unsupervised image categorization [C] // Proceedings of 11th International Conference on Computer Vision (ICCV). Riode Janeiro: IEEE Press, 2007: 1-8
- [17] Zhang Ren-yan, Zhao Hong-liang, Lu Xiao, et al. Grey image segmentation method based on affinity propagation clustering [J]. Journal of Naval University of Engineering, 2009, 21(3): 33-37
- [18] Givoni I E, Frey B J. Semi-supervised affinity propagation with instance-level constraints [C] // Proceedings of 12th International Conference on Artificial Intelligence and Statistics (AISTATS). Florida: Microtome Publishing, 2009: 161-168
- [19] Guan Ren-chu, Pei Zhi-li, Shi Xiao-hu, et al. Weight affinity propagation and its application to text clustering [J]. Journal of Computer Research and Development, 2010, 47(10): 1733-1740
- [20] Dueck D, Frey B J, Jojic N, et al. Constructing treatment portfolios using affinity propagation [C] // Proceedings of International Conference on Research in Computational Molecular Biology (RECOMB). Singapore: Springer, 2008: 360-371
- [21] Xu Wen-zhu, Xu Li-hong. Adaptive key-frame extraction based on affinity propagation clustering [J]. Computer Science, 2010(1): 268-270
- [22] Xiang Pei-su. New CBIR system based on the affinity propagation clustering algorithm [J]. Journal of Southwest University for Nationalities: Natural Science Edition, 2010, 36(4): 624-627
- [23] Wang Kai-jun, Zhang Jun-ying, Li Dan, et al. Adaptive affinity propagation clustering [J]. Acta Automatica Sinica, 2007, 33(12): 1242-1246
- [24] Wang Kai-jun, Li Jian, Zhang Jun-ying, et al. Semi-supervised affinity propagation clustering [J]. Computer Engineering, 2007, 33(23): 197-198, 201
- [25] Wang Kai-jun, Zheng Jie. Fast algorithm of affinity propagation clustering under given number of clusters [J]. Computer Systems & Applications, 2010, 19(7): 207-209
- [26] Xie Xin-xi, Wang Shi-tong. Affinity propagation clustering for symbolic interval data based on mutual distance [J]. Computer Application, 2008, 28(6): 1441-1443
- [27] Xiao Yu, Yu Jian. Semi-Supervised clustering based on affinity propagation algorithm [J]. Journal of Software, 2008, 19(11): 2803-2813
- [28] Gu Rui-jun, Wang Jia-cai, Chen Geng, et al. Affinity propagation clustering for large scale dataset [J]. Computer Engineering, 2010, 36(23): 22-24
- [29] Dong Jun, Wang Suo-ping, Xiong Fan-lun. Affinity propagation clustering based on variable-similarity measure [J]. Journal of Electronics & Information Technology, 2010, 32(3): 509-514
- [30] Li Ya-qin, Yang Hui-zhong. Multi-model modeling method based on affinity propagation clustering and Gaussian processes [J]. Computers and Applied Chemistry, 2010, 27(1): 51-54
- [31] Frey B J, Dueck D. Clustering by passing messages between data points [J]. Science, 2007, 315(5814): 972-976
- [32] Lin J, Zhu B Z. Improved Principal Component Analysis and Neural Network Ensemble Based Economic Forecasting [J]. Lecture Notes in Computer Science, 2006, 4113: 135-145
- [33] Lin Jian, Peng Min-jing. GDP forecasting model based on neural networks ensemble [J]. Chinese Journal of Management, 2005, 2(4): 434-436
- [34] Lin Jian, Zhu Bang-zhu. Neural network ensemble based on forecasting effective measure and its application [J]. Journal of Computational Information Systems, 2005, 1(4): 781-787
- [35] Perrone M P, Cooper L N. When Networks Disagree: Ensemble method for neural networks [C] // Mammone R J, eds. Artificial Neural Networks for Speed and Vision. New Yorks: Chapman & Hall, 1993: 126-142
- [36] Li Ling-ling, Liu Xi-yu, Lu Shu-qiang. Constructive methods for parallel learning neural network ensemble based on particle swarm optimization [J]. ShanDong Science, 2007, 20(4): 16-20
- [37] Opitz D, Vedelsby J. Actively searching for an efficient neural network ensemble [J]. Connection Science, 1996, 8(3/4): 337-353