A Dwindling Filter Line Search Method for Unconstrained Optimization *

Yannan Chen† and Wenyu Sun‡

January 12, 2011

Abstract

In this paper, we propose a new dwindling multidimensional filter second-order line search method for solving large-scale unconstrained optimization problems. Usually, the multidimensional filter is constructed with a fixed envelope, which is a strict condition for the gradient vectors. A dwindling multidimensional filter technique, which is a modification and improvement of the original multidimensional filter, is presented. Under some reasonable assumptions, the new algorithm is globally convergent to a second-order critical point, when negative curvature direction is exploited. Preliminary numerical experiments on a set of `CUTEr` test problems indicate that the new algorithm is more competitive than the traditional second-order line search algorithms.

Keywords: filter method, line search, negative curvature direction, second-order critical point, global convergence.

MSC(2010): 65K05, 90C30

1 Introduction

We consider the large-scale unconstrained optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}), \tag{1.1}$$

where f is a real valued function on $\mathbb{R}^n$. We assume that both the gradient $\boldsymbol{g}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ and the Hessian matrix $H(\boldsymbol{x}) = \nabla^2_{\boldsymbol{xx}} f(\boldsymbol{x})$ of f exist and are continuous. In this paper we use the following notations: $\|\cdot\|$ is the Euclidean norm, $|\cdot|$ is the absolute value, $\{\boldsymbol{x}_k\}$ a sequence of points generated by an algorithm, $f_k = f(\boldsymbol{x}_k)$, $\boldsymbol{g}_k = \boldsymbol{g}(\boldsymbol{x}_k)$ and $H_k = H(\boldsymbol{x}_k)$, and so on.

Filter method was first introduced for constrained nonlinear programming by Fletcher and Leyffer [8], which is a method without penalty function and guarantees global convergence [7, 9, 11, 18, 27, 28, 32, 33]. For unconstrained optimization problems, there are mainly two sorts of filter methods. The first approach is the straightforward application of the filter technique, see [20, 25, 34]. Particularly, Wang and Zhu [34]

*This work was supported by the National Natural Science Foundation of China No. 10871098 and 11071122, the Specialized Research Fund of Doctoral Program of Higher Education of China, and the Natural Science Fund of Jiangsu Province of China at Grant No. BK2009397.

†School of Mathematical Sciences, Nanjing Normal University, Nanjing 210046; and College of Science, Nanjing Forestry University, Nanjing 210037, China. E-mail: cyannan2008@163.com.

‡Corresponding author. School of Mathematical Sciences, Nanjing Normal University, Nanjing 210046, China. Email: wysun@njnu.edu.cn.

transform the simple unconstrained optimization problem (1.1) to a nonlinear programming problem with equality constraint,

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \qquad \text{s.t.} \quad \boldsymbol{g}(\boldsymbol{x}) = 0,$$

and then the filter technique for general nonlinear programming has been applied. The second approach is the multidimensional filter technique, which is proposed for nonlinear equations and nonlinear least-squares in [12], and for unconstrained optimization problems in [2, 15, 36]. The multidimensional filter has more freedom in accepting trial points, and has been studied in [17,19,23,35]. For the purpose of avoiding saddle points, the filter is reset to empty set if the nonconvexity of the objective function is detected [15, 35]. A brief history and review of filter methods can be consult in [10, 29].

A multidimensional filter $\mathcal{F}$ is defined as a list of n-tuples of the form $(\boldsymbol{g}_{k,1}, \ldots, \boldsymbol{g}_{k,n})$, where $\boldsymbol{g}_{k,i} \stackrel{\text{def}}{=} \boldsymbol{g}_i(\boldsymbol{x}_k)$, such that if $\boldsymbol{g}_k$ and $\boldsymbol{g}_l$ belong to $\mathcal{F}$, then

$$|\boldsymbol{g}_{k,j}| < |\boldsymbol{g}_{l,j}|, \qquad \text{for at least one} \qquad j \in \{1, \ldots, n\}.$$

Then, a trial point $\boldsymbol{x}_k^+$ is acceptable for the filter $\mathcal{F}$ if and only if

$$\forall\, \boldsymbol{g}_l \in \mathcal{F}, \qquad \exists\, j \in \{1, \ldots, n\} \quad : \quad |\boldsymbol{g}_j(\boldsymbol{x}_k^+)| - |\boldsymbol{g}_{l,j}| \le -\gamma_g \|\boldsymbol{g}_l\|, \tag{1.2}$$

where $\gamma_g \in (0, 1/\sqrt{n})$ is a small positive number. We say that $\gamma_g\|\boldsymbol{g}_l\|$ is the fixed measurement of the envelope of $\mathcal{F}$.

In this paper, we are going to construct a new algorithm which produces a second-order critical point $\boldsymbol{x}^*$, such that

$$\boldsymbol{g}(\boldsymbol{x}^*) = 0, \qquad \text{and} \qquad H(\boldsymbol{x}^*) \text{ is positive semidefinite.}$$

For this purpose, some second-order line search algorithms have been proposed (see [26,30]). The key idea is to determine, at each iterate $\boldsymbol{x}_k$, a pair of descent directions $(\boldsymbol{s}_k, \boldsymbol{d}_k)$, where $\boldsymbol{s}_k$ is a gradient-related direction satisfying

$$\boldsymbol{s}_k^\top \boldsymbol{g}_k \le -c_1 \|\boldsymbol{g}_k\|^2 \qquad \text{and} \qquad \|\boldsymbol{s}_k\| \le c_2 \|\boldsymbol{g}_k\|, \tag{1.3}$$

where c_1 and c_2 are some positive numbers, $\boldsymbol{d}_k$ is a negative curvature direction, such that

$$\begin{aligned} \boldsymbol{d}_k^\top \boldsymbol{g}_k \le 0, \qquad \boldsymbol{d}_k^\top H_k \boldsymbol{d}_k \le 0, \\ \boldsymbol{d}_k^\top H_k \boldsymbol{d}_k \to 0 \quad \text{implies} \quad \min[0,\ \lambda_{\min}(H_k)] \to 0, \end{aligned} \tag{1.4}$$

where $\lambda_{\min}(H)$ is the smallest eigenvalue of the matrix H. The second-order line search algorithms proposed in [22, 24] are based on curvilinear line search of the following form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k^2 \boldsymbol{s}_k + \alpha_k \boldsymbol{d}_k,$$

where $\alpha_k > 0$ is the step-length. In [6, 21, 31, 37, 38], this approach was embedded in a nonmonotonic framework and used to solving small and large-scale optimization problems. The scaling of two directions $\boldsymbol{s}_k$ and $\boldsymbol{d}_k$ are taken into account in [13]. Instead of using two directions synchronously, it selects the more promising one of two directions.

In what follows, we consider an algorithm where a new dwindling multidimensional filter technique is embedded in a second-order line search framework, which shows that it is competitive and efficient in theoretical and numeric aspects. This paper is organized as follows. We describe our motivation and the new dwindling multidimensional filter in Section 2, and propose a new dwindling filter line search algorithm in Section 3. In Section 4 the convergence analysis of our algorithm to a second-order critical point is proved. The preliminary numerical experiments are discussed in Section 5. Some conclusions are finally presented in Section 6.

2 The dwindling multidimensional filter technique

In this section, we describe the modification and improvement of existing multidimensional filter in line search context.

2.1 Our motivation

We consider the inexact line search method. Suppose that at current iterate $\boldsymbol{x}_k$, algorithms usually seek a step-length $\alpha > 0$ along the gradient-related direction $\boldsymbol{s}_k$ which satisfies (1.3), i.e., we define

$$\boldsymbol{x}_k^+(\alpha) \stackrel{\text{def}}{=} \boldsymbol{x}_k + \alpha \boldsymbol{s}_k$$

with

$$f(\boldsymbol{x}_k^+(\alpha)) - f_k \leq \alpha\mu \boldsymbol{g}_k^\top \boldsymbol{s}_k, \tag{2.1}$$

where $\mu \in (0,\ 1)$. Note that, from (1.3) and the Cauchy-Schwarz inequality, we have

$$-c_2\|\boldsymbol{g}_k\|^2 \leq \boldsymbol{g}_k^\top \boldsymbol{s}_k \leq -c_1\|\boldsymbol{g}_k\|^2.$$

Hence, two propositions are valid for (2.1):

(i) the objective function reduction must less than a preset quantity which is proportional to step-length α and $\mathcal{O}(\|\boldsymbol{g}_k\|^2)$, and

(ii) inequality (2.1) holds if α is small enough, see [26, 30].

When α is zero, both sides of (2.1) are zero and hence equality holds. If α is slightly larger than zero, the first-order derivatives on α of the left and right hand side of (2.1) are $\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha))^\top \boldsymbol{s}_k$ and $\mu \boldsymbol{g}_k^\top \boldsymbol{s}_k$, respectively. We then have that the inequality

$$\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha))^\top \boldsymbol{s}_k \leq \mu \boldsymbol{g}_k^\top \boldsymbol{s}_k$$

holds for α sufficiently small, since $\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha)) \to \boldsymbol{g}_k$ as $\alpha \to 0$, $\mu < 1$ and $\boldsymbol{g}_k^\top \boldsymbol{s}_k < 0$. Hence, we conclude that accepting trial points by means of information contained in the gradient vectors is a good idea. Moreover, the projections of gradient in some directions are ultimately decreased when step-length α vanishes.

Now, we consider the multidimensional filter mechanism in [15, 23]. A trial point $\boldsymbol{x}_k^+(\alpha)$ is acceptable to the filter $\mathcal{F}$ if and only if

$$\forall\, \boldsymbol{g}_l \in \mathcal{F}, \qquad \exists\, j \in \{1, \ldots, n\} \quad : \quad |\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha))| - |\boldsymbol{g}_{l,j}| \leq -\gamma_g \|\boldsymbol{g}_l\|. \tag{2.2}$$

Following this, we are now considering a simple example.

Example 2.1. *Suppose that $f(\boldsymbol{x})$ is an one-variable function, i.e., $n = 1$, and assume that $\boldsymbol{x}_k$ belongs to the filter. Then (2.2) can be rewritten as the following form*

$$|\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha))| - |\boldsymbol{g}_k| \leq -\gamma_g |\boldsymbol{g}_k|. \tag{2.3}$$

Without loss of generality, we suppose that $|\boldsymbol{g}_k|$ is nonzero. We find that:

(i) *the right hand side of (2.3) is irrespective to step-length α;*

(ii) *if α is sufficiently small, inequality (2.3) cannot be valid. The reason is that the left-hand side of (2.3) tends to zero as $\alpha \to 0$ while the right-hand side is strictly negative.*

This fact means that according to the rule of multidimensional filter (2.2), the trial point $\boldsymbol{x}_k^+(\alpha)$ must be rejected if $\boldsymbol{x}_k$ is contained in $\mathcal{F}$ and α is sufficiently small.

In out point of view, comparing to the rule of inexact line search (2.1), multidimensional filter's rule is strict on the gradient vectors. Hence, we give a modification and improvement of multidimensional filter, which is named dwindling multidimensional filter, in the next subsection.

2.2 The dwindling multidimensional filter

Our idea is to introduce the inexact line search step-length into the multidimensional filter and let the envelope of the filter to become thinner and thinner as the step-length approaches zero. We illustrate our idea for two-variable problem as Figure 1.

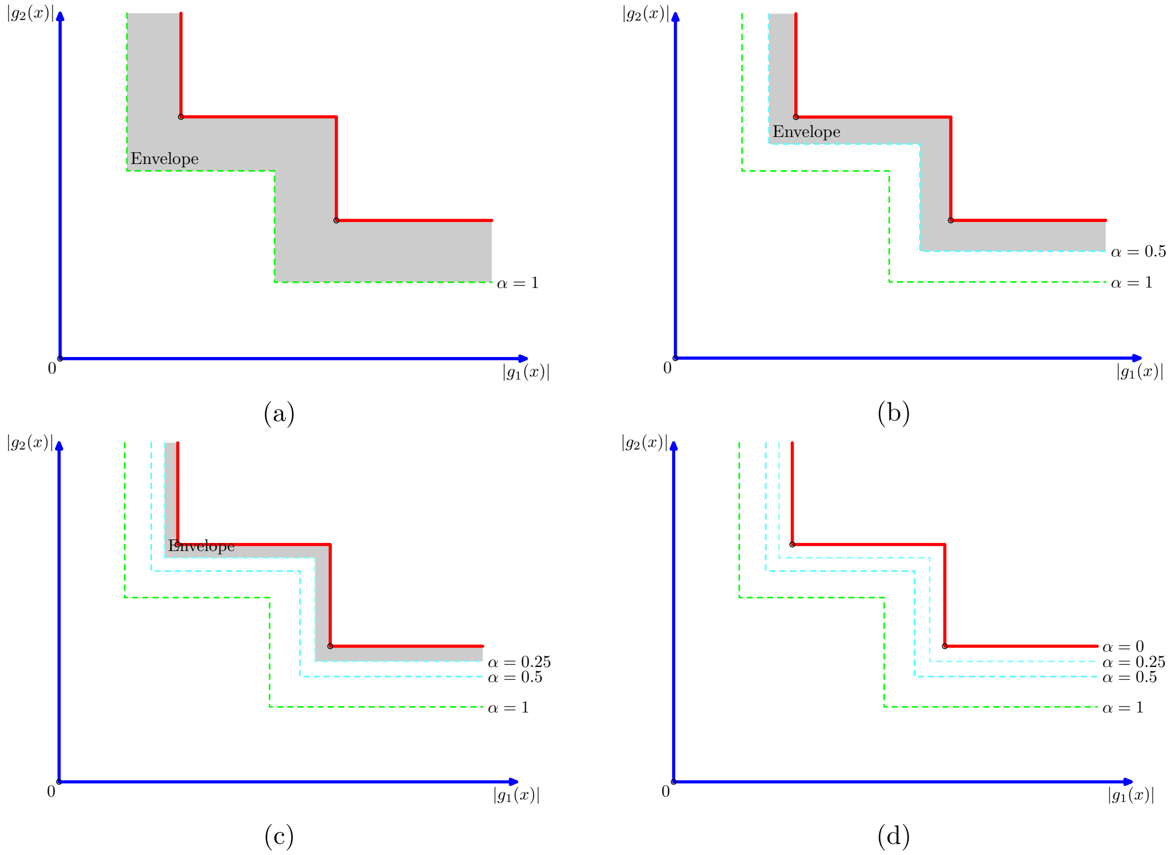


Figure 1: *The envelopes of the dwindling multidimensional filter vary as α decreases. (a) When α is a unit, the envelope is substantial and equals to the original multidimensional filter's envelope. The envelopes become thinner as $\alpha = 0.5$ in (b) and $\alpha = 0.25$ in (c). (d) If α vanishes, the envelope vanishes, too. (We set $\phi(\alpha) = \alpha$ in this figure.)*

Next, we describe the new rule for dwindling multidimensional filter. The trial point $\boldsymbol{x}_k^+(\alpha)$ is acceptable to the filter $\mathcal{F}$, if and only if it satisfies

$$\forall\, \boldsymbol{g}_l \in \mathcal{F}, \qquad \exists j \in \{1, \dots, n\} \quad : \qquad |\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha))| - |\boldsymbol{g}_{l,j}| \leq -\gamma_g \phi(\alpha) \|\boldsymbol{g}_l\|, \tag{2.4}$$

where $\phi(\alpha)$ is a dwindling function defined as follows.

Definition 2.2. $\phi(\alpha) : [0, 1] \mapsto \mathbb{R}$ *is a dwindling function if it is a monotonically increasing and continuous function such that* $\phi(\alpha) = 0$ *if and only if* $\alpha = 0$.

It is easy to see that if $\phi(\alpha) = 1$, the dwindling filter is reduced to the multidimensional filter. However, in this paper, we suppose that the dwindling function $\phi(\alpha)$ satisfies

$$\lim_{\alpha \to 0} \frac{\phi(\alpha)}{\alpha} = 0. \tag{2.5}$$

Under some assumptions, we can prove that the trial point $\boldsymbol{x}_k^+(\alpha)$ could not be rejected by iterate $\boldsymbol{x}_k$ when the dwindling filter is employed.

Lemma 2.3. *Suppose that there exists an index $j \in \{1, \dots, n\}$ such that*

$$\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha)) \cdot \boldsymbol{g}_{k,j} > 0, \qquad |\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha))| < |\boldsymbol{g}_{k,j}|, \qquad |\boldsymbol{e}_j^\top H_k \boldsymbol{s}_k| \neq 0$$

for all α sufficiently small, and that $\phi(\alpha)$ satisfies (2.5). Then, the trial point $\boldsymbol{x}_k^+(\alpha)$ could not be rejected by $\boldsymbol{x}_k$ if the step-length α is sufficiently small.

Proof. Since $|\boldsymbol{g}_{k,j}| > 0$, then $\|\boldsymbol{g}_k\| \neq 0$. Then, from the assumptions, we could obtain

$$\begin{aligned} |\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha))| - |\boldsymbol{g}_{k,j}| &= -\left| \boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha)) - \boldsymbol{g}_{k,j} \right| \\ &= -\left| \boldsymbol{e}_j^\top (\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha)) - \boldsymbol{g}_k) \right| \\ &= -\left| \alpha \boldsymbol{e}_j^\top H_k \boldsymbol{s}_k + o(\alpha) \right| \\ &= -\alpha |\boldsymbol{e}_j^\top H_k \boldsymbol{s}_k| + o(\alpha) \\ &\leq -\gamma_g \phi(\alpha) \|g_k\|, \end{aligned}$$

where the last inequality holds because of (2.5) when the step-length α is sufficiently small. So, $\boldsymbol{x}_k^+(\alpha)$ is not rejected by x_k according to new rule (2.4). □

Although we cannot ensure that the trial point $\boldsymbol{x}_k^+(\alpha)$ is acceptable to the new dwindling filter $\mathcal{F}$ when α is sufficiently small, because that $\mathcal{F}$ may contain other points besides x_k, this lemma shows that $\boldsymbol{x}_k^+(\alpha)$ cannot be rejected by current iterate $\boldsymbol{x}_k$ in some case.

If $\boldsymbol{x}_k^+(\alpha)$ is acceptable in the sense of (2.4), we add $\boldsymbol{g}(\boldsymbol{x}_k^+(\alpha))$ to $\mathcal{F}$ and remove from it every $\boldsymbol{g}_l$ with

$$|\boldsymbol{g}_{l,j}| > |\boldsymbol{g}_j(\boldsymbol{x}_k^+(\alpha))|, \qquad \forall\, j \in \{1, \dots, n\}.$$

3 The dwindling filter line search algorithm

We use the quadratic model m_k of $f(\boldsymbol{x})$ at iterate $\boldsymbol{x}_k$, where

$$m_k(\boldsymbol{x}_k + \boldsymbol{s}) \overset{\text{def}}{=} \frac{1}{2} \boldsymbol{s}^\top H_k \boldsymbol{s} + \boldsymbol{g}_k^\top \boldsymbol{s} + f_k.$$

If the nonconvexity of the quadratic model is detected, we reset the filter to empty set [15, 35]. At each iteration, since two search directions are obtained, we choose the more promising one by a switching condition used in [13] to perform line search.

We describe the new dwindling filter second-order line search algorithm as follows.

Algorithm 3.1 (The Dwindling Filter Line Search Algorithm (DFLS)).

Step 0. Initialization. *Given $\boldsymbol{x}_0 \in \mathbb{R}^n$. Set $\mu \in (0, \frac{1}{2})$, $\beta \in (0,1)$, $\tau > 0$, $\gamma_g > 0, \varepsilon > 0$, and $k = 0$. Initialize the filter $\mathcal{F} = \emptyset$, choose $F_0 \geq f(\boldsymbol{x}_0)$ and $\phi(\alpha)$, set* `nonconvex = False`.

Step 1. Test for convergence. *Compute $\boldsymbol{g}_k$. If $\|\boldsymbol{g}_k\| \leq \varepsilon$ and* `nonconvex = False`*, stop.*

Step 2. Compute search direction. *Compute a pair of directions $(\boldsymbol{s}_k,\ \boldsymbol{d}_k)$ which satisfies (1.3) and (1.4). Set* `nonconvex = True` *if m_k is nonconvex, otherwise set* `nonconvex = False`.

Step 3. Choice of the search direction. *If the switching condition*

$$\frac{\boldsymbol{g}_k^\top \boldsymbol{s}_k}{\|\boldsymbol{s}_k\|} \leq \tau \left[m_k \left(\boldsymbol{x}_k + \frac{\boldsymbol{d}_k}{\|\boldsymbol{d}_k\|} \right) - m_k(\boldsymbol{x}_k) \right]$$

holds, execute **Step 4**. *Otherwise execute* **Step 5**.

Step 4. Line search in gradient-related direction. *Compute* $\alpha_k = \beta^l$ *and* $\boldsymbol{x}_k^+ = \boldsymbol{x}_k + \alpha_k \boldsymbol{s}_k$*, where* l *is the smallest non-negative integer satisfies one of the following two cases.*

(a) **Truncated Newton iterations.** *If* $\boldsymbol{x}_k^+$ *satisfies*

$$f(\boldsymbol{x}_k^+) \leq f(\boldsymbol{x}_k) + \mu \left(\alpha_k \boldsymbol{g}_k^\top \boldsymbol{s}_k + \frac{1}{2} \alpha_k^2 \min[0, \boldsymbol{s}_k^\top H_k \boldsymbol{s}_k] \right), \tag{3.1}$$

set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k^+$*,* $F_{k+1} = F_k$ *and* $k := k+1$*, go to* **Step 1**.

(b) **Dwindling filter iterations.** *If* `nonconvex = False`*,* $f(\boldsymbol{x}_k^+) < F_k$ *and* $\boldsymbol{x}_k^+$ *is acceptable to* $\mathcal{F}$*, i.e., satisfies (2.4), set* $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k^+$ *and add* $\boldsymbol{g}(\boldsymbol{x}_k^+)$ *to* $\mathcal{F}$*. Set* $F_{k+1} = F_k$ *and* $k := k+1$*, go to* **Step 1**.

Step 5. Line search in negative curvature direction. *Choose* $\eta_k > 0$*. Compute the step length* $\alpha_k = \eta_k \beta^l$*, where* l *is the smallest non-negative integer such that*

$$f(\boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + \mu \left(\alpha_k \boldsymbol{g}_k^\top \boldsymbol{d}_k + \frac{1}{2} \alpha_k^2 \boldsymbol{d}_k^\top H_k \boldsymbol{d}_k \right).$$

Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$*,* $F_{k+1} = f(\boldsymbol{x}_{k+1})$*,* $k := k+1$*. Reset* $\mathcal{F} = \emptyset$*, and go to* **Step 1**.

Remark 1. At each iterate $\boldsymbol{x}_k$, we choose the parameter $\eta_k = \alpha_j$, where $j (< k)$ is the index of the last iteration at which the switching condition (3.1) fails.

Remark 2. Note that if the dimension n is large, a lot of memory may be used to store the filter. So in practical implementation, the components of the gradient vector may be grouped, and then the dwindling filter can be used in those groups. We suggest readers referring to [12, 16] for more details.

4 Convergence analysis

In this section, we study the convergence properties of our proposed algorithm. First, we give some general assumptions.

A1 f is twice continuously differentiable on $\mathbb{R}^n$, and

A2 the level set $\Omega \stackrel{\text{def}}{=} \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\}$ is compact.

Without loss of generality, we suppose that $F_0 = f(\boldsymbol{x}_0)$ at the beginning of the algorithm. Then the assumptions **A1** and **A2** imply that all iterates generated by the algorithm are in the compact level set Ω, and all objective functions and gradients of the iterates are bounded. Under these assumptions and the mechanism of our algorithm, we know that if there are only finitely many iterates, then the last point is a second-order critical point. So, we now assume that our algorithm produces infinitely many iterates.

For the convenience of the analysis, we define some index sets:

$$\begin{aligned} \mathcal{K}^F &\stackrel{\text{def}}{=} \{k \mid \text{ dwindling filter iterations }\}, \\ \mathcal{K}^N &\stackrel{\text{def}}{=} \{k \mid \text{ truncated Newton iterations }\}, \\ \mathcal{K}^L &\stackrel{\text{def}}{=} \{k \mid \text{ line search in negative curvature direction }\}. \end{aligned}$$

Clearly, every two sets of the three do not intersect.

Considering the values of F_k, we have a crucial property of the algorithm.

Lemma 4.1. *Denoting $\mathcal{K}^L = \{k_l\}$. We have that, $\{F_k\}$ is a monotonically decreasing sequence and $\{F_{k_l}\}$ is a strictly monotonically decreasing sequence. Moreover, for all $k \geq 0$,*

$$f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{k+1}) \geq \sum_{j=0}^{k} [F_j - F_{j+1}]. \tag{4.1}$$

Proof. We observe that the definition of F_k in the algorithm ensures that

$$F_{k_{l+1}} = F_j < F_{k_l}$$

for all l and all $k_l + 1 \leq j \leq k_{l+1}$. This directly implies that $\{F_k\}$ is monotonically decreasing and $\{F_{k_l}\}$ is strictly monotonically decreasing.

Suppose $k_{\bar{l}} \in \mathcal{K}^L$ is the largest integer which is not exceed k. If $k = k_{\bar{l}}$, then $F_{k+1} = f(\boldsymbol{x}_{k+1})$. Thus

$$f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{k+1}) = \sum_{j=0}^{k} [F_j - F_{j+1}]. \tag{4.2}$$

Otherwise, $k > k_{\bar{l}}$. By (4.2) and the definition of F_k in the algorithm, we have

$$\begin{aligned} f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{k+1}) &= f(\boldsymbol{x}_0) - f(\boldsymbol{x}_{k_{\bar{l}}+1}) + f(\boldsymbol{x}_{k_{\bar{l}}+1}) - f(\boldsymbol{x}_{k+1}) \\ &\geq \sum_{j=0}^{k_{\bar{l}}} [F_j - F_{j+1}] + F_{k_{\bar{l}}+1} - F_{k+1} \\ &= \sum_{j=0}^{k} [F_j - F_{j+1}]. \end{aligned}$$

Therefore, we have inequality (4.1). □

In the following, we will establish the convergence theorems. First, we prove that if there are infinitely many negative curvature iterations, the iterates produced by the algorithm must converge to second-order critical points.

Theorem 4.2. *Suppose that assumptions **A1-A2** hold and $|\mathcal{K}^L| = +\infty$. Assume that the direction $\boldsymbol{d}_k$ satisfies (1.4). Then the points $\{\boldsymbol{x}_k\}$ produced by the algorithm converge to second-order critical points.*

Proof. Consider $\{\boldsymbol{x}_{k_l} | k_l \in \mathcal{K}^L\}$. By Lemma 4.1, we know that $\{F_k\}$ is monotonously decreasing and $\{F_{k_l}\}$ is strictly monotonously decreasing. Then, by the proof of Theorem 3.1 of [13], the iterates produced by the algorithm must converge to second-order critical points. □

We now consider the case that there are only finitely many negative curvature iterations. In what follows, if there are finitely many dwindling filter iterations, then there must be infinitely many truncated Newton iterations. Similar to the proof of Theorem 3.1 of [13], we have the following theorem.

Theorem 4.3. *Suppose that assumptions **A1-A2** hold, $|\mathcal{K}^L| < +\infty$ and $|\mathcal{K}^F| < +\infty$, then $|\mathcal{K}^N| = +\infty$. Assume that the direction $\boldsymbol{s}_k$ satisfies (1.3). Then the points $\{\boldsymbol{x}_k\}$ produced by the algorithm converge to second-order critical points.*

The remainder case is that there are infinitely many dwindling filter iterations. We have the following theorem.

Theorem 4.4. *Suppose that assumptions **A1-A2** hold, $|\mathcal{K}^L| < +\infty$ and $|\mathcal{K}^F| = +\infty$. Assume that the directions $\boldsymbol{s}_k$ and $\boldsymbol{d}_k$ satisfy (1.3) and (1.4), respectively. Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{4.3}$$

Moreover, suppose that the sequence $\{\boldsymbol{x}_k\}$ converges to the unique limit point $\boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is a second-order critical point.

Proof. We first prove, by contradiction, that (4.3) is valid. Assume that there is a positive number $\epsilon > 0$ such that

$$\|\boldsymbol{g}_k\| \geq \epsilon \tag{4.4}$$

for all k.

Note $|\mathcal{K}^L| < +\infty$, then the filter $\mathcal{F}$ is no longer reset to empty set for k sufficiently large. Moreover, since our assumptions imply that $\{\|\boldsymbol{g}_k\|\}$ is bounded above and away from zero, there must exist a subsequence $\{k_i\} = \mathcal{K}_1 \in \mathcal{K}^F$ such that

$$\lim_{i\to\infty} \boldsymbol{g}_{k_i+1} = \boldsymbol{g}_\infty \qquad \text{with} \qquad \boldsymbol{g}_\infty \geq \epsilon. \tag{4.5}$$

By the definition of k_i, $\boldsymbol{x}_{k_i+1}$ is acceptable for $\mathcal{F}$ at each iteration k_i.

Since $\mathcal{F}$ is not reset for i sufficiently large, then there exists an index $j_i \in \{1, 2, \ldots, n\}$ such that

$$|\boldsymbol{g}_{k_i+1,j_i}| - |\boldsymbol{g}_{k_{i-1}+1,j_i}| < -\gamma_g \phi(\alpha_{k_i}) \|\boldsymbol{g}_{k_{i-1}+1}\|. \tag{4.6}$$

However, (4.4) implies that $\|\boldsymbol{g}_{k_{i-1}+1}\| \geq \epsilon$. Hence we deduce from (4.6) that

$$|\boldsymbol{g}_{k_i+1,j_i}| - |\boldsymbol{g}_{k_{i-1}+1,j_i}| < -\gamma_g \phi(\alpha_{k_i})\epsilon \leq 0$$

for all i sufficiently large. Since, by (4.5), the left hand side of this inequality tends to zero when $i \to \infty$, then $\phi(\alpha_{k_i})$ must tend to zero when $i \to \infty$, and we have, by Definition 2.2, that

$$\lim_{\substack{i\to\infty \\ k_i \in \mathcal{K}_1}} \alpha_{k_i} = 0. \tag{4.7}$$

Since $\boldsymbol{x}_{k_i+1} = \boldsymbol{x}_{k_i} + \alpha_{k_i}\boldsymbol{s}_{k_i}$ is acceptable to the filter $\mathcal{F}$ and (4.7), we have $\alpha_{k_i} < 1$ for all sufficiently large i. Define

$$\bar{\alpha}_{k_i} \stackrel{\text{def}}{=} \frac{\alpha_{k_i}}{\beta},$$

then $\bar{\alpha}_{k_i} \to 0$ as $i \to \infty$ and $k_i \in \mathcal{K}_1$. By Algorithm 3.1 and (3.1), we have that

$$f(\boldsymbol{x}_{k_i} + \bar{\alpha}_{k_i}\boldsymbol{s}_{k_i}) - f(\boldsymbol{x}_{k_i}) > \mu\bar{\alpha}_{k_i}\boldsymbol{g}_{k_i}^\top \boldsymbol{s}_{k_i} + \frac{\mu}{2}\bar{\alpha}_{k_i}^2 \min[0, \boldsymbol{s}_{k_i}^\top H_{k_i}\boldsymbol{s}_{k_i}].$$

Then by Taylor's theorem, we have

$$\bar{\alpha}_{k_i}\boldsymbol{g}_{k_i}^\top \boldsymbol{s}_{k_i} + \frac{1}{2}\bar{\alpha}_{k_i}^2 \boldsymbol{s}_{k_i}^\top H(\boldsymbol{x}_{k_i} + \theta\bar{\alpha}_{k_i}\boldsymbol{s}_{k_i})\boldsymbol{s}_{k_i} > \mu\bar{\alpha}_{k_i}\boldsymbol{g}_{k_i}^\top \boldsymbol{s}_{k_i} + \frac{\mu}{2}\bar{\alpha}_{k_i}^2 \min[0, \boldsymbol{s}_{k_i}^\top H_{k_i}\boldsymbol{s}_{k_i}]$$

for some $\theta \in (0, 1)$. Divided by $\bar{\alpha}_{k_i}$ and $\|\boldsymbol{s}_{k_i}\|$, the inequality can be written as

$$(1-\mu)\frac{\boldsymbol{g}_{k_i}^\top \boldsymbol{s}_{k_i}}{\|\boldsymbol{s}_{k_i}\|} > \frac{1}{2}\bar{\alpha}_{k_i}\|\boldsymbol{s}_{k_i}\| \left(\mu \min\left[0, \frac{\boldsymbol{s}_{k_i}^\top H_{k_i}\boldsymbol{s}_{k_i}}{\|\boldsymbol{s}_{k_i}\|^2}\right] - \frac{\boldsymbol{s}_{k_i}^\top H(\boldsymbol{x}_{k_i} + \theta\bar{\alpha}_{k_i}\boldsymbol{s}_{k_i})\boldsymbol{s}_{k_i}}{\|\boldsymbol{s}_{k_i}\|^2} \right). \tag{4.8}$$

Now, we extract a subsequence $\{\boldsymbol{x}_{k_i}\}$ whose indices $\{k_i\}$ lie in the set $\mathcal{K}_2 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}^F$ such that

$$\boldsymbol{x}_{k_i} \to \boldsymbol{x}_* \quad \text{and} \quad \frac{\boldsymbol{s}_{k_i}}{\|\boldsymbol{s}_{k_i}\|} \to \boldsymbol{s}_*.$$

By using (4.8), $\|\boldsymbol{s}_k\| \le c_2\|\boldsymbol{g}_k\|$, the bounds of $\|\boldsymbol{g}_k\|$ and $\|H_k\|$, and taking the limit $\bar{\alpha}_{k_i} \to 0$ as $i \to \infty$, we obtain that $(1-\mu)\boldsymbol{g}(\boldsymbol{x}_*)^\top \boldsymbol{s}_* \ge 0$, and hence that

$$\boldsymbol{g}(\boldsymbol{x}_*)^\top \boldsymbol{s}_* \ge 0, \tag{4.9}$$

where $k_i \in \mathcal{K}_2$, $\boldsymbol{x}_{k_i} \to \boldsymbol{x}_*$.

By condition (1.3), we have

$$\frac{\boldsymbol{g}_k^\top \boldsymbol{s}_k}{\|\boldsymbol{s}_k\|} \le -\frac{c_1}{c_2}\|\boldsymbol{g}_k\|.$$

Therefore, when $k \in \mathcal{K}_2$ and tends to infinity, we have

$$\boldsymbol{g}(\boldsymbol{x}_*)^\top \boldsymbol{s}_* \le -\frac{c_1}{c_2}\|\boldsymbol{g}(\boldsymbol{x}_*)\| \le -\frac{c_1}{c_2}\epsilon < 0,$$

which contradicts (4.9). The contradiction proves (4.3) holds.

Since we suppose that the whole sequence of iterates $\{\boldsymbol{x}_k\}$ converges to the unique limit point $\boldsymbol{x}^*$. Then $\boldsymbol{g}(\boldsymbol{x}^*) = 0$. According to $|\mathcal{K}^F| = +\infty$, then `nonconvex = False` holds infinitely many times. By assumption A1 and the fact that $f(\boldsymbol{x})$ is twice continuously differentiable, we obtain $H(\boldsymbol{x}^*)$ is positive semidefinite. Therefore, $\boldsymbol{x}^*$ is a second-order critical point. □

5 Numerical experiments

In order to evaluate the behavior of our new algorithm, we need a subroutine to solve the local quadratic subproblem and obtain a pair of directions $(\boldsymbol{s}_k, \boldsymbol{d}_k)$ which satisfy (1.3) and (1.4). In this paper, we employ the planar conjugate-gradient method [3–5] which has the following advantages.

- The planar conjugate-gradient method could simultaneously compute the pair of directions $(\boldsymbol{s}_k, \boldsymbol{d}_k)$, which satisfies (1.3) and (1.4).
- It is a Krylov subspace iterative method for computing an adequate negative curvature direction. It does not need to explicitly know the Hessian matrix, but only need to evaluate the product of the Hessian and any vector. So it is very suitable for large-scale optimization problems.
- Compared with the Lanczos method, the planar conjugate-gradient method does not need to store any matrix of Lanczos orthogonal basis.
- If the Hessian matrix is sufficiently positive definite, the planar conjugate-gradient method is equivalent to the normal conjugate-gradient method.

The local quadratic model is solved approximately and is terminated at the step $\boldsymbol{s}$ such that

$$\|\nabla m_k(\boldsymbol{x}_k + \boldsymbol{s})\| \le \min\left[0.1, \sqrt{\max(\epsilon_M, \|\nabla m_k(\boldsymbol{x}_k)\|)}\right] \|\nabla m_k(\boldsymbol{x}_k)\|,$$

where ϵ_M is a machine precision. We also terminate the planar conjugate-gradient iterations if it reaches the upper bound n or 100 iterations after finding the negative curvature direction.

All tests were performed in double precision on the Tsinghua Tongfang F5600 portable computer (1.5 GHz, 1 Gbyte of RAM) under Red Hat Fedora Core 3 operation system and the intel ifort compiler (version 9.0). All attempts to solve the test problems were limited to a maximum of 10000 iterations or 1000 seconds of CPU time.

For each problem, we use the default starting point and exact gradients. The product of exact Hessian and any vector is also used. Although, in our algorithm, the exact Hessian information is used, we don't employ them directly in matrix form in both computation and memory. Instead, the cheap vector-form for computing is used such that it is suitable for solving large-scale problems. We use the settings

$$\mu = 10^{-4}, \quad \beta = 0.5, \quad \tau = 2.0, \quad \eta_0 = 1,$$

and choose

$$F_0 = \min\left\{10^6 |f(\boldsymbol{x}_0)|, f(\boldsymbol{x}_0) + 1000\right\}.$$

In addition, we choose dwindling function

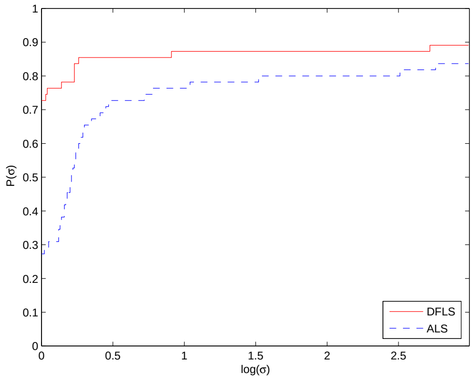
$$\phi(\alpha) = \alpha^{10} \quad \text{and} \qquad \gamma_g = \min\left\{0.001, \frac{1}{2\sqrt{n}}\right\}.$$

Finally, the algorithm stops if

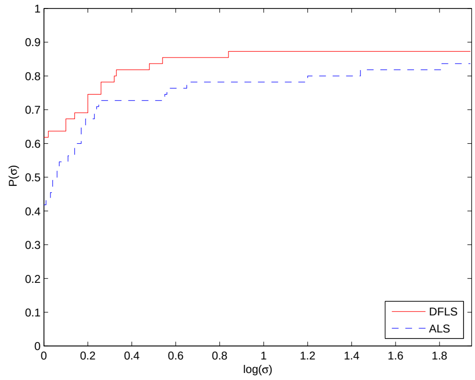
$$\|\nabla f(\boldsymbol{x}_k)\| \leq 10^{-6}\sqrt{n} \qquad \text{and} \qquad \texttt{nonconvex = False} \ .$$

No.	Problem	n	No.	Problem	n
1	BARD	3	29	HIMMELBH	2
2	BIGGS3	6	30	HUMPS	2
3	BIGGS5	6	31	LIARWHD	10000
4	BROWNDEN	4	32	LMINSURF	5625
5	CHAINWOO	4000	33	MARATOSB	2
6	CHNROSNB	50	34	MSQRTBLS	1024
7	CUBE	2	35	NLMSURF	5625
8	CURLY10	10000	36	NONDQUAR	10000
9	CURLY20	10000	37	OSBORNEA	5
10	DECONVU	61	38	OSBORNEB	11
11	DIXMAANA	3000	39	PALMER1C	8
12	DIXMAANF	3000	40	PENALTY1	1000
13	DIXMAANH	3000	41	PFIT1LS	3
14	DJTL	2	42	PFIT2LS	3
15	ERRINROS	50	43	PFIT3LS	3
16	EXTROSNB	2000	44	PFIT4LS	3
17	FLETCHCR	1000	45	ROSENBR	2
18	FMINSRF2	5625	46	SENSORS	100
19	FMINSURF	5625	47	SINEVAL	2
20	FREUROTH	5000	48	SINQUAD	10000
21	GENHUMPS	1000	49	SNAIL	2
22	GENROSE	10000	50	SPARSINE	2000
23	GROWTHLS	3	51	TOINTPSP	50
24	GULF	3	52	TQUARTIC	10000
25	HATFLDE	3	53	VIBRBEAM	8
26	HEART6LS	6	54	WOODS	4000
27	HEART8LS	8	55	YFITU	3
28	HELIX	3			

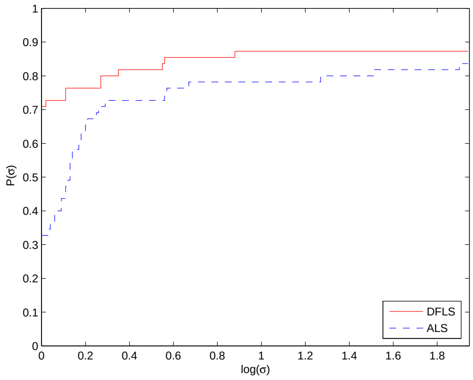
Table 1: *The test problems and their dimension.*



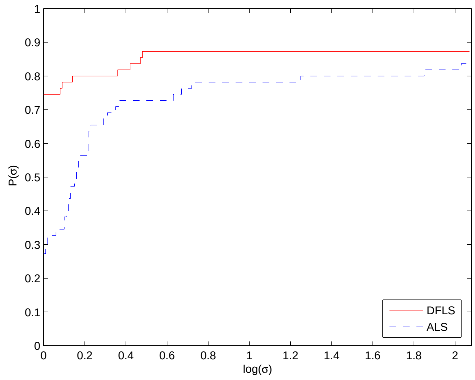
(a)



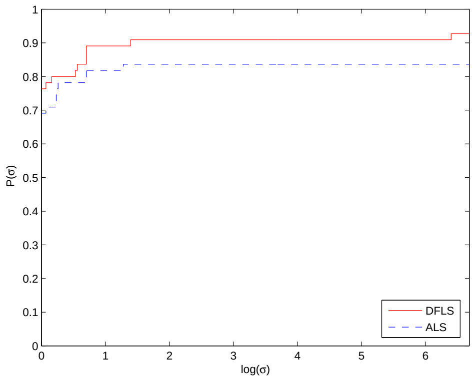
(b)



(c)



(d)



(e)

(a) Function evaluation.
(b) Gradient evaluation.
(c) Iteration performance.
(d) Conjugate-gradient iteration.
(e) CPU time.

Figure 2: *Performance profiles.*

For numerical comparison, two algorithms are tested. The first is the DFLS algorithm described in Section 3. The second is the ALS algorithm described in [13]. We report the numerical results obtained by running two algorithms on a set of unconstrained test problems from the CUTEr collection [1, 14]. There are 55 problems where the dwindling filter iterations have been detected. The names of these problems with the number of their variables are listed in Table 1. If no dwindling filter iteration been detected, DFLS is the same as ALS.

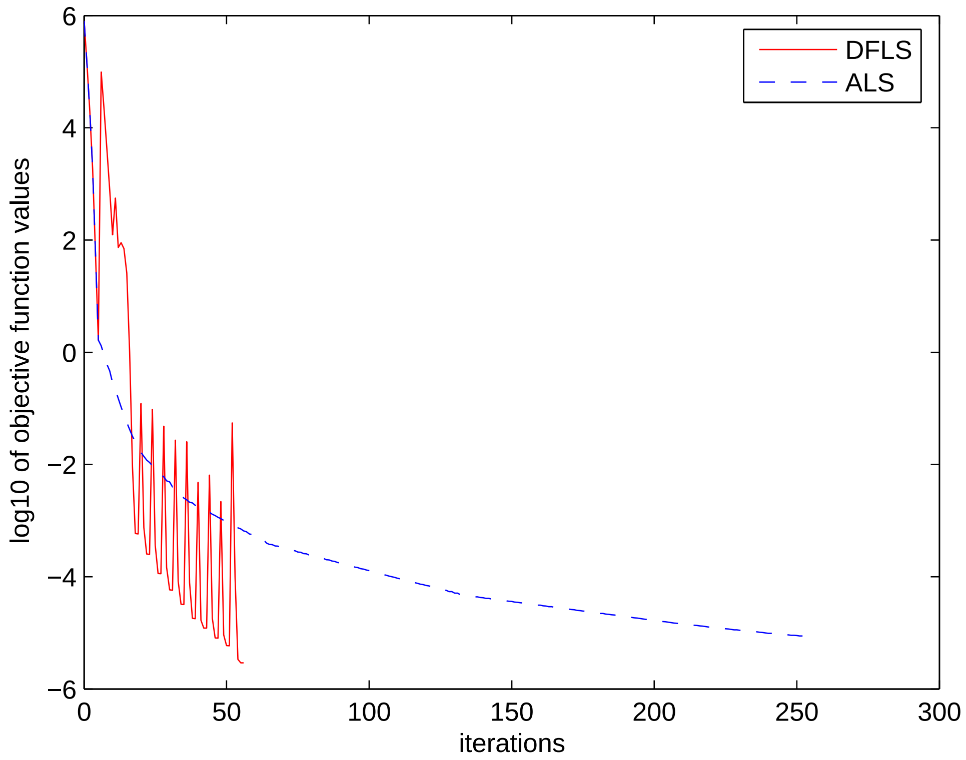


Figure 3: *The objective function value as a function of the iteration progress on the EXTROSNB problem.*

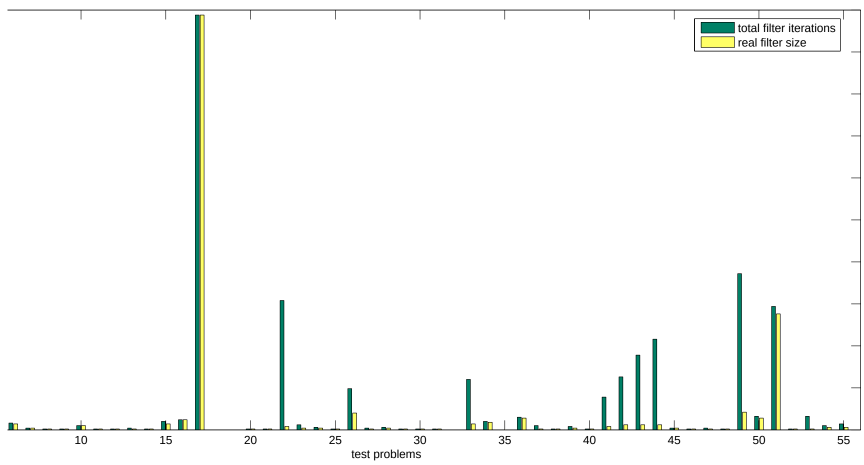


Figure 4: *Dwindling filter iterations and the amount of filter elements needed to store.*

For the problems where both algorithms succeed, they have the same final objective function values. In 55 test problems, algorithm DFLS successfully solves 93% and it fails on problems FMINSRF2, FMINSURF, LMINSURF and NIMSURF for exceeding CPU time. While algorithm ALS successfully solves 84% and it fails on problems BROWNDEN, DJTL, PALMER1C, VIBRBEAM for exceeding maximal iteration numbers and fails on problems CURLY10, CURLY20, FREUROTH, GENROSE, SINQUAD for exceeding CPU time.

Figure 2 shows the performance profiles of two algorithms for function and gradient evaluation, iterations, conjugate-gradient iterations and CPU time, respectively. Obviously, we can see that the algorithm DFLS is significantly efficient.

We present in Figure 3 a plot of the evolution of objective function values for two algorithms. This plot is typical, from which we see that the algorithm DFLS outperforms ALS. For algorithm DFLS, we note that there is large oscillation in objective values prior to convergence. From this figure, it is remarkable that the algorithm is convergent.

Since we cannot know the size of the filter in advance, the filter is adaptively managed by a serial structure using `pointer` language in FORTRAN 95 code, so the computer RAM could be distributed dynamically, and the memory could be saved as soon as possible. Figure 4 shows the dwindling filter iterations and the amount of filter elements needed to store. We can see that a majority of problems at most need to store 21 filter elements in our algorithm. But there need 494 and 138 filter elements in FLETCHCR and TOINTPSP respectively. It is easy to see that the memory needed to store filter entries does not exceed 4 MB for all test problems.

6 Conclusions

The dwindling multidimensional filter technique has been presented in a line search algorithm framework for solving large-scale unconstrained optimization problems. The envelope of dwindling multidimensional filter is dwindling when step-length of the inexact line search approaches zero. We have shown that, under mild assumptions, the sequence of iterates produced by the new algorithm converges to one second-order critical point. Preliminary numerical experiences indicate that the new algorithm is reliable and efficient for solving large-scale nonconvex and ill-conditioned problems.

Acknowledgements

We would like to thank Professor Ph.L. Toint for his helpful advice on using CUTEr.

References

[1] I. Bongartz, A.R. Conn, N.I.M. Gould and Ph.L. Toint, CUTE: Constrained and unconstrained testing environment, *ACM Transaction on Mathematical Software*, **21**(1995), 123-160.

[2] Y. Chen and W. Sun, A new nonmonotone optimization method with trust-region and second order line search, *Numerical Mathematics, A Journal of Chinese Universities*, **32**(2010), 369-386.

[3] G. Fasano, Planar-conjugate gradient algorithm for large scale unconstrained optimization, Part I: Theory, *J. Optim. Theory Appl.*, **125**(2005), 523-541.

[4] G. Fasano, Planar-conjugate gradient algorithm for large scale unconstrained optimization, Part 2: Application, *J. Optim. Theory Appl.*, **125**(2005), 543-558.

[5] G. Fasano and M. Roma, Iterative computation of negative curvature directions in large scale optimization, *Comput. Optim. Appl.*, **38**(2007), 81-104.

[6] M. C. Ferris, S. Lucidi and M. Roma, Nonmonotone curvilinear linesearch methods for unconstrained optimization, *Comput. Optim. Appl.*, **6**(1996), 117-136.

[7] R. Fletcher, N.I.M. Gould, S. Leyffer, Ph.L. Toint and A. Wächter, Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming, *SIAM J. Optim.*, **13**(2002), 635-659.

[8] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, *Math. Program.*, **91**(2002), 239-269.

[9] R. Fletcher, S. Leyffer, Ph.L. Toint, On the global convergence of a filter-SQP algorithm, *SIAM J. Optim.*, **13**(2002), 44-59.

[10] R. Fletcher, S. Leyffer, Ph.L. Toint, A brief history of filter methods, *SIAG/OPT Views-and-News*, **18**(2007), 2-12.

[11] C.C. Gonzaga, E. Karas and M. Vanti, A Globally convergent filter method for nonlinear programming, *SIAM J. Optim.*, **14**(2003), 646-669.

[12] N.I.M. Gould, S. Leyffer and Ph.L. Toint, A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares, *SIAM J. Optim.*, **15**(2004), 17-38.

[13] N.I.M. Gould, S. Lucidi, M. Roma and Ph.L. Toint, Exploiting negative curvature directions in linesearch methods for unconstrained optimization, *Optim. Methods Softw.*, **14**(2000), 75-98.

[14] N.I.M. Gould, D. Orban and Ph.L. Toint, CUTEr, a constrained and unconatrained testing environment, revisited, *ACM T. Math. Software*, **29**(2003), 373-394.

[15] N.I.M. Gould, C. Sainvitu and Ph. L. Toint, A filter-trust-region method for unconstrained optimization, *SIAM J. Optim.*, **16**(2005), 341-357.

[16] N.I.M. Gould and Ph.L. Toint, FILTRANE, a fortran 95 filter-trust-region package for solving nonlinear feasibility problems, *Report 03/17*, Dept of Mathematics, FUNDP, Namur (B), 2003.

[17] N. Krejić, Z. Lužanin and I. Stojkovska, Gauss-Newton-based BFGS method with filter for unconstrained minimization, *Appl. Math. Comput.*, **211**(2009), 354-362.

[18] C. Li and W. Sun, On filter-successive linearization methods for nonlinear semidefinite programming, *Science in China Series A*, **52**(2009), 2341-2361.

[19] S.J. Li and Z.H. Liu, A new trust region filter algorithm, *Appl. Math. Comput.*, **204**(2008), 485-489.

[20] J. Long and S.Y. Zeng, A new filter-Levenberg-Marquardt method with disturbance for solving nonlinear complementarity problems, *Appl. Math. Comput.*, **216**(2010), 677-688.

[21] S. Lucidi, F. Rochetich and M. Roma, Curvillinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization, *SIAM J. Optim.*, **8**(1998), 916-939.

[22] G.P. McCormick, A modification of Armijo's step-size rule for negative curvature, *Math. Program.*, **13**(1977), 111-115.

[23] W. Miao and W. Sun, A filter trust-region method for unconstrained optimization, *Numer. Math. J. Chinese Univ.*, **29**(2007), 88-96.

[24] J.J. Moré and D.C. Sorensen, On the use of directions of negative curvature in a modified Newton method, *Math. Program.*, **16**(1979), 1-20.

[25] P.Y. Nie, M.Y. Lai, S.J. Zhu and P.A. Zhang, A line search filter approach for the system of nonlinear equations, *Comput. Math. Appl.*, **55**(2008), 2134-2141.

[26] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, USA, 1999.

[27] A.A. Ribeiro, E.W. Karas and C.C. Gonzaga, Global convergence of filter methods for nonlinear programming, *SIAM J. Optim.*, **19**(2008), 1231-1249.

[28] C. Sainvitu and Ph.L. Toint, A filter-trust-region method for simple-bound constrained optimization, *Optim. Methods Softw.*, **22**(2007), 835-848.

[29] W. Sun, Filter methods for optimization: motivation and development, invited talk, International Conference on Engineering Mathematics and Computational Mathematics, Hong Kong Polytechnical University, December (2009).

[30] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, Springer Optimization and Its Applications (SOIA), Volume 1, Springer, New York, 2006.

[31] W. Sun and Q.Y. Zhou, An unconstrained optimization method using nonmonotone second order Goldstein's linesearch, *Science in China Ser. A* **50**(2007), 1389-1400.

[32] A. Wächter and L.T. Biegler, Line search filter methods for nonlinear programming: motivation and global convergence, *SIAM J. Optim.*, **16**(2005): 1-31.

[33] A. Wächter and L.T. Biegler, Line search filter methods for nonlinear programming: local convergence, *SIAM J. Optim.*, **16**(2005): 32-48.

[34] Z.J. Wang and D.T. Zhu, A filter-line-search method for unconstrained optimization, *J. Appl. Math. Comput.*, **34**(2010), 329-342.

[35] Z.H. Yang, W.Y. Sun and C.Y. Dang, A filter-trust-region method for LC^1 unconstrained optimization and its global convergence, *Anal. Theory Appl.*, **24**(2008), 55-66.

[36] Y. Zhang, W. Sun and L. Qi, A nonmonotone filter Barzilai-Borwein method for optimization, *Asia Pacific Journal of Operational Research*, **27**(2010), 55-69.

[37] Q. Zhou and W. Sun, Adaptive nonmonotone trust region method with curvilinear searches, *J. Comput. Math.*, **24**(2006), 761-770.

[38] Q. Zhou and W. Sun, A nonmonotone second order line search method for unconstrained optimization, *J. Comput. Math.*, **25**(2006), 104-112.