A Variable Metric and Nesterov Extrapolated Proximal DCA with Backtracking for A Composite DC Program

Yu You · Yi-Shuai Niu

Abstract In this paper, we consider a composite difference-of-convex (DC) program, whose objective function is the sum of a smooth convex function with Lipschitz continuous gradient, a proper closed and convex function, and a continuous concave function. This problem has many applications in machine learning and data science. The proximal DCA (pDCA), a special case of the classical DCA, as well as two Nesterov-type extrapolated DCA – ADCA (Phan et al. IJCAI:1369–1375, 2018) and pDCAe (Wen et al. Comput Optim Appl 69:297–324, 2018) – can solve this problem. The algorithmic step-sizes of pDCA, pDCAe, and ADCA are fixed and determined by estimating a prior the smoothness parameter of the loss function. However, such an estimate may be hard to obtain or poor in some real-world applications. Motivated by this difficulty, we propose a variable metric and Nesterov extrapolated proximal DCA with backtracking (SPDCAe), which combines the backtracking line search procedure (not necessarily monotone) and the Nesterov’s extrapolation for potential acceleration; moreover, the variable metric method is incorporated for better local approximation. Numerical simulations on sparse binary logistic regression and compressed sensing with Poisson noise demonstrate the effectiveness of our proposed method.

Keywords Difference-of-convex programming · Nesterov’s extrapolation · backtracking line search · variable metric · sparse binary logistic regression · compressed sensing with Poisson noise

Mathematics Subject Classification (2010) 65K05 · 90C26 · 90C30
1 Introduction

Difference-of-convex (DC) programming has been an active area of research in nonconvex and nonsmooth optimization [19,20,44,27,1,42]. They arise in extensive applications such as sparsity learning [55,28], clustering [23], molecular conformation [25], trust region subproblem [45,7], portfolio optimization [48,47,38], control theory [36], natural language processing [40], image denoising [56], mixed-integer optimization [22,35,31,39] and eigenvalue complementarity problem [24,37,34,33], to name a few; see [27] and the references therein for a comprehensive introduction about DC programming and its applications.

In this paper, we focus on a composite DC program in form of

$$\min \{ F(x) := f(x) + g(x) - h(x) : x \in \mathbb{R}^n \} ,$$  \hspace{1cm} (P)

where the following are assumed throughout the manuscript:

Assumption 1 (a) \( f \) and \( g \) are proper convex functions from \( \mathbb{R}^n \) to \(( -\infty, \infty]\), \( h : \mathbb{R}^n \to \mathbb{R} \) is convex; moreover, \( g \) is closed.

(b) There exists a nonempty closed and convex set \( Y \subseteq \mathbb{R}^n \) such that \( \text{dom}(g) \subseteq Y \subseteq \text{int}(\text{dom}(f)) \).

(c) \( f \) is \(L\)-smooth over \( Y \), i.e., the gradient of \( f \) is \( L\)-Lipschitz over \( Y \).  

(d) \( F \) is bounded from below, i.e., there exists scalar \( F^* \) such that \( F(x) \geq F^* \) for all \( x \in \mathbb{R}^n \).

Note that we do not require that the loss function \( f \) is real-valued over \( \mathbb{R}^n \) since the scope of the model can be limited in this way. Next, by setting \( h \equiv 0 \) in \((P)\) we derive the convex optimization problem

$$\min \{ \Phi(x) := f(x) + g(x) : x \in \mathbb{R}^n \} ,$$  \hspace{1cm} (P_c)$

for which we additionally assume the following:

Assumption 2 The optimal solution set of \((P_c)\) is nonempty, and its optimal value is denoted by \( \Phi^* \).

The classical DCA [44,45,26,46] is renowned for solving \((P)\), which has been introduced by Pham Dinh Tao in 1985 and extensively developed by Le Thi Hoai An and Pham Dinh Tao since 1994. One possible (commonly used) DC decomposition for \((P)\) takes the following form

$$\min \{ F(x) := G(x) - H(x)^1 : x \in \mathbb{R}^n \} ,$$  \hspace{1cm} (1)

where \( G(x) = L\|x\|^2/2 + g(x) \) and \( H(x) = L\|x\|^2/2 - f(x) + h(x) \) when \( x \in Y \) and \( \infty \) otherwise. Then DCA applied for this DC decomposition yields the proximal DC algorithm (pDCA) proposed in [17]. For the special case where \( Y = \mathbb{R}^n \), the accelerated variant of DCA (ADCA) [50], which incorporated the Nesterov’s extrapolation [29,30] for potential acceleration, can be applied for solving \((1)\); meanwhile, Wen et al. [54] used the same extrapolation technique for solving \((P)\).
A variable metric and extrapolated proximal DCA with backtracking

In some practical applications, the Lipschitz constant of the gradient of $f$ might be hard to estimate. The (monotone) backtracking line search procedure [6] is a common strategy for addressing this issue. However, in this way the step-size is nonincreasing, which may seriously affect the convergence speed if the local curvature of the loss function is relatively large near the initial iterates but small near the tail; see [53] for an illustration of this by the compressed sensing and the binary logistic regression problems. In addition, for those problems with the above property, the (global) Lipschitz constant is generally larger than the local one, thus the constant step-size strategy is not recommended. Taking this into account, [53] proposed a non-monotone backtracking strategy for addressing this kind of problems. Note that model ($P$) inherits the above issues in ($P_c$).

In this paper, in term of the success of Nesterov’s extrapolation techniques for acceleration, the variable metric method for better local approximation [10, 11], and the non-monotone backtracking for adaptive step-sizes selection, we incorporate all of these techniques to propose a scaled proximal DC algorithm with extrapolation and backtracking (SPDCAe) for solving problem ($P$). We prove that for suitable selections of the extrapolation parameters and the scaling matrices, if the generated sequence of SPDCAe is bounded, then any limit point of this sequence is a critical point of ($P$). Moreover, we also demonstrate that SPDCAe for the convex case ($P_c$), denote as SFISTA, enjoys the optimal $O(1/k^2)$ convergence rate in function values. This rate of convergence coincides with that of the well-known FISTA algorithm [6] and SFBEM [10] (a scaling version of FISTA with monotone backtracking). Finally, we point out that our SPDCAe (resp. SFISTA) actually covers pDCAe (resp. SFBEM).

The rest of the paper is organized as follows. Sect. 2 reviews some notations and preliminaries in convex analysis. In Sect. 3, we describe the DC optimization problem as well as its convex case we are concerned in this paper and present our algorithm SPDCAe and its convex version SFISTA. Sect. 4 focuses on establishing the subsequential convergence of SPDCAe and the
optimal convergence rate of SFISTA. Numerical results on problems of sparse binary logistic regression and compressed sensing with Poisson noise are summarized in Sect. 5.

2 Notations and preliminaries

Let $\mathbb{R}^n$ denote the $n$-dimensional column vector space endowed with canonical inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. For an extended real-valued function $p : \mathbb{R}^n \to (-\infty, \infty]$, the set

$$\text{dom}(p) := \{ x \in \mathbb{R}^n : p(x) < \infty \}$$

denotes its effective domain. If $\text{dom}(p) \neq \emptyset$ and $p$ does not attain the value $-\infty$, then $p$ is called a proper function. The notation $\text{int}(\text{dom}(p))$ denotes the interior set of $\text{dom}(f)$, and

$$\text{epi}(p) := \{ (x, t) \in \mathbb{R}^n \times \mathbb{R} : p(x) \leq t \}$$

denotes the epigraph of $p$. We have the definition that $p$ is closed (or convex) if $\text{epi}(p)$ is closed (or convex).

Let $p : \mathbb{R}^n \to (-\infty, \infty]$ be a proper function and $x \in \text{int}(\text{dom}(p))$. Then $p$ is said to be differentiable at $x$ if there exists $g \in \mathbb{R}^n$ such that

$$\lim_{d \to 0} \frac{p(x + d) - p(x) - \langle g, d \rangle}{\|d\|} = 0.$$ 

The unique vector $g$ as above is called the gradient of $p$ at $x$, and is denoted as $\nabla p(x)$. Finally, we say that $p$ is $L$-smooth over $Y$ if $p$ is differentiable over $Y$ and $\|\nabla p(x) - \nabla p(y)\| \leq L\|x - y\|$ for any $x$ and $y$. In addition, if $Y$ is convex, then we have the next descent lemma.

**Lemma 2.1 (see e.g., [5])** Let $p : \mathbb{R}^n \to (-\infty, \infty]$ be an $L$-smooth function ($L \geq 0$) over a given convex set $Y$. Then for any $x, y \in Y$,

$$p(y) \leq p(x) + \langle \nabla p(x), y - x \rangle + \frac{L}{2}\|x - y\|^2. \quad (2)$$

Now, let $p : \mathbb{R}^n \to (-\infty, \infty]$ be a proper closed and convex function. A vector $g \in \mathbb{R}^n$ is called a subgradient of $p$ at $x$ if

$$p(y) \geq p(x) + \langle g, y - x \rangle, \forall y \in \mathbb{R}^n,$$

and the subdifferential of $g$ at $x$ is defined by

$$\partial p(x) = \{ g \in \mathbb{R}^n : p(y) \geq p(x) + \langle g, y - x \rangle, \forall y \in \mathbb{R}^n \}. $$

The proximal mapping of $p$ is the operator defined by

$$\text{Prox}_p(x) = \arg\min_{u \in \mathbb{R}^n} \left\{ p(u) + \frac{1}{2} \|u - x\|^2 \right\}, \forall x \in \mathbb{R}^n.$$ 

Next, we summarize some properties about the proximal mapping.
Lemma 2.2 (see e.g., [5]) Let \( p : \mathbb{R}^n \to (-\infty, \infty] \) be a proper closed and convex function. Then

- \( \text{Prox}_p(x) \) is a singleton for any \( x \in \mathbb{R}^n \).
- \( \text{Prox}_p \) is nonexpansive, i.e., for any \( x \) and \( y \) it holds that
  \[ \|\text{Prox}_p(x) - \text{Prox}_p(y)\| \leq \|x - y\|. \]
- If \( z = \text{Prox}_p(x) \), then \( x - z \in \partial p(z) \).

Note that if \( p \) is the indicator function \( \delta_Y \) of a nonempty and closed convex set \( Y \), i.e., \( \delta_Y(x) = 0 \) if \( x \in Y \) and \( \infty \) otherwise, then \( \text{Prox}_{\delta_Y} \) is the projection operator over \( Y \), denoted as \( \Pi_Y \). The notations \( \text{Prox}_Q^p \) and \( \Pi_Y^Q \) (\( Q \) is a positive definite matrix) are adopted, the former actually means

\[
\text{Prox}_Q^p(x) = \arg\min_{u \in \mathbb{R}^n} \left\{ p(u) + \frac{1}{2} \|u - x\|_Q^2 \right\}, \quad \forall x \in \mathbb{R}^n.
\]

In this case, \( z = \text{Prox}_Q^p(x) \) implies that \( x - z \in Q^{-1}\partial p(z) \).

For two vectors \( x, y \in \mathbb{R}^n \), their Hardmard product, denoted as \( x \odot y \), is the vector comprising the component-wise products: \( x \odot y = (x_i y_i)_{i=1}^n \). The nonnegative orthant is denoted by \( \mathbb{R}^n_+ \). Next, \( \text{diag}(x) \) denotes the \( n \times n \) diagonal matrix with its diagonal vector being \( x \). The set of all \( n \times n \) symmetric matrices is denoted as \( \mathbb{S}^n \); moreover, \( \mathbb{S}^n_+ \) (or \( \mathbb{S}^n_{++} \)) means the set of all \( n \times n \) positive semidefinite (or positive definite) matrices. For \( D_1, D_2 \in \mathbb{S}^n \), \( D_1 \preceq D_2 \) means that \( D_2 - D_1 \in \mathbb{S}^n_+ \). For \( \mu > 0 \), \( D_\mu \) denotes the set \( \{ D \in \mathbb{S}^n : \mu I \preceq D \} \), where \( I \) is the identity matrix.

Finally, we present a technical lemma, adapted from [8, Proposition A.31], which will be utilized for establishing the convergence property of our proposed method SPDCAe for \((P)\) and its convex version SFISTA for \((P_c)\).

Lemma 2.3 ([8]) \( \{a_k\}, \{\xi_k\}, \{\zeta_k\}, \) and \( \{\gamma_k\} \) be nonnegative sequences of real numbers such that

\[ a_{k+1} \leq (1 + \zeta_k)a_k - \xi_k + \gamma_k, \quad k = 1, 2, \ldots, \]

and

\[ \sum_{k=1}^{\infty} \zeta_k < \infty, \quad \sum_{k=1}^{\infty} \gamma_k < \infty. \]

Then \( \{a_k\} \) converges to a finite value and \( \{\xi_k\} \) converges to 0.

3 Problem formulation and the proposed algorithms

In this section, we first introduce the DC optimization problem we focus on in this paper and present our proposed algorithm SPDCAe for \((P)\), which incorporates the Nesterov’s extrapolation, the backtracking line search (monotone and non-monotone), and the variable metric method. Next, we describe in
detail the algorithmic procedure of SFISTA, which is the convex version of SPDCAe with non-monotone backtracking for (P).

In this paper, we specifically focus on the composite DC program

\[
\min \{ F(x) := f(x) + g(x) - h(x) : x \in \mathbb{R}^n \}.
\]

Some assumptions for the involved functions are summarized in Assumption 1. Note that we do not follow the assumption of the two accelerated algorithms ADCA and pDCAe developed respectively in [50, 54] that \( f \) is real-valued over the whole space since the scope of the model can be limited in this way. For example, the loss function in the Poisson noise model [18] is the generalized Kullback-Leibler divergence, whose effective domain is larger than the non-negative orthant but not equal to the whole space. Indeed, their algorithms can be generalized for this situation by additionally projecting the extrapolation point onto the set \( Y \) at each iteration. We describe in Algorithm 1 our proposed algorithm SPDCAe for solving problem (P). The parameters \( \beta_k, L_k, \) and \( D_k \) remain to be specified. Later, we will prove that with suitable selections of parameters, any limit point of the sequence generated by SPDCAe, denoted as \( x^* \), is a critical point of problem (P), i.e.,

\[
0 \in \nabla f(x^*) + \partial g(x^*) - \partial h(x^*).
\]

At the moment, we just assume the following for the scaling matrices \( \{D_k\} \) and defer the specified selection to the numerical part in Sect. 5.

**Assumption 3** The scaling matrices \( \{D_k\} \) satisfy that

\[
\{D_k\} \subseteq D_\mu, \quad D_{k+1} \preceq (1 + \eta_k)D_k, \quad k = 1, 2, \ldots,
\]

where \( \mu > 0 \) and \( \{\eta_k\} \) is a nonnegative sequence such that \( \sum_{k=1}^{\infty} \eta_k < \infty \).
Remark 3.1 Assumption 3 implies that there exists $c > 0$ such that
\[ \mu I \preceq D_k \preceq cD_1, \quad k = 1, 2, \ldots, \]
where $c = \prod_{k=1}^{\infty} (1 + \eta_k)$.

To determine $\beta_k$ and $L_k$, we distinguish the next two backtracking strategies:

- **Non-monotone backtracking** Here, we use the heuristic method suggested in [53] to determine $\{L_k\}$. Specifically, let $T_1$ be a positive integer, the initial guess $L_k$ (as in line 2 of Algorithm 1) is set as the previously returned $L_{k-1}$ if $k$ is not divisible by $T_1$ and as $\rho L_k$ otherwise, where $\rho \in (0, 1)$. Next, two common methods for updating $\{\beta_k\}$ is as follows:
  - **Contract method** sets $\beta_1 = 0$ and $\beta_k = \theta_{k-1} - \theta_k$ for $k \geq 2$, where $\theta \in (0, 1)$ is a contractive factor (generally close to 1) and $\{\theta_k\}_{k \geq 1}$ is the sequence recursively defined by
    \[ \theta_1 = 1, \quad \theta_l = \frac{1 + \sqrt{1 + 4\theta_{l-1}^2L_l/L_{l-1}}}{2}, \quad l = 2, 3, \ldots. \]  
  - **Restart method** updates $\{\beta_k\}$ via the fixed restart scheme or the fixed and adaptive restart scheme proposed in [41]. Specifically, let $T_2$ be a positive integer. For the fixed restart scheme, one starts to update $\{\theta_l\}$ by (3) and obtain $\beta_1 = 0$ and $\beta_k = \frac{\theta_{k-1} - 1}{\theta_k}$ for $k \geq 2$, then reset $\theta_{k+1} = 1$ if $T_2$ is divisible by $k$. For the fixed and adaptive restart scheme, the restart condition shall be modified as either $T_2$ is divisible by $k$ or $\langle x^k - x^{k-1}, y^k - x^k \rangle > 0$.

- **Monotone backtracking** In this situation, the initial trail $L_k$ is set as the previously returned $L_{k-1}$. Then, about the update of $\{\beta_k\}$, the only difference from the non-monotone one is that (3) should be modified as
  \[ \theta_1 = 1, \quad \theta_l = (1 + \sqrt{1 + 4\theta_{l-1}^2})/2, \quad l = 2, 3, \ldots. \]

Remark 3.2 1. The restart schemes as described above were also used in [54] for their proposed algorithm pDCAe. Moreover, our SPDCAe with monotone backtracking is exactly pDCAe if the domain of $f$ is the whole space, $L_k$ is selected as the smoothness parameter of $f$, and the scaling matrix $D_k$ is set as the identity matrix at each iteration.
2. For SPDCAe with monotone backtracking, $x^{k-1} + \beta_k(x^{k-1} - x^{k-2})$ can be computed out of the inner loop.
3. The inner loop in Algorithm 1 can stop in finite step [10].
4. For the monotone backtracking, $L_k \geq L$ can be omitted, besides, the boundedness of $\{D_k\}$ as in Remark 3.1 can imply that $0 < \liminf L_k < \limsup L_k < \infty$; while for the non-monotone one, $L_k \geq L$ guarantees that $\liminf L_k > 0$, and thus $0 < \liminf L_k \leq \limsup L_k < \infty$. 

Algorithm 2: SFISTA with non-monotone backtracking for \((P_c)\)

```
Input: \(\theta_0 = 1, t_0 = 0, \eta > 1.\) Set \(x^0 = x^{-1} \in Y \subseteq \mathbb{R}^n.\)

for \(k = 1, 2, 3, \cdots\) do
  pick \(L_k > 0;\)
  set \(i = 0;\)
  repeat
    (1) update \(L_k\) by \(\eta^i L_k\) and set \(t_k = 1/L_k;\)
    (2) compute \(\theta_k = \frac{1 + \sqrt{1 + 4\theta_k^2 - 1/t^2_k - 1/t_{k-1}}}{2};\)
    (3) pick \(D_k \in S^{n}_{++},\) calculate \(y^k = \Pi_{D_k} Y (x^k - t_k D_k^{-1} \nabla f(y^k));\)
    (4) compute \(x^k = \text{Prox}_{D_k} (x^k - t_k D_k^{-1} \nabla f(y^k));\)
    (5) update \(i\) by \(i + 1.\)
  until \(f(x^k) \leq f(y^k) + \langle \nabla f(y^k), x^k - y^k \rangle + \frac{1}{2t_k} \|x^k - y^k\|^2_{D_k};\)
end
```

Next, we describe in Algorithm 2 the algorithmic procedure of SFISTA with non-monotone backtracking line search (the convex version of SPDCAe with non-monotone backtracking). Note that SFISTA with monotone backtracking is exactly SFBEM proposed in [10] and thus we omit to present it here. In [53], Scheinberg et al. showed that for some practical applications, such as the compressed sensing and the logistic regression problems, the local Lipschitz constant of the gradient of loss function is generally smaller than the global one, especially when approaching the tail. Thus, the monotone backtracking will lead to slow convergence near the solution set, while the non-monotone backtracking helps to overcome this issue to some extent. Later, we will show that SFISTA is equipped with the optimal convergence rate \(\Phi(x^k) - \Phi^* \leq O(1/k^2),\) which coincides with that of SFBEM and the classical FISTA [6].

4 Convergence analysis

In this section, we will investigate the convergence property of SPDCAe for \((P)\) and its convex version SFISTA for \((P_c)\). For the former, we demonstrate the subsequential convergence to a critical point; while for the latter, the optimal convergence rate \(\Phi(x^k) - \Phi^* \leq O(1/k^2)\) is established.

The convergence analyses of both SPDCAe and SFISTA are based on the following key inequality.

**Proposition 4.1** Suppose that \(f, g,\) and \(h\) satisfy properties (a), (b), and (c) of Assumption 1. For any \(x \in \mathbb{R}^n, h \in \partial h(x), y \in Y, D \in S^{n}_{++},\) and \(t > 0\) satisfying

\[
f(y) \leq f(y) + \langle \nabla f(y), y - y \rangle + \frac{1}{2t} \|y - y\|^2_D,
\]

where \(y := \text{Prox}_{D}^D (y - t D^{-1} \nabla f(y) - h)\), it holds that

\[
F(y) \leq F(x) + \frac{1}{2t} \|x - y\|^2_D - \frac{1}{2t} \|x - y\|^2_D.
\]
Proof It follows from $\bar{y} = \text{Prox}_{D}^{0}(y - tD^{-1}[\nabla f(y) - h])$ that $y - tD^{-1}[\nabla f(y) - h] - \bar{y} \in tD^{-1}\partial g(y)$ (term 3 of Lemma 2.2), which has the equivalent form

$$D(y - y) - t[\nabla f(y) - h] \in t\partial g(y).$$

Then, it holds that

$$tg(x) \geq tg(y) + \langle x - y, D(y - y) - t[\nabla f(y) - h] \rangle. \quad (5)$$

Thus, we have

$$F(y) = f(y) + g(y) - h(y)$$

$$\leq f(y) + \langle \nabla f(y), \bar{y} - y \rangle + \frac{1}{2t}\|\bar{y} - y\|_{D}^{2}$$

$$+ g(y) - \frac{1}{t}\langle x - \bar{y}, D(y - y) - t[\nabla f(y) - h] \rangle - h(y)$$

$$\leq f(y) + g(x) + \langle \nabla f(y), x - y \rangle + \frac{1}{t}\langle x - \bar{y}, D(y - y) \rangle + \frac{1}{2t}\|\bar{y} - y\|_{D}^{2}$$

$$- \langle x - \bar{y}, h \rangle - h(y)$$

$$\leq F(x) + \frac{1}{t}\langle x - \bar{y}, D(y - y) \rangle + \frac{1}{2t}\|\bar{y} - y\|_{D}^{2} - \langle x - \bar{y}, h \rangle - h(y) + h(x)$$

$$\leq F(x) + \frac{1}{t}\|x - y\|_{D}^{2} - \frac{1}{2t}\|x - \bar{y}\|_{D}^{2},$$

where the second inequality follows by $f(y) + \langle f(y), x - y \rangle \leq f(x)$, the last inequality holds since $h \in \partial h(x)$ and thus $h(\bar{y}) \geq h(x) + \langle \bar{y} - x, h \rangle$. $\square$

4.1 Subsequential convergence of SPDCAe

Now, we start to establish the subsequential convergence of SPDCAe with non-monotone backtracking and the update rule for $\{\beta_{k}\}$ as introduced in Sect. 3. The proof for SPDCAe with monotone backtracking follows similarly and thus omitted.

**Theorem 4.1** Suppose that Assumption 1 and Assumption 3 hold. Let $\{x_{k}\}$ be the sequence generated by Algorithm 1. Then, the following statements hold:

(i) The sequence $\{F(x_{k})\}$ converges and $\lim_{k \to \infty} \|x_{k} - x_{k-1}\| = 0$;

(ii) If $\{x_{k}\}$ is bounded, then any limit point of $\{x_{k}\}$ is a critical point of (P).

**Proof** (i) Let $k \geq 1$. Replace $x, h, y, \bar{y}, D, t$ in Proposition 4.1 by $x^{k-1}, h^{k-1}, y^{k}, x^{k}, D_{k}, t_{k}$, respectively. Then, we obtain

$$F(x^{k}) \leq F(x^{k-1}) + \frac{1}{2t_{k}}\|x^{k-1} - y^{k}\|_{D_{k}}^{2} - \frac{1}{2t_{k}}\|x^{k} - x^{k-1}\|_{D_{k}}^{2}. \quad (6)$$
Note that $y^k = \Pi^{D_k}_y (x^{k-1} + \beta_k (x^{k-1} - x^{k-2}))$ and $x^{k-1} \in Y$, thus we have
\[
\|x^{k-1} - y^k\|_D^2 \leq \beta_k^2 \|x^{k-1} - x^{k-2}\|_D^2.
\] (7)

Plunge (7) into (6) and then consider the update rule for $\{\beta_k\}$, it is easy to check that for $k \geq 2$,
\[
F(x^k) \leq F(x^{k-1}) + \frac{\alpha_k}{2t_{k-1}} \|x^{k-1} - x^{k-2}\|_D^2 - \frac{1}{2t_k} \|x^k - x^{k-1}\|_D^2,
\]
where $\alpha_k = \delta^2(1 - 1/\theta_k)(1 - 1/\theta_{k-1})^2$ (resp. $\alpha_k = (1 - 1/\theta_k)(1 - 1/\theta_{k-1})^2$) for 
the contract method (resp. restart method) for updating $\{\beta_k\}$ as introduced in Sect. 3. Clearly, for either of the selection methods, it holds that $\{\alpha_k\} \subseteq (0, 1)$ 
with $\sup_k \alpha_k < 1$. Next, denote $\tilde{F}(x^k) = F(x^k) - F^* + \frac{1}{2t_k} \|x^k - x^{k-1}\|_D^2$, we have
\[
\tilde{F}(x^k) = F(x^k) - F^* + \frac{1}{2t_k} \|x^k - x^{k-1}\|_D^2 \leq F(x^{k-1}) - F^* + \frac{1}{2t_{k-1}} \|x^{k-1} - x^{k-2}\|_D^2 - \frac{1}{2t_k} \|x^k - x^{k-1}\|_D^2 \leq F(x^{k-1}) - F^* + \frac{1}{2t_{k-1}} \|x^{k-1} - x^{k-2}\|_D^2 - \frac{1}{2t_k} \|x^k - x^{k-1}\|_D^2 \leq (1 + \eta_{k-1}) \tilde{F}(x^{k-1}) - \frac{1 - \alpha_k}{2t_{k-1}} \|x^{k-1} - x^{k-2}\|_D^2,
\]
where the second inequality follows by Assumption 3 for $\{D_k\}$. Based on the facts that $\sup_k \alpha_k < 1$ 
and $\{t_k\}$ is bounded from above (term 4 of Remark 3.2), it follows that the sequence $\{1 - \alpha_k/2t_{k-1}\}_{k\geq 2}$ has positive limit inferior. Then 
invoking Lemma 2.3, it holds that
\[
\lim_{k \to \infty} \|x^{k-1} - x^{k-2}\|_D^2 = 0,
\] (8)
and $\{\tilde{F}(x^k)\}$ converges to a finite value. Clearly, we have $\{F(x^k)\}$ converges to a finite value. Besides, $\{D_k\}$ satisfies that $\mu I \preceq D_k$ for $k \geq 1$. Thus, $\|x^{k-1} - x^{k-2}\|_D^2 \geq \mu \|x^{k-1} - x^{k-2}\|_D^2$, then (8) implies $\lim_{k \to \infty} \|x^{k-1} - x^{k-2}\| = 0$.

(ii) Take into account that $\lim_{k \to \infty} \|x^k - x^{k-1}\| = 0$ and $\mu I \preceq D_k$ for all $k$, we obtain from (7) that $\lim_{k \to \infty} \|x^k - y^k\| = 0$. Let $\bar{x}$ be any limit point of $\{x^k\}$, then there exists a convergent subsequence such that $\lim_{i \to \infty} x^{k_i} = \lim_{i \to \infty} y^{k_i} = \bar{x}$. Moreover, because the sequence $\{h^{k_i-1}\}$ is bounded in $\mathbb{R}^n$, we can assume without loss of generality that $\{h^{k_i-1}\}$ is convergent with limit $\bar{z}$. Then, it follows from $h^{k_i-1} \in \partial h(x^{k_i-1})$ for all $i$, $h^{k_i-1} \to \bar{z}$, $x^{k_i-1}$ \to $\bar{x}$, and the closedness of the graph $\partial h$ that $\bar{z} \in \partial h(\bar{x})$ [52, Theorem 24.4]. On the other hand, $x^k = \text{Prox}_{t_k D_k}^D(y^k - t_k D_k^{-1}(\nabla f(y^k) - h^{k_i-1}))$ implies that
\[
\frac{1}{t_{k_i}} D_k (y^{k_i} - x^{k_i}) - [\nabla f(y^{k_i}) - h^{k_i-1}] \in \partial g(x^{k_i}).
\] (9)
Consider also \( \{D_k\} \) bounded from above (Remark 3.1), thus the left-hand side of (9) converges to \(-\nabla f(\bar{x}) + \bar{z}\). Combining this with the facts that \( \{x_k\} \to \bar{x} \) and the closedness of the graph \( \partial g \), we have

\[
-\nabla f(\bar{x}) + \bar{z} \in \partial g(\bar{x}),
\]

thus \( 0 \in \nabla f(\bar{x}) + \partial g(\bar{x}) - \bar{z} \). Then we conclude that \( \bar{x} \) is a critical point of \( (P) \) since \( \bar{z} \in \partial h(\bar{x}) \). \( \square \)

4.2 Optimal convergence rate \( O(1/k^2) \) of SFISTA

In this part, we start to establish the optimal convergence rate of SFISTA with non-monotone backtracking (Algorithm 2) for \( (P_c) \).

**Theorem 4.2** Suppose that Assumption 2 holds. Let \( \{x_k\} \) be the sequence generated by Algorithm 2. Then, for any optimal solution \( x^* \) of \( (P_c) \), the following statements hold:

(i) For \( k \geq 0 \), denote \( v^k = x^k + \theta_k(x^k - x^{k-1}) \). Then, for \( k \geq 1 \) we have

\[
t_k \theta_k^2 (\Phi(x^k) - \Phi^*) + \frac{1}{2} \|x^* - v^k\|_D^2 \leq t_{k-1} \theta_{k-1}^2 (\Phi(x^{k-1}) - \Phi^*) + \frac{1}{2} \|x^* - v^{k-1}\|_D^2.
\]

(ii) If \( \{D_k\} \) satisfies Assumption 3, then there exists \( C > 0 \) such that for \( k \geq 2 \)

\[
\Phi(x^k) - \Phi^* \leq \frac{C}{(k+1)^2}.
\]

**Proof** (i) Let \( k \geq 1 \). Note that in this situation \( h \equiv 0 \) and \( \partial h(x) = \{0\} \). Replace \( h, y, \bar{y}, D, t \) in Proposition 4.1 by \( 0, y^k, x^k, D_k, t_k \), respectively. Then, we obtain

\[
\Phi(x^k) \leq \Phi(x) + \frac{1}{2t_k} \|x - y^k\|_D^2 - \frac{1}{2t_k} \|x - x^k\|_D^2
\]

\[
= \Phi(x) + \frac{1}{t_k} \langle x - x^k, D_k(x^k - y^k) \rangle + \frac{1}{2t_k} \|x^k - y^k\|_D^2. \tag{10}
\]

Set \( x = x^{k-1} \) and \( x = x^* \) respectively in (10), it follows that

\[
\Phi(x^k) \leq \Phi(x^{k-1}) + \frac{1}{t_k} \langle x^{k-1} - x^k, D_k(x^k - y^k) \rangle + \frac{1}{2t_k} \|x^k - y^k\|_D^2,
\]

\[
\Phi(x^k) \leq \Phi^* + \frac{1}{t_k} \langle x^* - x^k, D_k(x^k - y^k) \rangle + \frac{1}{2t_k} \|x^k - y^k\|_D^2.
\]
Moreover, the update of \( \theta_k^{-1} \) and \( \theta_k^{-1} \) respectively and next add them together, then we obtain

\[
\phi(x^k) - \Phi \leq (1 - \theta_k^{-1})(\phi(x^{k-1}) - \Phi) + \frac{1}{2\theta_k}||x^k - y^k||_D^2
\]

\[
+ \frac{1}{t_k}((1 - \theta_k^{-1})x^{k-1} - x^k + \theta_k^{-1}x^*, D_k(x^k - y^k))
\]

\[
= (1 - \theta_k^{-1})(\phi(x^{k-1}) - \Phi) + \frac{1}{2\theta_k}||(1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}x^* - y^k||_D^2
\]

\[
- \frac{1}{2t_k}||(1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}x^* - x^k||_D^2.
\] (11)

Note that

\[
y^k = \Pi_{D_k}(x^{k-1} + \frac{\theta_k^{-1}-1}{\theta_k}(x^{k-1} - x^{k-2}))
\]

\[
= \Pi_{D_k}((1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}v^{k-1})
\]

and \( (1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}x^* \in Y \), thus

\[
||(1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}x^* - y^k||_D \leq \|\theta_k^{-1}(x^* - v^{k-1})\|_D
\] (12)

moreover, \( v^k = x^{k-1} + \theta_k(x^k - x^{k-1}) \) implies that

\[
||(1 - \theta_k^{-1})x^{k-1} + \theta_k^{-1}x^* - x^k||_D = \|\theta_k^{-1}(x^* - v^k)\|_D.
\] (13)

Thus, combining (11), (12), and (13), we have

\[
t_k \theta_k^2(\phi(x^k) - \Phi) + \frac{1}{2}||x^* - v^k||_D^2 \leq t_k(\theta_k^2 - \theta_k) (\phi(x^{k-1}) - \Phi) + \frac{1}{2}||x^* - v^{k-1}||_D^2.
\]

Moreover, the update of \( \{\beta_k\} \) (line 6 of Algorithm 2) implies that \( \theta_k^2 - \theta_k - t_{k-1}/t_k \theta_k^{-2} = 0 \). Plunge this into the above inequality, we prove (i).

(ii) For \( k \geq 1 \), define \( a_k = t_k \theta_k^2(\phi(x^k) - \Phi) + \frac{1}{2}||x^* - v^k||_D^2 \). Thus,

\[
a_{k+1} = t_{k+1} \theta_{k+1}^2(\phi(x^{k+1}) - \Phi) + \frac{1}{2}||x^* - v^{k+1}||_D^2
\]

\[
\leq t_k \theta_k^2(\phi(x^k) - \Phi) + \frac{1}{2}||x^* - v^k||_D^2
\]

\[
\leq t_k \theta_k^2(\phi(x^k) - \Phi) + \frac{1 + \eta_k}{2}||x^* - v^k||_D^2
\]

\[
\leq (1 + \eta_k)(t_k \theta_k^2(\phi(x^k) - \Phi) + \frac{1}{2}||x^* - v^k||_D^2)
\]

\[
= (1 + \eta_k)a_k
\]

where the first inequality follows from the targeted inequality in term (i).

Then, invoking Lemma 2.3 we conclude that the sequence \( \{a_k\} \) converges and thus bounded. Let \( C_1 > 0 \) satisfy \( a_k \leq C_1 \) for all \( k \), then we have for \( k \geq 1 \)

\[
\phi(x^k) - \Phi \leq \frac{C_1}{t_k \theta_k^2}.
\]
Now, for \( k \geq 2 \) we have
\[
\sqrt{t_{k-1} \theta_{k-1}} = \sqrt{\theta_k (\theta_k - 1) t_k} \leq \sqrt{\theta_k} - \frac{\sqrt{\theta_k}}{2}.
\]
Summing the above inequality from 2 to \( k \) and then by rearranging, we have
\[
\frac{1}{\theta_k} \sqrt{\theta_k} \leq \frac{1}{(\sqrt{t_1} + \frac{1}{2} \sum_{i=2}^{k} \sqrt{t_i})^2} \leq \frac{4}{(k+1)^2 t_{\min}}.
\]
Then, let \( C = 4C_1/t_{\min} \), it holds that for all \( k \geq 2 \)
\[
\Phi(x^k) - \Phi^* \leq \frac{C}{(k+1)^2}.
\]
Therefore, we obtain the optimal convergence rate.

5 Numerical results

In this section, we conduct numerical simulations on problems of sparse binary logistic regression and compressed sensing with Poisson noise. All experiments are implemented in MATLAB 2019a on a 64-bit PC with an Intel(R) Core(TM) i5-6200U CPU (2.30GHz) and 8GB of RAM.

5.1 Sparse binary logistic regression

In this part, we consider the following DC regularized sparse binary logistic regression model:
\[
\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-b_i (\mathbf{a}_i^T x))) + \lambda \|x\|_1 - \lambda \|x\|_2 \right\},
\]
where \( \lambda > 0 \) is a regularization parameter and \( \{(\mathbf{a}_1, b_1), \cdots, (\mathbf{a}_m, b_m)\} \) is a training set with observations \( \mathbf{a}_i \in \mathbb{R}^n \) and labels \( b_i \in \{-1, 1\} \). This fits \((P)\) with \( f(x) = \sum_{i=1}^{m} \log(1 + \exp(-b_i (\mathbf{a}_i^T x))) \), \( g(x) = \lambda \|x\|_1 \), and \( h(x) = \lambda \|x\|_2 \).

Here, we use \( \ell_{1-2} \) regularizer \cite{55} for promoting sparsity; see \cite{12} for alternative nonconvex regularizers. For this model, \( Y \) in Algorithm 1 is the whole space and \( \Pi_{Y}^{D_k} \) is the identity operator. Besides, the computation about the proximal map \( \text{Prox}_{D_k}^t \) has closed form, which can be found in \cite[Example 6.8]{5}.

Two tested datasets, \texttt{w8a} and \texttt{CINA}, obtained from \texttt{libsvm} \cite{13} are utilized to show the performances of our proposed method SPDCAe (Algorithm 1). We compare four algorithms for (14): the first two are versions derived from SPDCAe, the other are pDCAe \cite{54} and ADCA \cite{50}. Note that all of these algorithms are equipped with Nesterov’s extrapolation, and here we do not compare that out of this type, such as pDCA \cite{17} and the general iterative
shrinkage and thresholding algorithm (GIST) [16], since in general these methods do not work better than that with extrapolation; see [54] for a comparison of pDCA, pDCAe, and GIST. Next, we discuss the implementation details of the involved algorithms below:

- **SPDCAe** is SPDCAe with scaling and non-monotone backtracking line search. The scaling matrix $D_k$ is selected by the following procedure. Let $g^k = \nabla f(y^k)$ and $\bar{g}^k = \sum_{i=1}^k g^k \circ g^k$, then set
  \[
  D_k = \text{diag} \left( \max \left( \frac{1}{\gamma_k}, \min \left( \gamma_k, \sqrt{\bar{g}^k + \varepsilon e_n} \right) \right) \right),
  \]
  where $\varepsilon = 10^{-6}$ and $\gamma_k = \sqrt{1 + \frac{10^{13}}{1+10^2}}$. Here, all of the above operators are conducted componentwise. Note that the update of the scaling matrices as above is motivated by that in the adaptive gradient (AdaGrad) algorithm [14]. The selection of $\{\gamma_k\}$ can guarantee that $\{D_k\}$ satisfies Assumption 3 since $\{D_k\}$ converges to the identity matrix. In addition, $L$ is set as $10^{-10}$. The factor $\eta$ is set as 2 and the initial guesses $L_1$ for the datasets are fixed as 1; then for $k \geq 2$, the initial guess $L_k$ is selected as $L_{k-1}/2$ if $k$ is not divisible by 5, otherwise as $L_{k-1}$. The selection of extrapolation parameter $\{\beta_k\}$ uses the fixed ($T_2 = 200$) and adaptive restart scheme as introduced in Sect. 3.

- **PDCAe** is SPDCAe without scaling and with non-monotone backtracking line search. Compared with that implementation of SPDCAe1, two modifications are in order. First, the initial guesses $L_1$ for the datasets are set as 0.1. Second, the scaling matrix is the identity matrix.

- **pDCAe** also uses the fixed ($T_2 = 200$) and restart scheme for updating $\{\beta_k\}$. The smoothness parameter of logistic loss $f$ is estimated by computing the largest eigenvalue of the Hessian matrix of $f$.

- **ADCA** uses the same bound of smoothness parameter of $f$ as that in pDCAe. Besides, the parameter $q$ in ADCA is set as 3.

In our experiments, $\lambda$ are all set as $10^{-3}$. The initial point for all of the involved algorithms is randomly generated by the MATLAB command $\texttt{rand(n,1)}$. We stop the algorithms via the relative error $(F(x^k) - F^*)/F^* \leq tol$, where $x^*$ is an approximately solution derived by running PDCAe1 for 10000 iterations and $F^*$ is its corresponding function value.

Table 1 demonstrates the performances for w8a and CINA datasets. The total number of iterations and CPU time (in seconds) averaged on 10 runs for $tol \in \{10^{-2} : i \in \{1, 2, 3, 4\}\}$ are reported. The mark “Max” means the number of iterations exceeds 10000. We visualize in Fig. 1 the trend of the relative error with respect to the CPU time. It is observed tat SPDCAe1 performs the best among the involved algorithms, hitting all of the tolerances with the least amount of time and the least number of iterations. On the contrary, pDCAe does not work well, especially for CINA dataset. For w8a dataset, PDCAe1 is better than ADCA; while for CINA dataset the result is opposite. It seems that ADCA converges very fast when approaching the
A variable metric and extrapolated proximal DCA with backtracking

Next, by comparing SPDCAe1 with PDCAe1, the benefit from scaling is indicated; while by comparing PDCAe1 and pDCAe, we observe that the non-monotone backtracking promotes the performance. Finally, consider that SPDCAe1 performs the best, the benefits from both scaling and the non-monotone backtracking are indicated.

**Table 1** Total number of iterations and CPU time (in seconds) averaged on 10 runs for w8a and CINA datasets. Bold values correspond to the best results for each dataset.

| Dataset | Algorithm | tol : $10^{-2}$ | tol : $10^{-4}$ | tol : $10^{-6}$ | tol : $10^{-8}$ |
|---------|-----------|-----------------|-----------------|-----------------|-----------------|
|         |           | Time | Iter. | Time | Iter. | Time | Iter. | Time | Iter. |
| w8a     | SPDCAe1   | 0.11  | 18    | 0.18  | 32    | 0.23  | 41    | 0.29  | 50    |
|         | PDCAe1    | 0.20  | 36    | 0.32  | 57    | 0.38  | 70    | 0.48  | 89    |
|         | pDCAe     | 0.84  | 148   | 2.93  | 587   | 5.84  | 1113  | 7.75  | 1571  |
|         | ADCA      | 1.03  | 153   | 2.12  | 356   | 3.08  | 486   | 4.24  | 739   |
| CINA    | SPDCAe1   | 0.16  | 186   | 0.49  | 684   | 0.72  | 1059  | 1.01  | 1524  |
|         | PDCAe1    | 0.47  | 743   | 1.42  | 1992  | 3.56  | 3799  | 3.89  | 5964  |
|         | pDCAe     | 3.43  | 5781  | Max   | Max   | Max   | Max   | Max   | Max   |
|         | ADCA      | 0.91  | 1082  | 1.29  | 1656  | 1.76  | 2467  | 2.28  | 3152  |

![Graph](image1.png)

(a) w8a  
(b) CINA

**Fig. 1** Trend of the relative error for w8a and CINA datasets

5.2 Compressed sensing with Poisson noise

In this subsection, we consider the problem of recovering the sparse signal corrupted by Poisson noise. Specifically, we assume the observed data $b \in \mathbb{R}^m$ is the realization of a Poisson random vector with expected values being $Ax_{true} + bg$, where $x_{true}$ is the (actually unknown) signal of interest, $A \in \mathbb{R}^{m \times n}$ is the measurement matrix, and $bg$ (a small scalar close to 0) is a positive background. In case of Poisson noise, the generalized Kullback-Leibler
KL(x) := ∑_{i=1}^{n} \left\{ b_i \log \frac{b_i}{(Ax + bg)_i} + (Ax + bg)_i - b_i \right\}

is used to measure the distance of x from the observed data b. To recover the true solution x_{true}, we use the following DC optimization model

$$\min_{x \in \mathbb{R}^n} \{KL(x) + \lambda \|x\|_1 - \lambda \|x\|_2\},$$  \hspace{1cm} (15)

where \( \lambda > 0 \) is a regularization parameter. In [10], the \( \ell_1 \) norm is utilized for inducing sparsity, here we use \( \ell_1 - 2 \) penalty instead. Then (15) matches \((P)\) with \( f(x) = KL(x) \), \( g(x) = \lambda \|x\|_1 + \delta_{\mathbb{R}_+}(x) \), and \( h(x) = \lambda \|x\|_2 \). Thus \( Y \) in Algorithm 1 is the nonnegative orthant and \( \Pi^{\mathbb{R}^+}_Y \) is not up to the underlying inner product. Besides, the computation about the proximal map \( \text{Prox}_{D_k}^{(g)} \) has closed form, which can be found in [5, Lemma 6.5].

We use the same procedure in [10] to generate the experimental data. The measurement matrix A actually depends on a probability, which is set as 0.9 in our test. For reader’s convenience, we describe the procedure below:

- The matrix A has been generated as detailed in [51] so that A preserves both the positivity and the flux of any signal (i.e., if \( z \geq 0 \), then \( (Az)_i \leq \sum_{i=1}^{n} z_i \)).
- The signal \( x_{true} \in \mathbb{R}^{5000} \) has all zeros except for 20 nonzeros entries drawn uniformly in the interval \([0, 10^5] \).
- The observed signal \( b \in \mathbb{R}^{1000} \) has been obtained by corrupting the vector \( Ax_{true} + bg \) (\( bg = 10^{-10} \)) by means of the MATLAB \texttt{imnoise} function.

Note that in this situation, the estimate of the bound of the smoothness parameter of KL is problematic since it is an extremely large number [18]. Thus, taking into account the efficiency, the aforementioned pDCAe and ADCA are not applicable; however, SPDCAe is still valid, from which we derive four versions: SPDCAe1, PDCAe1, SPDCAe0, and PDCAe0. The first two have already been introduced in the above subsection. Here, SPDCAe0 and PDCAe0 are their counterparts with monotone backtracking line search. Specifically, the first is SPDCAe with scaling and monotone backtracking, while the second is SPDCAe without scaling and with monotone backtracking. Our intention here is to show the effect from incorporating both the variable metric method and the non-monotone backtracking.

In our experiment, \( \lambda \) is set as \( 10^{-3} \). The scaling matrices for SPDCAe1 and SPDCAe0 have been selected by writing the gradient of KL as

$$-\nabla KL(x) = U_{KL}(x) - V_{KL}(x)$$

with \( U_{KL}(x) \geq 0 \) and \( V_{KL}(x) > 0 \); see [21] for a detailed description of such a decomposition. Then \( D_k \) is defined as

$$D_k = \text{diag} \left( \max \left( \frac{1}{\gamma_k}, \min \left( \gamma_k, \frac{y_k}{V_{KL}(y_k)} \right) \right) \right)^{-1}$$
with $\gamma_k = \sqrt{1 + \frac{10^{13}}{(k+1)^2}}$. Next for SPDCAe1 and PDCAe1, $L$ is set as $10^{-10}$; besides, the factor $\eta$ in SPDCAe1 and PDCAe1 is set as 2, while that in SPDCAe0 and PDCAe0 is set as 1.2. The initial guesses $L_1$ for SPDCAe1 and SPDCAe0 are set as 0.1, and that for PDCAe1 and PDCAe0 are set as $10^{-5}$. Moreover, for $k \geq 2$, the initial guesses for SPDCAe1 and PDCAe1 are selected as $L_{k-1}/2$ if $k$ is not divisible by 5, otherwise as $L_{k-1}$. Next for the involved algorithms, the initial point is the $n$-dimensional column vector of all ones and the extrapolation parameter $\{\beta_k\}$ is all updated via the fixed ($T_2 = 200$) and adaptive restart scheme. We stop the algorithms via the relative error $(F(x^k) - F^*)/F^* \leq tol$, where $x^*$ is an approximately solution derived by running PDCAe1 for 10000 iterations and $F^*$ is its corresponding function value.

In Table 2, we present in detail the total number of iterations and CPU time (in seconds) averaged on 10 runs for $tol \in \{10^{-i} : i \in \{1, 2, \cdots, 5\}\}$. The mark “Max” means the number of iterations exceeds 10000. Here, we observes that among all of the involved algorithms, PDCAe0 performs the worst, which requires 2910 steps to reach the tolerance $tol = 10^{-1}$ and always hits the maximum iterations 10000 for other tolerances; on the contrary, SPDCAe1 performs the best, hitting all of the tolerances with the least amount of time and the least number of iterations. Next in Fig. 2(a), the trend of the relative error with respect to the CPU time is plotted. Here, by comparing the trends of SPDCAe1 and SPDCAe0 (also PDCAe1 and PDCAe0) we observe the benefit from introducing the non-monotone backtracking; next by comparing SPDCAe1 and PDCAe1 (also SPDCAe0 and PDCAe0), the benefit from scaling is indicated; besides, by comparing SPDCAe0 and PDCAe1 we observe that the former benefits much from the scaling procedure and later is worse than PDCAe1. Anyway, SPDCAe1 works the best, indicating the benefit from both the scaling method and the non-monotone backtracking. Finally, we observe from Fig. 2(b) that the solution recovered by SPDCAe1 is very close to the true solution.

As a conclusion, for the Poisson denoising problem, SPDCAe1 could be a promising algorithm since the benefit from incorporating the inexpensive diagonally scaling procedure for better local approximation and using the non-monotone backtracking for adaptive step-sizes selection.

| Algorithm | $tol : 10^{-1}$ | $tol : 10^{-2}$ | $tol : 10^{-3}$ | $tol : 10^{-4}$ | $tol : 10^{-5}$ |
|-----------|----------------|----------------|----------------|----------------|----------------|
|           | Time | Iter. | Time | Iter. | Time | Iter. | Time | Iter. | Time | Iter. |
| SPDCAe1   | 0.22 | 28   | 0.29 | 38   | 0.31 | 42   | 0.39 | 54   | 0.40 | 55   |
| PDCAe1    | 0.57 | 78   | 0.80 | 101  | 0.81 | 104  | 0.96 | 110  | 0.97 | 112  |
| SPDCAe0   | 0.56 | 79   | 1.86 | 298  | 3.25 | 456  | 6.91 | 953  | 18.62| 2508 |
| PDCAe0    | 19.66| 2910 | Max  | Max  | Max  | Max  | Max  | Max  | Max  | Max  |
6 Conclusion

In this paper, we propose a DC programming algorithm, called SPDCAe, for solving a composite DC program \((P)\), which incorporates the variable metric method, the Nesterov’s extrapolation, and the backtracking line search (not necessarily monotone). We establish the subsequential convergence to a critical point of SPDCAe under suitable selections of the extrapolation parameters and the scaling matrices. Besides, we also demonstrate that the convex version of SPDCAe for \((P_c)\), denoted by SFISTA, enjoys the optimal \(O(1/k^2)\) convergence rate in function values. This rate of convergence coincides with that of the well-known FISTA algorithm \([6]\) and SFEM \([10]\) (a scaling version of FISTA with monotone backtracking). Numerical simulations on sparse binary logistic regression demonstrate the good performances of our methods (especially that with scaling and non-monotone backtracking) compared with pDCAE \([54]\) and ADCA \([50]\). Furthermore, for compressed sensing with Poisson noise problem, both pDCAe and ADCA are not applicable, while our algorithm is still valid, where we show the benefits of including scaling and non-monotone backtracking.

As further researches, first, the global sequential convergence of SPDCAe to a critical point under the Kurdyka-Lojasiewicz property \([9,3,4]\) is worth noting; second, the SPDCAe could be compared with other accelerated algorithms, e.g., boosted-DCA \([2,38]\), inertial-DCA \([43,56]\) and accelerated methods based on second-order ODE \([32,15]\) for some suitable applications; moreover, it deserves developing some ingenious procedures for both scaling and adaptive step-sizes selection. Finally, it might be meaningful for designing some inexact variants for addressing the case where the computation of the proximal mapping of \(g\) can not be exactly conducted.

Fig. 2 Performance for the Poisson denoising problem

(a) Trend of relative error

(b) The true solution and the solution obtained by SPDCAe under \(tol = 10^{-5}\)
Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant 11601327).

References

1. Aleksandrov, A.D.: On the surfaces representable as difference of convex functions. Sibirskie Elektronnye Matematicheskie Izvestiya 9, 360–376 (2012)
2. Artacho, F.J.A., Vuong, P.T.: The boosted dc algorithm for nonsmooth functions. SIAM J. Optim. 30(1), 980–1006 (2020)
3. Attouch, H., Bolte, J.: On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Mathematical Programming 116(1), 5–16 (2009)
4. Attouch, H., Bolte, J., Svaiter, B.F.: Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. Mathematical Programming 137(1), 91–120 (2015)
5. Beck, A.: First-order methods in optimization. SIAM, Philadelphia (2017)
6. Beck, A., Marc, T.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences 2(1), 183–202 (2009)
7. Beck, A., Vaisbourd, Y.: Globally solving the trust region subproblem using simple first-order methods. SIAM Journal on Optimization 28(3), 1951–1967 (2018)
8. Bertsekas, D.: Convex optimization algorithms. Athena Scientific (2015)
9. Bolte, J., Daniilidis, A., Lewis, A.: The lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization 17(4), 1205–1223 (2007)
10. Bonettini, S., Porta, F., Ruggiero, V.: A variable metric forward-backward method with extrapolation. SIAM J. Optim. 38(4), A2588–A2584 (2016)
11. Bonettini, S., Porta, F., Ruggiero, V., L., Z.: Variable metric techniques for forward-backward methods in imaging. Journal of Computational and Applied Mathematics 385, 113192 (2021)
12. Candès, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted ℓ1 minimization. Journal of Fourier Analysis and Applications 14, 877–905 (2008)
13. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines 2 (2011)
14. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res 12, 2121–2159 (2011)
15. França, G., Robinson, D.P., Vidal, R.: Gradient flows and proximal splitting methods: A unified view on accelerated and stochastic optimization. Physical Review E 103(5), 053304 (2021)
16. Gong, P., Zhang, C., Lu, Z., Huang, J., Ye, J.: A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. pp. 37–45. PMLR (2013)
17. Gotou J. Y., T.A., Tono, K.: DC formulations and algorithms for sparse optimization problems. Mathematical Programming 169(1), 141–176 (2018)
18. Harmany, Z.T., Marcia, R.F., M, R.: This is spiral-tap: Sparse poisson intensity reconstruction algorithms–theory and practice. In: IEEE Transactions on Image Processing, pp. 1084–1096 (2011)
19. Hiriart-Urruty, J.B.: Generalized differentiability/duality and optimization for problems dealing with differences of convex functions. In: Convexity and duality in optimization, pp. 37–70. Springer (1985)
20. Horst, R., Thoi, N.V.: DC programming: Overview. Journal of Optimization Theory and Applications 103, 1–43 (1999)
21. Lantéri, H., Roche, M., Cuevas, O., Aime, C.: A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints. Signal Processing 81(5), 945–974 (2001)
22. Le Thi, H., Pham Dinh, T.: A continuous approach for large-scale constrained quadratic zero-one programming. Optimization 45(3), 1–28 (2001)
23. Le Thi, H.A., Belghiti, M.T., Pham, D.T.: A new efficient algorithm based on dc programming and dca for clustering. Journal of Global Optimization 37(4), 593–608 (2007)
24. Le Thi, H.A., Moeini, M., Pham, D.T., Judice, J.: A dc programming approach for solving the symmetric eigenvalue complementarity problem. Computational Optimization and Applications 51(3), 1097–1117 (2012)
25. Le Thi, H.A., Pham, D.T.: Large-scale molecular optimization from distance matrices by a dc optimization approach. SIAM Journal on Optimization 14(1), 77–114 (2003)
26. Le Thi, H.A., Pham, D.T.: The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems. Annals of operations research 133(1-4), 23–46 (2005)
27. Le Thi, H.A., Pham, D.T.: DC programming and DCA: thirty years of developments. Mathematical Programming 169(1), 5–68 (2018)
28. Le Thi, H.A., Pham, D.T., Le, H.M., Vo, X.T.: DC approximation approaches for sparse optimization. European Journal of Operational Research 244(1), 26–46 (2015)
29. Nesterov, Y.E.: A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In: Dokl. akad. nauk Ssr, vol. 269, pp. 543–547 (1983)
30. Nesterov, Y.E.: Gradient methods for minimizing composite functions. Mathematical Programming 140(1), 125–161 (2013)
31. Niu, Y.S.: Programmation dc et dca en optimisation combinatoire et optimisation polynomiale via les techniques de sdp. Ph.D. thesis, INSA de Rouen, France (2010)
32. Niu, Y.S., Glowinski, R.: Discrete dynamical system approaches for boolean polynomial optimization. to appear in Journal of Scientific Computing, arXiv preprint arXiv:1912.10221 (2019)
33. Niu, Y.S., Judice, J., Le Thi, H.A., Pham, D.T.: Improved dc programming approaches for solving the quadratic eigenvalue complementarity problem. Applied Mathematics and Computation 353, 95–113 (2019)
34. Niu, Y.S., Pham, D.T.: A dc programming approach for mixed-integer linear programs. In: International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences, pp. 244–253. Springer (2008)
35. Niu, Y.S., Pham, D.T.: Dc programming approaches for bmi and qmi feasibility problems. In: Advanced Computational Methods for Knowledge Engineering, pp. 37–63. Springer (2014)
36. Niu, Y.S., Pham, D.T., Le Thi, H.A., Judice, J.J.: Efficient dc programming approaches for the asymmetric eigenvalue complementarity problem. Optimization Methods and Software 28(4), 812–829 (2013)
37. Niu, Y.S., Wang, Y.J., Le Thi, H.A., Pham, D.T.: Higher-order moment portfolio optimization via an accelerated difference-of-convex programming approach and sums-of-squares. arXiv :1906.01509 (2019)
38. O'Donoghue, B., Candès, E.: Adaptive restart for accelerated gradient schemes. Foundations of Computational Mathematics 15, 715–732 (2015)
39. de Oliveira, W.: The abc of dc programming. Set-Valued and Variational Analysis 28(4), 679–706 (2020)
40. de Oliveira, W., Tcheou, M.P.: An inertial algorithm for dc programming. Set-Valued and Variational Analysis 27(4), 895–919 (2019)
41. Pham, D.T., Le Thi, H.A.: Convex analysis approach to d.c. programming: theory, algorithms and applications. Acta Math. Vietnam. 22(1), 289–355 (1997)
42. Pham, D.T., Le Thi, H.A.: A dc optimization algorithm for solving the trust-region subproblem. SIAM Journal on Optimization 8(2), 476–505 (1998)
43. Pham, D.T., Le Thi, H.A.: Recent advances in DC programming and DCA. In: Transactions on Computational Intelligence XIII, pp. 1–37 (2014)
47. Pham, D.T., Le Thi, H.A., Pham, V.N., Niu, Y.S.: Dc programming approaches for discrete portfolio optimization under concave transaction costs. Optimization letters 10(2), 261–282 (2016)
48. Pham, D.T., Niu, Y.S.: An efficient dc programming approach for portfolio decision with higher moments. Computational Optimization and Applications 50(3), 525–554 (2011)
49. Pham, D.T., Souad, E.B.: Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. In: Fermat Days 85: Mathematics for Optimization, vol. 129, pp. 249–271 (1986)
50. Phan, D.N., Le, H.M., Le Thi, H.A.: Accelerated difference of convex functions algorithm and its application to sparse binary logistic regression. In: IJCAI, pp. 1369–1375 (2018)
51. Raginsky, M., Willett, R.M., Harmany, Z.T., Marcia, R.F.: Compressed sensing performance bounds under poisson noise. In: IEEE Transactions on Signal Process, vol. 58, pp. 3990–4002 (2010)
52. Rockafellar, R.T.: Convex analysis, vol. 36. Princeton university press (1970)
53. Scheinberg, K., Goldfarb, D., Bai, X.: Fast first-order methods for composite convex optimization with backtracking. Foundations of Computational Mathematics 14, 389–417 (2014)
54. Wen, B., Chen, X., Pong, T.K.: A proximal difference-of-convex algorithm with extrapolation. Computational optimization and applications 69(2), 297–324 (2018)
55. Yin, P., Lou, Y., He, Q., Xin, J.: Minimization of $\ell_{1-2}$ for compressed sensing. SIAM J. Sci. Comput. 37, A536–A563 (2016)
56. You, Y., Niu, Y.S.: A refined inertial dc algorithm for dc programming. Optimization and Engineering (2022)