J. Appl. Numer. Optim. 2 (2020), No. 1, pp. 3-13
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.2.2020.1.02

SUPERIORIZATION OF BLOCK ACCELERATED CYCLIC SUBGRADIENT METHODS

MOKHTAR ABBASI[1,*], TOURAJ NIKAZAD[2]

[1]*Department of Mathematics, University of Qom, 37161-46611, Qom, Iran*
[2]*Department of Applied Mathematics, School of Mathematics,*
Iran University of Science and Technology, 16846-13114 Tehran, Iran

Abstract. In this paper, we introduce a block version and a perturbed block version of the accelerated cyclic subgradient projections method with constraints and give their convergence analyses. The performance of the algorithm is illustrated with a numerical example from the computed tomography and six standard nonlinear test problems. We compare our algorithms with the accelerated cyclic subgradient projections method. Our algorithms produce better results than accelerated cyclic subgradient projections method and have ability to reduce the value of an objective function. Furthermore, the perturbed block version is able to control semiconvergence phenomenon comparing two other methods.

Keywords. Convex feasibility problem; Cyclic subgradient projections method; Iterative methods; Superiorization; Total variation.

1. INTRODUCTION

Let $\{C_i\}_{i=1}^m \subseteq \mathbb{R}^n$ be closed convex sets and $C = \bigcap_{i=1}^m C_i \neq \emptyset$. The problem of finding a point in C is called convex feasibility problem (CFP), see, e.g., [1] for more general setting. A consistent linear system of equations $Ax = b$ is a simple example of a CFP. Many problems in mathematics and physical sciences can be modeled as a CFP. Some real world inverse problems, which deal with image reconstruction, as computerized tomography, radiation therapy treatment planning, signal processing, impedance tomography and electron microscopy are modeled as a CFP, see [1] for more applications. It is easy to check that there exists at least one nonnegative convex function $g_i : \mathbb{R}^n \to \mathbb{R}^+$ for each C_i such that $C_i = \{x | g_i(x) = 0\}$. For example, one may consider $g_i(x) = \min_{y \in C_i} \|x - y\|$, where $\|x\|$ denotes a norm, e.g., the Euclidean norm. Therefore, a CFP can be written as the following consistent nonlinear system of equations

$$\begin{cases} g_1(x) = 0, \\ \vdots \\ g_m(x) = 0, \end{cases} \tag{1.1}$$

[*] Corresponding author.

E-mail addresses: m.abbasi78@gmail.com (M. Abbasi), tnikazad@iust.ac.ir (T. Nikazad).

Received August 27, 2019; Accepted December 14, 2019.

where $\{g_i\}_{i=1}^m$ are nonnegative convex functions, see [5, p. 24 and p. 28]. Thus, showing a CFP as (1.1) demonstrates the importance of nonlinear system of equations as (1.1). In this paper, we introduce a (perturbed) block version of ACSP for solving (1.1), where $\{g_i\}_{i=1}^m$ are any nonnegative convex functions.

Several iterative methods for solving CFP have been studied. The method of successive orthogonal projections (SOP) [11], the cyclic subgradient projections method (CSP) [8] and parallel subgradient projections method (PSP) [22] are some of those iterative methods. See [1, 6] for a review of the methods. Recently, an acceleration technique (introduced by De Pierro [9]) was applied to the fully sequential CSP method which is named accelerated CSP (ACSP) method, see [15].

In this paper, we study the constraint block version of the ACSP with perturbation for solving (1.1) and give its convergence analysis. Indeed, we consider the convergence analysis of the following perturbation iteration

$$x^{k+1} = U(x^k + \bar{\beta}_k \bar{v}_k),\ k = 1, 2, \ldots \tag{1.2}$$

where $U = U_p \cdots U_2 U_1$ and $U_j = P_Q T_j$ for $j = 1, 2, \cdots, p$. Here, $Q \neq \emptyset$ is a closed convex subset of $\mathbb{R}^n$ and P_Q denotes the metric projection operator onto Q. We consider P_Q in our algorithm to select solutions of (1.1) which satisfy some kind of constraint, see Section 3 for a specific case. The iteration operators T_j are picked from a finite pool of operators which is based on the ACSP method. Furthermore, the sequences $\{\bar{v}_k\}_{k=1}^\infty \subset \mathbb{R}^n$ and $\{\bar{\beta}_k\}_{k=1}^\infty \subset \mathbb{R}$ are given. An example of $\bar{v}_k$, which is based on total variation norm, is used in Section 3.

The perturbation iterative methods are relevant and important for their possible use in the framework of the recently developed superiorization methodology for the constrained minimization problems. The general idea of the superiorization is to combine optimization while seeking feasibility. This approach was first suggested in [2], but its origin goes back to [3, 4] and references therein. The superiorization principle does not try to find just a feasible point and not at the quest for a constrained minimum point. Instead, it tries to seek a feasible point that is superior over other accessible feasible points, with respect to a given objective function, see, e.g., [7, 16, 17, 21, 23]. As a simple example, consider a consistent linear system of equations $Ax = b$. The minimization problem $\min_{x\in\mathbb{R}^n} \|b - Ax\|$ may have many solutions. One may be interested to obtain a solution which optimizes a certain function, say total variation used in imaging applications, see [20]. In these special applications, the minimum value of the total variation is reached at zero which is not a feasible point. Therefore, the superiorization method seeks a feasible point which is better over other reachable feasible points and it gives probably a small enough value for total variation, see [2, 20].

1.1. **Organization.**

In Subsection 1.2, we give a brief introduction regarding Φ-class operators (see [17]) which is useful for our convergence analysis. In Section 2, we consider a block version of (1.1) and build up a block version of ACSP method. Furthermore, we define the operators T_j, see (1.2), based on block ACSP. Theorem 2.1 shows that the operators T_j belong to Φ-class. At the end of this section, we give a convergence analysis of the iterative method (1.2), i.e., superiorized version of block ACSP. To show the performance of the superiorized block ACSP, we give some numerical results which are taken from the field of tomography and standard nonlinear test problems in Section 3.

1.2. **Preliminaries.** Most of the operators used for solving CFPs belong to a new introduced class of operators namely Φ-class [17]. A subclass of Φ-class has been defined in [19].

Definition 1.1. An operator $T : D \subseteq \mathbb{R}^n \longrightarrow D$ with $Fix T \neq \emptyset$ belongs to Φ-class when satisfies two following conditions.

(I) For every $z \in Fix T$ there exist a non-negative real function ϕ_z such taht

$$\|z - T(x)\| \leq \|z - x\| - \phi_z(x) \tag{1.3}$$

(II) If the sequence $\{x_k\}$ converges to α and $\lim_{k\to\infty} \phi_z(x_k) = 0$ then $\alpha \in Fix T$

Φ-class is closed under composition and convex combination of operators. This property is very important because it allows to use Φ-class operators in different algorithmic structure such as (block) sequential, (block) simultaneous iterative methods, string averaging and recently introduced pattern structures [18].

Proposition 1.1. [17, Proposition 13] *Let $T, T_1, T_2 \in \Phi$ and $Fix T_1 \cap Fix T_2 \neq \emptyset$. Define $T_\lambda(x) = x + \lambda(T(x) - x)$ where $0 < \lambda < 1$. then*

(1) *$Fix T_\lambda = Fix T$ and $T_\lambda \in \Phi$.*
(2) *$Fix(\lambda T_1 + (1-\lambda)T_2) = Fix T_1 \cap Fix T_2$ and $\lambda T_1 + (1-\lambda)T_2 \in \Phi$.*
(3) *$Fix(T_1 T_2) = Fix T_1 \cap Fix T_2$ and $T_1 T_2 \in \Phi$.*

Remark 1.1. The Proposition 1.1 can be easily extended for finite number of operators.

2. Main Results

In this section, we consider a block version of (1.1) and define the operators $\{T_j\}_{j=1}^p$, which make the operator U in (1.2). Here p denotes the number of blocks. Let the nonlinear system of equations (1.1) be partitioned into p blocks of equations as

$$B_j = \{i_1^j, i_2^j, \cdots, i_{\ell_j}^j\} \subseteq \{1, 2, \cdots, m\}$$

such that $\bigcup_{j=1}^p B_j = \{1, 2, \cdots, m\}$. Indeed, based on B_j, the selected block includes the equations $g_{i_1^j}(x) = 0,\ g_{i_2^j}(x) = 0, \cdots, g_{i_{\ell_j}^j}(x) = 0$. Based on each block B_j, the operator T_j is defined as follows

$$T_j(x) = x - \alpha_j(x)\beta_j(x)v_j(x)/\|v_j(x)\|^2. \tag{2.1}$$

We define below the functions $\alpha_j(x)$, $\beta_j(x)$ and $v_j(x)$ for $j = 1, 2, \cdots, p$.

Let $x^0 = x$ be the starting point of the following recursion

$$x^r = x^{r-1} - \frac{g_{i_r^j}(x^{r-1})}{\|s_{i_r^j}(x^{r-1})\|^2} s_{i_r^j}(x^{r-1}),\ r = 1, \cdots, \ell_j, \tag{2.2}$$

where $s_{i_r^j}(x)$ is a subgradient of $g_{i_r^j}$ at point x. We should remind here that a vector $t \in \mathbb{R}^n$ is called a subgradient of a convex function g at a point y if $\langle t, x - y\rangle \leq g(x) - g(y)$ for every $x \in \mathbb{R}^n$. Note that the statement $g_{i_r^j}(x) \neq 0$ easily leads to $s_{i_r^j}(x) \neq 0$. Therefore, we assume $s_{i_r^j}(x) \neq 0$, where $g_{i_r^j}(x) = 0$ throughout the paper. We consider the following definitions

$$\beta_j(x) = \sum_{r=1}^{\ell_j} g_{i_r^j}(x) \frac{g_{i_r^j}(x^{r-1})}{\|s_{i_r^j}(x^{r-1})\|^2},\ \tilde{\beta}_j(x) = \sum_{r=1}^{\ell_j} \frac{(g_{i_r^j}(x^{r-1}))^2}{\|s_{i_r^j}(x^{r-1})\|^2}, \tag{2.3}$$

and

$$v_j(x)=\sum_{r=1}^{\ell_j} s_{i_r^j}(x^{r-1})\frac{g_{i_r^j}(x^{r-1})}{\|s_{i_r^j}(x^{r-1})\|^2},\ \alpha_j(x)=\lambda\,\frac{\|v_j(x)\|^2+\tilde{\beta}_j(x)}{\beta_j(x)}, \tag{2.4}$$

where $\lambda\in(0,1)$. Considering (2.3) and (2.4) is based on [15, Algorithm 3] which is mimicked by [10, p. 312]. We next show the following properties of T_j defined in (2.1).

Theorem 2.1. *The operator $T_j\in\Phi$ and Fix $T_j=\{x|g_{i_r^j}(x)=0,\forall r=1,\cdots,\ell_j\}$.*

Proof. Let $x^*\in\{x|g_{i_r^j}(x)=0,\forall r=1,\cdots,\ell_j\}$. Using [15, (59)], one easily gets

$$\|x^*-T_j(x)\|\le\|x^*-x\|-\lambda(1-\lambda)\left(\frac{\|v_j(x)\|^2+\tilde{\beta}_j(x)}{\|v_j(x)\|}\right)^2. \tag{2.5}$$

We first show

$$\text{Fix } T_j=\{x|g_{i_r^j}(x)=0,\forall r=1,\cdots,\ell_j\}.$$

For $x\in$ Fix T_j, using (2.5), we get

$$\|v_j(x)\|^2+\tilde{\beta}_j(x)=0 \tag{2.6}$$

and consequently $\tilde{\beta}_j(x)=0$. Therefore, using (2.3) and (2.2), we get $g_{i_r^j}(x)=0$ for all $r=1,\cdots,\ell_j$. It gives Fix $T_j\subseteq\{x|g_{i_r^j}(x)=0,\forall r=1,\cdots,\ell_j\}$. Conversely, let

$$x\in\{x|g_{i_r^j}(x)=0,\forall r=1,\cdots,\ell_j\}.$$

Using (2.2), we get $x=x^0=\cdots=x^{\ell_j}$ and consequently (using (2.3) and (2.1)) $x\in$ Fix T_j.

We next show $T_j\in\Phi$. Using (2.5), and based on Definition 1.1, we can define

$$\phi(x)=\lambda(1-\lambda)\left((\|v_j(x)\|^2+\tilde{\beta}_j(x))/\|v_j(x)\|\right)^2.$$

Let $x_q\to\alpha$ and $\lim_{q\to\infty}\phi(x_q)=0$. We claim that $\alpha\in$ Fix T_j. Since $\lim_{q\to\infty}\phi(x_q)=0$ and using the definition of ϕ, we get that $\lim_{q\to\infty}\tilde{\beta}_j(x_q)=0$. Consequently,

$$\lim_{q\to\infty}\sum_{r=1}^{\ell_j}\frac{(g_{i_r^j}(x_q^{r-1}))^2}{\|s_{i_r^j}(x_q^{r-1})\|^2}=0. \tag{2.7}$$

Using $x_q\to\alpha$, we show that $x_q^{r-1}\to\alpha$ for all $r=1,\cdots,\ell_j$ by induction. For $r=1$, using (2.2), we get $x_q^{r-1}=x_q^0=x_q$ and consequently $x_q^{r-1}\to\alpha$. Let $x_q^{r_0-1}\to\alpha$ for an arbitrary $r_0>1$.

We now show that $x_q^{r_0}\to\alpha$. Based on (2.2), we get

$$\lim_{q\to\infty}x_q^{r_0}=\alpha-\lim_{q\to\infty}\frac{g_{i_{r_0}^j}(x_q^{r_0-1})}{\|s_{i_{r_0}^j}(x_q^{r_0-1})\|^2}s_{i_{r_0}^j}(x_q^{r_0-1}). \tag{2.8}$$

Using $x_q^{r_0-1}\to\alpha$ and [1, Corollary 7.9, p. 412], there exists K such that $\|s_{i_{r_0}^j}(x_q^{r_0-1})\|\le K$ for all $q=1,2,\cdots$. Therefore, we get

$$\lim_{q\to\infty}\frac{g_{i_{r_0}^j}(x_q^{r_0-1})}{\|s_{i_{r_0}^j}(x_q^{r_0-1})\|^2}\|s_{i_{r_0}^j}(x_q^{r_0-1})\|\le K\lim_{q\to\infty}\frac{g_{i_{r_0}^j}(x_q^{r_0-1})}{\|s_{i_{r_0}^j}(x_q^{r_0-1})\|^2}. \tag{2.9}$$

Using (2.7), (2.8) and (2.9), one gets $\lim_{q\to\infty} x_q^{r_0} = \alpha$. Therefore, $\lim_{q\to\infty} x_q^{r-1} = \alpha$ for all $r = 1,2,\cdots,\ell_j$. Again, based on $x_q^{r-1} \to \alpha$ and [1, Corollary 7.9, p. 412], there exists K_1 such that $\|s_{i_r^j}(x_q^{r-1})\| \leq K_1$ for all $r = 1,2,\cdots,\ell_j$ and $q = 1,2,\cdots$. Thus, equality (2.7) leads to $\lim_{q\to\infty} g_{i_r^j}(x_q^{r-1}) = 0$ for all $r = 1,\cdots,\ell_j$. Now, based on continuity of $g_{i_r^j}$ and the fact that $\lim_{q\to\infty} x_q^{r-1} = \alpha$ for all $r = 1,2,\cdots,\ell_j$, one gets $0 = \lim_{q\to\infty} g_{i_r^j}(x_q^{r-1}) = g_{i_r^j}(\lim_{q\to\infty} x_q^{r-1}) = g_{i_r^j}(\alpha)$, which leads to $\alpha \in \mathrm{Fix}\, T_j$. Thus, $T_j \in \Phi$. □

We next give the main theorem of this paper.

Theorem 2.2. *Let Q be a closed convex subset of $\mathbb{R}^n$ and $\{g_i\}_{i=1}^m$ be nonnegative and convex functions on $\mathbb{R}^n$ such that*

$$S = Q \cap \{x | g_i(x) = 0, i = 1,2,\cdots,m\} \neq \emptyset.$$

Let $\{\bar{\beta}_k\}_{k=0}^\infty$ be a summable sequence of positive real numbers ($\sum_{k=1}^\infty \bar{\beta}_k < \infty$) and $\{\bar{v}^k\}_{k=0}^\infty$ be a bounded sequence in $\mathbb{R}^n$. Then the sequence x^k generated by (1.2) and (2.1)-(2.4) with any starting point converges to a point in S.

Proof. Based on [17, section 2.1, Remark 7], Theorem 2.1 and Proposition 1.1 (note $Fix T_j \cap Fix Q \neq \emptyset$), we get that $P_Q \in \Phi, T_j \in \Phi$ and $U_j = P_Q T_j \in \Phi$. Using Proposition 1.1 and [17, Remark 14], we get $U \in \Phi$ and $Fix U = Q \bigcap_{j=1}^p Fix T_j = S$. Based on [17, Remark 23], the proof is completed. □

3. Numerical Results

In this section, we give some numerical results which are taken from the field of tomography (image reconstruction from projections) and the standard tests from [10] and [13].

3.1. **Image reconstruction test.** The image reconstruction problem is a well known example, which can be modeled using the Radon transform. A large, sparse and ill-conditioned linear system takes place after expanding the solution into a finite series of basis functions. We consider the solution of such systems, i.e., $Ax = b$ where $A \in \mathbb{R}^{s\times n}$ and $b \in \mathbb{R}^s$. Furthermore, let $Q \cap \{x|Ax = b\} \neq \emptyset$. Let the linear system $Ax = b$ be partitioned into m row blocks of equations, which may contain common equations but each equation should appear in at least one of the blocks. Denote by A_t and b^t the t-th block of A and b, respectively. For the case of a disjoint partitioning the structure is

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{pmatrix}, \quad b = \begin{pmatrix} b^1 \\ b^2 \\ \vdots \\ b^m \end{pmatrix}. \tag{3.1}$$

Let $g_i(x) = 0.5\|b^i - A_i x\|^2$ for $i = 1,\cdots,m$. It is clear that

$$x \in Q \cap \{x|Ax = b\} \neq \emptyset \Leftrightarrow x \in Q \text{ and } g_i(x) = 0 \text{ for } i = 1,\cdots,m.$$

In this section we compare ACSP, block version of ACSP (i.e., the iteration (1.2) with $\bar{\beta}_k = 0$) and superiorized block ACSP methods. Since in our application the linear system $Ax = b$ has infinitely many solutions, we steer the superiorized block ACSP method to find an approximate solution which reduces value of the total variation (TV). Indeed, we consider

$$\bar{v}_k = -\frac{\nabla TV(x^k)}{\|\nabla TV(x^k)\|}$$

in (1.2) where $TV(y)$ denotes the total variation of y. For a $G \times H$ image y whose pixel values are denoted by $y_{g,h}$ $(1 \leq g \leq G, 1 \leq h \leq H)$, $TV(y)$ is defined as follows

$$TV(y) = \sum_{g=1}^{G-1} \sum_{h=1}^{H-1} \sqrt{(y_{g+1,h} - y_{g,h})^2 + (y_{g,h+1} - y_{g,h})^2}. \tag{3.2}$$

We used the (MATLAB-based) AIR tools software [12] to produce a phantom, the related coefficient matrices and corresponding right-hand sides. All the numerical results are performed with Intel(R) Core(TM) $i3$-$3110M$ CPU 2.4 GHz, 4GB RAM, and the codes have been written in MATLAB $2018a$.

We consider a geometry and use $m = 20$ which is the number of equations in (1.1) and the number of row blocks in matrix A. Furthermore, we consider $A_i \in \mathbb{R}^{900\times n}$ for all $i = 1, 2, \cdots, m$, see (3.1). We here use 'Shepp-Logan' phantom which is collected from AIR tools software. The geometry discretizes the phantom into 200×200 pixels and takes 90 projections (evenly distributed between 0 and 179 degrees) with 200 rays per projection. The resulting projection matrix A has dimension 18000×40000. We use the following parameters for the numerical results. We consider $\lambda = 0.99$ (see (2.4)), $\bar{\beta}_k = (0.99)^k$ (see (1.2)), $p = 5$ (indeed, (1.1) be partitioned into p blocks), $\ell_j = 4$ (number of equations in each block) (see Section 2), $Q = \{x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n | 0 \leq x_i \leq 1\}$ and zero starting point for three versions of ACSP. Indeed, we have $m = p\, \ell_j$ where constant ℓ_j is used.

Figure 1 shows relative error histories in ACSP, block ACSP and superiorized block ACSP methods. As it is seen, the superiorization methodology gives smaller relative error than two other methods. More precisely, block ACSP and superiorized block ACSP methods provide almost 22.5% and 66.7% better results than ACSP method, respectively. In Figure 2 we show the original image and the reconstructed images by three methods. Table 1 shows the values of the total variations for different methods within 20 iterations and the total variation of the original image. Furthermore, we report the CPU times in seconds. As it is seen in Table 1, the CPU times of three methods are almost same whereas the value of total variation is notably reduced by superiorized block ACSP method.

Figure 3 presents the percentage of improvement of block ACSP and superiorized block ACSP methods with respect to ACSP method. We define the percentage of improvement of relative error and total variation for block ACSP and superiorized block ACSP methods with respect to ACSP as follows

$$\text{PI-RE of 'method'} = 100 \times |\text{RE of 'method'-RE of ACSP}| / (\text{RE of ACSP})$$

where RE and PI-RE denote the relative error and the percentage of improvement of the relative error for a 'method', respectively. Similarly, the percentage of improvement of total variation is defined as

$$\text{PI-TV of 'method'} = 100 \times |\text{RTV of 'method'-RTV of ACSP}| / (\text{RTV of ACSP})$$

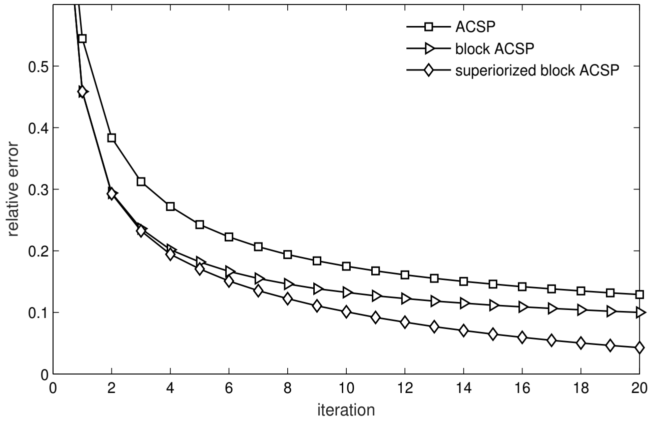

FIGURE 1. Relative error histories in ACSP, block ACSP and superiorized block ACSP methods, noiseless data.

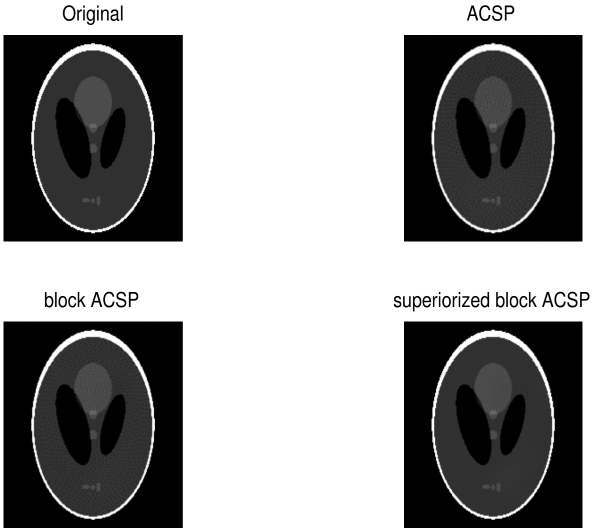

FIGURE 2. The reconstructed images by three methods and the original image, noiseless data.

TABLE 1. Second row: The value of the total variations for different methods within 20 iterations and the total variation of the original image. Third row: CPU time for different methods, noiseless data.

	original image	ACSP	block ACSP	superiorized block ACSP
TV	1143.7	1816.6	1784.1	1218.3
CPU time	–	4.69	4.79	4.8

where RTV denotes relative total variation and defines as

$$\text{RTV of `method'} = |\text{TV of `method'-TV of exact solution}| / (\text{TV of exact solution}).$$

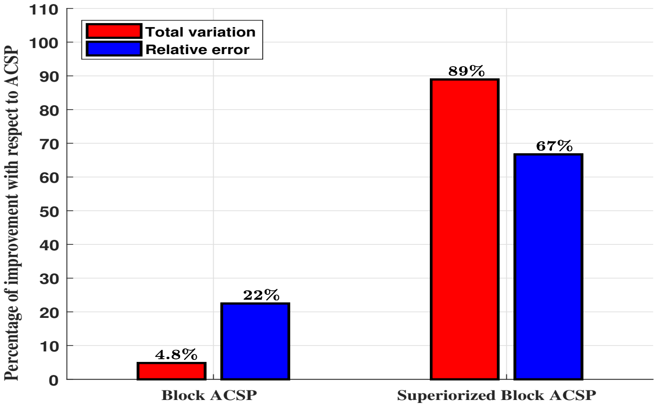

FIGURE 3. Percentage of improvement with respect to ACSP method, noiseless data.

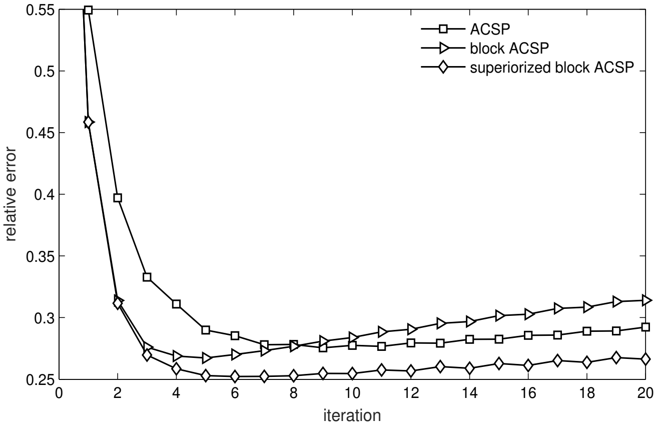

FIGURE 4. Relative error histories in ACSP, block ACSP and superiorized block ACSP methods with noisy data.

When applying an iterative method for image reconstruction problem with noisy data, one may view the iteration index as a regularizing parameter. Initially, the iteration vectors approach a regularized solution while continuing the iteration often leads to iteration vectors corrupted by noise, so-called semiconvergence, [14]. We now consider 5% white Gaussian noise for the right hand side vector b in (3.1). Relative error histories in ACSP, block ACSP and superiorized block ACSP methods with noisy data are presented in Figure 4. As it is seen, block ACSP method has the fastest semiconvergence and its superiorized version controls the phenomenon and gives the smallest relative error.

3.2. **Standard tests.** In this section, we compare ACSP and its block version using six standard test problems from [10, 13]. We use the notations $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$, $k = 1, 2, \cdots, m$ and $div(a,b) = \lfloor a/b \rfloor$ where $\lfloor a \rfloor$ denote the floor of a. Furthermore, *mod* denotes the modulo operation. Also, we define $g_i(x) = \max\{f_i(x), 0\}$ where f_i is defined as the following examples.

Example 3.1. The extended Powell singular function is defined as follows

$$f_k(x) = \begin{cases} x_i + 10x_{i+1}, & mod(k,4) = 1, \\ \sqrt{5}(x_{i+2} - x_{i+3}), & mod(k,4) = 2, \\ (x_{i+1} - 2x_{i+2})^2, & mod(k,4) = 3, \\ \sqrt{10}(x_i - x_{i+3})^2, & mod(k,4) = 0, \end{cases}$$

where $m = 2(n-2)$ and $i = 2div(k+3,4) - 1$.

Example 3.2. The chained Wood function.

$$f_k(x) = \begin{cases} 10(x_{i-1}^2 - x_i, & mod(k,6) = 1, \\ (x_{i-1} - 1), & mod(k,6) = 2, \\ \sqrt{90}(x_{i+1}^2 - x_{i+2})^2, & mod(k,4) = 3, \\ (x_{i+1} - 1)^2, & mod(k,6) = 4, \\ \sqrt{10}(-x_i - x_{i+2} + 2), & mod(k,6) = 5, \\ (-x_i - x_{i+2})/\sqrt{10}, & mod(k,6) = 0, \end{cases}$$

where $m = 3(n-2)$ and $i = 2\lfloor (k+5)/6 \rfloor$.

Example 3.3. The extended Rosenbrock function.

$$f_k(x) = \begin{cases} 10(x_i^2 - x_{i+1}), & mod(k,2) = 1, \\ x_i - 1, & mod(k,2) = 0, \end{cases}$$

where $m = 2(n-1)$ and $i = div(k+1,2)$.

Example 3.4. The generalized Broyden tridiagonal function.

$$f_k(x) = \begin{cases} (2x_i - 3)x_i + 2x_{i+1} - 1, & k = 1, \\ (2x_i - 3)x_i + x_{i-1} + 2x_{i+1} - 1, & k \neq 1, n, \\ (2x_i - 3)x_i + x_{i-1} - 1, & k = n, \end{cases}$$

where $m = n$ and $i = k$.

Example 3.5. Penalty function.

$$f_k(x) = \begin{cases} (x_k - 1)/\sqrt{10^5}, & k \leq n, \\ (\sum_{i=1}^n x_i^2) - 0.25, & k = n+1, \end{cases}$$

where $m = n+1$.

Example 3.6. Variably dimensioned function.

$$f_k(x) = \begin{cases} (x_k - 1), & k \leq n, \\ \sum_{i=1}^n i(x_i - 1), & k = n+1, \\ \left(\sum_{i=1}^n i(x_i - 1)^2\right)^2, & k = n+2, \end{cases}$$

where $m = n+2$.

TABLE 2. Number of iterations for ASCP and block ASCP methods with stopping criterion $g_i(x) < 10^{-4}$. The last right column denotes the percentage improvement of block ASCP with respect to ACSP.

Example	ASCP	block ACSP	percentage improvement
3.1	413	238	42.37%
3.2	1227	367	70.09%
3.3	456	492	−7.89%
3.4	5	6	−20%
3.5	4	4	0%
3.6	5	5	0%

In the numerical results of this section, we consider $m = 3000$, $p = 100$ and $\ell_j = 30$. Furthermore, we use (number of unknowns) $n = 1502, 1002, 1501, 3000, 2999$ and 2998 and the starting points $(3,-1,3,-1,\cdots,3,-1)$, $(3,-1,3,-1,\cdots,3,-1)$, $(-1.2,1,-1.2,\cdots,-1.2,1)$, $(-1,-1,\cdots,-1)$, $(1,2,3,\cdots,n)$ and $(1-1/n, 1-2/n, 1-3/n,\cdots,1-n/n)$ for Examples 3.1-3.6, respectively. The starting points are suggested by [10, 13]. Also we terminate the iteration where $g_i(x) < 10^{-4}$ for $i = 1,2,\cdots,m$. Table 2 reports the number of iterations for ACSP and its block version. In addition, we report the improvement percentage of block ACSP with respect to ACSP. Since the CPU times of ACSP and its block version were almost equal, we did not report them here. The mean value of iteration numbers of ACSP and its block version within six examples are 351.67 and 185.33, respectively. Therefore, the block ACSP reduces $47.3\%(= 100 \times (351.67 - 185.33)/351.67)$ of iteration numbers using the six test problems.

4. CONCLUSION

We introduce a block version of ACSP method and consider its perturbed version with constraints. We demonstrate the convergence analysis of the methods. The performance of the algorithms are illustrated with two series of numerical examples, which are taken from computed tomography and standard nonlinear test problems. The numerical results of tomography application show that superiorized block ACSP method, which is based on the idea of superiorization methodology, significantly reduces value of the total variation and has faster error reduction. Furthermore, it is able to control semiconvergence phenomenon. For the standard problems, block ACSP method reduces the number of iterations with almost the same computational time as ACSP method.

Acknowledgements
The authors are grateful to the reviewers for useful suggestions which improved the contents of this paper.

REFERENCES

[1] H.H. Bauschke, J.M. Borwein, On projection algorithms for solving convex feasibility problems, SIAM Rev. 38 (1996), 367–426.

[2] D. Butnariu, R. Davidi, G.T. Herman, I.G. Kazantsev, Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems, IEEE J. Sel. Topics in Signal Processing, 1 (2007), 540–547.

[3] D. Butnariu, S. Reich, A. J. Zaslavski, Convergence to fixed points of inexact orbits of Bregman-monotone and of nonexpansive operators in Banach spaces, Fixed Point Theory and its Applications, pp. 11–32, Yokohama Publishers, Yokohama, 2006

[4] D. Butnariu, S. Reich, A.J. Zaslavski, Stable convergence theorems for infinite products and powers of nonexpansive mappings, Numer. Funct. Anal. Optim. 29 (2008), 304–323.

[5] A. Cegielski, Iterative methods for fixed point problems in Hilbert spaces, vol. 2057, Springer, 2012.

[6] Y. Censor, Iterative methods for the convex feasibility problem, Ann. Discrete Math. 20 (1984), 83–91.

[7] Y. Censor, Can linear superiorization be useful for linear optimization problems? Inverse Probl. 33 (2017), 044006.

[8] Y. Censor, A. Lent, Cyclic subgradient projections, Math. Program. 24 (1982), 233–235.

[9] A.R. De Pierro, Methodos de projeçao para a resoluçao de sistemas gerais de equaçoes algébricas lineares, Doctotal Thesis, 1981.

[10] L.T. Dos Santos, A parallel subgradient projections method for the convex feasibility problem, J. Comput. Appl. Math. 18 (1987), 307–320.

[11] L.G. Gubin, B.T. Polyak, E.V. Raik, The method of projections for finding the common point of convex sets, USSR Comput. Math. Math. Phys. 7 (1967), 1–24.

[12] P.C. Hansen, J.S. Jørgensen, AIR Tools II: algebraic iterative reconstruction methods, improved implementation, Numerical Algo. 79 (2018), 107–137.

[13] J.J. Moré, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, Technical Report, Argonne National Lab., IL (USA), 1978.

[14] F. Natterer, The Mathematics of Computerized Tomography, SIAM, 2001.

[15] T. Nikazad, M. Abbasi, An acceleration scheme for cyclic subgradient projections method, Comput. Optim. Appl. 54 (2013), 77–91.

[16] T. Nikazad, M. Abbasi, Perturbation-resilient iterative methods with an infinite pool of mappings, SIAM J. Numer. Anal. 53 (2015), 390–404.

[17] T. Nikazad, M. Abbasi, A unifed treatment of some perturbed fxed point iterative methods with an infnite pool of operators, Inverse Probl. 33 (2017), 044002.

[18] T. Nikazad, M. Abbasi, Perturbed fixed point iterative methods based on pattern structure, Math. Methods Appl. Sci. 41 (2018), 5582–5592.

[19] T. Nikazad, M. Abbasi, M. Mirzapour, Convergence of string-averaging method for a class of operators, Optim. Methods Softw. 31 (2016), 1189–1208.

[20] T. Nikazad, R. Davidi, G.T. Herman, Accelerated perturbation-resilient block-iterative projection methods with application to image reconstruction, Inverse probl. 28 (2012), 035005.

[21] D. Reem, A. De Pierro, A new convergence analysis and perturbation resilience of some accelerated proximal forward–backward algorithms with errors, Inverse Probl. 33 (2017), 044001.

[22] L.T.D. Santos, Métodos de projeção do subgradiente para o problema de factibilidade convexa, Ph.D. thesis, 1985.

[23] M.V.W. Zibetti, C. Lin, G.T. Herman, Total variation superiorized conjugate gradient method for image reconstruction, Inverse Probl. 34 (2018), 034001.