

一类改进的 BFGS 拟牛顿法及与其他几种拟牛顿法的比较研究

杨 茜¹, 吴泽忠¹, 贺盛瑜²

(1. 成都信息工程大学应用数学学院, 四川 成都 610225; 2. 西昌学院, 四川 西昌 615000)

摘要: 拟牛顿法是最优化中一种重要的求解无约束问题的方法, 对拟牛顿算法的改进是一个重点研究内容。基于新拟牛顿方程 $\mathbf{B}_{k+1} \cdot \mathbf{s}^{(k)} = (\mathbf{y}^{(k)})^*$ 提出一个新的改进的 BFGS 算法, 结合以往几种改进的 BFGS 算法, 采用 Wolfe 线搜索准则对迭代步长进行搜索, 并通过选取一些测试函数利用 MATLAB 工具对这几种改进的 BFGS 拟牛顿法的收敛效果进行对比分析。实验结果表明, 提出的改进的 BFGS 算法相较于以往几种改进的 BFGS 算法, 收敛效果更好。

关键词: 无约束最优化; BFGS 拟牛顿法; 拟牛顿方程; Wolfe 线搜索准则

中图分类号: O221.2

文献标志码: A

doi: 10.16836/j.cnki.jcui.2023.02.015

0 引言

拟牛顿法是一种求解非线性优化问题方法^[1-4], 最早由美国 ARGONNE 国家实验室的物理学家 DAVI-DON 在 20 世纪 50 年代中期提出, 随后又由 FLETCH-ER 和 POWELL 不断改进和完善使其成为非线性规划问题研究的焦点。因为牛顿法需要计算 HESSIAN 矩阵的值及其逆矩阵, 应用牛顿法将产生巨大的计算量。因此拟牛顿法的本质思想即想要改善牛顿法这一缺陷, 它使用正定矩阵来近似 HESSIAN 矩阵的逆, 从而简化了运算的复杂度。拟牛顿法和最速下降法一样只需要求出每一步迭代时知道目标函数的梯度, 通过测量梯度的变化, 构造一个目标函数的模型使之足以产生超线性收敛性。另外, 因为拟牛顿法不需要二阶导数的信息, 所以有时比牛顿法更有效。

拟牛顿法有许多具体的方法, 其中 BFGS 算法是一种重要的拟牛顿法。对于该算法的研究有大量的文献, 通过文献可发现研究的重点大多是对 BFGS 算法进行修正改进从而得出新的 BFGS 拟牛顿算法。其中 Wei 等^[2]于 2006 年在国外一些学者的研究基础上提出了一类新的改进的拟牛顿方程, 即将拟牛顿条件中的 $\mathbf{y}^{(k)}$ 用 $(\mathbf{y}^{(k)})^*$ 替代, 其中 $(\mathbf{y}^{(k)})^* = \mathbf{y}^{(k)} + \mathbf{A}_k \mathbf{s}^{(k)}$, 从而得出一类新的 BFGS 算法。文献[2]提出了 3 种对 \mathbf{A}_k 选取的不同方法并证明了算法具有全局收敛性和超线性收敛性。在此基础上文献[5]对新拟牛顿方程中的

\mathbf{A}_k 进行改进。受以上研究的启发, 本文对文献[2]所提出的新拟牛顿方程中的 \mathbf{A}_k 进行改进, 进而提出一种新的 BFGS 算法, 并将这个新的 BFGS 算法与文献[2]和文献[5]提出的 BFGS 算法进行数值实验, 比较这几种改进的 BFGS 算法的收敛效果。

1 预备知识

考虑如下无约束的极小化问题:

$$\min f(\mathbf{x}) \quad (1)$$

其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbf{R}^n$, $f(\mathbf{x})$ 是 \mathbf{R}^n 上的一个连续可微函数。其迭代格式为

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)} \quad (2)$$

$$\mathbf{d}^{(k)} = -\mathbf{H}_k \mathbf{g}^{(k)} \quad (3)$$

其中, $\mathbf{d}^{(k)}$ 为 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 处的搜索方向, $\mathbf{g}^{(k)}$ 为 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 处的梯度, \mathbf{H}_k 为 HESSIAN 矩阵的近似, λ_k 为搜索步长。

1.1 拟牛顿条件

设函数经过 $k+1$ 次迭代后得到 $\mathbf{x}^{(k+1)}$, 再将目标函数 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k+1)}$ 附近作二阶泰勒展开, 取近似, 得到

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}^{(k+1)}) + \nabla f(\mathbf{x}^{(k+1)})^T \cdot (\mathbf{x} - \mathbf{x}^{(k+1)}) + \\ &\quad \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k+1)})^T \cdot \nabla^2 f(\mathbf{x}^{(k+1)}) \cdot (\mathbf{x} - \mathbf{x}^{(k+1)}) + \\ &\quad o(\|\mathbf{x} - \mathbf{x}^{(k+1)}\|^2) \end{aligned} \quad (4)$$

从而近似地有

$$\begin{aligned} f(\mathbf{x}) &\approx f(\mathbf{x}^{(k+1)}) + \nabla f(\mathbf{x}^{(k+1)})^T \cdot (\mathbf{x} - \mathbf{x}^{(k+1)}) + \\ &\quad \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k+1)})^T \cdot \nabla^2 f(\mathbf{x}^{(k+1)}) \cdot (\mathbf{x} - \mathbf{x}^{(k+1)}) \end{aligned} \quad (5)$$

在等式两端作用梯度算子 ∇ ,于是有

$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{(k+1)}) + \mathbf{H}_{k+1} \cdot (\mathbf{x} - \mathbf{x}^{(k+1)}) \quad (6)$$

在式(6)中令 $\mathbf{x} = \mathbf{x}^{(k)}$,并整理,可得

$$\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} \approx \mathbf{H}_{k+1} \cdot (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) \quad (7)$$

若引入记号

$$\mathbf{s}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \mathbf{y}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} \quad (8)$$

则式(7)可写作

$$\mathbf{y}^{(k)} \approx \mathbf{H}_{k+1} \cdot \mathbf{s}^{(k)} \quad (9)$$

则称式(9)为拟牛顿条件,因此,以 \mathbf{B}_{k+1} 对 \mathbf{H}_{k+1} 做近似,即

$$\mathbf{y}^{(k)} = \mathbf{B}_{k+1} \cdot \mathbf{s}^{(k)} \quad (10)$$

1.2 BFGS 拟牛顿法

BFGS 算法是由 Broyden、Fletcher、Goldfarb、Shanno 在 1970 年各自独立研究所提出的拟牛顿法,是目前最流行也是最有效的拟牛顿算法。BFGS 算法的核心思想是通过迭代用 \mathbf{B}_{k+1} 对 HESSIAN 矩阵 \mathbf{H}_{k+1} 做近似逼近,其迭代格式:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}^{(k)} (\mathbf{s}^{(k)})^T \mathbf{B}_k}{(\mathbf{s}^{(k)})^T \mathbf{B}_k \mathbf{s}^{(k)}} + \frac{\mathbf{y}^{(k)} (\mathbf{y}^{(k)})^T}{\mathbf{y}^{(k)} \mathbf{s}^{(k)}} \quad (11)$$

1.3 Wolfe 线搜索准则

对迭代步长的搜索分为精确线搜索^[6-7]和非精确线搜索。在实际计算过程中,采用精确线性搜索的计算将会产生巨大计算量,且从另一角度讲,优化的目的是求目标函数在整个区域中的最优值点,把主要精力集中于某个方向上的精确线搜索似乎没有必要。因此通常都会采用非精确步长搜索规则,其目的是使目标函数的下降量在每一次迭代时都达到最大。

Armijo 线搜索准则是最早的非线性搜索条件,但是采用 Armijo 线搜索准则会在迭代过程中出现较小的步长,且在后续的迭代中很可能连续出现很多的小步长。而 Wolfe 准则是由 Armijo 条件和曲线条件所组成,其中 Armijo 条件是使 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k+1)}$ 处的函数值比在 $\mathbf{x}^{(k)}$ 处的函数值小,曲线条件则是拒绝掉满足 Armijo 条件的小步长。

采用 Wolfe 线搜索准则对迭代格式 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)}$ 中的步长 λ_k 进行不精确搜索, Wolfe 线搜索准则如下:

$$f(\mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \sigma_1 \lambda_k (\mathbf{g}^{(k)})^T \mathbf{d}^{(k)} \quad (12)$$

$$\nabla f(\mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)})^T \mathbf{d}^{(k)} \geq \sigma_2 (\mathbf{g}^{(k)})^T \mathbf{d}^{(k)} \quad (13)$$

式(12)为 Armijo 条件,式(13)即为曲线条件,常数 σ_1, σ_2 满足 $0 < \sigma_1 < \frac{1}{2}, \sigma_1 < \sigma_2 < 1$ 。通过数值实验表明,由 Wolfe 准则进行线搜索得到的步长 λ_k 具有较好的性质。

2 几种改进的拟牛顿算法

2.1 改进的拟牛顿算法研究

标准的 BFGS 算法是一个传统的有效的拟牛顿算法,许多研究者对 BFGS 算法进行了改进^[8-11]。一方面对算法中的步长规则进行研究;另一方面,对拟牛顿条件改进,得到一些新的 BFGS 算法。由于本文采用 Wolfe 线搜索准则,因此不再对其他的步长规则研究作叙述,将主要分析讨论以下几种对拟牛顿条件作改进的研究。

Wei 等^[2]对传统的拟牛顿方程进行修正提出一种新的拟牛顿方程,即 $\mathbf{B}_{k+1} \cdot \mathbf{s}^{(k)} = (\mathbf{y}^{(k)})^*$,其中 $(\mathbf{y}^{(k)})^* = \mathbf{y}^{(k)} + \mathbf{A}_k \mathbf{s}^{(k)}$, \mathbf{A}_k 是一个对称正定矩阵,并提出了一种新的 BFGS 校正公式:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}^{(k)} (\mathbf{s}^{(k)})^T \mathbf{B}_k}{(\mathbf{s}^{(k)})^T \mathbf{B}_k \mathbf{s}^{(k)}} + \frac{(\mathbf{y}^{(k)})^* ((\mathbf{y}^{(k)})^*)^T}{(\mathbf{s}^{(k)})^T (\mathbf{y}^{(k)})^*}$$

文献[2]对这一类新的 BFGS 算法给出了全局收敛性证明,并给出了 3 种有关 \mathbf{A}_k 取法:

$$(i) \mathbf{A}_k = \mathbf{A}_k(1) = m_k \mathbf{I}$$

$$(ii) \mathbf{A}_k = \mathbf{A}_k(2) = \frac{\mathbf{v}_k}{\|\mathbf{s}^{(k)}\|^2} \mathbf{I}$$

$$(iii) \mathbf{A}_k = \mathbf{A}_k(3) = \frac{\mathbf{v}_k}{((\mathbf{s}^{(k)})^T \mathbf{y}^{(k)})^2} (\mathbf{y}^{(k)} (\mathbf{y}^{(k)})^T)$$

其中 $m_k = O(\|\mathbf{g}^{(k)}\|)$ 或者一个正数, $\mathbf{v}_k = 2[f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)})] + (\mathbf{g}^{(k)} + \mathbf{g}^{(k+1)})^T \mathbf{s}^{(k)}$ 。

陈奎琳^[5]在 Wei 等^[2]的基础上对拟牛顿方程中的参数进行 \mathbf{A}_k 改进,得到一种新的 BFGS 算法,即

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}^{(k)} (\mathbf{s}^{(k)})^T \mathbf{B}_k}{(\mathbf{s}^{(k)})^T \mathbf{B}_k \mathbf{s}^{(k)}} + \frac{(\mathbf{y}^{(k)})^* ((\mathbf{y}^{(k)})^*)^T}{(\mathbf{s}^{(k)})^T (\mathbf{y}^{(k)})^*}$$

其中 $(\mathbf{y}^{(k)})^* = \mathbf{y}^{(k)} + \mathbf{A}_k \mathbf{s}^{(k)}$, $\mathbf{A}_k = \|\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}\| \cdot \mathbf{I}$ 。

文献[5]对这一类新算法的收敛性给出了证明。

本文则将在 Wei 等^[2]和陈奎琳^[5]的基础上对新拟牛顿方程中的对称正定矩阵 \mathbf{A}_k 提出新的选取方法,从而得出一种新的 BFGS 算法如下:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}^{(k)} (\mathbf{s}^{(k)})^T \mathbf{B}_k}{(\mathbf{s}^{(k)})^T \mathbf{B}_k \mathbf{s}^{(k)}} + \frac{(\mathbf{y}^{(k)})^* ((\mathbf{y}^{(k)})^*)^T}{(\mathbf{s}^{(k)})^T (\mathbf{y}^{(k)})^*}$$

其中 $(\mathbf{y}^{(k)})^* = \mathbf{y}^{(k)} + \mathbf{A}_k \mathbf{s}^{(k)}$, $\mathbf{A}_k = \|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \cdot \mathbf{I}$,不难发现当 k 充分大时, $\mathbf{s}^{(k)} \rightarrow 0$,且 $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \rightarrow 0$,则新的拟牛顿方程充分接近原始的拟牛顿方程。

以上几种研究都是对拟牛顿条件做改进得到的新的 BFGS 算法,而文献[2]和文献[5]均未对提出的改进的 BFGS 算法做数值实验,因此本文将针对以上几种改进的 BFGS 算法做好数值实验,并进一步比较分析。

2.2 改进的 BFGS 算法步骤

Step1 给定初始点 $\mathbf{x}_0 \in \mathbf{R}^n$ 和初始对称正定矩阵

$B_0=I$,取终止标量 $\varepsilon>0$,令 $k=0$ 。

Step2 计算 $\mathbf{g}^{(k)}$ 。如果 $\|\mathbf{g}^{(k)}\| \leq \varepsilon$,则停止计算,输出 $\mathbf{x}^{(k)}$ 作为 $f(\mathbf{x})$ 极小点,否则,继续下一步。

Step3 解线性方程组 $B_k \mathbf{d}^{(k)} = -\mathbf{g}^{(k)}$,求得解 $\mathbf{d}^{(k)}$ 。

Step4 令 $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$,利用 Wolfe 线搜索准则求解搜索步长 λ_k ,给定 $\sigma_1 \in (0, \frac{1}{2}), \sigma_2 \in (\sigma_1, 1)$,若式(13)和式(14)同时成立则取 $\lambda_k = 1$,否则取 $\lambda_k > 0$ 满足式(13)和式(14)。

Step5 计算 A_k 和 $(\mathbf{y}^{(k)})^* = \mathbf{y}^{(k)} + A_k \mathbf{s}^{(k)}$,再将 $(\mathbf{y}^{(k)})^*$ 带入新的 BFGS 校正公式修正 B_k 得到 B_{k+1} 。

Step6 置 $k=k+1$,转 Step2。

2.3 改进的 BFGS 算法流程图

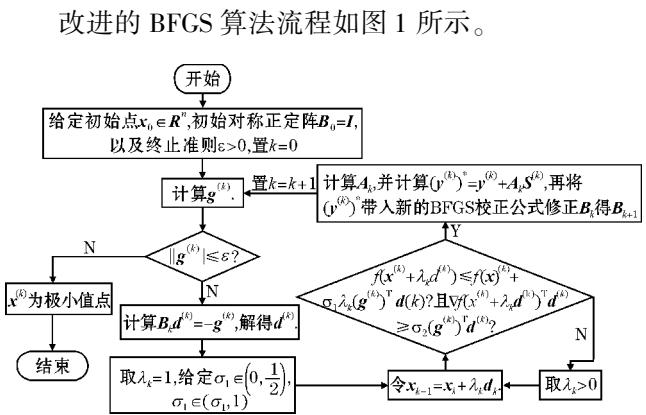


图 1 改进的 BFGS 算法流程图

3 数值实验

此次计算实验是基于 MATLAB (R2017b), 采用 Wolfe 线搜索准则,通过给定初始点,对一些函数实例分别使用以上几种改进的 BFGS 算法进行迭代求解最小值,并计算迭代过程中的迭代次数 k ,迭代最优解 $\mathbf{x}^{(k)}$,以及迭代函数最小值 val。并对这几种改进算法的迭代过程和迭代结果进行比较分析。为了准确分析比较这几种改进的拟牛顿算法的收敛速度,将对每一个实例取相同的参数和初始点进行迭代。为方便对各种改进的 BFGS 算法进行对比,将文献[2]、文献[5]中改进的 BFGS 算法和本文所提出的新的 BFGS 算法分别记为算法 1~3。

3.1 考虑二次函数无约束优化问题

实例 1 $\min f(\mathbf{x}) = \sum_{i=2}^n ix_i^2$

该函数为 Sum squares 测试函数,设置精度为 10^{-10} ,对于这个二次函数无约束优化问题,将分别对其取 n 个相同的值作为初始值。为了对不同的拟牛顿法的进行收敛性分析,分别对其取不同维度进行迭代。因此选取初始点为 $\mathbf{x}_0 = (1, 1, \dots, 1)^T$,分别选取 $n=3, n=10, n=20$ 和 $n=50$ 进行迭代,得到迭代计算过程见表 1。

表 1 实例 1 各种改进的 BGFS 算法的迭代计算结果

迭代采用算法	n	k	最优解 $\mathbf{x}^{(k)} (\times 10^{-13})$	最小值 val $(\times 10^{-23})$
算法 1	3	8	$(-0.5694, 1.3771)^T$	0.6337788
	10	16	$(-0.1319, 0.1163, 0.1950, -0.0385, \dots, -0.1608, 0.0564, 0.0593, -0.0905)^T$	0.0638036
	20	28	$(0.0073, 0.0045, 0.0023, -0.0129, \dots, -0.0080, -0.0014, -0.0028, -0.0019)^T$	0.0010466
	50	58	$(0.0044, -0.0048, -0.0053, 0.0117, \dots, -0.0005, -0.0007, -0.0001, 0.0013)^T$	1.2365456
算法 2	3	12	$(-0.2586, -0.5124)^T$	0.0921240
	10	23	$(-0.4768, 0.6399, -0.1615, -0.2813, \dots, -0.0114, -0.1530, -0.1021, -0.1779)^T$	0.2848355
	20	38	$(0.7096, -0.9417, 0.0916, -0.2156, \dots, -0.0579, 0.0936, -0.0141, 0.0184)^T$	1.0388962
	50	73	$(1.3154, 0.0449, 1.4362, -0.0480, \dots, -0.0321, -0.1211, -0.1255, 0.1122)^T$	6.6157481
算法 3	3	7	$(0.1013, -1.5007)^T$	0.6776828
	10	17	$(1.6339, 1.7617, 1.3067, -0.4565, \dots, -0.3643, -1.3493, 0.5769, 1.4391)^T$	6.3177110
	20	28	$(-0.3008, -0.1076, -0.1151, -0.0807, \dots, -0.0143, -0.0177, -0.0248, -0.0307)^T$	0.0373929
	50	49	$(-0.0009, -0.0005, -0.0001, 0.0001, \dots, 0.0000, 0.0000, 0.0000, -0.0000)^T$	0.0000029

比较分析:该测试函数是一个对简单的二次函数进行求和的函数,各种改进的 BGFS 算法的迭代收敛曲线如图 2 所示,该函数只有一个全局最小值 $f(\mathbf{x}^*) = 0$,其中 $\mathbf{x}^* = (0, 0, \dots, 0)^T$,在控制相同的初始点情况下,由表 1 中对于 $n=3, 10, 20, 50$ 的迭代结果可以看出,这 3 种改进的 BFGS 算法的收敛速度都相差不大,且都能计算出满足精度要求的最优值,其中算法 1 和

算法 3 的速度较快,算法 2 的速度较慢。特别地,当 $n=20$ 时,算法 1 和算法 3 具有相同的收敛速度,但算法 1 得到了精度更高的最优值,这说明当 $n=20$ 时,算法 1 的收敛效果优于算法 3。因此当 $n=10$ 和 $n=20$ 时,算法 1 的收敛效果优于算法 3;当 $n=3$ 和 $n=50$ 时,算法 3 的收敛速度快于算法 1。

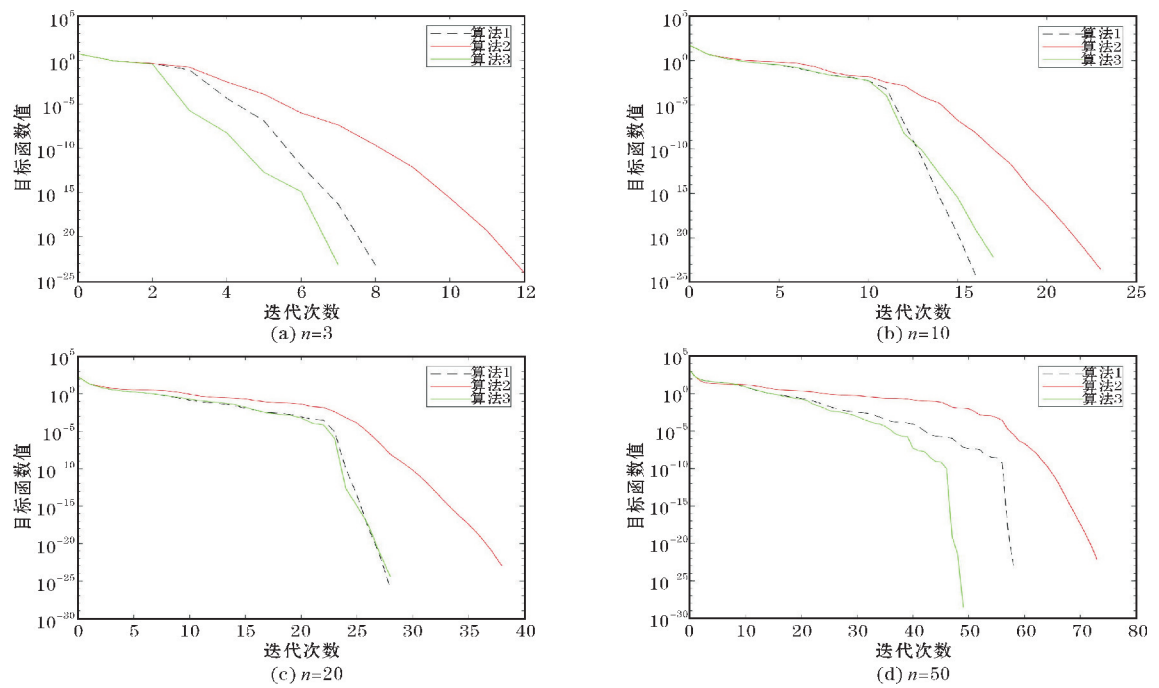


图2 实例1 各种改进的BGFS算法的迭代收敛曲线

3.2 考虑高次函数的无约束优化问题

实例2
$$\min f(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$$

该函数称为 Dixon & Price 函数,对于该函数,将精度设置为 10^{-10} ,同样选取不同维度进行迭代。选取初始点为 $\mathbf{x}_0 = (1.5, 1, 1.5, 1, \dots, 1.5, 1)^T$,再分别取 $n=2, n=10$, 和 $n=15$, 迭代结果见表2。

表2 实例2 各种改进的BGFS算法的迭代计算结果

迭代采用算法	n	k	最优解 $\mathbf{x}^{(k)}$	最小值 val($\times 10^{-23}$)
算法1	2	33	$(0.9999999, 0.7071068)^T$	2.2079971
	10	51	$(1.0000000, 0.7071068, 0.5946036, \dots, 0.5027150, 0.5013556, 0.5006774)^T$	2.1979343
	15	148	$(1.0000000, 0.7071068, 0.5946036, \dots, 0.5000846, 0.5000423, 0.5000211)^T$	2.9501049
算法2	2	35	$(0.9999999, 0.7071068)^T$	2.8040950
	10	53	$(1.0000000, 0.7071068, 0.5946036, \dots, 0.5027150, 0.5013556, 0.5006774)^T$	1.5296417
	15	70	$(1.0000000, 0.7071068, 0.5946036, \dots, 0.5000846, 0.5000423, 0.5000212)^T$	1.6780153
算法3	2	32	$(0.9999999, 0.7071068)^T$	2.6930876
	10	46	$(0.9999999, 0.7071068, 0.5946036, \dots, 0.5027150, 0.5013556, 0.5006774)^T$	2.2875534
	15	57	$(0.9999999, 0.7071068, 0.5946036, \dots, 0.5000846, 0.5000423, 0.5000212)^T$	1.4064852

比较分析:该函数是一个 n 元四次多项式函数,其各种改进的 BGFS 算法的迭代收敛曲线如图3所示,由文献[12]可知该测试函数具有全局最小值 $f(\mathbf{x}^*) = 0$,其中 $x_i = 2^{-\frac{2i-2}{2^i}}, i=1, 2, \dots, n$ 。由表2可以看出这三种改进的 BFGS 算法都能收敛到满足精度要求的最小

值,当 $n=2, n=10$ 时,三种算法迭代次数差别不大,速度快慢分别是算法3最快,其次为算法1再是算法2;而当 $n=15$ 时,算法3的收敛速度最快,算法2的收敛速度较快,而算法1的收敛速度则明显慢于另外两个算法。

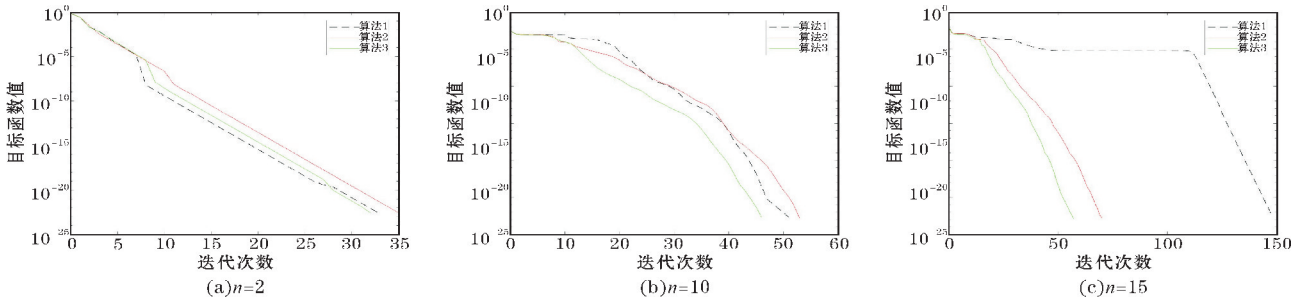


图3 实例2 各种改进的BGFS算法的迭代收敛曲线

实例 3 $\min f(\boldsymbol{x}) = \sum_{i=1}^{n-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

该函数称为 Rosenbrock 函数,也称为香蕉函数。在最优化中,Rosenbrock 函数是一个非凸函数,因此常被用作测试最优化算法性能。因此,采用此函数对这 3 种改进的 BFGS 算法进行对比分析。同样地,设置精

度为 10^{-5} ,为了对不同的改进的 BFGS 算法进行收敛性分析,将再分别对其取不同维度进行迭代。选取初始点为 $\boldsymbol{x}_0 = (-1.2, 0, -1.2, 0, \cdots, -1.2, 0)^T$,再选取 $n=2, n=10, n=20$ 和 $n=50$ 进行迭代,得到迭代计算结果见表 3。

表 3 实例 3 各种改进的 BGFS 算法的迭代计算结果

迭代采用算法	n	k	最优解 $\boldsymbol{x}^{(k)}$	最小值 val($\times 10^{-14}$)
算法 1	2	38	$(0.9999999, 0.9999999)^T$	0.4644797
	10	58	$(1.0000000, 0.9999999, 1.0000000, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.5652141
	20	125	$(0.9999999, 0.9999999, 0.9999999, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.7129615
	50	195	$(0.9999999, 0.9999999, 0.9999999, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.8784013
算法 2	2	32	$(1.0000000, 1.0000001)^T$	1.0367071
	10	49	$(0.9999999, 0.9999999, 0.9999999, \cdots, 0.9999999, 0.9999999, 0.9999999)^T$	4.0080599
	20	74	$(1.0000000, 0.9999999, 1.0000000, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.3025382
	50	142	$(1.0000000, 1.0000000, 1.0000000, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.9683717
算法 3	2	30	$(0.9999999, 0.9999999)^T$	0.0009107
	10	48	$(1.0000000, 0.9999999, 1.0000000, \cdots, 1.0000000, 1.0000000, 1.0000000)^T$	0.2936574
	20	63	$(0.9999999, 0.9999999, 0.9999999, \cdots, 0.9999999, 0.9999999, 1.0000000)^T$	0.8256109
	50	109	$(1.0000000, 1.0000000, 1.0000000, \cdots, 0.9999999, 0.9999999, 0.9999999)^T$	0.5087971

比较分析:该函数是一个非凸函数,它的各种改进的 BGFS 算法的迭代收敛曲线如图 4 所示,由文献 [12] 可知其具有全局最小值点 $\boldsymbol{x}^* = (1, 1, 1, \cdots, 1, 1)^T$,在点 $\boldsymbol{x}_m = (-1, 1, 1, \cdots, 1, 1)^T$ 附近有局部最小值。在控制相同的初始点情况下,由表 3

中对于 $n=2, 10, 20, 50$ 的迭代结果可以看出这 3 种改进的 BFGS 算法都能计算出满足精度要求的最优值,其中算法 3 的收敛速度最快,算法 2 的速度较慢,算法 1 的收敛速度最慢。

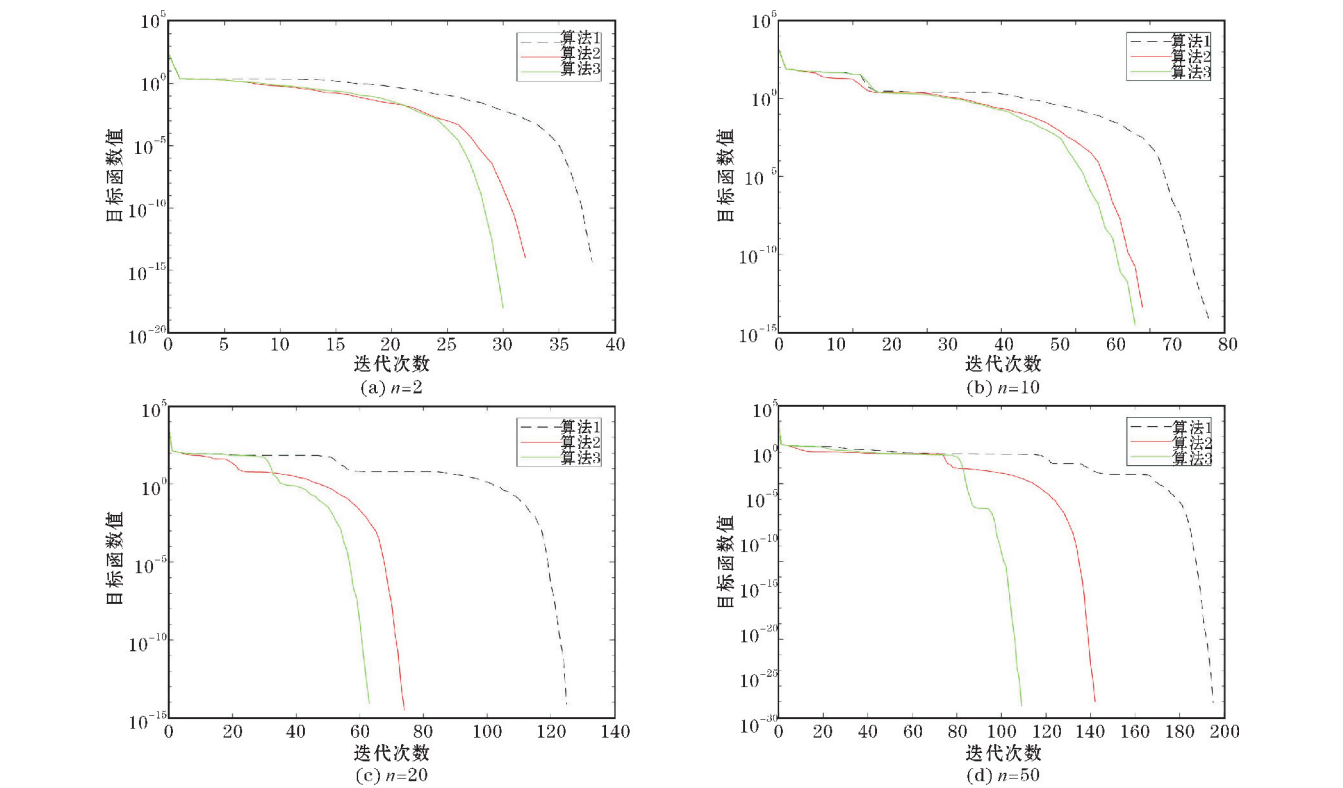


图 4 实例 3 各种改进的 BGFS 算法的迭代收敛曲线

实例 4 $\min f(\boldsymbol{x}) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$

该函数称为 Six-Hump Camel 函数,对于该函数,将精度设置为 10^{-8} ,由文献[12]可知该测试函数具有

两个全局最小值 $f(\boldsymbol{x}^*) = -1.0316$,其中两个全局最优
点分别为 $\boldsymbol{x}_1^* = (-0.08984, 0.71265)^T$ 和 $\boldsymbol{x}_2^* = (0.08984, -0.71265)^T$,由于该函数的两个最优值点关于原点具有对称性,因此分别选取初始点为 $\boldsymbol{x}_0 = (-2, 2)^T$ 和 $\boldsymbol{x}_0 = (2, -2)^T$,迭代结果见表 4。

表 4 实例 4 各种改进的 BGFS 算法的迭代计算结果

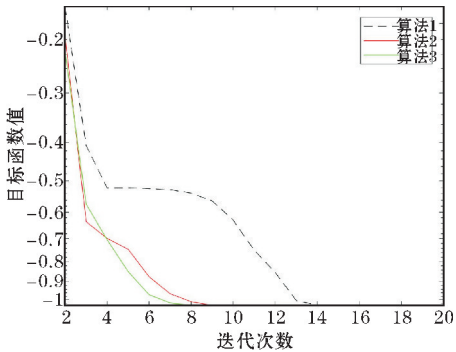
迭代采用算法	\boldsymbol{x}_0	k	最优解 $\boldsymbol{x}^{(k)}$	最小值 val
算法 1	$\boldsymbol{x}_0 = (-2, 2)^T$	19	$(-0.0898420130251718, 0.7126564030279365)^T$	-1.0316284534898774
	$\boldsymbol{x}_0 = (2, -2)^T$	19	$(0.0898420130251718, -0.7126564030279365)^T$	-1.0316284534898774
算法 2	$\boldsymbol{x}_0 = (-2, 2)^T$	13	$(-0.0898420130925821, 0.7126564029844342)^T$	-1.0316284534898774
	$\boldsymbol{x}_0 = (2, -2)^T$	13	$(0.0898420130925821, -0.7126564029844342)^T$	-1.0316284534898774
算法 3	$\boldsymbol{x}_0 = (-2, 2)^T$	12	$(-0.0898420131139434, 0.7126564030269821)^T$	-1.0316284534898774
	$\boldsymbol{x}_0 = (2, -2)^T$	12	$(0.0898420131139434, -0.7126564030269821)^T$	-1.0316284534898774

比较分析:该函数是一个二元六次函数,其各种改进的 BGFS 算法的迭代收敛曲线如图 5 所示。由表 4 的迭代结果可以看出,当初始点为 $\boldsymbol{x}_0 = (-2, 2)^T$ 时,这 3 种改进的 BGFS 算法均收敛到 $\boldsymbol{x}_1^* = (-0.08984, 0.71265)^T$;而当初始点为 $\boldsymbol{x}_0 = (2, -2)^T$ 时,这 3 种改进的 BGFS 算法均收敛到 $\boldsymbol{x}_2^* = (0.08984, -0.71265)^T$,且对于不同的初始点,它们的迭代次数和最优值分别相同,这说明对于该函数取不同的初始点对 3 种改进的 BGFS 算法的收敛速度影响不大,但是会决定它们收敛的最优值点。在控制初始点相同的情况下,这 3 种算法能收敛到同一最优值点,其中算法 3 的收敛速度最快,算法 2 的收敛速度较快,算法 1 的收敛速度最慢。

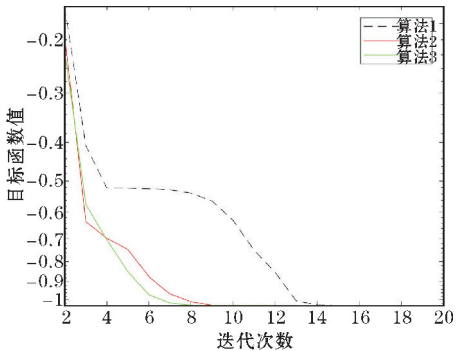
3.3 考虑非多项式函数的无约束优化问题

实例 5 $\min f(\boldsymbol{x}) = \sin(x_1)e^{(1-\cos(x_2))} + \cos(x_2)e^{(1-\sin(x_1))} + (x_1 - x_2)^2$

该函数称为 Bird 函数,设置精度为 10^{-5} ,选取初始点为 $\boldsymbol{x}_0 = \{(10, 0)^T, (2, -5)^T\}$,迭代结果见表 5。



(a) $\boldsymbol{x}_0 = (-2, 2)$



(b) $\boldsymbol{x}_0 = (2, -2)$

图 5 实例 4 各种改进的 BGFS 算法的迭代收敛曲线

表 5 实例 5 各种改进的 BGFS 算法的迭代计算结果

迭代采用算法	\boldsymbol{x}_0	k	最优解 $\boldsymbol{x}^{(k)}$	最小值 val
算法 1	$\boldsymbol{x}_0 = (10, 0)^T$	14	$(4.7010431, 3.1529385)^T$	-106.76453675
	$\boldsymbol{x}_0 = (2, -5)^T$	16	$(-1.5821422, -3.1302468)^T$	-106.76453675
算法 2	$\boldsymbol{x}_0 = (10, 0)^T$	10	$(4.7010431, 3.1529385)^T$	-106.76453675
	$\boldsymbol{x}_0 = (2, -5)^T$	10	$(-1.5821422, -3.1302468)^T$	-106.76453675
算法 3	$\boldsymbol{x}_0 = (10, 0)^T$	9	$(4.7010431, 3.1529385)^T$	-106.76453675
	$\boldsymbol{x}_0 = (2, -5)^T$	9	$(-1.5821422, -3.1302468)^T$	-106.76453675

比较分析:该函数是一个同时含有指数函数和三角函数的非多项式函数,其各种改进的 BGFS 算法的迭代收敛曲线如图 6 所示,由文献^[12]可知该测试函数具有两个全局最小值 $f(\mathbf{x}^*)=-106.764537$,其中两个全局最优值点分别为 $\mathbf{x}_1^*=(4.70104,3.15294)^T$ 和 $\mathbf{x}_2^*=(-1.58214,-3.13024)^T$ 。由表 5 可以看出,当初始点为 $\mathbf{x}_0=(10,0)^T$ 时,这 3 种改进的 BFGS 算法均收敛到 $\mathbf{x}_1^*=(4.70104,3.15294)^T$;而当初始点为 $\mathbf{x}_0=(2,-2)^T$ 时,这 3 种改进的 BFGS 算法均收敛到 $\mathbf{x}_2^*=(-1.58214,-3.13024)^T$,这说明对于该函数取不同的初始点对 3 种改进的 BFGS 算法的收敛最优值没有太大影响。在控制初始点相同的情况下,这 3 种算法能收敛到同一最优值点,其中算法 3 的收敛速度最快,算法 2 的收敛速度较快,算法 1 的收敛速度最慢。

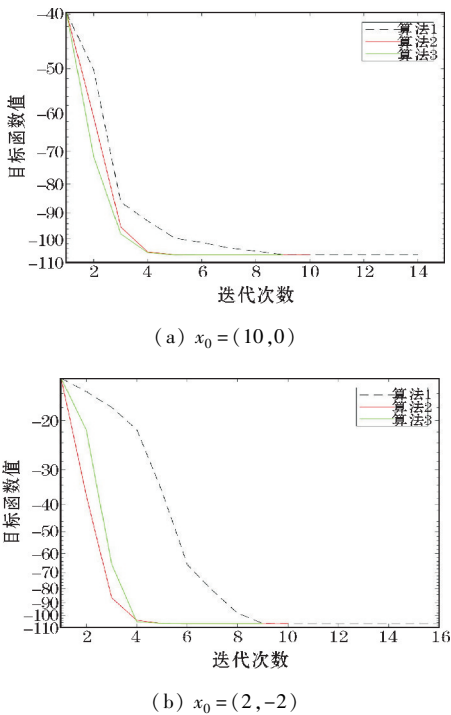


图 6 实例 5 各种改进的 BGFS 算法的迭代收敛曲线

实例 6 $\min f(\mathbf{x})=(\sum_{i=2}^5 i\cos((i+1)x_1+i))\cdot(\sum_{i=2}^5 i\cos((i+1)x_2+i))$

该函数称为 Shubert 函数,对于该函数,将精度设置为 10^{-5} ,由于该函数有多个极小值点,因此将分别选取初始点为 $\mathbf{x}_0=\{(-1,-1)^T,(-8,-1)^T,(-1,-8)^T,(5,-3)^T\}$,迭代结果见表 6。

比较分析:Shubert 函数具有多个局部极小和多个全局极小,其各种改进的 BGFS 算法的迭代收敛曲线如图 7 所示。由文献^[12]可知当 $-10\leq x_i\leq 10$ 时该函数具有 18 个全局最小值 $f(\mathbf{x}^*)\simeq-186.7039$ 。其中, $\mathbf{x}^*=\{(-7.0835,4.8580)^T,(-7.0835,-7.7083)^T,(4.8580,-7.0835)^T,(-1.4251,-0.8003)^T,(5.4828,4.8580)^T,(-7.7083,-0.8003)^T,(-7.7083,-7.0835)^T,(-7.0835,-1.4251)^T,(4.8580,5.4828)^T,(-0.8003,-7.7083)^T,(-0.8003,4.8580)^T,(5.4828,-7.7083)^T,(5.4828,-1.4251)^T,(-0.8003,-1.4251)^T,(-1.4251,5.4828)^T,(-7.7083,5.4828)^T,(-1.4251,-7.0835)^T,(4.8580,-0.8003)^T\}$ 。

由表 6 的迭代结果可以看出,这 3 种改进的 BFGS 算法对于初始点取 $\mathbf{x}_0=\{(-1,-1)^T,(-8,-1)^T,(-1,-8)^T,(5,-3)^T\}$ 均可迭代到全局最优值点,且其中算法 3 速度最快,算法 2 速度较快,算法 1 速度最慢。特别地,当取 $\mathbf{x}_0=(-1,-1)^T$ 时,算法 1 和算法 2 的迭代最优值点为 $\mathbf{x}^*=(-1.4251284,-0.8003211)^T$,而算法 3 的迭代最优值点为 $\mathbf{x}^*=(-0.8003211,-1.4251284)^T$,这是因为这两个最优值点距离初始点 $\mathbf{x}_0=(-1,-1)^T$ 均较为接近,所以采用的校正公式不同会导致得到的最优值点不同。对于其他的不同初始点,3 种改进的 BFGS 算法均得到了相同的最优值点。

表 6 实例 6 各种改进的 BGFS 算法的迭代计算过程

迭代采用算法	\mathbf{x}_0	k	最优解 $\mathbf{x}^{(k)}$	最小值 val
算法 1	$\mathbf{x}_0=(-1,-1)^T$	17	$(-1.4251284,-0.8003211)^T$	-186.73090883
	$\mathbf{x}_0=(-1,-8)^T$	10	$(-0.8003211,-7.7083137)^T$	-186.73090883
	$\mathbf{x}_0=(-8,-1)^T$	10	$(-7.7083137,-0.8003211)^T$	-186.73090883
	$\mathbf{x}_0=(5,-3)^T$	15	$(4.8580569,-0.8003211)^T$	-186.73090883
算法 2	$\mathbf{x}_0=(-1,-1)^T$	10	$(-1.4251284,-0.8003211)^T$	-186.73090883
	$\mathbf{x}_0=(-1,-8)^T$	9	$(-0.8003211,-7.7083137)^T$	-186.73090883
	$\mathbf{x}_0=(-8,-1)^T$	9	$(-7.7083137,-0.8003211)^T$	-186.73090883
	$\mathbf{x}_0=(5,-3)^T$	9	$(4.8580569,-0.8003211)^T$	-186.73090883
算法 3	$\mathbf{x}_0=(-1,-1)^T$	10	$(-0.8003211,-1.4251284)^T$	-186.73090883
	$\mathbf{x}_0=(-1,-8)^T$	7	$(-0.8003211,-7.7083137)^T$	-186.73090883
	$\mathbf{x}_0=(-8,-1)^T$	7	$(-7.7083137,-0.8003211)^T$	-186.73090883
	$\mathbf{x}_0=(5,-3)^T$	8	$(4.8580569,-0.8003211)^T$	-186.73090883

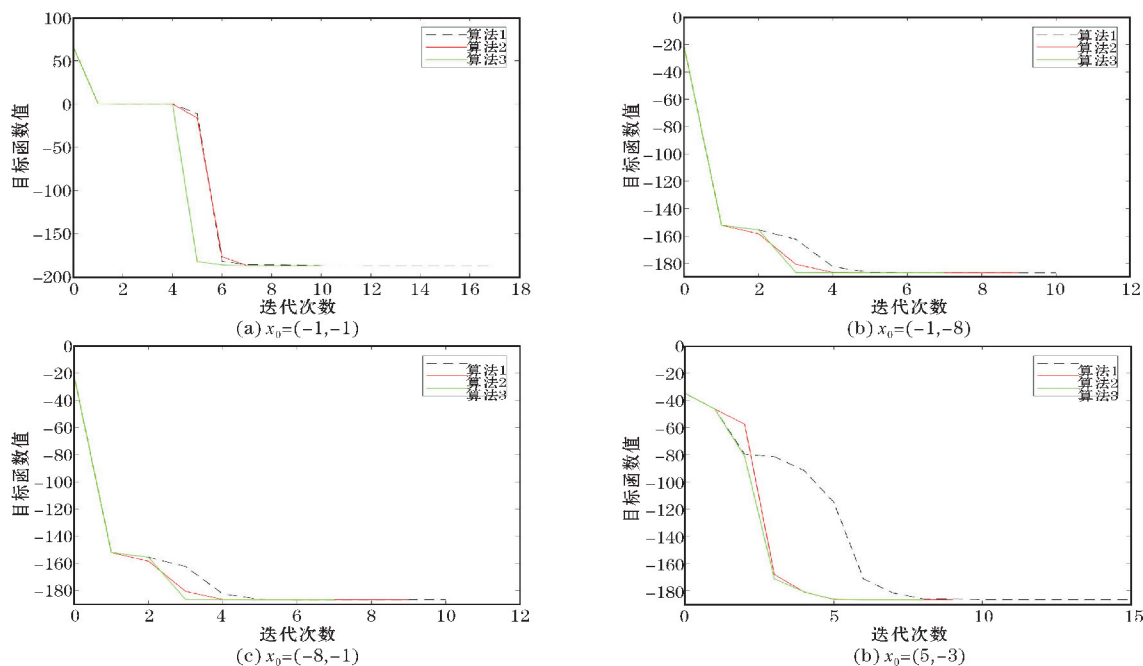


图7 实例6各种改进的BFGS算法的迭代收敛曲线

4 结束语

采用基于 Wolfe 线搜索准则的拟牛顿算法来求解无约束优化中的最小值问题,讨论了3种不同的改进的BFGS算法,再分别用这3种改进的BFGS算法对不同的无约束优化问题求最优解进行对比分析。通常情况下,为了不浪费精力去寻找迭代步长的精确极小值,多会采用非精确线搜索准则对迭代步长进行搜索。因为搜索方向很少恰好是正确的方向,通常只要能靠近就足够。而采用 Armijo 线搜索准则会在迭代过程中出现较小的步长,采用 Wolfe 线搜索准则能够有效地避免出现类似这样的小步长。

通过数值实验计算以及对比分析,发现对于二次函数的实例中,使用这3种改进BFGS算法的收敛速度都较为接近,其中算法1和算法3的收敛速度对于不同的函数维度分别表现出不同的优越性,而算法2的收敛速度则稍显缓慢。在所测试的3个有关高次函数的实例中,发现在控制初始点相同的情况下,这3种算法能收敛到同一最优值点,其中算法3的收敛速度均优于算法1和算法2,算法1和算法2在不同的函数中对于不同的函数维度各有优势。且在实例4中,发现对于具有多个全局最优值的函数,通过选取不同的初始点,会得到不同的最优值点,这说明不同的初始点选择会决定3种改进的BFGS算法的收敛最优值点。最后在有关非多项式函数实例中,两个实例都表明算法3的收敛速度最快,算法2的收敛速度较快,算法1

的收敛速度最慢。且发现对于具有多个全局最优值的函数,在控制初始点相同下,这3种改进的BFGS算法基本都能得到相同的最优值点,只有在实例6中取 $x_0 = (-1, -1)^T$ 时,算法3得到了与算法1和算法2不同的最优值点,这是因为所得到的最优值点均离初始点较为接近,而选取不同的初始点也会得到不同的最优值点。综上所述,在求解无约束优化问题时,算法3的改进相较于算法1和算法2的改进是一个收敛效果较为不错的改进。

参考文献:

- [1] Mokhtar S. Bazaraa, Hanif D. Sherali, C. M. Shetty. Nonlinear programming: theory and algorithms [M]. Library of Congress Publication, 2005.
- [2] Wei Z, Li G, Qi L. New quasi-Newton methods for unconstrained optimization problems[J]. Computational Optimization and Applications, 2006, 175: 1156-1188.
- [3] Wei Z, Yu G, Lian Z. The superlinear convergence of a modified BFGS-type method for unconstrained optimization[J]. Computational Optimization and Applications, 2004, 29: 315-332.
- [4] Yuan G, Wei Z. Convergence analysis of a modified BFGS method on convex minimizations[J]. Computational Optimization and Applications, 2010, 47: 237-255.

- [5] 陈奎林. 一种改进的 BFGS 算法及其收敛性分析[J]. 重庆理工大学学报, 2011, 25(11): 111–114.
- [6] Forsgren A, Odland T. On exact linesearch quasi-Newton methods for minimizing a quadratic function[J]. Computational Optimization and Applications, 2018, 69: 225–241.
- [7] Ek D, Forsgren A. Exact linesearch limited-memory quasi-Newton methods for minimizing a quadratic function[J]. Computational Optimization and Applications, 2021, 79: 789–816.
- [8] 景慧丽. 无约束最优化问题中修改的 BFGS 方法[J]. 科技信息, 2008, 26: 209–210.
- [9] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1997.
- [10] Yuan G, Wei Z, Lu X. Global convergence of BFGS and PRP methods under a modified weak Wolfe-Powell line search[J]. Appl. Math. Modelling, 2017, 47: 811–825.
- [11] LI D, FU M, KUSHIMA. A modified BFGS method and its global convergence in nonconvex minimization[J]. Journal of Computational and Applied Mathematics, 2001, 129: 15–35.
- [12] Momin Jamil, Yang Xin She. A Literature Survey of Benchmark Functions For Global Optimization Problems[J]. International Journal of Mathematical Modelling & Numerical Optimisation, 2013, 4(2): 150–194.

A Kind of Improved BFGS Quasi-Newton Method and its Comparison with other Quasi-Newton Methods

YANG Qian¹, WU Zezhong¹, HE Shengyu²

(1. College of Applied Mathematics, Chengdu University of Information Technology, Chengdu 610225, China; 2. Xichang University, Xichang 615000, China)

Abstract: Quasi-Newton method is an important method to solve unconstrained problems in optimization, and the improvement of the quasi-Newton algorithm is a key research content. In this paper, a new improved BFGS algorithm is proposed based on the new quasi-Newton equation $\mathbf{B}_{k+1} \cdot \mathbf{s}^{(k)} = (\mathbf{y}^{(k)})^*$. Combined with several previous improved BFGS algorithms, the iterative step size is searched by using the Wolfe line search criterion. By selecting some test functions, the convergence effects of these improved BFGS quasi Newton methods are compared and analyzed by using MATLAB tools. The experimental results show that the improved BFGS algorithm proposed in this paper is an improvement with good convergence effect compared with several previous improved BFGS algorithms.

Keywords: unconstrained optimization; BFGS quasi-Newton method; quasi-Newton equation; Wolfe line search criterion