



非线性规划 U-D 分解方法及其在神经网络训练中的应用

史 忠 科

(西北工业大学自控系 西安 710072)

摘 要

提出一种有效的 U-D 分解 DFP 和 BFGS 算法。该算法解决了 H 阵的正定性问题,保证了算法的数值稳定性,并大大提高了计算效率。对 H 阵的计算量分析表明,该算法的计算效率比普通方法高 20%,比普通平方根方法高 $0.4n$ (n 为 H 阵维数) 倍。神经网络训练的应用表明,新算法比普通 DFP 和 BFGS 方法更有效、更准确。

关键词: 非线性规划,大型规划问题,神经网络,学习算法,无约束优化。

1 引言

非线性规划在管理工程、自动控制等领域有着极为广泛地应用。变尺度法是非线性规划的一种有效方法。自从 1959 年 Davidon W C 提出这一方法后, Fletcher R 和 Powell M J D 将其发展与修正,成为一种成熟的算法。目前,变尺度方法已发展成为一大类算法,并被认为是解决无约束优化问题的最有效算法之一。特别是 BFGS (Broyden-Fletcher-Goldfarb-Shannon) 方法,由于算法的稳定性和超线性收敛特性,而被当作目前最好的一种算法,并得到了极为广泛地应用^[1]。

然而,标准的 BFGS 方法在求解较大型的规划问题时,常常效率很低或不能得到实用的结果^[1,2]。为了解决这一问题, Powell, Griewank 及 Toint 等人进行了一系列研究^[1,4],但都没有从根本上解决数值稳定性对大型规划结果的影响。本文采用一种新的 U-D 分解方法,从数值稳定性的角度解决大型规划所存在的问题,并提高整个规划算法的计算效率。

2 变尺度方法的稳定结构

多变量函数

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) \quad (1)$$

的极值常常使用变尺度方法计算。在各种变尺度方法中,其差别仅在于 Hessian 矩阵之

逆阵的迭代计算式不同,其中以 DFP 和 BFGS 方法最为有名.

在 DFP 算法中, H_k 的迭代计算式为^[1]

$$H_{k+1} = H_k + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T g_k} - \frac{H_k g_k g_k^T H_k}{g_k^T H_k g_k}. \quad (2)$$

式中

$$\Delta x_k = x_{k+1} - x_k; \quad (3)$$

$$g_k = G_{k+1} - G_k; \quad (4)$$

G_k 为 $f(x)$ 在 $x = x_k$ 处的梯度.

在 BFGS 算法中, H_k 的迭代计算式为^[1]

$$\left. \begin{aligned} H_{k+1} &= H_k + (b_k \Delta x_k \Delta x_k^T - H_k g_k \Delta x_k^T - \Delta x_k g_k^T H_k) / a_k, \\ b_k &= 1 + g_k^T H_k g_k / a_k. \end{aligned} \right\} \quad (5)$$

式中 $a_k = \Delta x_k^T g_k$.

在(2),(5)两式中,均存在矩阵的相减运算,且要求 H_k 为正定矩阵.上述算法在接近最优值处难以保证收敛性,特别是求解大型非线性规划时容易出现计算发散.为了解决这一问题,容易得出(2),(5)式的稳定结构.

DFP 算法的稳定结构为

$$H_{k+1} = A_k H_k A_k^T + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T g_k}, \quad (6)$$

式中

$$A_k = I - \frac{H_k g_k g_k^T}{g_k^T H_k g_k}.$$

BFGS 算法的稳定结构为

$$H_{k+1} = B_k H_k B_k^T + \Delta x_k \Delta x_k^T / a_k, \quad (7)$$

其中 $B_k = I - \Delta x_k g_k^T / a_k$.

(6),(7)两式均可用平方根方法计算,可以保证计算稳定性,但计算量太大,为 n^3 数量级^[5].为此,本文给出下面的递推方法.

3 递推 U-D 分解计算方法

为了便于研究,将(2)式写为

$$H_{k+1/k} = H_k - \frac{H_k g_k g_k^T H_k}{g_k^T H_k g_k}, \quad (8)$$

$$H_{k+1} = H_{k+1/k} + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T g_k^T}. \quad (9)$$

令

$$UDU^T = H_{k+1/k}, \quad U_{k+1} D_{k+1} U_{k+1}^T = H_{k+1}. \quad (10)$$

其中 U 为单位上三角阵, D 为对角阵.(8)式采用 Bierman 方法^[6]计算,(9)式可由 Thornton-Jacobson^[6]方法得到.只要 $\Delta x_k \neq 0, g_k \neq 0$,则 D_{k+1} 的对角元素均大于零,

这样就保证了 DFP 方法的数值稳定性,而且计算量为 n^2 数量级^[6]。

对于 BFGS 算法,这里采用下述方法处理。由(5)式可得

$$H_{k+1} = H_k + (\sqrt{b_k} \Delta x_k - H_k g_k / \sqrt{b_k})(\sqrt{b_k} \Delta x_k^T - g_k^T H_k / \sqrt{b_k}) / a_k - \frac{H_k g_k g_k^T H_k}{a_k b_k}, \quad (11)$$

而

$$a_k b_k = a_k + g_k^T H_k g_k. \quad (12)$$

这样,可将(11)式写成

$$H_{k+1/k} = H_k - \frac{H_k g_k g_k^T H_k}{a_k + g_k^T H_k g_k}, \quad (13)$$

$$H_{k+1} = H_{k+1/k} + (\Delta x_k - H_k g_k / b_k)(\Delta x_k^T - g_k^T H_k / b_k) \frac{b_k}{a_k}. \quad (14)$$

(13),(14) 式可分别按 Bierman 和 Thornton-Jacobson 方法计算^[6]。只要 $\Delta x_k \neq 0$, $\Delta g_k \neq 0$, BFGS 算法是数值稳定的,且计算量为 n^2 数量级。

4 计算效率分析

为了说明本文提出的 U-D 分解递推算法的有效性,下面对所列举方法的计算量进行分析比较。因为这里涉及的算法只有 H_{k+1} 的计算式不同,所以在计算量和存贮量比较时,仅考虑 H_{k+1} 。表 1 列出各算法关于 H_{k+1} 计算量的比较。表中 MDFP 指采用 MWG-S 正交化进行 U-D 分解的 DFP 算法,即(6)式;MBFGS 指采用 MWG-S 方法进行 U-D 分解的 BFGS 算法,即(7)式;RDFP 指(8),(9)式表示的递推 U-D 分解 DFP 算法;RBFSG 指(13),(14)式表示的递推 U-D 分解 BFGS 算法。

表 1 H_{k+1} 计算量的比较

算 法	加 法	乘 法	除 法
DFP	$3n^2 + n - 2$	$3n^2 + 2n$	$2n$
BFGS	$4n^2 + 2n - 2$	$3n^3 + 3n$	$2n$
MDFP	$n^3 + n^2 + n - 1$	$n^3 + 2n^2 - n$	$2n - 1$
MBFGS	$n^3 + 2n - 1$	$n^3 + n^2$	$2n - 1$
RDFP	$2.5n^2 + 2.5n - 1$	$2.5n^2 + 8.5n - 1$	$3n - 2$
RBFSG	$2.5n^2 + 3.5n - 1$	$2.5n^2 + 8.5n - 1$	$4n - 1$

5 在神经网络训练中的应用

若飞机降落时采用神经网络控制,舵偏角 δ_{zk} 的描述为

$$\delta_{zk} = \sum_{i=1}^3 p_i \sum_{j=1}^3 w_{ij} q_j \sum_{l=1}^3 r_{il} u_{il}, \quad (15)$$

其中

$$w_{ij} = \begin{cases} 0 & \text{不选时,} \\ 1 & \text{选取时;} \end{cases}$$

p_i, q_j, r_l 为待定权系数。

$$\begin{aligned} u_{11} &= \omega_z(k), u_{12} = \omega_z(k) - \omega_z(k-1), \\ u_{13} &= \omega_z(k) - 2\omega_z(k-1) + \omega_z(k-2), \\ u_{21} &= \alpha(k), u_{22} = \alpha(k) - \alpha(k-1), \\ u_{23} &= \alpha(k) - 2\alpha(k-1) + \alpha(k-2), \\ u_{31} &= \theta(k), u_{32} = \theta(k) - \theta(k-1), \\ u_{33} &= \theta(k) - 2\theta(k-1) + \theta(k-2). \end{aligned}$$

其中 α 为气流迎角, ω_z 为俯仰角速度, θ 为俯仰角, δ_z 为升降舵舵偏角。

着陆控制结果如图 1 所示。

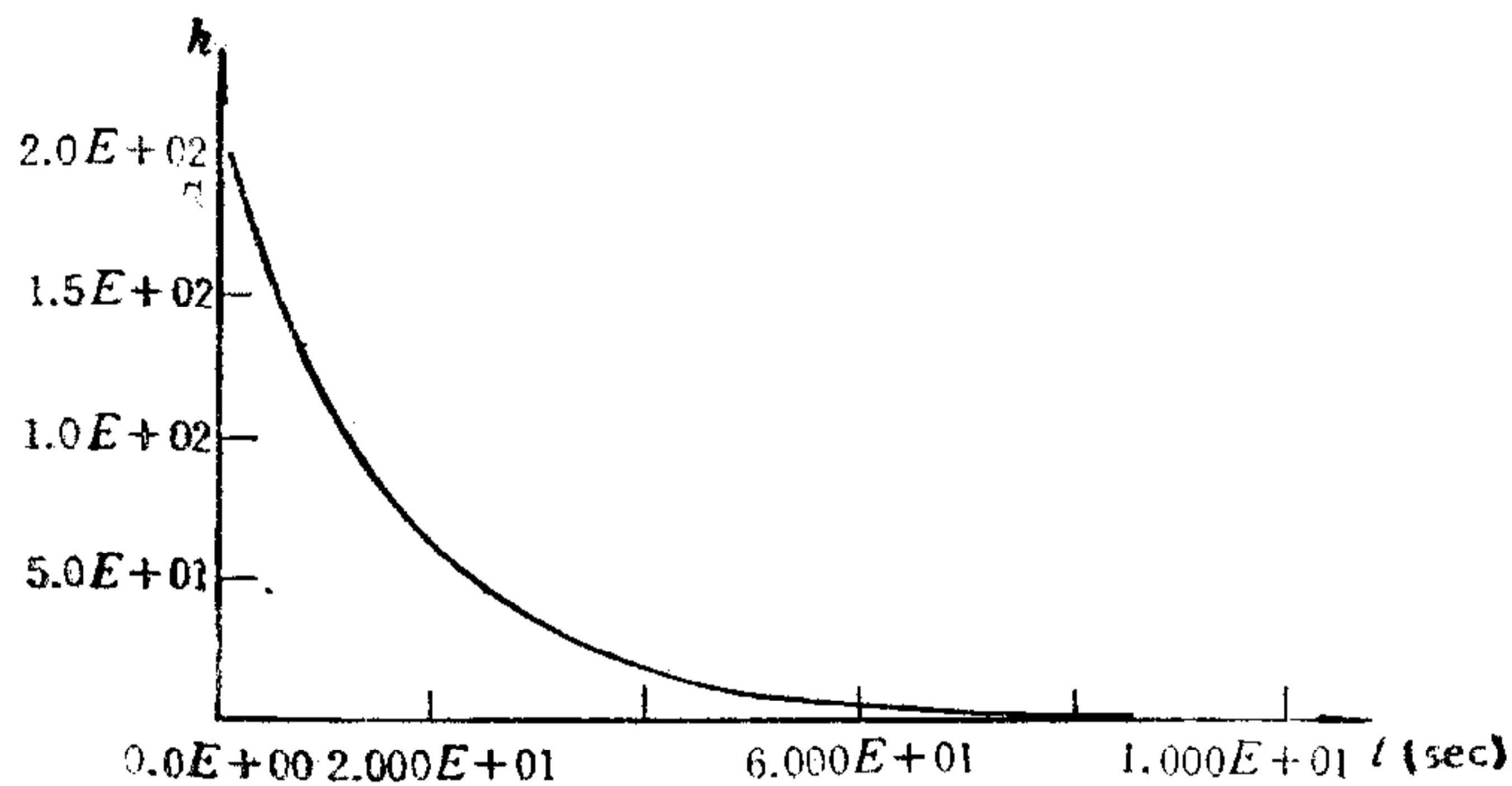


图 1 飞机着陆控制结果

6 结束语

本文给出的递推算法不仅数值稳定性好,而且精度和计算效率高。对于 $n \geq 100$ 时,计算效率提高 20%,比普通平方根方法高 $0.4n$ 倍。因此,该方法在求解非线性规划问题及神经网络的训练中,可以推广使用。当未知参数的维数越高时,其优越性更明显。

参 考 文 献

- [1] Toint Ph L. Global convergence of the partitioned BFGS algorithm for convex partially separable optimization. *Mathematical Programming*, 1986, **36**(3): 290—306.
- [2] Powell M J D. How bad are the BFGS and DFP methods when the objective function is quadratic? *Mathematical Programming*, 1986, **34**(1): 34—47.
- [3] Powell M J D. On the global convergence of trust region algorithms for unconstrained minimization. *Mathematical Programming*, 1984, **29**(3): 297—303.
- [4] Chamberlain R M et al. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Studies*, 1982, **16**(1): 1—17.
- [5] Mayback P S. *Stochastic models Estimation and Control*, New York: Academic Press, 1979, 368—405.
- [6] 史忠科. 固定区间平滑新算法及其在飞行试验中的应用. *自动化学报*, 1991, **17**(3): 323—329.

U-D FACTORIZATION-BASED NONLINEAR PROGRAMMING METHOD AND ITS APPLICATION IN NEURAL NETWORK TRAINING

SHI ZHONGKE

(Department of Automatic Control, Northwestern Polytechnical University Xi'an 710072)

ABSTRACT

To solve convergence rate problems of often used DFP and BFCS methods, the stable construction of inverse Hessian matrix are presented. To get high numerical stability and computational efficiency, U-D factorization-based DFP and BFGS algorithms are developed. In the new methods the positive definiteness of the inverse matrix H is ensured and both the stability and convergence of the algorithm is improved. By using rank-one U-D factorization updates of H , the numerical accuracy and efficiency are increased. Operational counts for computing H show that the efficiency of the new algorithm is increased by 20% and the storages of matrix H is reduced by 50%. Results of several numerical example show that the optimization problems can be solved by using the programming methods presented in this paper and accurate results may be obtained.

Key words: Nonlinear programming, large scale problem, neural network, learning algorithm, unconstrained optimization.