Comput Optim Appl (2013) 56:559–572
DOI 10.1007/s10589-013-9564-5

A customized proximal point algorithm for convex minimization with linear constraints

Bingsheng He · Xiaoming Yuan · Wenxing Zhang

Received: 14 November 2010 / Published online: 7 May 2013

Abstract This paper demonstrates a customized application of the classical proximal point algorithm (PPA) to the convex minimization problem with linear constraints. We show that if the proximal parameter in metric form is chosen appropriately, the application of PPA could be effective to exploit the simplicity of the objective function. The resulting subproblems could be easier than those of the augmented Lagrangian method (ALM), a benchmark method for the model under our consideration. The efficiency of the customized application of PPA is demonstrated by some image processing problems.

Keywords Convex minimization · Proximal point algorithm · Resolvent operator · Augmented Lagrangian method

B. He was supported by the NSFC grant 91130007 and the grant of MOE of China 20110091110004.

X. Yuan was supported by the General Research Fund from Hong Kong Research Grants Council: 203712.

B. He
International Center of Management Science and Engineering and Department of Mathematics, Nanjing University, Nanjing, 210093, China
e-mail: hebma@nju.edu.cn

X. Yuan (✉)
Department of Mathematics, Hong Kong Baptist University, Hong Kong, China
e-mail: xmyuan@hkbu.edu.hk

W. Zhang
School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China
e-mail: wxzh1984@126.com

1 Introduction

We consider a basic optimization model, i.e., the convex minimization problem with linear equality constraints,

$$\min\{\theta(x) \mid Ax = b, x \in \mathcal{X}\}, \tag{1.1}$$

where $\theta(x) : \Re^n \to \Re$ is a convex (not necessarily smooth) function, $A \in \Re^{m\times n}$, $b \in \Re^m$ and $\mathcal{X} \subseteq \Re^n$ is a nonempty closed convex set. Throughout, the solution set of (1.1) denoted by $\mathcal{X}^*$ is assumed to be nonempty.

The augmented Lagrangian method (ALM) in [15, 20] plays a fundamental theoretical and algorithmic role in solving the model (1.1). The iterative scheme of ALM for (1.1) reads

$$\begin{cases} x^{k+1} = \text{Argmin}\{\theta(x) - (\lambda^k)^T(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2 \mid x \in \mathcal{X}\}, \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - b), \end{cases} \tag{1.2}$$

where $\lambda \in \Re^m$ is the Lagrange multiplier and $\beta > 0$ is a penalty parameter. We refer to, e.g., [3, 19], for the advantages of ALM over some Lagrange based methods or penalty methods. One particular advantage is that the penalty parameter β is not required to tend to infinity by the ALM. Thus, the subproblem in (1.2) is more likely to be well-conditioned and it becomes possible to apply sophisticated methods (e.g., Newton-like solvers) to tackle the ALM subproblems.

Our discussion is for the specific case of (1.1) where the objective function $\theta(x)$ is simple in the sense that following problem is easy to solve (e.g., a closed-form solution exits or efficient solvers are available)

$$x := \text{Argmin}\left\{\theta(x) + \frac{r}{2}\|x - a\|^2 \mid x \in \mathcal{X}\right\}, \tag{1.3}$$

for any given $a \in \Re^n$ and $r > 0$. Our interest in this specific scenario is inspired mainly by the increasingly popular applications of sparse optimization in various areas such as compressed sensing, image processing, and statistics, all of which are the cases of (1.3) with $\theta(x) = \|x\|_1 := \sum_{i=1}^m |x_i|$. For the case of (1.3) where $\mathcal{X} = \Re^m$, the simplicity assumption (1.3) essentially means the resolvent operator of $\theta(x)$, which is defined as $(I + \frac{1}{r}\partial\theta)^{-1}(a)$, has a closed-form representation. Here, $\partial(\cdot)$ denotes the subdifferential of a convex but nonsmooth function.

Under the simplicity assumption (1.3), however, the ALM subproblem in (1.2) could be still hard whenever the coefficient matrix A is not a multiple of the identity. The main purpose of this paper is to show that if we apply the very classical proximal point algorithm (PPA) [18] to the saddle-point reformulation of (1.1) but with a customized proximal parameter in accordance with the specific structure of (1.1), then the difficulty of solving the resulting subproblem at each iteration is the same as solving (1.3). Therefore, once the objective function $\theta(x)$ in (1.1) has the simple property given in (1.3), our customized application of PPA is able to exploit it while the direct application of ALM (1.2) does not. More concretely, if $\theta(x)$ is simple enough such that the problem in (1.3) has a closed-form solution (which is the case when $\theta(x) = \|x\|_1$ and $\mathcal{X} = \Re^m$), so are the subproblems of our proposing customized application of PPA. Given the insightful explanation in [21] of how the ALM is related

the PPA, i.e., the ALM for the primal problem (1.1) is exactly the application of PPA to its dual problem, this paper demonstrates the close relationship between these two fundamental methods from another perspective. That is, we show that the PPA can be directly applied to solve the primal problem (1.1), and this primal application could be more effective than its dual application for some specific scenarios of (1.1).

The rest of the paper is organized as follows. In Sect. 2, we review the application of PPA to the saddle-point reformulation of (1.1), based on which our idea of exploiting the simplicity in (1.3) via a customized proximal parameter can be clearly illustrated. In Sect. 3, we specify the implementation details of the resulting customized PPA for (1.1). In Sect. 4, we show that our discussion for (1.1) can be easily extended to a general case with linear inequality constraints. Then, in Sect. 5, we apply the derived customized algorithms to solve some image processing problems and report some preliminary numerical results. Finally, in Sect. 6, we make some conclusions.

2 Motivation

We provide a saddle-point reformulation of (1.1); and specify the idea of identifying a customized application of PPA with the purpose of exploiting the simplicity in (1.3).

By deriving the optimality condition as a corresponding saddle-point problem, or the primal-dual reformulation, of (1.1), we can easily see that solving (1.1) amounts to finding a pair of (x^*, λ^*) such that

$$\begin{cases} x^* \in \mathcal{X}, \quad (x - x^*)^T \{f(x^*) - A^T\lambda^*\} \geq 0, \quad \forall x \in \mathcal{X}, \\ \qquad\qquad Ax^* - b = 0, \end{cases} \tag{2.1}$$

where $f(x^*) \in \partial\theta(x^*)$. Further, by denoting

$$u = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F(u) = \begin{pmatrix} f(x) - A^T\lambda \\ Ax - b \end{pmatrix} \quad \text{and} \quad \Omega = \mathcal{X} \times \Re^m, \tag{2.2}$$

the system (2.1) can be characterized by the following variational inequality problem (VIP): Finding $u^* \in \Omega$ and $f(x^*) \in \partial\theta(x^*)$ such that

$$\left(u - u^*\right)^T F\left(u^*\right) \geq 0, \quad \forall u \in \Omega. \tag{2.3}$$

We denote (2.3) by VI(Ω, F). Clearly, (2.3) is a monotone VIP because of the monotonicity of $\partial\theta(x)$. In addition, because of the assumed nonemptiness of $\mathcal{X}^*$, the solution set of VI(Ω, F) denoted by Ω^* is also nonempty. We can further reformulate (2.3) as the problem of finding a root of a maximal monotone operator. But the variational form (2.3) is sufficient for our later analysis.

Applying the PPA in [18] for (2.3) results in the iterative scheme

$$\left(u - u^{k+1}\right)^T \left\{F\left(u^{k+1}\right) + G\left(u^{k+1} - u^k\right)\right\} \geq 0, \quad \forall u \in \Omega, \tag{2.4}$$

where the metric proximal parameter $G \in \Re^{(n+m)\times(n+m)}$ is required to be positive definite, see e.g., [5]. A particular choice of G is $G = \beta \cdot I$ where $\beta > 0$ and I is the identity matrix, which means the proximal terms $(u^{k+1} - u^k)$ is regularized uniformly. For the general case of (2.4) with a generic proximal parameter G, the proximal subproblem (2.4) could be as difficult as the original problem (2.3), and

sophisticated strategies for seeking approximate solutions of the PPA subproblem should be investigated.

Taking into account the specific structure in (2.2), we find that if we choose G as

$$G = \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix}, \tag{2.5}$$

where $r > 0$, $s > 0$, and they satisfies $rs > \|A^T A\|$ in order to ensure the positive definiteness of G, then the proximal subproblem (2.4) with this particular proximal parameter reduces to

$$\begin{pmatrix} x - x^{k+1} \\ \lambda - \lambda^{k+1} \end{pmatrix}^T \left\{ \begin{pmatrix} f(x^{k+1}) - A^T\lambda^{k+1} \\ Ax^{k+1} - b \end{pmatrix} + \begin{pmatrix} rI_n & -A^T \\ -A & sI_m \end{pmatrix} \begin{pmatrix} x^{k+1} - x^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} \right\} \geq 0, \tag{2.6}$$

for any $(x, \lambda) \in \Omega$. Equivalently, we need to solve

$$\big(Ax^{k+1} - b\big) - A\big(x^{k+1} - x^k\big) + s\big(\lambda^{k+1} - \lambda^k\big) = 0, \tag{2.7}$$

and

$$x^{k+1} \in \mathcal{X}, \quad \big(x - x^{k+1}\big)^T \big\{ f\big(x^{k+1}\big) - A^T\big(2\lambda^{k+1} - \lambda^k\big) + r\big(x^{k+1} - x^k\big) \big\} \geq 0, \quad \forall x \in \mathcal{X}. \tag{2.8}$$

Note that the solution of (2.7) is given explicitly by

$$\lambda^{k+1} = \lambda^k - \frac{1}{s}\big(Ax^k - b\big) \tag{2.9}$$

and (2.8) amounts to solving the convex programming problem

$$x^{k+1} = \operatorname{Argmin}\left\{ \theta(x) + \frac{r}{2}\left\| \big(x - x^k\big) - \frac{1}{r}A^T\big(2\lambda^{k+1} - \lambda^k\big) \right\|^2 \,\middle|\, x \in \mathcal{X} \right\}. \tag{2.10}$$

Thus, the subproblem (2.4) with the particular proximal parameter (2.5) is specified into (2.9) and (2.10), and it becomes easy if the simplicity assumption (1.3) holds. In this sense, this primal application of PPA to (1.1) is more effective than its dual application (1.2), which has harder subproblems than (2.10).

3 Algorithm

Now, based on our customized application of PPA, we can specify an algorithm for (1.1) with the simplicity assumption (1.3). Obviously, acceleration strategies in the literature of generic PPA can be combined with our customized PPA. We focus on the acceleration scheme in [13] (see also [9]) which combines a computationally trivial relaxation step with the original PPA.

From now on, we denote by $\tilde{u}^k = (\tilde{x}^k, \tilde{\lambda}^k)$ the output of the proximal subproblem (2.4) with the given iterate $u^k = (x^k, \lambda^k)$, and by $u^{k+1} = (x^{k+1}, \lambda^{k+1})$ the new iterate combined with the relaxation step in [13]. Equations (2.9) and (2.10) are thus rewritten as (3.1a) and (3.1b), respectively.

Algorightm 1: A customized PPA for (1.1)

Step 0. Choose G according to (2.5), let $\gamma \in (0,2)$, and take $(x^0, \lambda^0) \in \Re^n \times \Re^m$.
Step k. $(k \geq 0)$ Let

$$\tilde{\lambda}^k = \lambda^k - \frac{1}{s}\left(Ax^k - b\right), \tag{3.1a}$$

and

$$\tilde{x}^k = \operatorname{Argmin}\left\{\theta(x) + \frac{r}{2}\left\|\left(x - x^k\right) - \frac{1}{r}A^T\left(2\tilde{\lambda}^k - \lambda^k\right)\right\|^2 \,\middle|\, x \in \mathcal{X}\right\}. \tag{3.1b}$$

Set

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}. \tag{3.2}$$

Remark 3.1 Since Algorithm 1 is just a customized version of the relaxed PPA in [13], its convergence is guaranteed by classical theory of PPA. We thus omit it. At the same time, we would mention that it is easy to show that the sequence $\{x^k, \lambda^k\}$ generated by Algorithm 1 is contractive with respective to the solution set Ω^*. Thus, its convergence can also be easily established under the analytic framework of contraction type methods in [4].

4 Extension

Our analysis can be easily extended to the case with linear inequality constraints:

$$\min\left\{\theta(x) \mid Ax \geq b,\ x \in \mathcal{X}\right\}. \tag{4.1}$$

With analogous analysis in Sect. 2, we can propose a customized PPA for (4.1) whose subproblems are also of the same difficulty as (1.3). The detail is similar to Algorithm 1; thus omitted.

Algorightm 2: A customized PPA for (4.1)
Step 0. Choose G according to (2.5), let $\gamma \in (0,2)$, and take $(x^0, \lambda^0) \in \Re^n \times \Re^m$.
Step k. $(k \geq 0)$ Let

$$\tilde{\lambda}^k = P_{\Re^m_+}\left[\lambda^k - \frac{1}{s}\left(Ax^k - b\right)\right],$$

where $P_{\Re^m_+}$ denotes the projection onto $\Re^m_+$ under the Euclidean norm, and

$$\tilde{x}^k = \operatorname{Argmin}\left\{\theta(x) + \frac{r}{2}\left\|\left(x - x^k\right) - \frac{1}{r}A^T\left(2\tilde{\lambda}^k - \lambda^k\right)\right\|^2 \,\middle|\, x \in \mathcal{X}\right\}.$$

Set

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \gamma \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

5 Numerical experiments

We now apply the proposed algorithms to solve some image reconstruction problems and report the numerical results. Our algorithms were coded by MATLAB 7.9 and all the numerical experiments were conducted on a Lenovo personal computer with Intel Core(TM) CPU 2.30 GHz and 8 G memory.

5.1 Comparison of Algorithm 1 with ALM

Since the purpose of investigating customized applications of PPA is mainly to alleviate the ALM subproblem in (1.2) when the simplicity assumption (1.3) holds for (1.1), we first focus on the comparison of Algorithm 1 with the direct application of ALM (1.2).

A specific application of (1.1) is the basis pursuit (BP) problem arising from many areas such as compressed sensing and image processing (see e.g., [7]). In this subsection, we apply Algorithm 1 ("C-PPA" for short) to the wavelet-based image processing problem. A brief introduction for the wavelet-based image processing is as follows, and for more details the reader is referred to some monographs such as [23]. Let $\mathbf{x} \in \Re^l$ represent an $l_1 \times l_2$ image with $l = l_1 \cdot l_2$ (the two-dimensional images are tackled by vectorizing them as one-dimensional vectors, e.g., in the lexicographic order) and $W \in \Re^{l\times n}$ be a wavelet dictionary, i.e., each column of W be the elements of a wavelet frame. Commonly, the image $\mathbf{x}$ possesses a sparse representation under the dictionary W, i.e., $\mathbf{x} = Wx$ with x being a sparse vector. The wavelet-based image processing thus considers recovering the real image $\mathbf{x}$ from an observation b which might have some missing pixels or convolutions. Specifically, the model for the wavelet-based image processing can be casted as

$$\min\big\{\|x\|_1 \mid BWx = b\big\}, \tag{5.1}$$

where $B \in \Re^{m\times l}$ is a matrix representation of convolution or downsampling operators, and it is typically ill-conditioned; and $\|x\|_1$ is to deduce a sparse representation under the wavelet dictionary. Thus, the wavelet-based image deconvolution, vignetting, inpainting, zooming, and their combinations are covered by (5.1) when B is specified into some concrete operators, see e.g., [7, 8, 10, 17, 23].

Our experiments focus on the wavelet-based image inpainting and zooming problems. For the inpainting problem, the matrix $B \in \Re^{l\times l}$ (also called "mask") in (5.1) is a diagonal matrix whose diagonal elements are either 0 or 1, where the locations of 0 correspond to missing pixels in images and locations of 1 correspond to the pixels to be kept. For the wavelet-based image zooming problem, the matrix B can be expressed as $B = SH$ where $S \in \Re^{m\times l}$ is a downsampling matrix and $H \in \Re^{l\times l}$ is a blurry matrix. We adopt the reflective boundary condition for both image reconstruction problems. Hence, H can be diagonalized by the discrete consine transform (DCT), i.e., $H = \mathcal{C}^{-1}\Lambda\mathcal{C}$ where $\mathcal{C}$ represents the DCT and Λ is a diagonal matrix whose diagonal entries are eigenvalues of H (see e.g., [14] for details). Under this circumstance, the subproblem (1.3) possesses the following closed-form solution

$$\left(I + \frac{1}{r}\partial\theta\right)^{-1}(a) = \mathrm{sign}(a) \circ \max\big\{|a| - 1/r, 0\big\}, \quad \forall a \in \Re^n, \tag{5.2}$$

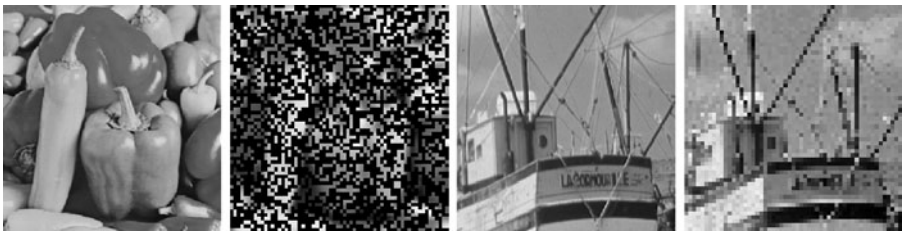

Fig. 1 Original Peppers, degraded Peppers, original Boat, and degraded Boat

where the operator "$\circ$" stands for component wise scalar multiplication. Since the dictionary W has the property $WW^T = I$, the blurry matrix H can be diagonal by DCT and the binary matrix (both mask and downsampling matrices) S satisfies $\|S\| = 1$, we have $\|A^T A\| = 1$ (where $A := BW$) for the wavelet-based image inpainting and zooming problems. Therefore, the requirement $rs > \|A^T A\|$ in (2.5) reduces to $rs > 1$ in order to implement C-PPA for our experiments.

We test the 256×256 images of Peppers.png and Boat.png for the image inpainting and image zooming problems, respectively. The dictionary W is chosen as the inverse discrete Haar wavelet transform with a level of 6 (see e.g., [1, 2]). Below we give the detail of how the tested images are degraded. Both the clean and degraded images are displayed in Fig. 1.

- For the image inpainting problem, the original image Peppers is first blurred by the out-of-focus kernel with a radius of 7. Then 60 % pixels of the blurred images are lost by implementing a mask operator S. The positions of missing pixels are located randomly.
- For the image zooming problem, the original image Boat is downsampled by a downsampling matrix S with a factor of 4. Then, the downsampled image is corrupted by a convolution whose kernel is generated by `fspecial('gaussian',9,2.5)` of MATLAB.

For solving the concrete application (5.1) by ALM (1.2), we need to employ certain algorithms to solve the x-subproblem in (1.2) iteratively. We choose the popular solvers "ISTA" in [8] and "FISTA" in [2] for this purpose, and they both allow for a maximal number of 10 iterations for the internal iteration. The ALM (1.2) embedded by ISTA and FISTA are denoted by "ALM-ISTA" and "ALM-FISTA", respectively. The implementation details including parameter values are described as below. We refer to [16] for more details about how to choose the involved parameters.

- For the image inpainting problem, we take $r = 0.6$, $s = 1.02/r$, $\gamma = 1.9$, and $(x^0, \lambda^0) = (W^T(b), \mathbf{0})$ to implement C-PPA; $\beta = 10$ and $\lambda^0 = \mathbf{0}$ in (1.2).
- For the image zooming problem, we take $r = 0.55$, $s = 1.02/r$, $\gamma = 1.2$, and $(x^0, \lambda^0) = (\mathbf{0}, \mathbf{0})$ to implement C-PPA; $\beta = 10$ and $\lambda^0 = \mathbf{0}$ in (1.2).

As usual, the quality of the reconstruction is measured by the signal-to-noise ratio (SNR) in decibel (dB)

$$\text{SNR} := 20 \log_{10} \frac{\|\mathbf{x}\|}{\|\bar{\mathbf{x}} - \mathbf{x}\|},$$

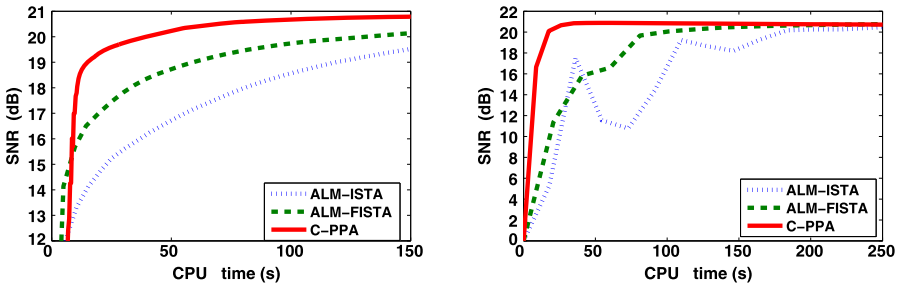

Fig. 2 Evolutions of SNR w.r.t. computing time. *Left*: for image inpainting. *Right*: for image zooming

where $\bar{\mathbf{x}}$ is a reconstructed image and $\mathbf{x}$ is a clean image. We plot the evolutions of SNR with respect to computing time for ALM-ISTA, ALM-FISTA and C-PPA in Fig. 2. It shows that all these three methods are effective to reconstruct images, while C-PPA has the farthest reconstruction speed. It verifies our theoretical assertion: The primal application of PPA to solve with (1.1) could be more efficient than its dual application at least for some special cases where the simplicity assumption (1.3) holds.

5.2 Comparison of Algorithm 1 with some other existing methods

We take the wavelet-based image processing model (5.1) as an example to compare Algorithm 1 ("C-PPA") with some efficient specialized algorithms in the literature, including the linearized Bregman iterative scheme ("L-Bregman") in [25], the primal-dual Bregman scheme ("PD-Bregman") in [26] and the gradient projection for sparse reconstruction ("GPSR") in [11].

The iterative scheme of L-Bregman in [25] for solving (5.1) is

$$\begin{cases} x^{k+1} = \operatorname{Arg\,min}\{\|x\|_1 + \frac{1}{2\alpha}\|x - \alpha v^k\|^2\}, \\ v^{k+1} = v^k + (BW)^T(b - BWx^{k+1}), \end{cases} \tag{5.3}$$

where $\alpha > 0$ is a penalty parameter. Also, for solving (5.1), the iterative scheme of PD-Bregman in [26] reads as

$$\begin{cases} x^{k+1} = \operatorname{Arg\,min}\{\theta(x) + \frac{1}{2\delta}\|x - [1 - \delta(BW)^T(BW)]x^k \\ \qquad\qquad - \delta(BW)^T(y^k + b)\|^2\}, \\ y^{k+1} = y^k - \gamma BW(x^{k+1} - b), \end{cases} \tag{5.4}$$

where $\delta > 0$ is a penalty parameter and it requires to be $\delta \in (0, 1)$ for the wavelet-based imaging problem (5.1) (recall $\|(BW)^T(BW)\| = 1$); and $\gamma \in (0, 2)$ is a relaxation factor. For GPSR in [11], note it mainly deals with the baisis pursuit denoising problem and solves the unconstrained l_1-l_2 model

$$\min\left\{\tau\|x\|_1 + \frac{1}{2}\|BWx - b\|^2\right\}, \tag{5.5}$$

where $\tau > 0$ is a trade-off parameter. For solving (5.5), GPSR first uses the fact $\|x\|_1 = u - v$ where $u_i = (x_i)_+$ and $v_i = (-x_i)_+$ with $(x_i)_+ := \max\{0, x_i\}$; and then reformulates it as

$$\min\left\{\tau \mathbf{1}^T(u-v) + \frac{1}{2}\left\|BW(u-v) - b\right\|^2 \mid u \geq 0, v \geq 0\right\}, \tag{5.6}$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$. We will compare the nonmonotone version of GPSR in [11], denoted by "GPSR-BB(n)", with the proposed method. To implement the GPSR-BB(n) approach, we use the original code downloaded from http://www.lx.it.pt/~mtf/GPSR. As mentioned in [11], theoretically the unconstrained model (5.5) with an appropriately small value of τ is equivalent to the constrained model (5.1) in the sense they have the same solution. But empirically it is hard to determine this value. For comparison purpose with other methods, we implement GPSR-BB(n) and test the cases where $\tau = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and $\tau = \{5 \times 10^{-5}, 5 \times 10^{-4}, 5 \times 10^{-3}, 5 \times 10^{-2}, 5 \times 10^{-1}\}$ for the inpainting and zooming scenarios of (5.5), respectively; and compare the SNR values of the reconstructed images. Then we select $\tau = 0.01$ for the inpainting scenario and $\tau = 5 \times 10^{-3}$ for the zooming scenario in the unconstrained model (5.5), since with these values the solutions of (5.5) are closest to the approximate solutions of (5.1) approached by C-PPA. We plot the evolution of SNR value with respect to iteration and computing time in Fig. 3. The evolution of C-PPA is also plotted for comparison.

We still test the 256×256 images of Peppers.png and Boat.png; and the inpainting and zooming operators are the same as Sect. 5.1, see Fig. 1 for the clean and corrupted images. The implementation details of the methods to be tested are described as below (For parameters of L-Bregman and PD-Bregman, we choose values as suggested in [25, 26]).

- For the image inpainting problem, we take $\tau = 0.01$ for GPSR-BB(n); $\alpha = 2$ and $v^0 = \mathbf{0}$ for L-Bregman; $\delta = 0.9$, $\gamma = 1.8$ and $(x^0, y^0) = (W^T(b), \mathbf{0})$ for PD-Bregman; $r = 0.6$, $s = 1.02/r$, $\gamma = 1.9$ and $(x^0, \lambda^0) = (W^T(b), \mathbf{0})$ for C-PPA.
- For the image zooming problem, we take $\tau = 0.005$ for GPSR-BB(n); $\alpha = 5$ and $v^0 = \mathbf{0}$ for L-Bregman; $\delta = 0.9$, $\gamma = 1.2$ and $(x^0, y^0) = (\mathbf{0}, \mathbf{0})$ for PD-Bregman; $r = 0.55$, $s = 1.02/r$, $\gamma = 1.2$ and $(x^0, \lambda^0) = (\mathbf{0}, \mathbf{0})$ for C-PPA.

In Fig. 4, we plot the evolution of SNR with respect to iteration and computing time for all the tested algorithms. These plots show that the proposed Algorithm 1 is efficient for solving the specific wavelet-based image processing problem (5.1). In Fig. 5 we display the reconstructed images by executing the tested algorithms for 150 seconds.

5.3 Test of Algorithm 2

In this subsection, we focus on an application of the model (4.1) and verify the efficiency of the proposed Algorithm 2.

We consider the total variation (TV) uniform noise removal model:

$$\min\left\{\left\||\nabla \mathbf{x}|\right\|_1 \mid \left\|H\mathbf{x} - \mathbf{x}^0\right\|_\infty \leq \sigma\right\}, \tag{5.7}$$

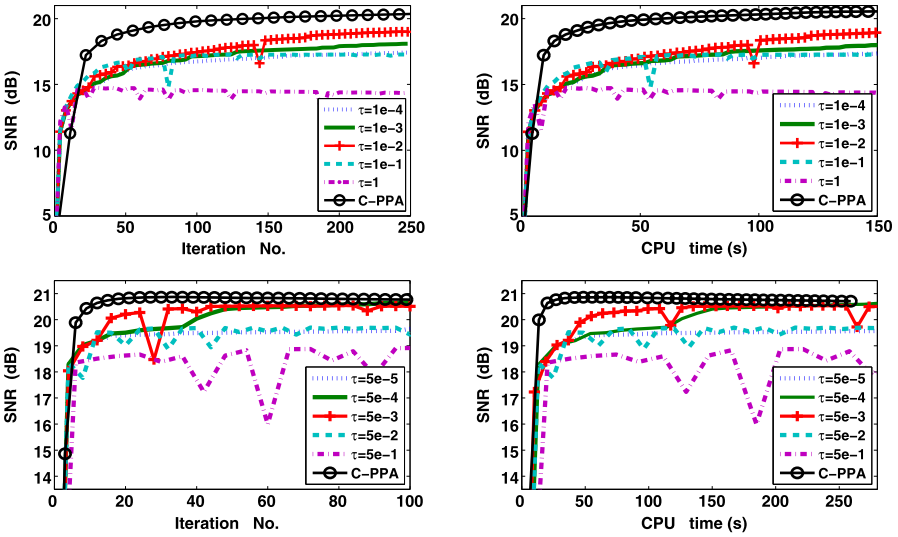

Fig. 3 Evolutions of SNR w.r.t. iterations and computing time for GPSR-BB(n) with different τ. *Top row*: for image inpainting. *Bottom row*: for image zooming

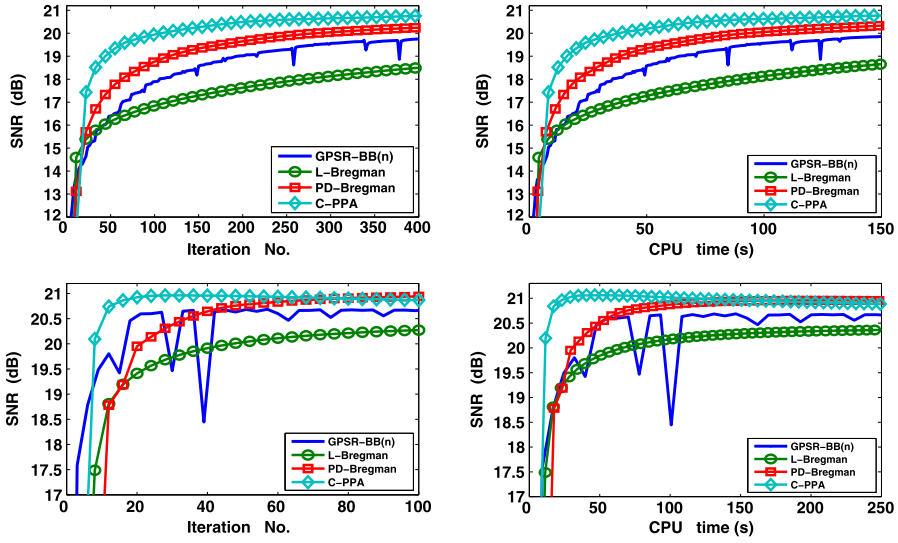

Fig. 4 Evolutions of SNR w.r.t. iterations and computing time. *Top row*: for image inpainting. *Bottom row*: for image zooming

where $\mathbf{x}^0 \in \Re^l$ is an observed image corrupted by a zero-mean uniform noise; $H \in \Re^{l\times l}$ is the matrix representation of a blurry operator as in Sect. 4, $\|\|\nabla \cdot \|\|_1$ is the TV norm (see e.g., [22]), σ is a parameter indicating the uniform noise level and $\|\mathbf{x}\|_\infty := \max_{1\le i\le l} |\mathbf{x}_i|$. For applications of (5.7), the reader is referred to, e.g., [24].

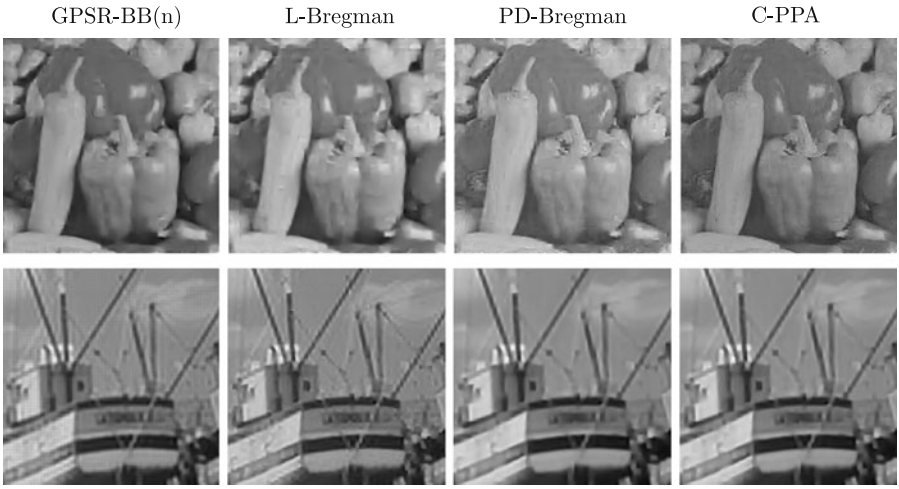

Fig. 5 Reconstructed images by tested methods. *Top row*: image inpainting. *Bottom row*: image zooming

We first show that the model (5.7) is a special case of (4.1); thus the proposed Algorithm 2 is applicable. By denoting

$$A := \begin{bmatrix} H \\ -H \end{bmatrix} \quad \text{and} \quad b := \begin{bmatrix} \mathbf{x}^0 - \sigma \mathbf{1} \\ -\mathbf{x}^0 - \sigma \mathbf{1} \end{bmatrix},$$

with $\mathbf{1} = (1, 1, \ldots, 1)^T \in \Re^l$, the model (5.7) amounts to

$$\min\left\{ \left\| |\nabla \mathbf{x}| \right\|_1 \mid A\mathbf{x} \geq b \right\}, \tag{5.8}$$

which is a special case of (4.1). Applying Algorithm 2 to (5.8), the main subproblem at each iteration is to solve the following minimization problem

$$\min\left\{ \left\| |\nabla \mathbf{x}| \right\|_1 + \frac{r}{2}\|\mathbf{x} - a\|^2 \right\}, \quad \forall a \in \Re^l, \tag{5.9}$$

which can be easily solved up to a high precision by existing methods. Thus, the simplicity assumption (1.3) holds for the application (5.7). In our numerical experiments, we use the method in [6] and execute 10 iterations.

For this TV uniform noise removal model, we compare Algorithm 2 (still abbreviated as "C-PPA") with the specialized algorithm in [1] (denoted by "C-SALSA") for the model (5.7). By introducing auxiliary variables $\mathbf{y}$ and $\mathbf{z}$, the model (5.7) can be reformulated as

$$\min\left\{ \left\| |\nabla \mathbf{y}| \right\|_1 \mid \mathbf{x} = \mathbf{y}, H\mathbf{x} = \mathbf{z}, \mathbf{z} \in \mathcal{Z} \right\}, \tag{5.10}$$

with $\mathcal{Z} := \{\mathbf{z} \in \Re^n \mid \|\mathbf{z} - \mathbf{x}^0\|_\infty \le \sigma\}$. Then, C-SALSA solves (5.10) by the alternating direction method of multipliers in [12] and obtains the following iterative scheme

$$\begin{cases} \mathbf{x}^{k+1} = \operatorname{Arg\,min}\{\mu_1 \|H\mathbf{x} - \mathbf{z}^k - \frac{\lambda_1^k}{\mu_1}\|^2 + \mu_2 \|\mathbf{x} - \mathbf{y}^k - \frac{\lambda_2^k}{\mu_2}\|^2\}, \\ \mathbf{y}^{k+1} = \operatorname{Arg\,min}\{\||\nabla \mathbf{y}|\|_1 + \frac{\mu_2}{2}\|\mathbf{x}^{k+1} - \mathbf{y} - \frac{\lambda_2^k}{\mu_2}\|^2\}, \\ \mathbf{z}^{k+1} = P_{\mathcal{Z}}[H\mathbf{x}^{k+1} - \frac{\lambda_1^k}{\mu_1}], \\ \lambda_1^{k+1} = \lambda_1^k - \mu_1(H\mathbf{x}^{k+1} - \mathbf{z}^{k+1}), \\ \lambda_2^{k+1} = \lambda_2^k - \mu_2(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}), \end{cases} \tag{5.11}$$

where λ_1 and λ_2 are Lagrangian multipliers; μ_1 and μ_2 are penalty parameters; $P_{\mathcal{Z}}$ is the projection onto $\mathcal{Z}$ which is defined as

$$\big(P_{\mathcal{Z}}[a]\big)_i = \mathbf{x}_i^0 + \min\left\{1, \frac{\sigma}{|a_i - x_i^0|}\right\}\big(a_i - \mathbf{x}_i^0\big), \quad \forall a \in \Re^l.$$

For the $\mathbf{y}$-subproblem in (5.11) whose closed-form solution is not available, we also use the method in [6] and execute 10 iterations.

We test the 256-by-256 images of Peppers.png and Boat.png. The clean images are degraded by either the Gaussian (`fspecial('gaussian',9,2.5)`) or the out-of-focus (`fspecial('disk',3)`) convolution. Then, the degraded images are further corrupted by the zero-mean uniform noise with $\sigma = 0.2$ or 0.5. To implement C-PPA, we take $r = 0.6$, $s = 1.02/r$ and $\gamma = 1.8$. For C-SALSA, as mentioned in [1], the parameters μ_1 and μ_2 should be tuned and our experiments show that $\mu_1 = 50$ and $\mu_2 = 10$ is a good choice for the application (5.7). The initial points for both algorithms are taken as zeros and the stopping criterion is set as

$$\text{Tol} = \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|}{\|\mathbf{x}^k\|} < 10^{-3}. \tag{5.12}$$

In Table 1, we report the numerical results when C-PPA and C-SALSA are applied to solve the TV uniform noise removal problem (5.7). Each set of "$\cdot/\cdot/\cdot$" represents the number of iterations, the computing time in seconds and the restored SNR when the stopping criterion (5.12) is reached. This table also shows that the C-PPA proposed in generic setting is even competitive to a specialized algorithm when a concrete application of (4.1) is considered. In Fig. 6, for Boat.png we show the corrupted images and the restored images by C-PPA.

6 Conclusions

This paper is a demonstration of how the classical proximal point algorithm (PPA) can be customized into an efficient algorithm for solving the convex minimization problem with linear constraints. The primal application of PPA with a customized proximal parameter, is shown to be more efficient than its dual application which is essentially the augmented Lagrangian method, another classical method in the area of convex optimization. We are interested in further investigating customizations of

Table 1 Numerical results on uniform noise removal

Blur	σ	Images	C-SALSA	C-PPA
Gaussian	0.2	Peppers	42/21.3/17.91	28/12.2/17.95
		Boat	40/20.2/17.98	24/11.6/18.02
	0.5	Peppers	52/28.4/16.87	26/14.9/16.95
		Boat	53/32.5/16.97	29/16.1/16.98
Out-of-focus	0.2	Peppers	35/19.4/18.77	22/11.3/18.97
		Boat	36/20.5/18.87	23/12.2/18.90
	0.5	Peppers	47/28.4/17.38	31/15.8/17.42
		Boat	43/25.5/17.53	25/14.9/17.55

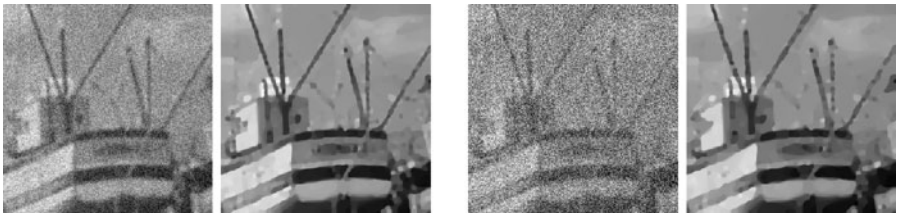

Fig. 6 Degraded image with $\sigma = 0.2$ and restored image by C-PPA, Degraded image with $\sigma = 0.5$ and restored image by C-PPA

the PPA for models other than (1.1) or some specific cases of (1.1) with separable structures, and designing highly customized algorithms which can exploit fully the structures and properties of the models under consideration.

References

1. Afonso, M., Bioucas-Dias, J., Figueiredo, M.: An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. IEEE Trans. Image Process. **20**, 681–695 (2010)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
3. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Method. Academic Press, New York (1982)
4. Blum, E., Oettli, W.: Mathematische Optimierung, Econometrics and Operations Research XX. Springer, Berlin (1975)
5. Burke, J.V., Qian, M.J.: A variable metric proximal point algorithm for monotone operators. SIAM J. Control Optim. **37**, 353–375 (1998)
6. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**, 89–97 (2004)
7. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. **43**, 129–159 (2001)
8. Daubechies, I., Defrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**, 1413–1457 (2004)
9. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program., Ser. A **55**, 293–318 (1992)
10. Fadili, J., Starck, J.L., Elad, M., Donoho, D.: Mcalab: reproducible research in signal and image decomposition and inpainting. Comput. Sci. Eng. **12**(1), 44–63 (2010)

11. Figueiredo, M.A.T., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Process. **1**(4), 586–597 (2007)
12. Glowinski, R., Marrocco, A.: Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires. RAIRO. Rech. Opér. **R2**, 41–76 (1975)
13. Gol'shtein, E.G., Tret'yakov, N.V.: Modified Lagrangian in convex programming and their generalizations. Math. Program. Stud. **10**, 86–97 (1979)
14. Hansen, P., Nagy, J., O'Leary, D.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)
15. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**, 303–320 (1969)
16. He, B.S., Yuan, X.M.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci. **5**, 119–149 (2012)
17. Malgouyres, F., Guichard, F.: Edge direction preserving image zooming: a mathematical and numerical analysis. SIAM J. Numer. Anal. **39**(1), 1–37 (2001)
18. Martinet, B.: Regularisation, d'inéquations variationelles par approximations succesives. Rev. Francaise Inf. Rech. Oper. **4**, 154–159 (1970)
19. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Berlin (1999)
20. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: Fletcher, R. (ed.) Optimization, pp. 283–298. Academic Press, New York (1969)
21. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **1**, 97–116 (1976)
22. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
23. Starck, J.L., Murtagh, F., Fadili, J.M.: Sparse Image and Signal Processing, Wavelets, Curvelets, Morphological Diversity. Cambridge University Press, Cambridge (2010)
24. Weiss, P., Aubert, G., Blanc-Féraud, L.: Some applications of l_∞-constraints in image processing. INRIA Research Report (2006)
25. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l^1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**, 143–168 (2008)
26. Zhang, X.Q., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. J. Sci. Comput. **46**, 20–46 (2011)