

求解无约束优化问题的Aitken 加速算法

谢亚君¹, 姚洁¹, 马昌凤²

(1. 福州外语外贸学院 理工学院, 福建福州 350202;
2. 福建师范大学 数学与信息学院, 福建福州 350007)

摘要: 拟牛顿法是求解无约束优化问题的经典且有效方法. 基于Neculai(2019)提出的所谓对角拟牛顿更新技巧, 构造了一种新的修正Aitken加速算法来求解无约束优化问题. 理论上保证了该方法相比于对角拟牛顿更新技巧有更高阶的收敛性能. 数值测试结果也验证了所提出算法的高效性.

关键词: 无约束优化问题; 修正Aitken加速迭代; 高阶收敛性; 数值实验

中图分类号: O224.2

文献标识码: A **文章编号:** 1000-4424(2021)04-0481-11

§1 引言

基于Neculai^[1]的工作, 本文提出一种求解如下无约束优化问题

$$\min_{x \in \mathbf{R}^n} f(x) \quad (1)$$

的Aitken加速方法, 其中 $f \in \mathbf{R}^n \rightarrow \mathbf{R}$ 为连续可微的函数.

至今为止, 无约束优化问题的求解已获得广泛关注, 并取得丰硕的研究成果(见[2-6]), 其中, 包括共轭梯度法(见[7]), 牛顿法与拟牛顿法等(见[8-12]). 虽然牛顿法具有二阶收敛速度的优势, 但是需要计算二阶导数. 因此在求解大规模问题时, 该方法往往因为计算代价过高而变得不再适用. 同时当Hessian阵非正定时, 无法保证所产生的方向是目标函数的下降方向. 尤其是当Hessian阵奇异时, 算法无法继续循环迭代. 修正牛顿法可在一定程度上克服此缺陷, 然而修正参数的选取较难把握, 对收敛速度亦有较大影响. 众所周知, 拟牛顿法可以避免这些缺点, 在迭代过程中不需要计算目标函数的Hessian阵, 却在某种意义上具有Hessian阵的作用, 因此该方法在相关领域得到广泛的应用. 例如, 均衡问题, 神经网络, 激光检测, 非线性方程组等(见[13-16]).

假设 x_0 为某个初值点. 求解问题(1)的迭代格式有如下表达式

$$x_{k+1} = x_k + \alpha_k d_k, \quad (2)$$

收稿日期: 2020-06-27 修回日期: 2020-11-01

基金项目: 福建省自然科学基金(2019J01879); 福州外语外贸学院高层次人才计划(FWKQJ202002); 国科大实验室技术合作研发项目(20A01246ZY; H2020003)

其中 α_k 是由某种线搜索,例如Wolfe或Armijo线搜索获得的步长, d_k 表示搜索方向.通常对问题(1)的迭代求解格式有两个主要切入点,一个是针对步长因子 α_k 进行合理设计,另一个是考虑有效地选取较优的搜索方向 d_k .详细可参看文献[17-20].

牛顿法是通过求解如下方程得到牛顿搜索方向 d_k ,

$$G_k d_k = -g_k, \quad (3)$$

其中 G_k, g_k 分别表示函数 f 在点 x_k 处的Hessian矩阵和梯度值.鉴于Hessian阵 G_k 高昂的计算代价,可寻求一个矩阵 B_{k+1} 作为Hessian阵 G_{k+1} 的一种近似替代.将函数 f 在点 x_{k+1} 处利用二次Taylor展开得到近似方程

$$G_{k+1} s_k \approx y_k,$$

进而构造所谓的割线方程

$$B_{k+1} s_k = y_k, \quad (4)$$

其中 $s_k = x_{k+1} - x_k, y_k = g_{k+1} - g_k, B_{k+1} = B_k + \Delta_k, \Delta_k$ 可视为误差校正矩阵.利用割线方程推导近似Hessian阵 B_{k+1} ,关于矩阵 B_{k+1} 的有效选取方面已有一系列较好的研究成果.例如,秩1校正公式或BFGS算法等(见[21-23]).

在文献[1]中, Neculai证明了一个简单而有效的对角拟牛顿更新公式来求解

$$\min f(x) = \frac{1}{2} \|\Delta_k\|_F^2 + \text{tr}(B_k + \Delta_k), \quad (5)$$

约束条件为弱割线方程 $s_k^T B_{k+1} s_k = s_k^T y_k$ (见[8]).误差校正矩阵 $\Delta_k := \text{diag}(\delta_k^1, \delta_k^2, \dots, \delta_k^n)$ 被考虑为一个对角矩阵.同时, Neculai特别给了详细理由来阐述引入弱割线方程的优势和特点.然而注意到目标函数的迹算子项“ $\text{tr}(B_k + \Delta_k)$ ”可以暂时被忽略的.原因是矩阵 $B_k + \Delta_k$ 在每一步更新中总是都保持着对角阵的形式.事实上,可通过极小化第一个偏差项的Frobenius范数,即 $\|\Delta_k\|_F$,来达到一种最小化的目的.进而仅考虑一个更为简洁的问题

$$\min f(x) = \frac{1}{2} \|\Delta_k\|_F^2, \text{ s.t. } s_k^T B_{k+1} s_k = s_k^T y_k. \quad (6)$$

针对问题(6),基于对角拟牛顿更新迭代法(DNRTR)(见[1]),提出一个修正的Aitken加速迭代格式来求解无约束优化问题(1),该方法在与前者对比的过程中显示了非常良好且具有明显优势的收敛性能.

本文剩余部分组织如下: §2提出求解无约束优化问题(1)的修正Aitken加速迭代方法,同时证明了其具有良好的高阶收敛性能.数值实验和结论分别安排在§3和§4.

§2 算法及收敛性

本节考虑改善对角拟牛顿更新方法(DNRTR)(见[1]),进而引入一个修正Aitken加速迭代格式来求解优化问题(1).最后提供重要的理论证明和收敛性结果.

首先注意到问题(6)亦可转化为如下形式

$$\begin{aligned} \min f(x) &= \frac{1}{2} \sum_{i=1}^n (\delta_k^i)^2 \\ \text{s.t. } & \sum_{i=1}^n (s_k^i)^2 \delta_k^i = s_k^T y_k - \sum_{i=1}^n (s_k^i)^2 b_k^i, \end{aligned} \quad (7)$$

其中 δ_k^i, b_k^i 分别是对角矩阵 Δ_k, B_k 的第 i 个元素.

问题(7)的Lagrangian函数为

$$L(\delta_k^i, \lambda) = \frac{1}{2} \sum_{i=1}^n (\delta_k^i)^2 + \lambda \left(\sum_{i=1}^n (s_k^i)^2 \delta_k^i - s_k^T y_k + \sum_{i=1}^n (s_k^i)^2 b_k^i \right), \quad (8)$$

其中 λ 表示其Lagrangian乘子. 由最优性条件可得

$$\frac{\partial L(\delta_k^i, \lambda)}{\partial \delta_k^i} = \delta_k^i + \lambda (s_k^i)^2 = 0, i = 1, \dots, n, \quad (9)$$

即

$$\delta_k^i = -\lambda (s_k^i)^2, i = 1, \dots, n. \quad (10)$$

将(10)式代入(7)式, 即可得Lagrangian乘子

$$\lambda = \frac{\sum_{i=1}^n (s_k^i)^2 b_k^i - s_k^T y_k}{\sum_{i=1}^n (s_k^i)^4} = \frac{s_k^T B_k s_k - s_k^T y_k}{\text{tr}(S_k^2)}, \quad (11)$$

其中 $S_k = \text{diag}((s_k^1)^2, \dots, (s_k^n)^2)$. 结合(11)式以及(10)式可得如下对角矩阵更新格式

$$B_{k+1} = B_k + \Delta_k = B_k + \frac{s_k^T y_k - s_k^T B_k s_k}{\text{tr}(S_k^2)} S_k. \quad (12)$$

其中

$$\Delta_k := \frac{s_k^T y_k - s_k^T B_k s_k}{\text{tr}(S_k^2)} S_k. \quad (13)$$

从而线搜索方向可由式子 $d_{k+1}^i = -(B_{k+1})_{ii}^{-1} g_{k+1}^i (i = 1, \dots, n)$ 所确定. 注意到, 该迭代格式是由相互独立的“点对点”形式进行更新. 鉴于此特殊性, 考虑对DNRTR方法引入一种修正的Aitken加速技巧从而可望获得更具优越的收敛性能.

算法 1 (基于拟牛顿更新的Aitken加速算法(AADQN))

步1 给定初始点 $x_0 \in \mathbf{R}^n$, 参数 $0 < \beta < 1, 0 < \sigma < 0.5, m = 0$ 及充分小的正数 $\varepsilon_1, \varepsilon_2 > 0$. 计算向量 $g_0 = \nabla f(x_0)$. 令 $d_0 = -g_0, k = 0$.

步2 若 $\|g(x_k)\|_2 < \varepsilon_1$ 停算; 否则, 转到步3.

步3 计算满足线搜索的步长 α_k , 若不等式

$$f(x_k + \beta^m d_k) \leq f(x_k) + \sigma \beta^m g_k^T d_k \quad (14)$$

成立, 令 $m_k = m, \alpha_k = \beta^{m_k}$; 否则, $m := m + 1$, 然后循环(14). 再计算

$$x_{\bar{k}} = x_k + \alpha_k d_k, f_{\bar{k}} = f(x_{\bar{k}}), g_{\bar{k}} = \nabla f(x_{\bar{k}}). \quad (15)$$

令 $s_k = x_{\bar{k}} - x_k, y_k = g_{\bar{k}} - g_k$.

步4 利用公式(12)更新对角矩阵 B_{k+1} .

步5 令

$$\varphi(x) := x - \alpha_k \frac{g(x)}{(B_{k+1})_{ii}}. \quad (16)$$

计算

$$x_{\bar{k}+1} = \varphi(x_{\bar{k}}) = x_{\bar{k}} - \alpha_k \frac{g(x_{\bar{k}})}{(B_{k+1})_{ii}}, x_{\bar{k}+2} = \varphi(x_{\bar{k}+1}) = x_{\bar{k}+1} - \alpha_k \frac{g(x_{\bar{k}+1})}{(B_{k+1})_{ii}}, \quad (17)$$

$$\bar{x}^i = x_{\bar{k}+2}^i - (x_{\bar{k}+1}^i - x_{\bar{k}+2}^i)^2 / (x_{\bar{k}+2}^i - 2x_{\bar{k}+1}^i + x_{\bar{k}}^i), \quad (18)$$

其中 $i = 1, \dots, n$, $\bar{x}^i = (\bar{x})_i$, $p_k^i = (p_k)_i$, $q_k^i = (q_k)_i$. 置 $x_k = \bar{x}$.

步6 更新搜索方向

$$(d_{k+1})_i = \begin{cases} -\frac{(g(x_{k+1}))_i}{(B_{k+1})_{ii}}, & (B_{k+1})_{ii} \geq \varepsilon_2, \\ -(g(x_{k+1}))_i, & \text{否则.} \end{cases} \quad (19)$$

步7 置 $k := k + 1$, 返回步2.

注记 1 显然经过计算可知(19)所确定的搜索方向是下降的, 即 $g(x_{k+1})^T d_{k+1} < 0$. 事实上有

$$\begin{aligned} g(x_{k+1})^T d_{k+1} &= \sum_{i=1}^n (g(x_{k+1}))_i (d_{k+1})_i \\ &= \begin{cases} -\sum_{i=1}^n \frac{((g(x_{k+1}))_i)^2}{(B_{k+1})_{ii}}, & (B_{k+1})_{ii} \geq \varepsilon_2 > 0, \\ -\sum_{i=1}^n ((g(x_{k+1}))_i)^2, & \text{否则.} \end{cases} \end{aligned}$$

由上式可得, 若 $\|g(x_{k+1})\| \neq 0$, 则有 $g(x_{k+1})^T d_{k+1} < 0$.

注记 2 在近似Hessian阵 B_{k+1} 为对角阵的条件下, (12) 可视为秩 n 拟牛顿法修正公式.

注记 3 为了避免在更新迭代中对角矩阵 B_{k+1} 可能出现非正定或奇异的情形, 算法要求 $b_{k+1}^i = b_k^i + \delta_k^i > 0, i = 1, 2, \dots$. 具体执行中考虑若 $b_{k+1}^i > \varepsilon_2 > 0$, 则 $d_{k+1}^i = -\frac{(g(x_{k+1}))_i}{b_{k+1}^i}$. 否则, $d_{k+1}^i = -(g(x_{k+1}))_i$.

注记 4 算法考虑了Aitken加速的思想. 设 x_k 为 x_* 的某个近似, 令 $x_{k+1} = \varphi(x_k)$, $x_{k+2} = \varphi(x_{k+1})$. 则 $x_{k+1} - x_* \approx \eta(x_k - x_*)$, $x_{k+2} - x_* \approx \eta(x_{k+1} - x_*)$, 其中 $\eta \approx \varphi'(x)$. 以上两式相除可得

$$\frac{x_{k+1} - x_*}{x_{k+2} - x_*} \approx \frac{x_k - x_*}{x_{k+1} - x_*},$$

进而可解得

$$x_* \approx x_{k+2} - \frac{(x_{k+2} - x_{k+1})^2}{x_{k+2} - 2x_{k+1} + x_k}. \quad (20)$$

当 $k \rightarrow \infty$, 若 $x_k \rightarrow x_*$ 且 $\|g_*\| \rightarrow 0$, 则由算法步5定义的格式 $x = \varphi(x)$ 即为不动点迭代. 关于 $\|g_*\| \rightarrow 0$ 的事实将在后面收敛性分析给出详细证明.

下面先给出算法的适定性证明.

引理 1 序列 $\{\Delta_k\}, \{B_k\}$ 由更新格式(12)产生. 此外, 若 $\|s_k\| \neq 0$, 且存在 $\underline{b} \leq b_0^i \leq \bar{b} (i = 1, 2, \dots, n)$, 其中 b_0^i 为对角矩阵 B_0 的第 i 个对角元素, 则对所有的正整数 k , 序列 $\{\Delta_k\}, \{B_k\}$ 有界.

证 当 $k = 0, y_0 = \nabla^2 f(s_0) s_0$, 其中 $x_0 < s_0 < x_1$. 令 \tilde{s}_0 为向量 s_0 的最大值分量, 由(13)式可得

$$\begin{aligned} |\delta_0^i| &= \left| \frac{s_0^T y_0 - s_0^T B_0 s_0}{\text{tr}(S_0^2)} (s_0^i)^2 \right| \leq \frac{|s_0^T y_0 - s_0^T B_0 s_0|}{\text{tr}(S_0^2)} (s_0^i)^2 \\ &\leq \frac{|s_0^T \nabla^2 f(s_0) s_0 - s_0^T B_0 s_0|}{\text{tr}(S_0^2)} (\tilde{s}_0)^2 \leq \frac{|s_0^T \nabla^2 f(s_0) s_0 - s_0^T B_0 s_0|}{\text{tr}(S_0^2) (\tilde{s}_0)^2} (\tilde{s}_0)^4 \\ &\leq \frac{|s_0^T \nabla^2 f(s_0) s_0 - s_0^T B_0 s_0|}{(\tilde{s}_0)^2}, \end{aligned} \quad (21)$$

其中用到不等式 $(\tilde{s}_0)^4 < \text{tr}(S_0^2)$.

注意到 $\|s_0\|^2 \leq n(\tilde{s}_0)^2$ 且由已知条件

$$\underline{b}\|s_0\|^2 \leq s_0^T B_0 s_0 \leq \bar{b}\|s_0\|^2, \tag{22}$$

可得

$$|\delta_0^i| \leq \frac{\tilde{C}\|s_0\|^2 + \delta_0\|s_0\|^2}{(\tilde{s}_0)^2} \leq \frac{(\tilde{C} + \delta_0)n(\tilde{s}_0)^2}{(\tilde{s}_0)^2} = (\tilde{C} + \delta_0)n := C_0. \tag{23}$$

其中 $\tilde{C} = \max\{c_1, c_2\}$ (假设4) 且 $\delta_0 = \max\{|\underline{b}|, |\bar{b}|\}$, 因此 Δ_0 是有界的.

由于 $b_1^i = b_0^i + \delta_0^i (i = 1, 2, \dots)$, 故有

$$\underline{b} - C_0 \leq b_1^i \leq \bar{b} + C_0,$$

即 B_1 也是有界的. 从而由归纳法可得引理结论.

文献[24]中给出了所谓界退化性质

$$\|B_{k+1} - \nabla^2 f(x_*)\|_F \leq \|B_k - \nabla^2 f(x_*)\|_F + \kappa \nu_k, \tag{24}$$

其中 B_{k+1} 是前一步迭代矩阵 B_k 的更新, $\nu_k = \max\{\|x_{k+1} - x_*\|, \|x_k - x_*\|\}$, κ 为某个常数. 该文描述了Hessian矩阵与其近似矩阵 B_{k+1} 可由前一步迭代中二者的差值来控制的事实, 既满足界退化性质, 同时也给出了满足此性质的算法至少具有 q -线性收敛的结果(详见[24]). 下面证明更新公式(12)满足界退化性质.

引理 2 若 $\|s_k\| \neq 0$, 则更新公式(12)满足界退化性质.

证 由更新公式(12)可得

$$\begin{aligned} \|B_{k+1} - \nabla^2 f(x_*)\|_F &= \|B_k + \Delta_k - \nabla^2 f(x_*)\|_F \\ &\leq \|B_k - \nabla^2 f(x_*)\|_F + \|\Delta_k\|_F. \end{aligned} \tag{25}$$

由于 $\|s_k\| \neq 0$, 因此必存在 $\rho > 0$, 使得 $\|s_k\| > \rho$, 同时有 $(\tilde{s}_k)^2 \geq \frac{\rho^2}{n}$ 且

$$\text{tr}(S_k^2) = \sum_{i=1}^n (s_k^i)^4 \geq (\tilde{s}_k)^4 = (\tilde{s}_k)^2 (n(\tilde{s}_k)^2) \frac{1}{n} \geq \frac{\rho^2}{n} \|s_k\|^2 \frac{1}{n}, \tag{26}$$

其中 \tilde{s}_k 为向量 s_k 的最大值分量. 注意到

$$\begin{aligned} \|s_k\| &= \|x_{k+1} - x_k\| = \|x_{k+1} - x_* + x_* - x_k\| \\ &\leq \|x_{k+1} - x_*\| + \|x_* - x_k\| \leq 2\nu_k. \end{aligned} \tag{27}$$

由引理1 可得 $\|\Delta_k\|_F = \|B_{k+1} - B_k\|_F < C_0\sqrt{n}$, 进而有

$$\begin{aligned} \|\Delta_k\|_F &= \left\| \frac{s_k^T B_{k+1} s_k - s_k^T B_k s_k}{\text{tr}(S_k^2)} S_k \right\|_F = \left\| \frac{s_k^T (B_{k+1} - B_k) s_k}{\text{tr}(S_k^2)} S_k \right\|_F \\ &= \frac{\|s_k^T (B_{k+1} - B_k) s_k\|_F \|S_k\|_F}{\text{tr}(S_k^2)} \leq \frac{\|(B_{k+1} - B_k)\|_F \|s_k\|^2}{\sqrt{\text{tr}(S_k^2)}} \leq 2C_0 \frac{n^{\frac{3}{2}}}{\rho} \nu_k. \end{aligned} \tag{28}$$

取(24)中 $\kappa = 2C_0 \frac{n^{\frac{3}{2}}}{\rho}$, 则引理结论得证.

综合以上引理1-2 可得下面结论.

定理 1 若存在两个正的常数 ε 与 δ 满足不等式 $\|x_0 - x_*\| < \varepsilon$ 及 $\|B_0 - \nabla^2 f(x_*)\|_F < \delta$. 那么由更新格式(12)产生的序列 $\{x_k\}$ 为适定的.

定理 2 设 $\{x_k\}$ 是由算法1产生的序列, $f(x)$ 有下界且对任意的 $x_0 \in \mathbf{R}^n$, $\nabla f(x)$ 在水平集

$$L(x_0) = \{x \in \mathbf{R}^n | f(x) \leq f(x_0)\}$$

上存在且一致连续. 则

- (a) 下降方向 d_k 满足与 $-g_k$ 的夹角 θ_k 的余弦值大于某个常数 $\lambda \in (0, 1)$;
 (b) $\{x_k\}$ 的任意聚点 x_* 都满足 $\nabla f(x_*) = 0$.

证 (a) 由(19)式中步长 d_k 的选取可知, 若 $\|g_k\| \neq 0$, 当 $(B_k)_{ii} \geq \varepsilon_2 > 0 (i = 1, 2, \dots)$, 则

$$\cos \theta_k = \frac{-g_k^T d_k}{\|g_k\| \|d_k\|} = \frac{\sum_{i=1}^n \frac{(g_k)_i^2}{(B_k)_{ii}}}{\sqrt{\sum_{i=1}^n (g_k)_i^2} \sqrt{\sum_{i=1}^n \frac{(g_k)_i^2}{(B_k)_{ii}}}} > 0. \quad (29)$$

否则当 $(B_k)_{ii} \leq 0$, 由算法可知取步长 $d_k = -g_k$, 从而 $\cos \theta_k = 1$. 综上可知 $\cos \theta_k > \lambda > 0$, 其中 $\lambda \in (0, 1)$.

(b) 用反证法, 设 x_* 是 $\{x_k\}$ 的聚点且 $\nabla f(x_*) \neq 0$. 由条件可知 $f(x_k) \rightarrow f(x_*)$ 且 $f(x_k) - f(x_{k+1}) \rightarrow 0$. 又由(14)式可得

$$-\sigma g_k^T s_k \rightarrow 0, g_k^T s_k \rightarrow 0, \quad (30)$$

其中 $s_k = \beta^{m_k} d_k$. 若 $g_k \rightarrow 0$, 则由(30)式可知, $\|s_k\| \rightarrow 0$. 又由(14)可得, 对于 $\beta^{m_k-1} = \frac{\beta^{m_k}}{\beta}$, 不等式

$$f(x_k + \beta^{m_k-1} d_k) - f(x_k) > \sigma \beta^{m_k-1} g_k^T d_k. \quad (31)$$

注意到 $\beta^{m_k-1} d_k = \frac{s_k}{\beta}$, 因此(31)式可写成如下形式

$$f(x_k + \frac{s_k}{\beta}) - f(x_k) > \sigma g_k^T (\frac{s_k}{\beta}). \quad (32)$$

若令 $p_k = \frac{s_k}{\|s_k\|}$, 则 $\frac{s_k}{\beta} = \frac{\|s_k\|}{\beta} p_k$. 由 $\|s_k\| \rightarrow 0$, 可得 $\hat{\alpha}_k = \frac{\|s_k\|}{\beta} \rightarrow 0$. 同时(32)式可进一步写成

$$\frac{f(x_k + \hat{\alpha}_k p_k) - f(x_k)}{\hat{\alpha}_k} > \sigma g_k^T p_k. \quad (33)$$

由于 $\|p_k\| = 1$, 因此 $\{p_k\}$ 有界, 从而存在收敛子列. 不失一般性, 仍记为 $\{p_k\} \rightarrow p_*$ ($\|p_*\| = 1$). 将(33)两边取极限得

$$\nabla f(x_*)^T p_* \geq \sigma \nabla f(x_*)^T p_*. \quad (34)$$

故有

$$\nabla f(x_*)^T p_* \geq 0. \quad (35)$$

注意到 $p_k = \frac{s_k}{\|s_k\|} = \frac{d_k}{\|d_k\|}$, 因此

$$-g_k^T p_k = -g_k^T \left(\frac{d_k}{\|d_k\|} \right) = \|g_k\| \cos \theta_k \geq \lambda \|g_k\|. \quad (36)$$

对上式求极限可得

$$-\nabla f(x_*)^T p_* \geq \lambda \|\nabla f(x_*)\| > 0.$$

即 $\nabla f(x_*)^T p_* < 0$, 与(35)矛盾. 因此有 $\nabla f(x_*) = 0$.

下面的定理将说明: 在算法1中的步5, 借助(20)式中的Aitken迭代格式, 将对算法DNRTR^[1]执行了有益的改善和加速.

定理 3 假设算法1(即AADQN)满足引理1-2以及定理2的条件, 则由算法产生的序列 $\{x_k\}$ 收敛于优化问题(1)的解 x_* 且收敛阶高于算法DNRTR.

证 显然由引理1-2以及定理2可知算法1产生的序列 $\{x_k\}$ 收敛于优化问题(1)的解 x_* . 由文献[1]可知算法DNRTR至少为 q -线性收敛的. 下面证明算法AADQN所产生的序列 $\{x_k\}$ 收敛阶高于算法DNRTR.

由算法AADQN可知

$$\lim_{k \rightarrow \infty} \frac{x_{k+2}^i - x_*^i}{x_{k+1}^i - x_*^i} = \lim_{k \rightarrow \infty} \frac{x_{k+1}^i - x_*^i}{x_k^i - x_*^i} = \theta, 0 < \theta < 1, k = 0, 1, \dots \quad (37)$$

进一步, 由极限的性质可得

$$\begin{aligned} x_{k+2}^i - x_*^i &= (\theta + \epsilon_{k+1})(x_{k+1}^i - x_*^i) \\ &= (\theta + \epsilon_{k+1})(\theta + \epsilon_k)(x_k^i - x_*^i) \\ &:= \hat{\theta}_k(x_k^i - x_*^i), \end{aligned} \quad (38)$$

其中 $\hat{\theta}_k := \theta_{k+1}\theta_k, \theta_{k+1} := \theta + \epsilon_{k+1}, \theta_k := \theta + \epsilon_k$ 且 $\epsilon_k \rightarrow 0$, 当 $k \rightarrow \infty$.

为了方便, 记算法AADQN 步5 中的 $k = \bar{k}$. 由(38)以及(17)则有

$$\bar{x}^i - x_*^i = \hat{\theta}_k(x_k^i - x_*^i) - \frac{(\hat{\theta}_k - \theta_k)^2 (x_k^i - x_*^i)^2}{((\theta_{k+1} - 1)\theta_k - (\theta_k - 1))(x_k^i - x_*^i)} \quad (39)$$

$$= \left[\hat{\theta}_k - \frac{(\hat{\theta}_k - \theta_k)^2}{((\theta_{k+1} - 1)\theta_k - (\theta_k - 1))} \right] (x_k^i - x_*^i). \quad (40)$$

因此

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{\bar{x}^i - x_*^i}{(x_k^i - x_*^i)} &= \lim_{k \rightarrow \infty} \hat{\theta}_k - \frac{(\hat{\theta}_k - \theta_k)^2}{((\theta_{k+1} - 1)\theta_k - (\theta_k - 1))} \\ &= \theta^2 - \frac{(\theta(\theta - 1))^2}{(\theta - 1)\theta - (\theta - 1)} \\ &= 0. \end{aligned} \quad (41)$$

§3 数值实验

本节给出一些数值例子来验证§2提出的算法1(AADQN)在求解无约束优化问题(1)时的高效性. 通过与文献[1]中的算法DNRTR 在更方面性能指标进行详细对比. 这些指标包括迭代次数(记为“IT”), 消耗的CPU时间(记为“CPU”), 梯度范数(记为“GN”)以及函数值(记为“VAL”). 终止准则为 $GN = \|g(x_k)\|_2 < 10^{-6}$ 或迭代次数超过 $k_{\max} = 500$, 其中 x_k 表示第 k 步迭代点. 数值实验的机器环境为Intel(R)Core(TM) i7-2670QM, CPU 2.20GHZ, 内存8GB的Windows10操作系统.

为了全面比较两个算法的综合性能, 选择了无约束优化问题的10个不同测试函数, 这些测试函数源于文献[25], 其中10个不同的测试函数也充分地展现了其多样性和非线性程度. 将这些测试函数的具体形式列在表1中, 同时给出不同的初始点信息.

表 1 测试函数

函数名称	函数解析式	初始点
Quadratic QF1	$f(x) = \frac{1}{2} \sum_{i=1}^n ix_i^2 - x_n$	$x_0 = [1, 1, \dots, 1]$
Hager	$f(x) = \sum_{i=1}^n (\exp(x_i) - \sqrt{ix_i})$	$x_0 = [1, 1, \dots, 1]$
LIARWHD	$f(x) = \sum_{i=1}^n 4(-x_1 + x_i^2)^2 + \sum_{i=1}^n (x_i - 1)^2$	$x_0 = [4, 4, \dots, 4]$
Diagonal 6	$f(x) = \sum_{i=1}^n \exp(x_i) - (1 - x_i) $	$x_0 = [1, 1, \dots, 1]$
QUARTC	$f(x) = \sum_{i=1}^n (x_i - 1)^4$	$x_0 = [2, 2, \dots, 2]$
Pertured Quadratic	$f(x) = \sum_{i=1}^n ix_i^2 + \frac{1}{100} (\sum_{i=1}^n x_i)^2$	$x_0 = [0.5, 0.5, \dots, 0.5]$
Raydan 2	$f(x) = \sum_{i=1}^n (\exp(x_i) - x_i)$	$x_0 = [1, 1, \dots, 1]$
EG2	$f(x) = \sum_{i=1}^{n-1} \sin(x_1 + x_i^2 - 1) + \frac{1}{2} \sin(x_n^2)$	$x_0 = [1, 1, \dots, 1]$
TRIDIA	$f(x) = \gamma(\delta x_1 - 1)^2 + \sum_{i=2}^n i(\alpha x_i - \beta x_{i-1})^2$	$x_0 = [1, 1, \dots, 1]$
FIETCHCR	$f(x) = c \sum_{i=1}^{n-1} i(x_{i+1} - x_i + 1 - x_i^2)^2$	$x_0 = [2, 2, \dots, 2]$

所有的数值测试结果在表2-4以及图1-2中展示.

表2-4中列出两种算法的收敛性对比结果. 事实上, 可以给出更多不同参数下的收敛性状况. 本节给出当参数 $\alpha = 2, \beta = 1, \gamma = 1, \delta = 2, c = 100$ 的情形. 从相关收敛性指标中, 注意到VAL未必都会趋向于零, 原因是有些测试函数并非在零处取得其最小值. 在图1-图2中, 显示了在不用维数下算法AADQN的迭代序列 $\{x_k\}$ 的梯度范数(GN)下降速度非常迅速, 且明显优于算法DNRTR, 特别是针对Pertured Quadratic函数维数较高($n = 10000$)情形, 算法AADQN的迭代次数均小于30次, 而算法DNRTR都超过了500次. 这种高阶收敛的理论依据在定理3中已给出具体证明. 同时在表2-表4中也注意到对于大部分测试函数, 在计算成本方面, 算法AADQN明显小于算法DNRTR.

总而言之, 两种方法均收敛到无约束优化问题(1)的解 x_* . 然而, 无论从CPU时间与迭代次数, 亦或者是GN与VAL, 都明显说明了算法AADQN优于算法DNRTR. 这也意味着算法AADQN在求解无约束优化问题(1)时是可行且有效的.

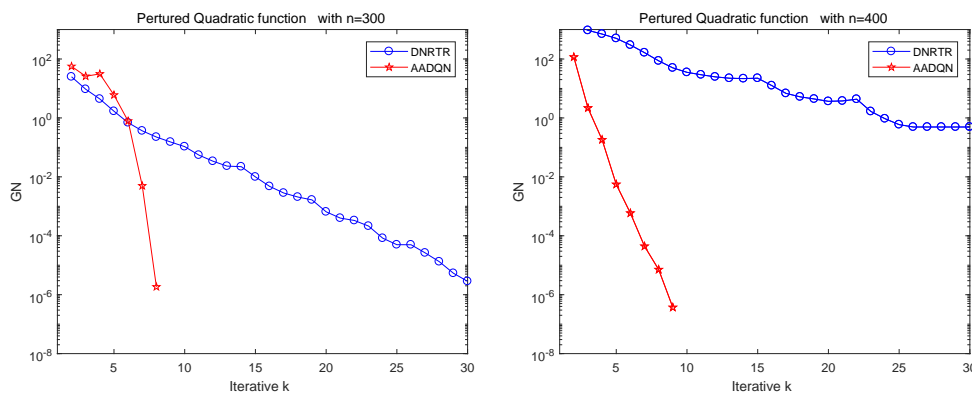


图 1 算法DNRTR 与AADQN 的收敛性轨迹, $n = 300, 400$

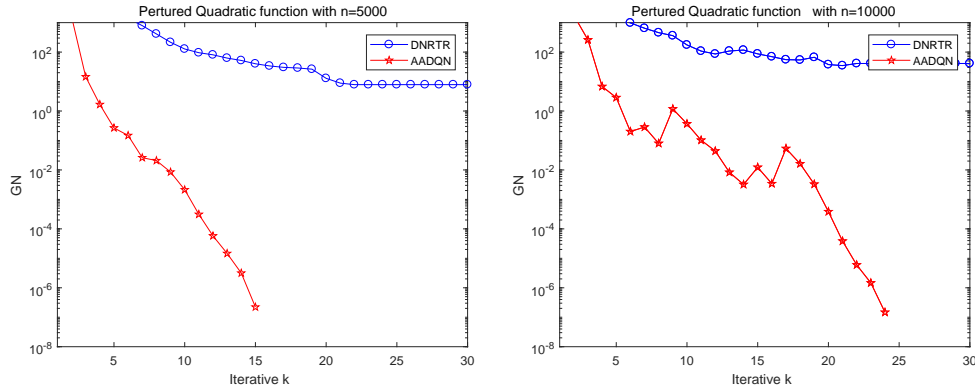


图2 算法DNRTR 与AADQN 的收敛性轨迹, $n = 5000, 10000$

表2 数值结果 $n = 300$

方法	性能	Quadratic QF1	Hager	LIARWHD	Diagonal 6	QUARTC
DNRTR	IT	46	56	9	26	16
	CPU	0.2144	0.2757	0.1047	0.1425	0.1380
	GN	8.1542e-07	6.0347e-07	6.0934e-07	7.3537e-07	5.3493e-07
	VAL	-9.9750e-01	-2.4932e+03	1.5343e-16	1.0400e-05	1.9713e-11
AADQN	IT	2	8	6	5	9
	CPU	0.0373	0.0864	0.0743	0.0647	0.0889
	GN	2.4292e-13	8.0418e-07	1.4949e-08	1.7945e-09	1.4784e-09
	VAL	-9.9750e-01	-2.4932e+03	4.5059e-18	1.3092e-08	4.1663e-14

表3 数值结果 $n = 300$

方法	性能	Pertured Quadratic	Raydan 2	EG2	TRIDIA	FIETCHCR
DNRTR	IT	55	26	136	500	500
	CPU	0.2768	0.1377	1.345	0.2818	0.5384
	GN	6.4819e-07	7.3537e-07	7.1281e-07	1.3759e-02	1.5385e-02
	VAL	2.8509e-15	2.0000e+02	-2.9900e+02	2.9668e-05	3.9259e-04
AADQN	IT	9	4	46	148	27
	CPU	0.0739	0.0459	0.1831	0.0831	0.1358
	GN	2.0916e-07	1.8375e-08	1.6651e-07	9.8115e-07	9.3134e-07
	VAL	3.6236e-17	2.0000e+02	-1.9900e+02	7.8500e-14	4.8987e-16

表 4 Pertured Quadratic 函数在不同维数情形下的数值结果

方法	性能	$n=10000$	$n=5000$	$n=3000$	$n=2000$	$n=1000$
DNRTR	IT	500	500	500	500	500
	CPU	88.8881	69.8945	19.1826	9.7055	8.8819
	GN	2.4688e-01	2.6674e-01	2.3535e-02	2.4551e-02	2.1042e-02
	VAL	3.4860e-03	6.0295e-03	2.7675e-04	3.8784e-05	3.103e-05
AADQN	IT	24	15	11	11	10
	CPU	15.0198	11.4774	1.2449	0.9443	0.3841
	GN	1.4704e-07	2.2155e-07	5.1081e-07	5.6918e-07	4.2393e-07
	VAL	3.3837e-15	2.4842e-15	5.4153e-17	3.2043e-17	8.4398e-17

§4 结论

本文基于对角拟牛顿更新, 提出一个求解无约束优化问题的修正Aitken加速迭代算法. 这种新思想的获得是源于在搜索方向更新时借助了对角矩阵, 从而具有互相独立的“点对点”更新特性, 因此易于考虑引入一些有效的加速技巧. 算法的高阶收敛性优点在理论上得到验证. 数值实验结果也进一步说明了, 所提出的算法在求解无约束优化问题时是很有意义且非常高效的.

参考文献:

- [1] Andrei N. A diagonal quasi-Newton updating method for unconstrained optimization[J]. Numer Algor, 2019, 81: 575-590.
- [2] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1997.
- [3] 李董辉, 童小娇, 万中. 数值最优化理论与方法[M]. 北京: 科学出版社, 2010.
- [4] 张立卫, 单峰. 最优化方法[M]. 北京: 科学出版社, 2010.
- [5] 王宜举, 修乃华. 非线性最优化理论与方法[M]. 北京: 科学出版社, 2012.
- [6] 马昌凤, 柯艺芬, 谢亚君. 最优化计算方法及其Matlab 程序设计[M]. 北京: 国防工业出版社, 2015.
- [7] 戴琰虹, 袁亚湘. 非线性共轭梯度法[M]. 上海: 上海科学技术出版社, 2000.
- [8] Dennis J E Jr, More J J. Quasi-newton methods, motivation and theory[J]. SIAM Rev, 1977, 19: 46-89.
- [9] Nash S G. Preconditioning of truncated-Newton methods[J]. SIAM J Sci Stat Comput, 1985, 6: 599-616.
- [10] Dennis J, Wolkowicz H. Sizing and least-change secant methods[J]. SIAM J Numer Anal, 1993, 30: 1291-1314.
- [11] Amini K, Rizi A G. A new structured quasi-Newton algorithm using partial information on Hessian[J]. J Comput Appl Math, 2010, 234: 805-811.
- [12] Weerakoon S, Fernando T G I. A variant of Newton's method with accelerated third-order convergence[J]. Appl Math Lett, 2000, 13(8): 87-93.
- [13] Jiang Yu, Liu Qiang, Cao Hui. Peak detection based on FPGA using quasi-Newton optimization method for Femtosecond laser ranging[J]. IEEE Access, 2020, 8(99): 47776-47786.
- [14] Liu Jianmei, Ma Shuaiqi. A modified truncated quasi-Newton method based on BFGS formula for the stochastic user equilibrium problem[J]. J Shandong University(Engineering Science), 2018, 48(1): 36-41.

- [15] Liu Jia, Liu Qiang. Speed and resource optimization of BFGS quasi-Newton implementation on FPGA using inexact line search method for neural network training[A]// 2018 International Conference on Field-Programmable Technology (FPT)[C]. 2018.
- [16] Zhou Weijun. A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems[J]. J Comput Appl Math, 2020, 367: 112454.
- [17] Byrd R H, Gilbert J C, Nocedal J. A trust region method based on interior point techniques for nonlinear programming[J]. Math Prog, 2000, 89(1): 149-185.
- [18] Chen Xiaojun, Niu Lingfeng, Yuan Yaxiang. Optimality conditions and a smoothing trust region newton method for nonlipschitz optimization[J]. SIAM J Optim, 2013, 23(3): 1528-1552.
- [19] Yuan Yaxiang. Recent advances in trust region algorithms[J]. Math Prog, 2015, 151(1): 249-281.
- [20] Dai Yuhong, Yuan Jinyun, Yuan Yaxiang. Modified two-point stepsize gradient methods for unconstrained optimization[J]. Comput Optim Appl, 2002, 22(1): 103-109.
- [21] Yuan Yaxiang. A modified BFGS algorithm for unconstrained optimization[J]. IMA J Numeri Analysis, 1991, 11(3): 325-332.
- [22] Liu Dong, Nocedal J. On the limited memory BFGS method for large scale optimization[J]. Math Prog, 1989, 45(3): 503-528.
- [23] Nocedal J. Updating quasi-Newton matrices with limited storage[J]. Math Comput, 1980, 35(151): 773-782.
- [24] Dennis J E Jr, Schnabel R B. Numerical methods for unconstrained optimization and nonlinear equations[M]. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.
- [25] Andrei N. An unconstrained optimization test functions collection[J]. Adv Model Optim, 2008, 10(1): 147-161.

An Aitken acceleration iteration method for solving unconstrained optimization problem

XIE Ya-Jun¹, YAO Jie¹, MA Chang-Feng²

- (1. School of Technology, Fuzhou University of International Studies and Trade, Fuzhou 350202, China;
2. College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China)

Abstract: Quasi-Newton iteration is a class of popular method for unconstrained optimization problem. In this research, based on so-called diagonal quasi-Newton updating method proposed by Neculai(2019), a modified Aitken acceleration approach is firstly investigated to solve unconstrained optimization problem. The performance is ensured theoretically at least higher-order rate of convergence in comparison to diagonal quasi-Newton updating method. Numerical experiments illustrate the efficiency of the proposed method.

Keywords: unconstrained optimization; modified Aitken acceleration iteration; higher-order rate of convergence; numerical tests

MR Subject Classification: 65B