A theoretical analysis on efficiency of some Newton-PCG methods

DENG Naiyang[1], ZHANG Jianzhong[2] & ZHONG Ping[1]

1. China Agricultural University, Beijing 100083, China;
2. City University of Hong Kong, Hong Kong, China
Correspondence should be addressed to Zhang Jianzhong (email: mazhang@cityu.edu.hk)
Received April 19, 2004; revised January 30, 2005

Abstract In this paper, we study the efficiency issue of inexact Newton-type methods for smooth unconstrained optimization problems under standard assumptions from theoretical point of view by discussing a concrete Newton-PCG algorithm. In order to compare the algorithm with Newton's method, a ratio between the measures of their approximate efficiencies is investigated. Under mild conditions, it is shown that first, this ratio is larger than 1, which implies that the Newton-PCG algorithm is more efficient than Newton's method, and second, this ratio increases when the dimension n of the problem increases and tends to infinity at least at a rate $\ln n/\ln 2$ when $n \to \infty$, which implies that in theory the Newton-PCG algorithm is much more efficient for middle- and large-scale problems. These theoretical results are also supported by our preliminary numerical experiments.

Keywords: **unconstrained optimization, Newton's method, preconditioned conjugate gradient method, efficiency.**

DOI: 10.1360/04ys0090

1 Introduction

We consider local algorithms for solving middle- and large-scale unconstrained optimization problems

$$\min f(x), \qquad x \in R^n, \tag{1.1}$$

where $f : R^n \to R$ is smooth and its Hessian $\nabla^2 f$ is available. A popular choice for the purpose is truncated Newton methods in which the Newton equation is allowed to be solved approximately. In fact, some variants of the linear conjugate gradient or preconditioned conjugate gradient (CG or PCG) methods are usually used for the purpose, and thus we have the Newton-CG method or Newton-PCG method. Over the past two decades, a great achievement has been obtained in this field: many elegant and powerful algorithms have been developed and a solid convergence theory has been derived (see refs. [1–9]). However, we hardly see any theoretical analysis on the efficiency issue of such algorithms, and in particular there is no any clear conclusion that the Newton-PCG method is more efficient than Newton's method in theory.

In order to show the theoretical superiority in efficiency, a particular Newton-PCG method is constructed in refs. [10, 11]. Their main contribution is to confirm that Newton-

PCG method can be more efficient than Newton's method under rather strong conditions from theoretical point of view. In fact, the method is compared with Newton's method in the following two aspects:

(1) Computation cost: Notice that the cost in every iteration of both Newton's method and the method in ref. [10] consists of two parts: forming the Newton equation by computing the derivative information of the objective function (i.e. evaluating a Hessian and a gradient) and solving the Newton equation by CF method or PCG method. It is proved that, for the method in ref. [10], the average cost of the second part is less than the corresponding one of Newton's method while the cost of the first part keeps the same.

(2) Convergence rate: It is proved that, if Newton's method is precisely quadratically convergent, the method in ref. [10] is also precisely quadratically convergent.

In a word, in theory the method in ref. [10] is more efficient than Newton's method, but only for the particular problem (1.1) that meets the following two restrictions:

(1) The cost of arithmetic computation to solve the Newton equation is dominant and the cost to compute a Hessian and a gradient is negligible.

(2) Newton's method is precisely quadratically convergent.

Also, the method of ref. [10] has not been verified by any numerical computation.

This paper removes the above restrictions and shows the theoretical efficiency of Newton-PCG method for the general problems (1.1).

Removing the first restriction implies that the cost to compute the derivative information has to be reduced. This can be implemented by the fact (see, e.g. Subsection 12.3 of ref. [7]) that in a PCG step the Hessian-vector product $\nabla^2 f \cdot q$ can be approximated by a difference quotient of ∇f along the direction q, which needs only one extra gradient evaluation. As such difference quotient provides only some approximation to the product $\nabla^2 f \cdot q$, in order to support this modification, the theoretical discussion and numerical experiments are necessary.

Removing the second restriction implies that the problem (1.1) is solved under the standard assumptions, which only ensure that the local convergence rate of Newton's method is at least quadratic. So we have to investigate the progress of the preconditioned conjugate gradient subiterations under some uncertain circumstance, and find how the progress is influenced by the preconditioner and the subiterations. This is completed by a careful theoretical analysis via introducing a concept "progress index".

Instead of proposing an all-purpose and fully complete algorithm, the main aim of this paper is to discuss the efficiency problem for the Newton-PCG method based on the definition of the efficiency measure given by Ostrowski in ref. [12]. We are going to build an algorithm model that contains several parameters. Note that, according to ref. [12], the efficiency of an algorithm is a local property in a neighborhood of a solution because it de-

pends on the local convergence rate and is irrelevant to its global performance. Therefore, for simplicity, our algorithm model is a local one without any globalization strategy. For the algorithm model, by an extension of the efficiency measure in ref. [12] and using the "progress index" introduced in this paper, a lower bound of its efficiency is obtained. This bound depends on the parameters in the algorithm model. Maximizing this bound leads to a concrete realization of the algorithm model. The theoretical superiority of Newton-PCG method in efficiency is shown by proving that this concrete algorithm is significantly more efficient than Newton's method for middle- and large-scale problems in terms of the efficiency bound.

Compared with ref. [10], this paper is not only able to deal with the general problem (1.1) without the above two restrictions, but also an improvement on the result of ref. [10] even for the particular problem (1.1) with the above two restrictions. Furthermore, differently from ref. [10], the theoretical conclusion of this paper is supported by numerical experiments.

This paper is organized as follows: In Section 2, an algorithm model with some parameters is established. Its efficiency is estimated in Section 3. A concrete realization of the model is proposed in Section 4. In Section 5, the efficiency of the concrete algorithm is studied. Some conclusions are given in Section 6.

2 An algorithm model

Obviously, we cannot analyze the efficiency quantitatively without specifying a particular type of Newton-PCG method. So, to start with this research, we are going to describe a local Newton-PCG algorithm model in this section. The algorithm model consists of cycles. Every cycle contains two kinds of steps:

(1) CF step: solving the Newton equation,

$$\nabla^2 f(x) \cdot s = \nabla f(x), \tag{2.1}$$

by the Cholesky factorization method;

(2) PCG$_d$ step: solving the Newton equation (2.1) by the preconditioned conjugate gradient method, where the matrix-vector product $\nabla^2 f(x)q$ is approximated by the difference quotient

$$\nabla^2 f(x)q \approx \frac{\nabla f(x+hq) - \nabla f(x)}{h}.$$

More precisely, as an approximate solution to the Newton equation (2.1), we find $\widetilde{s}$ by the following algorithm PCG$_d$(C, x, l_c, e, h), where C is the preconditioner, x is the current point, l_c is the maximum number of subiterations, h is the step size, and e is a scalar used in the termination criterion which is introduced by the following consideration: Suppose that $\{x_k\}$ is generated by $x_{k+1} = x_k + s_k$, where s_k is an approximate solution to the Newton equation (2.1). According to ref. [2], the progress of every step can be controlled by the corresponding residual $r(s_k)$. For example, Theorem 3.3 in ref. [2]

says that $\{x_k\}$ converges with at least q-order $1+e$ if and only if the residual satisfies $\|r_k(s_k)\| = O(\|\nabla f(x_k)\|^{1+e})$. This observation motivates us to use e in the formula (2.2) below.

Algorithm PCG$_d$(C, x, l_c, e, h)

Step 0. Initial data: set initial $s_0 = 0$ and $r_0 = \nabla f(x)$. Set $i = 0$.

Step 1. Termination test: if the approximate residual r_i at s_i satisfies

$$\|r_i\| \leqslant \|\nabla f(x)\|^{1+e} \qquad \text{or} \qquad i = l_c, \tag{2.2}$$

go to Step 4.

Step 2. Subiteration:

(a) set $z_i = Cr_i$, $t_i = z_i^T r_i$;

(b) if $i = 0$, then $q_0 = -z_0$; else

$$\beta_i = z_i^T w_{i-1}/q_{i-1}^T w_{i-1}, \tag{2.3}$$

and

$$q_i = -z_i + \beta_i q_{i-1}; \tag{2.4}$$

(c) set $s_{i+1} = s_i + \lambda_i q_i$, where $\lambda_i = t_i/q_i^T w_i$ and

$$w_i = \frac{\nabla f(x + hq_i/\|q_i\|) - \nabla f(x)}{h}\|q_i\|; \tag{2.5}$$

(d) set

$$r_{i+1} = r_i + \lambda_i w_i. \tag{2.6}$$

Step 3. Set $i = i + 1$ and go to Step 1.

Step 4. Set $\tilde{s} = s_i$, and stop. □

Our algorithm model has the parameters

$$p, \qquad l = (l_1, \cdots, l_p), \qquad \text{and} \qquad \alpha = (\alpha_1, \cdots, \alpha_p), \tag{2.7}$$

where p is such a nonnegative integer that each cycle contains $p+1$ steps—one CF step is followed by p PCG$_d$ steps; the positive integers $l_1, \cdots, l_p$ are respectively the maximum numbers of subiterations in these p PCG$_d$ steps; the positive scalars $\alpha_1, \cdots, \alpha_p$ are used in the termination test of these PCG$_d$ steps. Using Algorithm PCG$_d(\cdot)$, the following algorithm model, called Algorithm CP(p, l, α), is established.

Algorithm CP(p, l, α)

The algorithm consists of cycles. Each cycle generates $p+1$ iterates x_1, x_2, $\cdots$, x_{p+1} from a starting point x_{CF}. For the first cycle, the starting point is the initial point (guess) x^0. For the later cycles, the starting point is the last iterate x_{p+1} obtained by the previous cycle. All of the cycles have the same structure. Thus, to show the algorithm model, it is enough to describe one cycle in detail.

The cycle

Step 0. Initial data: set the starting point $x_0 = x_{CF} \in R^n$. Set $k = 0$.

Step 1. Termination test: if $k = p + 1$ or $\nabla f(x_k) = 0$, stop.

Step 2. Switch test: if $k = 0$, go to Step 3; otherwise go to Step 4.

Step 3. CF step: find the solution s_k to the Newton equation

$$\nabla^2 f(x_k)s = -\nabla f(x_k) \tag{2.8}$$

by Cholesky factorization $\nabla^2 f(x_k) = L_k D_k L_k^T$. Set

$$B = L_k D_k L_k^T. \tag{2.9}$$

Set $m = 0$, and go to Step 5.

Step 4. PCG$_d$ step: set $m := m + 1$. Select the step size h such that

$$0 < h \leqslant \|\nabla f(x_k)\|^{\max\{\frac{l_m}{\alpha_m}, \frac{2}{\alpha_m}\}}. \tag{2.10}$$

Find $\tilde{s}$ by Algorithm PCG$_d$(B^{-1}, x_k, l_m, l_m/α_m, h). Set $s_k = \tilde{s}$.

Step 5. Update the iterate: set $x_{k+1} = x_k + s_k$. Set $k := k + 1$ and go to Step 1.

□

By repeating the cycles described above, we get the sequence $\{x^k\}$ generated by Algorithm CP(p, l, α):

$$\begin{aligned}\{x^k\} \;=\; & \{x_{\mathrm{CF}}^{0(p+1)}, x^{0(p+1)+1}, \cdots, x^{0(p+1)+p}, x^{0(p+1)+p+1} = x_{\mathrm{CF}}^{1(p+1)}, \cdots, \\ & x^{(j-1)(p+1)+(p+1)} = x_{\mathrm{CF}}^{j(p+1)}, x^{j(p+1)+1}, \cdots, x^{j(p+1)+i}, \\ & \cdots, x^{j(p+1)+p}, x^{j(p+1)+(p+1)} = x_{\mathrm{CF}}^{(j+1)(p+1)}, \cdots\},\end{aligned} \tag{2.11}$$

where the subscript CF is to show that at such points we take a CF step to obtain the next point. For brevity, we often express the above sequence as

$$\begin{aligned}\{x^k\} \;=\; & \{x^{j,i}\} = \{x^{0,0}, x^{0,1}, \cdots, x^{0,p}, x^{1,0}, \cdots, x^{j,0}, x^{j,1}, \\ & \cdots, x^{j,i}, \cdots, x^{j,p}, x^{j+1,0}, \cdots\}.\end{aligned} \tag{2.12}$$

3 An efficiency analysis on algorithm CP(p, l,α)

The efficiency of Algorithm CP(p,l,α) is analyzed in this section under the following standard assumptions:

Assumption 1. $\nabla^2 f(x)$ is Lipschitz continuous with the constant L in a neighborhood of the solution x^* to (1.1).

Assumption 2. $\nabla^2 f(x^*)$ is symmetric positive definite.

Our main purpose in this section is to derive a lower bound for the efficiency measure of Algorithm CP(p, l, α), which will be given in Theorem 3.4 below.

An early study on the efficiency of an algorithm appeared in ref. [12], where an intuitive and clear definition was given.

Definition 3.1. Efficiency $\Gamma_{\rm O}$ (Ostrovski[12]). Suppose that the sequence $\{x^k\}$ generated by an algorithm converges to the solution x^* (at least) with q-order ϱ, and the computation cost, $Q[x^k, x^{k+1}]$, required to compute x^{k+1} from x^k, does not depend on k, i.e. $Q[x^k, x^{k+1}] = Q$, where Q is a constant. Then the efficiency $\Gamma_{\rm O}$ of the algorithm (at least) is

$$\Gamma_{\rm O} = \frac{\varrho}{Q}. \tag{3.1}$$

The definition of the efficiency $\Gamma_{\rm O}$ is rather restrictive and cannot be used in this research. We now extend this efficiency measure by introducing "progress index" and "generalized convergence order".

Definition 3.2. Progress index. Let both $x_{\rm CF}$ and x_c be near the solution x^*. The progress index ν from $x_{\rm CF}$ to x_c with respect to x^* is defined as

$$\nu = \nu[x_{\rm CF}, x_c, x^*] = \frac{\ln \|x_c - x^*\|}{\ln \|x_{\rm CF} - x^*\|}. \tag{3.2}$$

Definition 3.3. Generalized convergence q-order. Suppose that the sequence $\{y^k\}$ converges to x^*. Then its generalized convergence q-order is defined as

$$\varrho(\{y^k\}) = \liminf_{k\to\infty} \nu[y^{k-1}, y^k, x^*], \tag{3.3}$$

where $\nu[\cdot,\cdot,\cdot]$ is the progress index defined by (3.2).

Remark. It is not difficult to see that if the sequence $\{y^k\}$ converges to x^* precisely (or at least) with q-order ϱ, then its generalized convergence q-order $\varrho(\{y^k\})$ is equal to (or not smaller than) ϱ. Therefore the generalized convergence q-order is an extension of the usual convergence q-order.

Note that if a sequence $\{x^k\}$ is generated by an algorithm, its any subsequence $\{y^k\}$ can also be considered as a sequence generated by this algorithm. Keeping this observation in mind, Definition 3.1 can be extended as

Definition 3.4. Efficiency Γ. The efficiency Γ of an algorithm is defined by

$$\Gamma = inf_{\{x^k\}} sup_{\{y^k\}\subset\{x^k\}} \frac{\ln \varrho(\{y^k\})}{\limsup_{k\to\infty} Q[y^{k-1}, y^k]}, \tag{3.4}$$

where $\{x^k\}$ is any sequence generated by the algorithm and converges to solution x^*, $\{y^k\}$ is any subsequence of $\{x^k\}$, $\varrho(\{y^k\})$ is the generalized convergence q-order of $\{y^k\}$, and $Q[y^{k-1}, y^k]$ is the computation cost required to compute y^k from y^{k-1}.

For Newton's method, its efficiency is trivial.

Theorem 3.1 The efficiency $\Gamma = \Gamma^{\rm Newton}$ of Newton's method with Cholesky factorization satisfies

$$\Gamma^{\rm Newton} \geqslant \frac{\ln 2}{Q_{\rm Hg} + Q_{\rm CF}} \stackrel{\rm def}{=} v^{\rm N}, \tag{3.5}$$

where $Q_{\rm Hg} = Q_{\rm H} + Q_{\rm g}$, $Q_{\rm H}$ and $Q_{\rm g}$ are respectively the computation costs to evaluate a Hessian and a gradient, and Q_{CF} is the computation cost to solve the Newton equation by Cholesky factorization.

Consider a subsequence of the sequence (2.11) or (2.12) which consists of the last iterates in every cycle, i.e.

$$\{y^j\} = \{x^{(j-1)(p+1)+p+1}\} = \{x_{\mathrm{CF}}^{j(p+1)}\} = \{x^{j,0}\}, \qquad j = 0, 1, 2, \cdots. \tag{3.6}$$

Our next lemma and theorem are concerned with the generalized convergence q-order of the above sequence.

Lemma 3.2. Suppose that Assumptions 1 and 2 are valid. Consider the sequence (2.12) generated by Algorithm CP$(\cdot,\cdot,\cdot)$. Then for any fixed i with $1 \leqslant i \leqslant p+1$,

$$\lim_{j\to\infty} \nu[x^{j,0}, x^{j,i}, x^*] \geqslant \underline{\nu}_i, \tag{3.7}$$

where $\underline{\nu}_i = \underline{\nu}_i(l_1, \cdots, l_{i-1}, \alpha_1, \cdots, \alpha_{i-1})$ is recursively defined by

$$\underline{\nu}_1 = 2, \tag{3.8}$$

$$\underline{\nu}_{q+1} = \psi(l_q, \alpha_q, \underline{\nu}_q), \qquad q = 1, \cdots, p \tag{3.9}$$

with

$$\psi(\xi,\eta,\zeta) = \begin{cases} \zeta + \min(1, \zeta/\eta)\xi, & \text{if } \xi \leqslant \max(\zeta,\eta); \\ 2\zeta, & \text{otherwise.} \end{cases} \tag{3.10}$$

Proof[1)]**.** To give a brief outline, first we state a proposition that is used in the proof.

Proposition. Assume that the progress index ν from $x^{j,0}$ to $x^{j,m}$ $(m = 1, \cdots, p)$ with respect to x^* satisfies

$$\nu = \nu[x^{j,0}, x^{j,m}, x^*] \geqslant 1.$$

Assume also that the increment $\tilde{s} = s^{j,m}$ at $x^{j,m}$ is obtained by a PCG$_d$ step in Algorithm CP$(\cdot,\cdot,\cdot)$, where the step size h satisfies

$$0 < h \leqslant \|\nabla f(x^{j,m})\|^{\max\{\frac{l_m}{\alpha_m}, \frac{2}{\alpha_m}\}}.$$

Then there exists $\delta > 0$ such that when $\|x^{j,0} - x^*\| \leqslant \delta$ and $\|x^{j,m} - x^*\| \leqslant \delta$, the residual $r(\tilde{s}) = \nabla^2 f(x^{j,m})\tilde{s} + \nabla f(x^{j,m})$ satisfies

$$\|r(\tilde{s})\| \leqslant \gamma \|\nabla f(x^{j,m})\|^{1+l_m/\max\{\nu,\alpha_m\}},$$

where γ is a constant which depends only on $\nabla^2 f(x^*)$ and the Lipschitz constant L.

According to the termination condition (2.2), there are two possibilities: either

$$\|r_i\| \leqslant \|\nabla f(x^{j,m})\|^{1+l_m/\alpha_m} \tag{3.11}$$

with $i < l_m$, or the number of the subiterations is

$$i = l_m. \tag{3.12}$$

The above proposition can be proved in the case (3.11) and the case (3.12) respectively.

In order to prove Lemma 3.2, we should find the lower bounds to $\nu[x^{j,0}, x^{j,i}, x^*]$, $i = 1, \cdots, p+1$. It is easy to see that

$$\nu[x^{j,0}, x^{j,1}, x^*] \geqslant 2 + \theta_1^j = \underline{\nu}_1 + \theta_1^j, \tag{3.13}$$

1) For the detail of the proof, see Deng, N. Y., Zhang, J. Z., Zhong, P., On efficiency of Newton-PCG methods with difference quotient—a detailed version, available at http://www.cityu.edu.hk/ma/staff/zhang/DZZ.PDF.

where $\underline{\nu}_1$ is defined by (3.8), and $\lim_{j\to\infty}\theta_1^j = 0$. For $\nu[x^{j,0}, x^{j,2}, x^*]$, we first estimate $\nu[x^{j,1}, x^{j,2}, x^*]$. The Proposition with $m = 1$ says that

$$x^{j,2} = x^{j,1} + s^{j,1}, \tag{3.14}$$

where

$$\begin{aligned}\|r(s^{j,1})\| &= \|\nabla^2 f(x^{j,1})s^{j,1} + \nabla f(x^{j,1})\| \\ &\leqslant \gamma\|\nabla f(x^{j,1})\|^{1+\frac{l_1}{\max\{\nu[x^{j,0},x^{j,1},x^*],\alpha_1\}}},\end{aligned} \tag{3.15}$$

and γ is a constant. Next, using (3.14)–(3.15), we deal with two cases separately:

(1) When $l_1 \leqslant \max\{\nu[x^{j,0}, x^{j,1}, x^*], \alpha_1\}$, it can be shown that, for $\nu[x^{j,0}, x^{j,1}, x^*] \leqslant \alpha_1$ and $\nu[x^{j,0}, x^{j,1}, x^*] > \alpha_1$, we respectively have

$$\nu[x^{j,0}, x^{j,2}, x^*] \geqslant \underline{\nu}_1 + (\underline{\nu}_1/\alpha_1)l_1 + \theta_3^j \tag{3.16}$$

and

$$\nu[x^{j,0}, x^{j,2}, x^*] \geqslant \underline{\nu}_1 + l_1 + \theta_5^j, \tag{3.17}$$

where $\lim_{j\to\infty}\theta_3^j = 0$ and $\lim_{j\to\infty}\theta_5^j = 0$.

(2) When $l_1 > \max\{\nu[x^{j,0}, x^{j,1}, x^*], \alpha_1\}$, it can be shown that

$$\nu[x^{j,0}, x^{j,2}, x^*] \geqslant 2\underline{\nu}_1 + \theta_7^j, \tag{3.18}$$

where $\lim_{j\to\infty}\theta_7^j = 0$.

Thus, combining (3.16), (3.17) and (3.18) yields

$$\nu[x^{j,0}, x^{j,2}, x^*] \geqslant \underline{\nu}_2 + \theta_8^j, \tag{3.19}$$

where $\underline{\nu}_2 = \underline{\nu}_2(l_1, \alpha_1)$ is defined by (3.8)–(3.10), and $\lim_{j\to\infty}\theta_8^j = 0$.

Up to now, we have got the estimates (3.13) and (3.19), namely $\nu[x^{j,0}, x^{j,1}, x^*]$ and $\nu[x^{j,0}, x^{j,2}, x^*]$. In a similar way, we can obtain

$$\nu[x^{j,0}, x^{j,i}, x^*] \geqslant \underline{\nu}_i + \theta_i^j, \qquad i = 3, \cdots, p+1,$$

where $\underline{\nu}_i = \underline{\nu}_i(l_1, \cdots, l_{i-1}, \alpha_1, \cdots, \alpha_{i-1})$ is defined by (3.8)–(3.10), and $\lim_{j\to\infty}\theta_i^j = 0$. □

Theorem 3.3. The subsequence defined by (3.6) is convergent provided that x^0 is close enough to x^*. Furthermore, its generalized convergence q-order, $\varrho(\{y^j\})$, satisfies

$$\varrho(\{y^j\}) \geqslant \underline{\nu}_{p+1} = \underline{\nu}_{p+1}(l, \alpha), \tag{3.20}$$

where $\underline{\nu}_{p+1}(l, \alpha)$ is defined by (3.8)–(3.10) with $l = (l_1, \cdots, l_p)$ and $\alpha = (\alpha_1, \cdots, \alpha_p)$.

Proof. The equalities (3.8)–(3.10) imply that $\underline{\nu}_i \geqslant 2$, $i = 1, \cdots, p+1$. Therefore, using Lemma 3.2 repeatedly, we conclude that the subsequence (3.6) is convergent when x_0 is close enough to x^*. In addition, noticing Definition 3.3, the conclusion (3.20) is a direct result of the particular case of Lemma 3.2 with $i = p+1$. □

The next theorem gives a lower bound of the efficiency Γ to Algorithm CP$(\cdot,\cdot,\cdot)$.

Theorem 3.4. Suppose that Assumptions 1 and 2 are valid. Then the sequence (2.12) generated by Algorithm CP$(\cdot,\cdot,\cdot)$ is convergent provided that x^0 is close enough to x^*. Furthermore, the efficiency Γ of Algorithm CP$(\cdot,\cdot,\cdot)$ satisfies

$$\Gamma \geqslant \underline{\Gamma}(p,l,\alpha) \stackrel{\text{def}}{=} \frac{\ln \underline{\nu}_{p+1}(l,\alpha)}{Q_{\rm Hg}+Q_{\rm CF}+pQ_{\rm g}+(\sum_{t=1}^{p} l_t)(Q_{\rm g}+Q_{\rm I}^-)}, \tag{3.21}$$

where $Q_{\rm Hg}$, $Q_{\rm CF}$ and $\underline{\nu}_{p+1}(l,\alpha)=\underline{\nu}_{p+1}(l_1,\cdots,l_p,\alpha_1,\cdots,\alpha_p)$ are respectively defined in Theorem 3.1 and (3.8)–(3.10), and Q_I^- is the maximum computation cost of one subiteration in a PCG$_d$ step.

Proof. Similar to the proof of Theorem 3.3, the convergence property of the sequence (3.12) comes from Lemma 3.2 directly. So we only need to show (3.21). In fact, we consider the subsequence (3.6). By Theorem 3.3, we conclude that its generalized convergence q-order satisfies

$$\varrho(\{y^k\}) \geqslant \underline{\nu}_{p+1}, \tag{3.22}$$

where $\underline{\nu}_{p+1}$ is defined in Lemma 3.2. On the other hand, $Q[y^{k-1},y^k]$ is the sum of the computation costs of one CF step and p PCG$_d$ steps. In a CF step, it is required to form the Newton equation via evaluating a Hessian and a gradient and to solve this equation by Cholesky factorization. So the computation cost is $Q_{\rm Hg}+Q_{\rm CF}$. However, in a PCG$_d$ step, we need not evaluate the Hessian any more. In fact, in the t-th PCG$_d$ step, an upper bound of the computation cost is $Q_{\rm g}+l_t(Q_{\rm g}+Q_{\rm I}^-)$. Therefore,

$$Q[y^{k-1},y^k] \leqslant Q_{\rm Hg}+Q_{\rm CF}+pQ_{\rm g}+(\sum_{t=1}^{p} l_t)(Q_{\rm g}+Q_{\rm I}^-). \tag{3.23}$$

Combining Definition 3.4, (3.22) and (3.23), we get the inequality in (3.21). The theorem is proved. □

4 A concrete algorithm

In order to propose an implementable and efficient version from the algorithm model, Algorithm CP$(\cdot,\cdot,\cdot)$, it is natural to specify its parameters p, l and α such that these parameters maximize the corresponding efficiency Γ, or its lower bound $\underline{\Gamma}=\underline{\Gamma}(p,l,\alpha)$ given in Theorem 3.4. Note that, besides the parameters p, l and α, $\underline{\Gamma}$ also depends on the quantities $Q_{\rm H}$, $Q_{\rm g}$, $Q_{\rm CF}$ and $Q_{\rm I}^-$. For simplicity, these quantities are measured by the corresponding numbers of multiplicative operations involved, i.e.

$$\begin{aligned} Q_{\rm H}(Q_{\rm g}) = {} & \text{the number of multiplicative operations to} \\ & \text{evaluate a Hessian } \nabla^2 f \text{ (a gradient } \nabla f), \end{aligned} \tag{4.1}$$

$$Q_{\rm CF}=\frac{1}{6}n^3+\frac{3}{2}n^2-\frac{2}{3}n, \qquad Q_{\rm I}^-=n^2+8n+3. \tag{4.2}$$

This leads to the following

Algorithm CP

It is the same as Algorithm CP(p, l, α) except that the parameters and h in (2.10) are specified as follows:

(1) Find the global solution σ^* to the one-dimensional optimization problem

$$\max v(\sigma) = \frac{\ln(2+\sigma)}{Q_{\rm CF} + Q_{\rm Hg} + \bar{p}(\sigma)Q_{\rm g} + \sigma(Q_{\rm g} + Q_{\rm I}^-)}, \tag{4.3}$$

$$\text{s.t. } \sigma \text{ is a nonnegative integer,} \tag{4.4}$$

where $Q_{\rm Hg} = Q_{\rm H} + Q_{\rm g}$, $Q_{\rm H}$, $Q_{\rm g}$, $Q_{\rm CF}$ and $Q_{\rm I}^-$ are defined in (4.1) and (4.2), $\bar{p}(\sigma)$ is the smallest integer not smaller than $\frac{\ln(2+\sigma)}{\ln 2} - 1$.

(2) Define p: $p = p^* = \bar{p}(\sigma^*)$.

(3) Define $l = l^* = (l_1^*, \cdots, l_p^*)$ as follows: if $\sigma^* = 1$, $l_1 = l_1^* = 1$; if $\sigma^* \geqslant 2$,

$$l_m = l_m^* = \begin{cases} 2^m, & m = 1, \cdots, p-1; \\ \sigma^* - 2^p + 2, & m = p. \end{cases}$$

(4) Define $\alpha = (\alpha_1, \cdots, \alpha_p) = \alpha^* = (\alpha_1^*, \cdots, \alpha_p^*)$: $\alpha_m = \alpha_m^* = 2^m$, $m = 1, \cdots, p$.

(5) Select h in (2.10) such that

$$0 < h \leqslant \|\nabla f(x_k)\|. \tag{4.5}$$

Note that since $\max\{\frac{l_m}{\alpha_m}, \frac{2}{\alpha_m}\} \leqslant 1$, for $m = 1, \cdots, p$, (4.5) implies (2.10) when x_k is close to x^*. In order to show the optimality of the selections p^*, l^* and α^* given in the above algorithm, we need the following

Lemma 4.1. Using the convention

$$\sum_{n_1}^{n_2} \cdots = 0, \qquad \text{if} \quad n_1 > n_2, \tag{4.6}$$

the function $\underline{\nu}_{p+1} = \underline{\nu}_{p+1}(l, \alpha)$ defined in Lemma 3.2 has the following properties:

(1) Suppose that α and l respectively satisfy

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \qquad m = 1, \cdots, p, \tag{4.7}$$

and

$$l_m \leqslant 2 + \sum_{t=1}^{m-1} l_t, \qquad m = 1, \cdots, p. \tag{4.8}$$

Then

$$\underline{\nu}_{p+1}(l, \alpha) = 2 + \sum_{t=1}^{p} l_t. \tag{4.9}$$

(2) For any $l = (l_1, \cdots, l_p)$, define $l' = (l'_1, \cdots, l'_p)$ by

$$l'_m = \min\left\{l_m, 2 + \sum_{t=1}^{m-1} l'_t\right\}, \qquad m = 1, \cdots, p. \tag{4.10}$$

Then

$$\underline{\nu}_{p+1}(l, \alpha) = \underline{\nu}_{p+1}(l', \alpha) \leqslant 2 + \sum_{t=1}^{p} l'_t. \tag{4.11}$$

Furthermore, if $\alpha = (\alpha_1, \cdots, \alpha_p)$ satisfies

$$\alpha_m \leqslant 2 + \sum_{t=1}^{m-1} l'_t, \qquad m = 1, \cdots, p, \tag{4.12}$$

then (4.11) is strengthened to

$$\underline{\nu}_{p+1}(l, \alpha) = \underline{\nu}_{p+1}(l'\alpha) = 2 + \sum_{t=1}^{p} l'_t. \tag{4.13}$$

Proof[1)]**.** Its outline is as follows: In order to show the conclusion (1) of Lemma 4.1, it is sufficient to prove, by induction, that

$$\underline{\nu}_{q+1}(l_1, \cdots, l_q, \alpha_1, \cdots, \alpha_q) = 2 + \sum_{t=1}^{q} l_t, \qquad q = 0, 1, \cdots, p.$$

For the conclusion (2) of Lemma 4.1, we prove (4.13) first, and then prove (4.11) by showing that $\underline{\nu}_{p+1}(l, \alpha) = \underline{\nu}_{p+1}(l_1, \cdots, l_p, \alpha_1, \cdots, \alpha_p)$ is nonincreasing with respect to α_t $(t = 1, \cdots, p)$. □

Theorem 4.2. The parameters p^*, l^* and α^* given in Algorithm CP maximize $\underline{\Gamma}(p, l, \alpha)$, i.e. the lower bound of the efficiency of Algorithm CP(p, l, α) given in Theorem 3.4.

Proof[1)]**.** Here we only list its two steps. The first step is to show, by the conclusion (1) of Lemma 4.1, that for (p^*, l^*, α^*), we have

$$\underline{\Gamma}(p^*, l^*, \alpha^*) = v^*, \tag{4.14}$$

where v^* is the optimal value to the problem (4.3)–(4.4).

The second step is to define $\Omega, l' = (l'_1, \cdots, l'_p)$ and $\alpha' = (\alpha'_1, \cdots, \alpha'_p)$ by

$$\begin{aligned}\Omega = \{&(p, l, \alpha) \mid p \in R^1, l = (l_1, \cdots, l_p) \in R^p, \alpha = (\alpha_1, \cdots, \alpha_p) \in R^p,\\ &p \geqslant 0 \text{ integer}, l_1, \cdots, l_p \geqslant 0 \text{ integers}, \alpha_1, \cdots, \alpha_p > 0 \text{ scalars}\},\end{aligned}$$

$$l'_m = \min\left\{l_m, 2 + \sum_{t=1}^{m-1} l'_t\right\}, \quad m = 1, \cdots, p,$$

and

$$\alpha'_m = 2 + \sum_{t=1}^{m-1} l'_t, \qquad m = 1, \cdots, p,$$

1) For the detail of the proof, see Deng, N. Y., Zhang, J. Z., Zhong, P., On efficiency of Newton-PCG methods with difference quotient—a detailed version, available at http://www.cityu.edu.hk/ma/staff/zhang/DZZ.PDF.

and then prove, by the conclusion (2) of Lemma 4.1, that

$$\underline{\Gamma}(p,l,\alpha) \leqslant \underline{\Gamma}(p,l',\alpha) \leqslant \underline{\Gamma}(p,l^{'},\alpha^{'}) \leqslant v^*. \tag{4.15}$$

□

Theorem 4.3. Algorithm CP is well defined if the initial point x^0 is close enough to the solution x^*.

Proof. To prove that Algorithm CP is well-defined, we only need to show the existence of the global solution σ^* to the problems (4.3)–(4.4). In fact, it is easy to see that the objective function $v(\sigma)$ satisfies

$$v(0) = \frac{\ln 2}{Q_{\mathrm{CF}} + Q_{\mathrm{Hg}}} > 0, \tag{4.16}$$

and $v(\sigma) \leqslant \frac{1}{Q_I^-} \cdot \frac{\ln(2+\sigma)}{\sigma}$, which implies that

$$\lim_{\sigma\to\infty} v(\sigma) = 0. \tag{4.17}$$

Combining (4.16) and (4.17) yields the existence of a finite global solution σ^* to (4.3)–(4.4). □

5 Comparison of efficiencies

In this section, we compare the efficiency of Algorithm CP with that of Newton's method under Assumptions 1 and 2 given in Section 3. For Newton's method, a lower bound of the efficiency is given in Theorem 3.1. The corresponding result for Algorithm CP is shown in the following theorem.

Theorem 5.1. The efficiency $\Gamma = \Gamma^{\mathrm{CP}}$ of Algorithm CP satisfies

$$\Gamma^{\mathrm{CP}} \geqslant v^*, \tag{5.1}$$

where $v^* = v(\sigma^*)$ is the optimal value and σ^* is the global solution to the problem (4.3)–(4.4).

Proof. By (4.14) and Theorem 3.4, we have

$$\Gamma^{\mathrm{CP}} \geqslant \underline{\Gamma}(p^*,l^*,\alpha^*) = v^*. \qquad \square$$

For Algorithm CP and Newton's method, Theorem 5.1 and Theorem 3.1 respectively give the lower bounds of their efficiency measures

$$v^* \qquad \text{and} \qquad v^N = \ln 2/(Q_{\mathrm{Hg}} + Q_{\mathrm{CF}}),$$

where v^* is the optimal value to (4.3)–(4.4). In order to compare their efficiency measures, the ratio

$$R^* = R^*(Q_{\mathrm{H}}, Q_{\mathrm{g}}, n) \stackrel{\mathrm{def}}{=} v^*/v^N \tag{5.2}$$

is introduced. Note that both Algorithm CP and Newton's method are the members of our algorithm model Algorithm CP (p,l,α). The estimates of their efficiency measures $v^* = \underline{\Gamma}(p^*,l^*,\alpha^*)$ and $v^N = \underline{\Gamma}(0,\cdot,\cdot)$ are obtained in the same way, where $\underline{\Gamma}(\cdot,\cdot,\cdot)$

is defined in (3.21). Therefore, in a sense, this ratio reflects the relationship between their efficiency measures. If this ratio is larger than one, we can reasonably regard that Algorithm CP is more efficient than Newton's method. The larger this ratio is, the much more efficient Algorithm CP will be. The remainder of this section will estimate this ratio from theoretical point of view.

It is not difficult to observe that R^* defined by (5.2) is the optimal value to the one-dimensional optimization problem

$$\max\ R(\sigma, Q_{\rm H}, Q_{\rm g}, n) = \frac{Q_{\rm CF} + Q_{\rm Hg}}{Q_{\rm CF} + Q_{\rm Hg} + \overline{p}(\sigma)Q_{\rm g} + \sigma(Q_{\rm g} + Q_{\rm I}^-)} \cdot \frac{\ln(2+\sigma)}{\ln 2}, \quad (5.3)$$

$$\text{s.t. } \sigma \text{ is a nonnegative integer}, \quad (5.4)$$

where $\overline{p}(\sigma)$ is defined in the problems (4.3)–(4.4). The estimates to $R^* = R^*(Q_{\rm H}, Q_{\rm g}, n)$ below are based on this observation. The next theorem gives a sufficient condition which ensures that $R^* > 1$.

Theorem 5.2. If

$$n \geqslant 11 \qquad \text{and} \qquad Q_{\rm H} \geqslant 2.42\ Q_{\rm g},$$

we have

$$R^*(Q_{\rm H}, Q_{\rm g}, n) > 1. \quad (5.5)$$

Particularly, for the case $Q_{\rm H} = Q_{\rm g} = 0$, if $n \geqslant 11$ we have

$$R^*(0, 0, n) > 1. \quad (5.6)$$

Proof. To prove (5.5), it is sufficient to prove that, when $n \geqslant 11$ and $Q_{\rm H} \geqslant 2.42\ Q_{\rm g}$,

$$R(1, Q_{\rm H}, Q_{\rm g}, n) > 1, \quad (5.7)$$

where $R(\cdot, \cdot, \cdot, \cdot)$ is defined by (5.3). Or by the fact

$$R(1, Q_{\rm H}, Q_{\rm g}, n) = \frac{\ln 3}{\ln 2} \cdot \frac{Q_{\rm CF} + Q_{\rm H} + Q_{\rm g}}{Q_{\rm CF} + Q_{\rm H} + 3Q_{\rm g} + Q_{\rm I}^-}, \quad (5.8)$$

we only need to show that

$$\left(\ln \frac{3}{2}\right) Q_{\rm CF} - (\ln 2) Q_{\rm I}^- > 0, \quad \text{and} \quad \left(\ln \frac{3}{2}\right) Q_{\rm H} - \left(2\ln 2 - \ln \frac{3}{2}\right) Q_{\rm g} > 0. \quad (5.9)$$

It can be verified that the above two inequalities are valid when $n \geqslant 11$ and $Q_{\rm H} \geqslant 2.42\ Q_{\rm g}$. Therefore, (5.7) is true, and the conclusion (5.5) is proved. The conclusion (5.6) is obtained from (5.5) immediately. □

Next we investigate the behavior of $R^* = R^*(Q_{\rm H}, Q_{\rm g}, n)$ when n increases.

Theorem 5.3. For the case $Q_{\rm H} = Q_{\rm g} = 0$, $R^*(Q_{\rm H}, Q_{\rm g}, n) = R^*(0, 0, n)$ is the optimal value of the problem

$$\max \quad R(\sigma, 0, 0, n) = \frac{\ln(2+\sigma)}{\ln 2} \cdot \frac{1}{1 + \sigma\phi(n)}, \quad (5.10)$$

$$\text{s.t.} \quad \sigma \text{ is a nonnegative integer}, \quad (5.11)$$

where

$$\phi(n) = Q_{\rm I}^{-}/Q_{\rm CF}. \tag{5.12}$$

Furthermore, when $n \geqslant 11$, $R^*(0,0,n)$ is strictly increasing with respect to n.

Proof. By (5.3)–(5.4), $R^*(0,0,n)$ is obviously the optimal value of the problems (5.10)–(5.12). So we only need to show that when $n \geqslant 11$,

$$R^*(0,0,n+1) > R^*(0,0,n). \tag{5.13}$$

Denoting the global solution to (5.10)–(5.12) by $\sigma^*(n)$, (5.13) is equivalent to

$$\frac{\ln(2+\sigma^*(n+1))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n+1)\phi(n+1)} > \frac{\ln(2+\sigma^*(n))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n)\phi(n)}. \tag{5.14}$$

Now we prove (5.14). Since $\sigma^*(n+1)$ is the global solution corresponding to $n+1$,

$$\frac{\ln(2+\sigma^*(n+1))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n+1)\phi(n+1)} \geqslant \frac{\ln(2+\sigma^*(n))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n)\phi(n+1)}.$$

However, as (5.6) implies that when $n \geqslant 11$, $\sigma^*(n) \geqslant 1$ and $\phi(n)$ is strictly decreasing with respect to $n \geqslant 1$, we have

$$\frac{\ln(2+\sigma^*(n))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n)\phi(n+1)} > \frac{\ln(2+\sigma^*(n))}{\ln 2}\cdot\frac{1}{1+\sigma^*(n)\phi(n)}.$$

Combining the above two inequalities, (5.14) is proved. □

Remark. Recall that, for the algorithm in ref. [10], the efficiency is established only for particular problems with the two restrictions mentioned in Section 1, including the restriction $Q_{\rm H} = Q_{\rm g} = 0$. So the above theorem can be used to show the superiority of Algorithm CP over the algorithm in ref. [10]. In fact, the efficiency of Algorithm CP is measured by $R^*(0,0,n)$ and, as shown by the above theorem, $R^*(0,0,n)$ is the optimal value of the problems (5.10)–(5.12). For the algorithm in ref. [10], the counterpart of the ratio $R^*(0,0,n)$, say $r^*(n)$, was also examined. In the language of this paper, $r^*(n)$ was the optimal value of the problem

$$\max \frac{1+p}{1+(2^{p+1}+p-2)Q_{\rm I}/Q_{\rm CF}}, \tag{5.15}$$

$$\text{s.t. } p \text{ is a nonnegative integer}, \tag{5.16}$$

where

$$Q_{\rm I} = 2n^2 + 6n + 2. \tag{5.17}$$

Now we show that Algorithm CP is more efficient in the sense that, when $n \geqslant 12$,

$$R^*(0,0,n) > r^*(n). \tag{5.18}$$

Obviously,

$$\{2^{p+1}-2 \mid p = 0,1,2,\cdots\} \subset \{\sigma | \sigma = 0,1,2,\cdots\}.$$

Therefore, the optimal value $R^*(0,0,n)$ of (5.10)–(5.12) is not smaller than the optimal

value $\underline{R}^*(n)$ of the problem

$$\max \ \frac{\ln(2+\sigma)}{\ln 2} \cdot \frac{1}{1+\sigma Q_{\mathrm{I}}^-/Q_{\mathrm{CF}}}, \tag{5.19}$$

$$\text{s.t.} \quad \sigma \in \{2^{p+1}-2 \mid p=0,1,2,\cdots\}, \tag{5.20}$$

which is equivalent to

$$\max \frac{1+p}{1+(2^{p+1}-2)Q_{\mathrm{I}}^-/Q_{\mathrm{CF}}}, \tag{5.21}$$

$$\text{s.t.} \quad p \text{ is a nonnegative integer.} \tag{5.22}$$

Therefore, in order to prove (5.18), we only need to show that when $n \geqslant 12$,

$$\underline{R}^*(n) > r^*(n). \tag{5.23}$$

Note that $n \geqslant 12$ implies that

$$\frac{1+p}{1+(2^{p+1}-2)Q_{\mathrm{I}}^-/Q_{\mathrm{CF}}}|_{p=1} > 1,$$

and therefore $\underline{R}^*(n) > 1$. Thus the conclusion (5.23) comes from the inequality

$$\frac{1+p}{1+(2^{p+1}-2)Q_{\mathrm{I}}^-/Q_{\mathrm{CF}}} > \frac{1+p}{1+(2^{p+1}+p-2)Q_{\mathrm{I}}/Q_{\mathrm{CF}}}, \quad \forall p \geqslant 1, \tag{5.24}$$

and the superiority of Algorithm CP is proved.

Now let us deal with the general case $Q_{\mathrm{H}} \geqslant 0$ and $Q_{\mathrm{g}} \geqslant 0$. For simplicity, we assume that there is a relationship between Q_{H} and Q_{g}. It is usually assumed that

$$Q_H = nQ_g. \tag{5.25}$$

Here we consider a more general case

$$Q_{\mathrm{H}} = \mu n Q_{\mathrm{g}}, \tag{5.26}$$

where $\mu \in (0,1]$ is a constant. In order to estimate R^*, similar to strengthening the constraint (5.11) to (5.20), we restrict the integer σ in (5.3)–(5.4) in the set $\{2^{p+1}-2 \mid p = 0,1,2,\cdots\}$. Thus, under the assumption (5.26), the problem (5.3)–(5.4) is transformed into the one-dimensional problem

$$\max \ \underline{R}(p,\mu,Q_{\mathrm{g}},n) = \frac{p+1}{1+\varphi(p,\mu,Q_{\mathrm{g}},n)}, \tag{5.27}$$

$$\text{s.t.} \quad p \quad \text{is a nonnegative integer,} \tag{5.28}$$

where

$$\varphi(p,\mu,Q_{\mathrm{g}},n) = \frac{(p+2^{p+1}-2)Q_{\mathrm{g}} + (2^{p+1}-2)Q_{\mathrm{I}}^-}{Q_{\mathrm{CF}} + (\mu n+1)Q_{\mathrm{g}}}. \tag{5.29}$$

Let $\underline{p}^*$ and $\underline{R}^* = \underline{R}^*(\mu, Q_{\mathrm{g}}, n) = \underline{R}(\underline{p}^*, \mu, Q_{\mathrm{g}}, n)$ be the global solution and the optimal value to (5.27)–(5.29), respectively. Obviously,

$$R^*(Q_{\mathrm{H}}, Q_{\mathrm{g}}, n) \geqslant \underline{R}^*(\mu, Q_{\mathrm{g}}, n). \tag{5.30}$$

In other words, the ratio R^* defined by (5.2) has a lower bound $\underline{R}^*$. So our task is transferred to estimating $\underline{R}^*$, the optimal value to the problems (5.27)–(5.29).

Theorem 5.4. Suppose that the assumption (5.26) holds. Then for $n \geqslant 11$ and any $\mu \in [\underline{\mu}, 1]$, where $\underline{\mu} \in (0, 1/16)$, we have

$$R^*(Q_{\mathrm{H}}, Q_{\mathrm{g}}, n) \geqslant \underline{R}^*(\mu, Q_{\mathrm{g}}, n) \geqslant \underline{R}^*(\underline{\mu}, \infty, n). \tag{5.31}$$

Proof. The first inequality of (5.31) is just (5.30). Now we show the second one. It suffices to prove

$$\underline{R}^*(\mu, Q_{\rm g}, n) \geqslant \underline{R}^*(\underline{\mu}, Q_{\rm g}, n) \geqslant \underline{R}^*(\underline{\mu}, \infty, n). \tag{5.32}$$

By (5.27)–(5.29), it is easy to see that the objective function $\underline{R}(p, \mu, Q_{\rm g}, n)$ is increasing with respect to μ. Therefore, for any $\mu \in [\underline{\mu}, 1]$, the first inequality of (5.32) is valid. In order to prove its second one, it suffices to show that $\underline{R}^*(\underline{\mu}, Q_{\rm g}, n)$ is strictly decreasing with respect to $Q_{\rm g}$, or to show

$$\frac{\partial \underline{R}(p, \underline{\mu}, Q_{\rm g}, n)}{\partial Q_{\rm g}} < 0, \tag{5.33}$$

which is equivalent to

$$\frac{Q_{\rm I}^-}{Q_{\rm CF}} < \left(1 + \frac{p}{2^{p+1} - 2}\right)\left(\frac{1}{\underline{\mu} n + 1}\right), \tag{5.34}$$

where $Q_{\rm CF}$ and $Q_{\rm I}^-$ are defined by (4.2). When $\underline{\mu} < 1/16$, we have

$$\left(1 + \frac{p}{2^{p+1} - 2}\right)\left(\frac{1}{\underline{\mu} n + 1}\right) \geqslant \frac{1}{(1/16)n + 1} > \frac{Q_{\rm I}^-}{Q_{\rm CF}}.$$

Therefore, the second inequality of (5.32) is valid. □

Theorem 5.5. $\underline{R}^*(\underline{\mu}, \infty, n)$, defined in Theorem 5.4, is strictly increasing with respect to n when $n > \frac{2}{\underline{\mu}}$.

Proof. By (5.27)–(5.29), it is easy to see that $\underline{R}^*(\underline{\mu}, \infty, n)$ is the optimal value to the optimization problem

$$\max \quad \underline{R}(p, \underline{\mu}, \infty, n) = \frac{(p+1)(\underline{\mu} n + 1)}{\underline{\mu} n + p + 2^{p+1} - 1}, \tag{5.35}$$

$$\text{s.t.} \quad p \text{ is a nonnegative integer.} \tag{5.36}$$

The objective function can be rewritten as

$$\underline{R}(p, \underline{\mu}, \infty, n) = \frac{p+1}{1 + (p + 2^{p+1} - 2)\tilde{\phi}(n)}, \tag{5.37}$$

where

$$\tilde{\phi}(n) = \frac{1}{\underline{\mu} n + 1}. \tag{5.38}$$

When $n > \frac{2}{\underline{\mu}}$, we have

$$\underline{R}(1, \underline{\mu}, \infty, n) = \frac{2}{1 + \frac{3}{\underline{\mu} n + 1}} > 1.$$

Therefore,

$$\underline{R}^*(\underline{\mu}, \infty, n) \geqslant \underline{R}(1, \underline{\mu}, \infty, n) > 1, \tag{5.39}$$

which implies that $\underline{p}^* = \underline{p}^*(n) \geqslant 1$ when $n > \frac{2}{\underline{\mu}}$. Therefore, noticing that $\tilde{\phi}(n)$ defined by (5.38) is strictly decreasing with respect to $n \geqslant 1$, we have that, when $n > \frac{2}{\underline{\mu}}$,

$$\begin{aligned} \underline{R}^*(\underline{\mu}, \infty, n+1) &= \underline{R}(\underline{p}^*(n+1), \underline{\mu}, \infty, n+1) \\ &\geqslant \underline{R}(\underline{p}^*(n), \underline{\mu}, \infty, n+1) \\ &> \underline{R}(\underline{p}^*(n), \underline{\mu}, \infty, n) = \underline{R}^*(\underline{\mu}, \infty, n). \end{aligned}$$

Hence, the conclusion is proved. □

Theorem 5.6. Suppose that the assumption (5.26) holds. Then for any $\widetilde{\mu} \in (0, 1]$,

$$R^*(Q_{\rm H}, Q_{\rm g}, n)|_{Q_{\rm H}=\mu n Q_{\rm g}}$$

tends to infinity at least at a rate $\ln n/\ln 2$ uniformly for $\mu \in [\widetilde{\mu}, 1]$ and $Q_g \geqslant 0$. More precisely, for any $\epsilon > 0$ and $\widetilde{\mu} \in (0, 1]$, there exists N such that when $n > N$,

$$\frac{R^*(Q_{\rm H}(n), Q_{\rm g}(n), n)|_{Q_{\rm H}(n)=\mu n Q_{\rm g}(n)}}{\ln n/\ln 2} \geqslant 1 - \epsilon, \tag{5.40}$$

provided that $\mu \in [\widetilde{\mu}, 1]$ and $Q_{\rm g}(n) \geqslant 0$.

Proof. To prove (5.40), by (5.31), it suffices to prove

$$\underline{R}^*(\underline{\mu}, \infty, n) \sim \ln n/\ln 2. \tag{5.41}$$

In fact, let us consider maximizing $\underline{R}(p, \underline{\mu}, \infty, n)$ defined by (5.35) with a continuous variable p. Then its maximizer $\widehat{p}^*$ satisfies the equation

$$z \ln z = \beta, \tag{5.42}$$

where

$$z = \frac{2^{p+1}}{e}, \tag{5.43}$$

and

$$\beta = \frac{\underline{\mu} n - 2}{e}. \tag{5.44}$$

Notice that $z\ln z$ is an increasing function when $z > e$. So it is easy to see that when $\beta > e$, the solution z^* to (5.42) satisfies that

$$\frac{\beta}{\ln \beta} < z^* < \frac{\beta}{\ln \beta - \ln\ln \beta}. \tag{5.45}$$

Therefore, when $\beta \to \infty$,

$$z^* \sim \frac{\beta}{\ln \beta}, \tag{5.46}$$

and hence by (5.46), (5.44) and (5.43) with z and p there being replaced by z^* and $\widehat{p}^*$ respectively, we have that, when $n \to \infty$,

$$2^{\widehat{p}^*+1} \sim \underline{\mu} n/\ln n, \tag{5.47}$$

and

$$\widehat{p}^* \sim \frac{\ln n}{\ln 2}. \tag{5.48}$$

Obviously, the optimal solution p^* to (5.35)—(5.36) satisfies

$$\widehat{p}^* - 1 \leq p^* \leqslant \widehat{p}^* + 1, \tag{5.49}$$

which implies that

$$2^{p^*+1} \leqslant 2 \cdot 2^{\widehat{p}^*+1}. \tag{5.50}$$

Now we are in a position to prove (5.41). By (5.35)—(5.36),

$$\underline{R}^*(\underline{\mu}, \infty, n) = \frac{p^* + 1}{\frac{\underline{\mu} n - 1}{\underline{\mu} n + 1} + \frac{p^*}{\underline{\mu} n + 1} + \frac{2^{p^*+1}}{\underline{\mu} n + 1}}. \tag{5.51}$$

Obviously, by (5.47)—(5.50) we have that, when $n \to \infty$,

$$p^* + 1 \sim \frac{\ln n}{\ln 2}, \tag{5.52}$$

$$\frac{\underline{\mu} n - 1}{\underline{\mu} n + 1} + \frac{p^*}{\underline{\mu} n + 1} + \frac{2^{p^*+1}}{\underline{\mu} n + 1} \sim 1. \tag{5.53}$$

Thus the validity of (5.41) is obtained from (5.51), (5.52) and (5.53). □

6 Conclusion

A Newton-PCG algorithm model was taken into consideration for analyzing its efficiency by a measure which is an extension of the efficiency defined by Ostrowski in ref. [12]. To implement and enhance the performance of such model, we chose particular values for the parameters in the model and thus a concrete local algorithm is given. In order to compare this algorithm with Newton's method, we introduce the ratio R^* between their approximate efficiency measures:

$$R^* = R^*(Q_{\rm H}, Q_{\rm g}, n) \approx \frac{\text{The efficiency of the concrete algorithm}}{\text{The efficiency of Newton's method}}, \tag{6.1}$$

where $Q_{\rm H} = Q_{\rm H}(n)$ and $Q_{\rm g} = Q_{\rm g}(n)$ are respectively the number of multiplicative operations to evaluate a Hessian $\nabla^2 f$ and a gradient ∇f, and n is the dimension of the problem. In a sense, the value of R^* indicates the superiority of the concrete algorithm: if $R^* > 1$, the algorithm is superior to Newton's method; the larger the value of R^*, the more remarkable the superiority. Theorem 5.2 shows that, under the mild conditions

$$n \geqslant 11, \qquad \text{and} \qquad Q_{\rm H} \geqslant 2.42\, Q_{\rm g}, \tag{6.2}$$

we have $R^*(Q_{\rm H}, Q_{\rm g}, n) > 1$, which implies that this particular PCG-Newton algorithm is better than Newton's method if the efficiency is concerned. In addition, for the case

$$Q_{\rm H}(n) = \mu n Q_{\rm g}(n) \quad \text{with} \quad \mu \in (0, 1], \tag{6.3}$$

it is proved in Theorems 5.4–5.5 that, when $n > \frac{2}{\underline{\mu}}$ with $\underline{\mu} \in (0, 1/16)$, $R^*(Q_{\rm H}, Q_{\rm g}, n) = R^*(\mu n Q_{\rm g}(n), Q_{\rm g}(n), n)$ has a tendency to increase when n increases. This implies that the larger the scale of the problem, the better the concrete algorithm. Furthermore, when $n \to \infty$, Theorem 5.6 shows that if μ in (6.3) satisfies that $\mu \in [\widetilde{\mu}, 1]$ with $\widetilde{\mu} > 0$, $R^*(\mu n Q_{\rm g}(n), Q_{\rm g}(n), n)$ tends to infinity uniformly at least at a rate $\ln n / \ln 2$. These theoretical results are supported by our preliminary numerical experiments[1)]. So, this paper is an essential extension and improvement of ref. [10]. Through this research we confirm from theoretical point of view that a carefully designed PCG type method indeed outperforms Newton's method, especially for medium- and large-size problems, which reinforces many reports on numerical performances of these two kinds of methods.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 10371131) and the Strategic Research of City University of Hong Kong (Grant No. 7001713).

1) see Section 6 of the following paper: Deng, N. Y., Zhang, J. Z., Zhong, P., On efficiency of Newton-PCG methods with difference quotient—a detailed version, available at http://www.cityu.edu.hk/ma/staff/zhang/DZZ.PDF.

References

1. Conn, A., Gould, N., Toint, Ph., LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), Berlin: Springer-Verlag, 1992.
2. Dembo, R. S., Eisenstat S. C., Steihaug, T., Inexact Newton methods, SIAM Journal on Numerical Analysis, 1982, 19(2): 400–408.
3. Eisenstat, S. C., Walker, H. F., Choosing the forcing terms in an inexact Newton method, SIAM Journal on Scientific Computing, 1996, 17(1): 16–32.
4. Gould, N. I. M., Lucidi, S., Roma, M., et al., Solving the trust-region subproblem using the Lanczos method, SIAM Journal on Optimization, 1999, 9(2): 504–525.
5. Lucidi, S., Rochetich, F., Roma, M., Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization, SIAM Journal on Optimization, 1998, 8(4): 916–939.
6. Morales, J. L., Nocedal, J., Automatic preconditioning by limited memory quasi-Newton updating, SIAM Journal on Optimization, 2000, 10(4): 1079–1096.
7. Nash, S. G., A survey of truncated Newton methods, Journal of Computational and Applied Mathematics, 2000, 124(1-2): 45–59.
8. Steihaug, T., The conjugate gradient method and trust regions in large scale optimization, SIAM Journal on Numerical Analysis, 1983, 20(3): 626–637.
9. Toint, P. L., Towards an efficient sparsity exploiting newton method for minimization, in Sparse Matrices and Their Uses (ed. Duff, I. S.), New York: Academic Press, 1981, 57–88.
10. Deng, N. Y., Wang, Z. Z., Theoretical efficiency of an inexact Newton method, Journal of Optimization Theory and Applications, 2000, 105(1): 97–112.
11. Dixon, L. C. W., On Deng-Wang theorem, OR Transactions, 2000, 4: 42–48.
12. Ostrowski, A., Solution of Equations and Systems of Equations, New York: Academic Press, 1960.