Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 13, No. 4, pp. 665–681, October 2017

A RELAXATION OF THE PARAMETER IN THE FORWARD-BACKWARD SPLITTING METHOD*

Zehui Jia† and Xingju Cai‡

Abstract: The forward-backward splitting method finds a zero of the monotone operator $T = A + B$ via a forward step involving only B and a backward step involving the resolvent of A. Under the assumption that T has a zero point, and the parameter c_k involved in the algorithm lies in $(0, 2\sigma)$, where $\sigma > 0$ is the co-coercive modulus of B, then the method converges globally. In this paper, we prove that, with a simple relaxation step, the convergence can be guaranteed for the range of the parameter c_k being enlarged doubly, i.e., from $(0, 2\sigma)$ to $(0, 4\sigma)$, and the convergence can also be guaranteed in the case that the resolvent of A is inaccurately evaluated. With further assumptions on the mapping A or B, we prove the R-linear convergence rate for the sequence produced by the forward-backward splitting method with a relaxation step.

Key words: *forward-backward splitting algorithm, maximal monotone operators, relaxation, convergence*

Mathematics Subject Classification: *90C25, 90C33, 90C30*

1 Introduction

One of the classical methods for finding a zero of a monotone set-valued mapping T, i.e.,

$$0 \in T(x) \tag{1.1}$$

is the proximal point algorithm (PPA for short) proposed by Martinet [17] and Rockafellar [20]. PPA generates the iterates $\{x^k\}$ via the following recursion:

$$x^{k+1} = (I + c_k T)^{-1}(x^k),$$

where $c_k > 0$ are parameters and I denotes the identity mapping. The main difficulty in practical implementation of the method is that $I + c_k T$ may be hard to invert, depending on the nature of T. In many applications, the operator T is given by the sum of two other maximal monotone mappings: $T = A + B$, and $I + c_k A$ or $I + c_k B$ are easier to invert than $I + c_k T$. For this case, one can devise an algorithm that only uses $(I + c_k A)^{-1}$ and/or $(I + c_k B)^{-1}$ rather than $(I + c_k T)^{-1}$. Such an approach is called splitting method. There

*The research was supported by the PAPD of Jiangsu Higher Education Institutions(14KJB110015);NSFC (11401315);Jiangsu Provincial NSFC (BK20140914).

†The first author is supported by the Startup Foundation for Introducing Talent of NUIST, Grant No. 2017r059.

‡Corresponding author. The research of the second author was supported by NSFC (11401315), China Postdoctoral Science Foundation funded project (2015T80566) and Postdoctoral Science Foundation of Jiangsu Province (1401020B).

are many such procedures [9], which can be summarized into four principal classes: forward-backward [10, 19], double-backward [15, 19], Peaceman-Rachford [16] and Douglas-Rachford [16]. The forward-backward method is a useful method, and there are many applications such as [14, 13]. In this paper we pay our attention on the first class, the forward-backward splitting method.

Suppose that

$$T = A + B,$$

where A is a maximal monotone mapping and B is a single-valued function. Recall that the forward-backward method proposed by Lions and Mercier [16], generates the iterative sequence $\{x^k\}$ via the following recursion

$$x^{k+1} = (I + c_k A)^{-1}((I - c_k B)(x^k)). \tag{1.2}$$

Note that $0 \in T(x)$ if and only if

$$-cB(x) \in cA(x) \Longleftrightarrow (I - cB)(x) \in (I + cA)(x) \Longleftrightarrow x = (I + cA)^{-1}((I - cB)(x)).$$

Thus (1.2) is simply a fixed-point iteration of the latter equation with a varying multiplier c_k at each iteration. There are many other studies on this method [1, 3, 4, 5, 6, 8, 12, 21, 11, 7]. Based on the classical forward-backward splitting method for the subgradients in Hilbert space, Bredies [3] proposed a general method which involves the iterative solution of simpler subproblems to find a minimizer of the sum of a smooth and a non-smooth convex function, and then applied it to the minimization of Tikhonov-functionals associated with linear inverse problems and semi-norm penalization in Banach spaces; Combettes [5] proposed a variable metric forward-backward splitting algorithm and proved its convergence in real Hilbert spaces, and then applied this framework to derive primal-dual splitting algorithms for solving various classes of monotone inclusions in duality; He [12] studied the relationship between the forward-backward splitting method and the extra-gradient method for monotone variational inequalities, and gave some modifications for the two methods; Tseng [21] made a modification of the forward-backward method which entails an additional forward step and a projection step at each iteration; David and Yin [7] show that the forward-backward operator is averaged and under the regular step size, the convergence rate is $o(\frac{1}{t})$.

Forward-backward method can cover some useful methods, such as gradient projection method, in the case of $A = N_C$ and $B = F$, where C is a nonempty closed convex set and F is the gradient of a differentiable convex function. This method was analyzed by Mercier [18] and Gabay [10] and they showed that if B is co-coercive, then the iterates x^k converge to a solution if $0 < m \le c_k \le M < 2\sigma$, where $\sigma > 0$ is the co-coercive modulus of B and m, M are two constants which make c_k be bounded away from 0 and 2σ. Note that the parameter c_k plays the role of step-size, and from the numerical point of view, a larger c_k may usually lead to higher efficiency than a smaller one. Thus, in this paper, we propose a relaxation version of forward-backward algorithm and show that with a simple relaxation step, the convergence range for the parameter c_k involved in the forward-backward algorithm can be enlarged doubly, i.e., from $0 < m \le c_k \le M < 2\sigma$ to $0 < m \le c_k \le M < 4\sigma$. The iteration of the forward-backward splitting method with a relaxation step is as follows:

$$\bar{x}^{k+1} = (I + c_k A)^{-1}((I - c_k B)(x^k)), \tag{1.3}$$

$$x^{k+1} = x^k - \alpha_k(x^k - \bar{x}^{k+1}), \tag{1.4}$$

where α_k is a relaxation factor satisfied $0 < m' \le \alpha_k \le 2(1-\frac{c_k}{M}) < 2(1-\frac{c_k}{4\sigma})$, here α_k be bounded away from 0 and $2(1-\frac{c_k}{4\sigma})$. Note that comparing with the step (1.3), the computational load of (1.4) is ignorable, i.e., we achieve the goal with nearly no cost.

This paper is organized as follows. In Section 2, we summarize some necessary preliminaries. In Section 3, we present the main results, which fully demonstrates the convergence range for the parameter involved in the algorithm can be enlarged doubly and under some further conditions on the mapping of T, i.e., either A or B is strongly monotone, we give a R-linear convergence rate. In order to illustrate the efficiency introduced by the new parameter strategy of enlarging its range and the relaxation step. In Section 4, we report some numerical results on two examples, where we compare the new method with the modified forward-backward method in [8]. In Section 5, we give a short conclusion.

2 Preliminaries

In this section, we summarize some basic concepts and necessary results that will be useful for further discussion.

Let $\Phi : \mathcal{R}^n \to [-\infty, \infty]$ be a mapping, the effective domain of Φ is defined as

$$\mathrm{dom}\Phi = \{x \in \mathcal{R}^n \mid \Phi(x) < +\infty\},$$

and the graph of Φ is defined as

$$\mathrm{gph}\Phi := \{(x,u) \mid x \in \mathrm{dom}\Phi, u \in \Phi(x)\}.$$

A mapping $\Phi : \mathcal{R}^n \to \mathcal{R}^n$ is monotone if

$$(x-y)^T(u-v) \ge 0, \quad \forall x,y \in \mathrm{dom}\Phi, \quad \forall u \in \Phi(x), \quad \forall v \in \Phi(y).$$

A mapping $\Phi : \mathcal{R}^n \to \mathcal{R}^n$ is strongly monotone if there exists a constant $c > 0$ such that

$$(x-y)^T(u-v) \ge c\|x-y\|^2, \quad \forall x,y \in \mathrm{dom}\Phi, \quad \forall u \in \Phi(x), \quad \forall v \in \Phi(y).$$

A monotone mapping Φ is maximal monotone if the graph of Φ is not properly contained in the graph of any other monotone operator Ψ.

If we identify the map Φ with its graph, maximal monotone maps are monotone maps that are maximal with respect to set inclusion.

Definition 2.1. Let T be a mapping from a set $\Omega \subset \mathcal{R}^n \to \mathcal{R}^n$, then

(1) T is said to be co-coercive on Ω with modulus $\sigma > 0$, if

$$(u-v)^T(T(u)-T(v)) \ge \sigma\|T(u)-T(v)\|^2, \quad \forall u,v \in \Omega.$$

(2) T is said to be nonexpansive on Ω, if

$$\|T(x)-T(y)\| \le \|x-y\|, \quad \forall x,y \in \Omega.$$

(3) T is said to be firmly nonexpansive on Ω, if

$$\|T(x)-T(y)\|^2 \le (x-y)^T(T(x)-T(y)), \quad \forall x,y \in \Omega.$$

Obviously, if T is firmly nonexpansive, then T is nonexpansive. The converse is not true in general.

The following lemma will be used to prove the main theorem in this paper.

Lemma 2.2 ([2]). *Suppose that the mapping $A : \mathcal{R}^n \to \mathcal{R}^n$ is maximal monotone, then the resolvent $J_{cA} = (I+cA)^{-1}$ is firmly nonexpansive and* $\mathrm{dom}(I+cA)^{-1} = \mathcal{R}^n$ *, where c is a positive scalar.*

$\boxed{3}$ Convergence

This section presents the main result of this paper, which fully demonstrates the convergence range for the parameter involved in the algorithm can be enlarged doubly even with approximate evaluation of the resolvent. Furthermore, by adding some conditions on the mapping, we can get the linear rate of convergence.

Lemma 3.1. *Suppose that the mapping A is maximal monotone and B is co-coercive with modulus σ. If*

$$\bar{x} = J_{cA}(I - cB)(x),$$

with

$$0 < m \le c \le M < 4\sigma,$$

then $-(x - \bar{x})$ is a descent direction of the merit function $\frac{1}{2}\|x - x^*\|^2$ at x, where $x^* \in \Omega^*$ and $\Omega^* := \{x \mid 0 \in A(x) + B(x)\}$ is the solution set of (1.1) with $T = A + B$.

Proof. Since the mapping A is maximal monotone, it follows from Lemma 2.2 that the resolvent of A is firmly nonexpansive, i.e.,

$$(J_{cA}(u) - J_{cA}(v))^T(u - v) \ge \|J_{cA}(u) - J_{cA}(v)\|^2, \quad \forall u, v \in \mathcal{R}^n.$$

Equivalently,

$$((u - J_{cA}(u)) - (v - J_{cA}(v)))^T(J_{cA}(u) - J_{cA}(v)) \ge 0, \quad \forall u, v \in \mathcal{R}^n. \tag{3.1}$$

Setting $u := x - cB(x)$ and $v := x^* - cB(x^*)$ in (3.1), we have

$$\begin{aligned} &((x - cB(x) - J_{cA}(I - cB)(x)) - (x^* - cB(x^*) - J_{cA}(x^* - cB(x^*))))^T \\ &\qquad\qquad\qquad (J_{cA}(I - cB)(x) - J_{cA}(x^* - cB(x^*))) \ge 0. \end{aligned}$$

Using the identities

$$x^* = J_{cA}(x^* - cB(x^*)) \quad \text{and} \quad \bar{x} = J_{cA}(I - cB)(x),$$

we obtain

$$((x - \bar{x}) - c(B(x) - B(x^*)))^T(\bar{x} - x^*) \ge 0.$$

Then,

$$(\bar{x} - x^*)^T(x - \bar{x}) \ge c(x - x^*)^T(B(x) - B(x^*)) - c(x - \bar{x})^T(B(x) - B(x^*)). \tag{3.2}$$

Rearranging terms, from (3.2) we have

$$\begin{aligned} &(x - x^*)^T(x - \bar{x}) \\ &= \|x - \bar{x}\|^2 + (\bar{x} - x^*)^T(x - \bar{x}) \\ &\ge \|x - \bar{x}\|^2 + c(x - x^*)^T(B(x) - B(x^*)) - c(x - \bar{x})^T(B(x) - B(x^*)) \\ &\ge \|x - \bar{x}\|^2 + c\sigma\|B(x) - B(x^*)\|^2 - c(x - \bar{x})^T(B(x) - B(x^*)), \end{aligned} \tag{3.3}$$

where the second inequality is due to the co-coercivity of B. Using Young's inequality

$$ab \le \frac{a^p}{p} + \frac{b^q}{q}, \tag{3.4}$$

where $p > 1, \frac{1}{p} + \frac{1}{q} = 1$, and by setting $a := \sigma^{\frac{1}{2}}(B(x) - B(x^*))$, $b := \frac{1}{2}\sigma^{-\frac{1}{2}}(x - \bar{x})$ and $p = q = 2$, we have:

$$(B(x) - B(x^*))^T(x - \bar{x}) \leq \sigma\|B(x) - B(x^*)\|^2 + \frac{1}{4\sigma}\|x - \bar{x}\|^2. \tag{3.5}$$

Thus, it follows from (3.3) and (3.5) that

$$(x - x^*)^T(x - \bar{x}) \geq \left(1 - \frac{c}{4\sigma}\right)\|x - \bar{x}\|^2 > 0, \tag{3.6}$$

whenever x is not a solution, which means that $-(x - \bar{x})$ is a descent direction for $\frac{1}{2}\|x - x^*\|^2$ at x. This completes the proof. □

The next theorem gives the convergence result of the iteration (1.3) and (1.4). We allow the possibility of inaccurate evaluation of the resolvent of A (but assume exact evaluation of B).

Theorem 3.2. *Let T be a set-valued map with a nonempty, closed convex domain and at least one zero. Assume that $T = A + B$, where A is maximal monotone and B is a single-valued function. Let $\{x^k\}$ be a sequence such that*

(a) $\|\tilde{x}^{k+1} - J_{c_k A}(I - c_k B)(x^k)\| \leq \varepsilon_k$,

(b) $x^{k+1} = x^k - \alpha_k(x^k - \tilde{x}^{k+1})$; *and suppose that* $\sum_{k=0}^{\infty} \varepsilon_k < \infty$. *If B is co-coercive with modulus σ, and*

$$0 < m \leq c_k \leq M < 4\sigma, \quad 0 < m' \leq \alpha_k \leq 2(1 - \frac{c_k}{M}) < 2(1 - \frac{c_k}{4\sigma}), \quad \forall k.$$

Then $\{x^k\}$ converges to a point x^* such that $0 \in T(x^*)$.

Proof. We prove first for the case of exact evaluation of the resolvent of A, that is $\varepsilon_k = 0$ for all k. In this case x^{k+1} is given by (1.3) and (1.4). Then we have

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &= \|x^k - \alpha_k(x^k - \bar{x}^{k+1}) - x^*\|^2 \\
&= \|x^k - x^*\|^2 + \alpha_k^2\|x^k - \bar{x}^{k+1}\|^2 - 2\alpha_k(x^k - x^*)^T(x^k - \bar{x}^{k+1}) \\
&\leq \|x^k - x^*\|^2 + \alpha_k^2\|x^k - \bar{x}^{k+1}\|^2 - 2\alpha_k(1 - \frac{c_k}{4\sigma})\|x^k - \bar{x}^{k+1}\|^2 \\
&= \|x^k - x^*\|^2 + \left[\alpha_k^2 - 2\alpha_k\left(1 - \frac{c_k}{4\sigma}\right)\right]\|x^k - \bar{x}^{k+1}\|^2,
\end{aligned} \tag{3.7}$$

where the inequality follows from (3.6). Since the relaxation factor α_k satisfied the inequality $0 < m' \leq \alpha_k \leq 2(1 - \frac{c_k}{M}) < 2(1 - \frac{c_k}{4\sigma})$, we have that $\alpha_k^2 - 2\alpha_k(1 - \frac{c_k}{4\sigma}) < 0$ for all k, and $\alpha_k^2 - 2\alpha_k(1 - \frac{c_k}{4\sigma})$ is bounded away from zero. This shows that $\{x^k\}$ is bounded, and

$$\lim_{k\to\infty} \|x^k - \bar{x}^{k+1}\| = 0. \tag{3.8}$$

Using $\bar{x}^{k+1} = J_{c_k A}((I - c_k B)(x^k))$ by (1.3), we get

$$\lim_{k\to\infty} \|x^k - J_{c_k A}((I - c_k B)(x^k))\| = 0. \tag{3.9}$$

Since $\{x^k\}$ is bounded, it has at least one cluster point. Supposing that x^∞ is one of the cluster points, there is an index set κ, such that $\{x^k : k \in \kappa\}$ converges to x^∞. Since $\{c_k\}$

is also bounded, it has at least a cluster point $\bar{c} > 0$. Without loss of generality, we assume that $\{c_k : k \in \kappa\}$ converges to $\bar{c}$. Taking limit in (3.9), we have $x^\infty = J_{\bar{c}A}((I - \bar{c}B)(x^\infty))$, which means that x^∞ is a solution of the inclusion $0 \in T(x)$. It remains to show that x^∞ is the only limit point of the sequence x^k. By (3.7), with $x^* = x^\infty$, we see that the entire sequence $\{\|x^k - x^\infty\|\}$ is nonincreasing and bounded, therefore it is convergent. Since the subsequence $\{x^k : k \in \kappa\}$ converges to x^∞, $\{\|x^k - x^\infty\| : k \in \kappa\} \to 0$. It follows that $\{\|x^k - x^\infty\|\} \to 0$, i.e., the entire sequence $\{x^k\}$ converges to $x^\infty (= x^*)$.

We next consider the case of inaccurate evaluation of the resolvent. For convenience, we denote $\bar{x}^{k+1} = J_{c_k A}((I - c_k B)(x^k))$ and $z^{k+1} = x^k - \alpha_k(x^k - \bar{x}^{k+1})$. From (3.7), we get

$$\|z^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 + \left[\alpha_k^2 - 2\alpha_k \left(1 - \frac{c_k}{4\sigma}\right)\right] \|x^k - \bar{x}^{k+1}\|^2, \tag{3.10}$$

and it follows that

$$\|z^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2, \qquad \forall \alpha_k \in [m', 2\left(1 - \frac{c_k}{M}\right)] \subset (0, 2\left(1 - \frac{c_k}{4\sigma}\right)). \tag{3.11}$$

Since

$$\|x^{k+1} - z^{k+1}\| = \alpha_k \|\tilde{x}^{k+1} - \bar{x}^{k+1}\|,$$

and $\|\tilde{x}^{k+1} - \bar{x}^{k+1}\| \leq \varepsilon_k$, we get

$$\|x^{k+1} - z^{k+1}\| \leq \alpha_k \varepsilon_k. \tag{3.12}$$

Using the following inequality that for any two vectors on the same space

$$\|a + b\|^2 \leq (1 + \tau)\|a\|^2 + \left(1 + \frac{1}{\tau}\right) \|b\|^2, \quad \forall \tau > 0,$$

we have (setting $a := z^{k+1} - x^*$, $b := x^{k+1} - x^*$, and $\tau := \varepsilon_k$)

$$\begin{aligned}
&\|x^{k+1} - x^*\|^2 \\
&= \|z^{k+1} - x^* + (x^{k+1} - z^{k+1})\|^2 \\
&\leq (1 + \varepsilon_k)\|z^{k+1} - x^*\|^2 + (1 + \frac{1}{\varepsilon_k})\|x^{k+1} - z^{k+1}\|^2 \\
&\leq (1 + \varepsilon_k)(\|x^k - x^*\|^2 + [\alpha_k^2 - 2\alpha_k(1 - \frac{c_k}{4\sigma})]\|x^k - \bar{x}^{k+1}\|^2) + 4(1 + \varepsilon_k)\varepsilon_k \\
&\leq \left[\prod_{i=0}^{k}(1 + \varepsilon_i)\right] \|x^0 - x^*\|^2 + \sum_{t=0}^{k} \left[\prod_{i=t+1}^{k} (1 + \varepsilon_i)\right] \left[\alpha_t^2 - 2\alpha_t(1 - \frac{c_t}{4\sigma})\right] \|x^t - \bar{x}^{t+1}\|^2 \\
&\quad + 4\sum_{t=0}^{k} \left[\prod_{i=t+1}^{k} (1 + \varepsilon_i)\right] (1 + \varepsilon_t)\varepsilon_t,
\end{aligned} \tag{3.13}$$

where the second inequality follows from (3.10) and (3.12) and the fact that $\alpha_k < 2$ for all k. From the assumption that $\sum_{k=0}^{\infty} \varepsilon_k < \infty$, we have $\prod_{i=0}^{k}(1 + \varepsilon_i) < \infty$ and

$$\sum_{t=0}^{k} \left[\prod_{i=t+1}^{k} (1 + \varepsilon_i)\right] (1 + \varepsilon_t)\varepsilon_t \leq \prod_{i=0}^{k}(1 + \varepsilon_i) \sum_{t=0}^{k} \varepsilon_t < \infty,$$

which together with the fact that $\alpha_k^2 - 2\alpha_k(1 - \frac{c_k}{4\sigma}) < 0$ and (3.13), imply that

$$\|x^{k+1} - x^*\|^2 < +\infty,$$

and as a consequence, $\{x^k\}$ is bounded. The coefficient $\alpha_t^2 - 2\alpha_t(1 - \frac{c_t}{4\sigma}) < 0$, which is due to $0 < m' \le \alpha_t \le 2(1 - \frac{c_k}{M}) < 2(1 - \frac{c_k}{4\sigma})$. For $1 < \prod_{i=0}^k (1 + \varepsilon_i)$, rearranging terms in (3.13) yields

$$\begin{aligned}
&-\sum_{t=0}^{k} \left[\alpha_t^2 - 2\alpha_t \left(1 - \frac{c_t}{4\sigma}\right)\right] \|x^t - \bar{x}^{t+1}\|^2 \\
&\le \left[\prod_{i=0}^{k}(1+\varepsilon_i)\right] \|x^0 - x^*\|^2 + 4\sum_{t=0}^{k}\left[\prod_{i=t+1}^{k}(1+\varepsilon_i)\right](1+\varepsilon_t)\varepsilon_t - \|x^{k+1} - x^*\|^2 \\
&\le \left[\prod_{i=0}^{k}(1+\varepsilon_i)\right] \|x^0 - x^*\|^2 + 4\sum_{t=0}^{k}\left[\prod_{i=t+1}^{k}(1+\varepsilon_i)\right](1+\varepsilon_t)\varepsilon_t,
\end{aligned}$$

which implies that

$$-\sum_{t=0}^{k} \left[\alpha_t^2 - 2\alpha_t \left(1 - \frac{c_t}{4\sigma}\right)\right] \|x^t - \bar{x}^{t+1}\|^2 < +\infty.$$

Then we have $\lim_{k\to\infty} \|x^k - \bar{x}^{k+1}\| = 0$, a similar result as (3.8). The rest of the proof is similar to the case of exactly evaluating the resolvent, and we omit it here. □

If B is strongly monotone, we can establish a R-linear convergence rate for the sequence $\{x^k\}$ produced by (1.3) and (1.4). This result asserts that $\{x^k\}$ converges to x^* at least R-linearly, that is, there exists positive constants c and η with $\eta < 1$ such that $\|x^{k+1} - x^*\| \le c\eta^k$ for all k sufficiently large.

Corollary 3.3. *Assume the same setting of Theorem 3.2. If in addition B is strongly monotone, the sequence $\{x^k\}$ converges at least R-linearly.*

Proof. Let x^* be the unique solution of $0 \in T(x)$ and let $\eta_B > 0$ be the modular of strong monotonicity of B. Using the same notation as in the proof of Theorem 3.2, we have

$$(B(x^k) - B(x^*))^T(x^k - x^*) \ge \eta_B \|x^k - x^*\|^2,$$

so that

$$\|B(x^k) - B(x^*)\| \ge \eta_B \|x^k - x^*\|.$$

Using Young's inequality (3.4), and by setting $p = q = 2$,

$$a := \left(\frac{M}{4}\right)^{\frac{1}{2}} \left(B(x^k) - B(x^*)\right),\ b := \frac{1}{2}\left(\frac{M}{4}\right)^{-\frac{1}{2}} (x^k - \bar{x}^{k+1}),$$

we have

$$(B(x^k) - B(x^*))^T(x^k - \bar{x}^{k+1}) \le \frac{M}{4}\|B(x^k) - B(x^*)\|^2 + \frac{1}{M}\|x^k - \bar{x}^{k+1}\|^2. \tag{3.14}$$

Together with (3.3), we have

$$\begin{aligned}
(x^k - x^*)^T(x^k - \bar{x}^{k+1}) &\ge \left(1 - \frac{c_k}{M}\right)\|x^k - \bar{x}^{k+1}\|^2 + c_k\left(\sigma - \frac{M}{4}\right)\|B(x^k) - B(x^*)\|^2 \\
&\ge \left(1 - \frac{c_k}{M}\right)\|x^k - \bar{x}^{k+1}\|^2 + c_k\eta_B^2\left(\sigma - \frac{M}{4}\right)\|x^k - x^*\|^2. \quad (3.15)
\end{aligned}$$

The second inequality is due to strongly monotonicity of B. Referring to (3.7), we deduce

$$\begin{aligned}
\|x^{k+1}-x^*\|^2 &= \|x^k-\alpha_k(x^k-\bar{x}^{k+1})-x^*\|^2 \\
&= \|x^k-x^*\|^2+\alpha_k^2\|x^k-\bar{x}^{k+1}\|^2-2\alpha_k(x^k-x^*)^T(x^k-\bar{x}^{k+1}) \\
&\le \|x^k-x^*\|^2+\alpha_k^2\|x^k-\bar{x}^{k+1}\|^2 \\
&\quad -2\alpha_k\left[\left(1-\frac{c_k}{M}\right)\|x^k-\bar{x}^{k+1}\|^2+c_k\eta_B^2\left(\sigma-\frac{M}{4}\right)\|x^k-x^*\|^2\right] \\
&= \|x^k-x^*\|^2-2\alpha_k c_k\eta_B^2\left(\sigma-\frac{M}{4}\right)\|x^k-x^*\|^2 \\
&\quad +\left[\alpha_k^2-2\alpha_k\left(1-\frac{c_k}{M}\right)\right]\|x^k-\bar{x}^{k+1}\|^2. \qquad (3.16)
\end{aligned}$$

Due to

$$0<m\le c_k\le M<4\sigma,\quad 0<m'\le\alpha_k\le 2(1-\frac{c_k}{M})<2\Big(1-\frac{c_k}{4\sigma}\Big),\qquad \forall k,$$

we have

$$\begin{aligned}
\|x^{k+1}-x^*\|^2 &\le \|x^k-x^*\|^2-2\alpha_k c_k\eta_B^2\left(\sigma-\frac{M}{4}\right)\|x^k-x^*\|^2 \\
&\le \left[1-2m'm\eta_B^2\left(\sigma-\frac{M}{4}\right)\right]\|x^k-x^*\|^2. \qquad (3.17)
\end{aligned}$$

Since η_B is positive, this complete the proof. □

If A is strongly monotone, we also can establish a R-linear convergence rate for the sequence $\{x^k\}$ produced by (1.3) and (1.4).

Corollary 3.4. *Assume the same setting of Theorem 3.2. If in addition A is strongly monotone, the sequence $\{x^k\}$ converges at least R-linearly.*

Proof. Let x^* be the unique solution of $0\in T(x)$ and let η_A be the modular of strong monotonicity of A. Note that η_A is positive. Using the same notation as in the proof of Theorem 3.2, we have

$$(A(x^{k+1})-A(x^*))^T(x^{k+1}-x^*)\ge\eta_A\|x^{k+1}-x^*\|^2,$$

so that

$$\|A(x^{k+1})-A(x^*)\|\ge\eta_A\|x^{k+1}-x^*\|,$$

$$(J_{c_kA}(u)-J_{c_kA}(v))^T(u-v)\ge(1+c_k\eta_A)\|J_{c_kA}(u)-J_{c_kA}(v)\|^2,\quad \forall u,v\in\mathcal{R}^n.$$

Equivalently, for all $u,v\in\mathcal{R}^n$

$$((u-J_{c_kA}(u))-(v-J_{c_kA}(v)))^T(J_{c_kA}(u)-J_{c_kA}(v))\ge c_k\eta_A\|J_{c_kA}(u)-J_{c_kA}(v)\|^2. \quad (3.18)$$

Setting $u:=x^k-c_kB(x^k)$ and $v:=x^*-c_kB(x^*)$ in (3.18), and using the identities

$$x^*=J_{c_kA}(x^*-c_kB(x^*))\quad\text{and}\quad \bar{x}=J_{c_kA}(I-c_kB)(x^k),$$

we have

$$((x^k-\bar{x}^{k+1})-c_k(B(x^k)-B(x^*)))^T(\bar{x}^{k+1}-x^*)\ge c_k\eta_A\|\bar{x}^{k+1}-x^*\|^2.$$

Then

$$\begin{aligned}(\bar{x}^{k+1}-x^*)^T(x^k-\bar{x}^{k+1}) \geq\ & c_k(x^k-x^*)^T(B(x^k)-B(x^*)) \\ & - c_k(x^k-\bar{x}^{k+1})^T(B(x^k)-B(x^*)) \\ & + c_k\eta_A\|\bar{x}^{k+1}-x^*\|^2.\end{aligned} \tag{3.19}$$

Similar to the proof of Corollary 3.3, we have

$$(x^k-x^*)^T(x^k-\bar{x}^{k+1}) \geq \left(1-\frac{c_k}{4\sigma}\right)\|x^k-\bar{x}^{k+1}\|^2 + c_k\eta_A\|\bar{x}^{k+1}-x^*\|^2 > 0. \tag{3.20}$$

Referring to (3.7), we deduce

$$\begin{aligned}\|x^{k+1}-x^*\|^2 &= \|x^k-\alpha_k(x^k-\bar{x}^{k+1})-x^*\|^2 \\ &= \|x^k-x^*\|^2+\alpha_k^2\|x^k-\bar{x}^{k+1}\|^2-2\alpha_k(x^k-x^*)^T(x^k-\bar{x}^{k+1}) \\ &\leq \|x^k-x^*\|^2+\left[\alpha_k^2-2\alpha_k\left(1-\frac{c_k}{4\sigma}\right)\right]\|x^k-\bar{x}^{k+1}\|^2 \\ &\quad -2\alpha_k c_k\eta_A\|\bar{x}^{k+1}-x^*\|^2.\end{aligned} \tag{3.21}$$

Set $A=\min\{A_1,A_2\}$, where $A_1=-\alpha_k^2+2\alpha_k\left(1-\frac{c_k}{4\sigma}\right)$, $A_2=2\alpha_k c_k\eta_A$. So we obtain

$$\begin{aligned}\|x^{k+1}-x^*\|^2 &\leq \|x^k-x^*\|^2-A_1\|x^k-\bar{x}^{k+1}\|^2-A_2\|\bar{x}^{k+1}-x^*\|^2 \\ &\leq \left(1-\frac{A}{2}\right)\|x^k-x^*\|^2.\end{aligned} \tag{3.22}$$

With the inequality $0<A<1$, we complete the proof. □

4 Nuemerical experiments

In this section, we test the performance of the relaxed forward-backword method, denoting it RFB for short. We consider the following two cases to be tested, complementarity problems and generalized Nash equilibrium problems. In addition, we compare RFB with the algorithm in [8], which is denoted MFB for short (we will describe the algorithm simply later), and the original forward-backward method (FB for short). All codes were written in Matlab 2008b and run on an HP personal computer with Pentium Dual-Core processor 2.66 GHz and 2 GB memory. To demonstrate the efficiency of RFB, we report the numerical results in terms of the number of iterations ("Iter.") and computing time in seconds ("Time").

Algorithm 1 Modified forward-backward method (MFB)

1: Initialization. Choose an initial stepsize c_0, $\epsilon > 0$, $\gamma \in (0,2)$ and $x^0 \in \mathcal{X}$.

2: Find the smallest nonnegative integer l_k such that $c_k = c_0\beta^{l_k}$, and get $\bar{x}^k$ satisfying

$$\bar{x}^k = (I + c_k A)^{-1}((I - c_k B)(x^k)),$$

and

$$c_k\|B(x^k) - B(x^k)\| \le \|x^k - x^k\|, \quad c_k < \sigma,$$

where σ is the co-coercive modulus of B.

3: Update the new iteration point through the following iterative scheme

$$x^{k+1} = P_{\mathcal{X}}[x^k - \gamma\beta_k g_k],$$

where $g_k = x^k - \bar{x}^k - \alpha_k(A(x^k) - A(\bar{x}^k))$ and $\beta_k = \frac{(x^k - \bar{x}^k)^T g_k}{\|g_k\|^2}$. If the stop criterion does not attained, set $k := k + 1$ and go to 2.

4.1 Complementarity problem

We now consider a special case of the problem (1.1), that is the complementarity problem, which is to find a vector $x \in \mathcal{R}^n$ such that

$$x \ge 0, \qquad F(x) \ge 0 \qquad \text{and} \qquad x^T F(x) = 0. \tag{4.1}$$

Below, we describe the details of the underlying mapping $F(x)$.

The underlying mapping F consists of a linear part and a nonlinear part. Concretely,

$$F(x) = Mx + D(x) + q,$$

where $Mx + q$ is the linear part and $D(x)$ is the nonlinear part. We form the linear part as described in [8], i.e., $M = A^T A + B$, where A is an $n \times n$ matrix whose entries are randomly generated in the interval $(-5, 5)$ and the skew-symmetric matrix B is generated in the same way; the vector q is generated randomly in the interval $(-500, 0)$. For the nonlinear part $D(x)$, each component is $D_j(x) = a_j \cdot \arctan(x_j)$ $(j = 1, 2, \cdots, n)$, where a_j is a uniformly random variable in $(0, 1)$.

In the experiment, we set the stopping criterion as $\|x - P_+(x - F(x))\| \le 10^{-7}$. The elements of the initial point x_0 is randomly distributed on $(0, 10)$. We choose the parameter c_k to be $0.3 \cdot alf$ in our algorithm, and $0.15 \cdot alf$ in the FB method, where alf is a constant we deduce from the same armijo rule, which ensure the parameter c_k to satisfy the inequalities $0 < m \le c_k \le M < 4\sigma$ and $0 < m \le c_k \le M < 2\sigma$respectively, and we choose the relaxation factor $\alpha_k = 1.35$.

In Table 1, we give the number of iterations and the CPU time of the three algorithms, when they reach the same stopping rule. And we illustrate and compare the three algorithms in Figure 1 by plotting the residual as function of the number of iterations in different dimensions, which shows the convergent ratio between the three algorithms as the increase of number of iterations.

From Table1 and Figure 1, it is easy to see that for all cases, our algorithm(RFB) needs less number of iterations and less CPU time to achieve the same stopping rule than the other two algorithms, which needs only half iterations of the FB algorithm. The parameter c_k in RFB is larger than that in FB method. Just like our analysis mentioned before, the parameter c_k plays the role of step-size, and from the numerical point of view, a larger c_k

Table 1: The comparison of the iteration number and time among the three algorithms

Dimension	Method	Iter.	Time
n=100	MFB	357	0.3940
	FB	515	0.3798
	RFB	269	0.2325
n=300	MFB	537	1.6029
	FB	632	1.4347
	RFB	384	0.9184
n=400	MFB	444	1.9861
	FB	634	2.0903
	RFB	318	1.1195
n=600	MFB	488	6.8107
	FB	542	5.7401
	RFB	344	3.8860

may usually lead to higher efficiency than a smaller one. So the simple relaxation step at each iteration is worthy.

$\boxed{4.2}$ Transportation equilibrium problem

In this example, we consider a transportation equilibrium problem. There are N nodes, and L directed links in the network $[N, L]$. Assuming that a, b stand for the links, p, q stand for paths, w means the pair of OD, P_w means the set of all paths which link w. Δ and Γ mean the correlation matrix of path-link and path-OD respectively, which are determined by the network.

We use x_p to denote the flow on the path p, and f_a the flow on the link a, so we have the equality:

$$f = \Delta^T x.$$

We denote the demand of each pair of OD as d_w, and it meets the flow equilibrium:

$$d_w = \sum_{p \in P_w} x_p.$$

So we have the equality:

$$d = \Gamma^T x.$$

Assuming that $t(f) = t_a, a \in l$ is the cost function on the link. We set θ_p as the cost function on path p. Given the cost function t, we can deduce the cost function θ_p on path p:

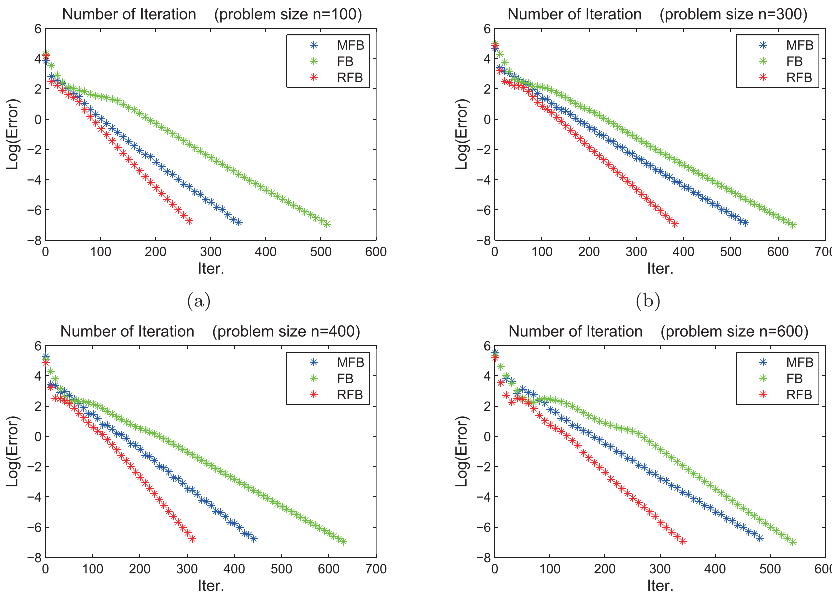
$$\theta = \Delta t(f) \ \ and \ \ \theta(x) = \Delta t(\Delta^T x).$$

For each OD pair w, we have a disutility function $\lambda_w(d)$. The utility function and the disutility function are all the function of x. The transportation equilibrium is to find the flow $x^* \in S$, which satisfys:

$$(y - x^*)^T F(x^*) \geq 0, \qquad \forall y \in K. \tag{4.2}$$

Since

$$F_p(x) = \theta_p x - \lambda_w(d(x)), \quad \forall w, p \in P_w,$$



(a) (b)

(c) (d)

Figure 1: The number of iterations for problem (4.1), for dimension of the problem: (a) $n = 100$; (b) $n = 300$; (c) $n = 400$; (d) $n = 600$.

so we have

$$F(x) = \Delta t(\Delta^T x) - \Gamma\lambda(\Gamma^T x).$$

In our experiment, we specify this transportation equilibrium problem by a given network $[N, L]$ and demand quantities of each pair of OD. In the network $[N, L]$, we assume there are 25 nodes, 37 directed links and 6 pairs of OD, we depict the network in Figure 2, there are 55 paths, so the dimension of variable x is 55, and the dimension of Δ and Γ are 55×37 and 55×6 respectively. The cost of the users are given in Table 2 and the disutility function is given as follows:

$$\lambda_w(d) = -m_w d_w + q_w,$$

where the parameters m_w and q_w are given in Table 3.

In this specific example, we consider fixed demand quantities:

$$\{x \in \mathcal{R}^n \mid B^T x = d,\ x \geq 0\},$$

where the demand quantity $d = (10, 25, 10, 60, 100, 20)$ is a given vector.

In this experiment, we also choose the MFB algorithm in [8] and the FB method as the comparing algorithms, with different error bounds eps.. All the elements of the initial point x_0 is one, We choose the parameter c_k to be $0.12 \cdot alf$ in our algorithm, and $0.06 \cdot alf$ in the FB method, where alf is a constant deduced from the same armijo rule. And we choose the relaxation factor $\alpha_k = 1.7$.

In Table 4, we give the iterative numbers and the CPU time of the three algorithms, with the same accuracy. And we illustrate and compare the three algorithms in Figure 3 by plotting the residual of problem (4.2) as function of the number of iterations in different

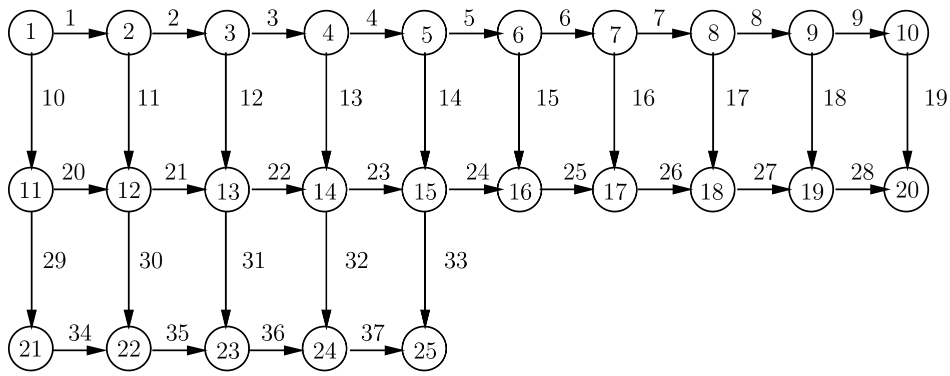


Figure 2: The transportation network

Table 2: The cost function on each link in the transportation network

$t_1(f) = 5 \cdot 10^{-6} f_1^4 + 0.5f_1 + 0.2f_2 + 50$	$t_2(f) = 3 \cdot 10^{-6} f_2^4 + 0.4f_2 + 0.4f_1 + 20$
$t_3(f) = 5 \cdot 10^{-6} f_3^4 + 0.3f_3 + 0.1f_4 + 35$	$t_4(f) = 3 \cdot 10^{-6} f_4^4 + 0.6f_4 + 0.3f_5 + 40$
$t_5(f) = 6 \cdot 10^{-6} f_5^4 + 0.6f_5 + 0.4f_6 + 60$	$t_6(f) = 0.7f_6 + 0.3f_7 + 50$
$t_7(f) = 8 \cdot 10^{-6} f_7^4 + 0.8f_7 + 0.2f_8 + 40$	$t_8(f) = 4 \cdot 10^{-6} f_8^4 + 0.5f_8 + 0.2f_9 + 65$
$t_9(f) = 10^{-6} f_9^4 + 0.6f_9 + 0.2f_{10} + 70$	$t_{10}(f) = 0.4f_{10} + 0.1f_{12} + 80$
$t_{11}(f) = 7 \cdot 10^{-6} f_{11}^4 + 0.7f_{11} + 0.4f_{12} + 65$	$t_{12}(f) = 0.8f_{12} + 0.2f_{13} + 70$
$t_{13}(f) = 10^{-6} f_{13}^4 + 0.7f_{13} + 0.3f_{18} + 60$	$t_{14}(f) = 0.8f_{14} + 0.3f_{15} + 50$
$t_{15}(f) = 3 \cdot 10^{-6} f_{15}^4 + 0.9f_{15} + 0.2f_{14} + 20$	$t_{16}(f) = 0.8f_{16} + 0.5f_{12} + 30$
$t_{17}(f) = 3 \cdot 10^{-6} f_{17}^4 + 0.7f_{17} + 0.2f_{15} + 45$	$t_{18}(f) = 0.5f_{18} + 0.1f_{16} + 30$
$t_{19}(f) = 0.8f_{19} + 0.3f_{17} + 60$	$t_{20}(f) = 3 \cdot 10^{-6} f_{20}^4 + 0.6f_{20} + 0.1f_{21} + 30$
$t_{21}(f) = 4 \cdot 10^{-6} f_{21}^4 + 0.4f_{21} + 0.1f_{22} + 40$	$t_{22}(f) = 2 \cdot 10^{-6} f_{22}^4 + 0.6f_{22} + 0.1f_{23} + 50$
$t_{23}(f) = 3 \cdot 10^{-6} f_{23}^4 + 0.9f_{23} + 0.2f_{24} + 35$	$t_{24}(f) = 2 \cdot 10^{-6} f_{24}^4 + 0.8f_{24} + 0.1f_{25} + 40$
$t_{25}(f) = 3 \cdot 10^{-6} f_{25}^4 + 0.9f_{25} + 0.3f_{26} + 45$	$t_{26}(f) = 6 \cdot 10^{-6} f_{26}^4 + 0.7f_{26} + 0.8f_{27} + 30$
$t_{27}(f) = 3 \cdot 10^{-6} f_{27}^4 + 0.8f_{27} + 0.3f_{28} + 50$	$t_{28}(f) = 3 \cdot 10^{-6} f_{28}^4 + 0.7f_{28} + 65$
$t_{29}(f) = 3 \cdot 10^{-6} f_{29}^4 + 0.3f_{29} + 0.1f_{30} + 45$	$t_{30}(f) = 4 \cdot 10^{-6} f_{30}^4 + 0.7f_{30} + 0.2f_{31} + 60$
$t_{31}(f) = 3 \cdot 10^{-6} f_{31}^4 + 0.8f_{31} + 0.1f_{32} + 75$	$t_{32}(f) = 6 \cdot 10^{-6} f_{32}^4 + 0.8f_{32} + 0.3f_{31} + 60$
$t_{33}(f) = 4 \cdot 10^{-6} f_{33}^4 + 0.9f_{33} + 0.2f_{31} + 75$	$t_{34}(f) = 6 \cdot 10^{-6} f_{34}^4 + 0.7f_{34} + 0.3f_{30} + 55$
$t_{35}(f) = 3 \cdot 10^{-6} f_{35}^4 + 0.8f_{35} + 0.3f_{32} + 60$	$t_{36}(f) = 2 \cdot 10^{-6} f_{36}^4 + 0.8f_{36} + 0.4f_{31} + 75$
$t_{37}(f) = 6 \cdot 10^{-6} f_{37}^4 + 0.5f_{37} + 0.1f_{36} + 35$	

Table 3: The parameters in the disutility function

the pair of OD	(1,20)	(1,25)	(2,25)	(3,25)	(1,24)	(11,25)
m_w	1	6	10	5	7	9
q_w	100	80	200	600	800	700

Table 4: The comparison of the iteration number and time among the three algorithms

Error bound	Method	Iter.	Time
eps.=10^{-1}	MFB	1219	1.0646
	FB	2365	1.3118
	RFB	997	0.5321
eps.=10^{-2}	MFB	1527	1.3030
	FB	3143	1.6810
	RFB	1502	0.8031
eps.=10^{-3}	MFB	1903	1.6063
	FB	4194	2.1519
	RFB	2355	1.1968
eps.=10^{-4}	MFB	2493	2.1366
	FB	5324	2.5963
	RFB	3176	1.9812

error bounds, which shows the convergent ratio between the three algorithms as the increase of iteration number.

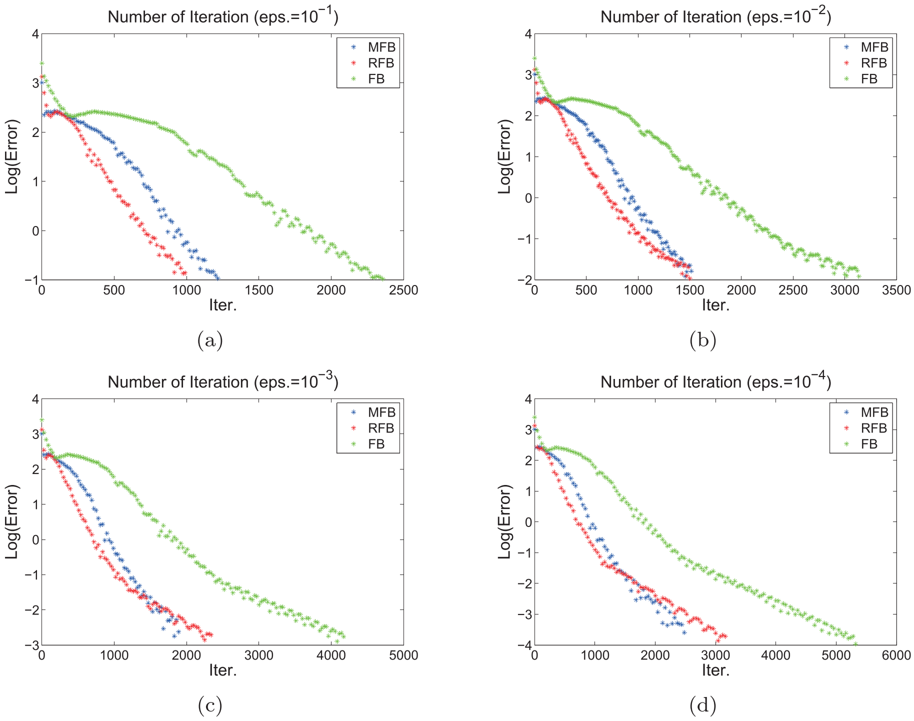


Figure 3: The iterative number of problem (4.2) among the three algorithms, in case of the error bound of the problem is setting: (a) eps.=10^{-1}; (b) eps.=10^{-2}; (c) eps.=10^{-3}; (d) eps.=10^{-4}.

From Table 4 and Figure 3, it is easy to see that for all cases, our algorithm(RFB) needs less number of iterations to achieve the same accuracy than the FB algorithm, actually we only needs almost half iterations of the FB algorithm. And the CPU time of RFB is less than the other two algorithms. The parameter c_k of RFB is larger than the c_k in the FB method, which plays the role of step-size. As our analysis, a larger c_k may usually lead to higher efficiency than a smaller one, which is verified by this experiment again. So the simple relaxation step at each iteration is worthy. We also find that, when the value of eps. becomes small, the iteration number of our algorithm is becoming larger than the iteration number of MFB, although still smaller than the FB method. While we find that we use less CPU time than MFB, which means the calculative cost of each iteration in RFB is smaller than MFB.

5 Conclusion

In this paper, we successfully enlarged the convergence range of the famous forward-backward splitting method from $0 < m \le c_k \le M < 2\sigma$ to $0 < m \le c_k \le M < 4\sigma$, with a slight cost of combining the iterative point generated by the classical forward-backward splitting method and the current iterative point. Under the same conditions as those in the classical forward-backward splitting method, we prove the global convergence of the new algorithm. We also consider the possibility of inaccurate evaluation of the resolvent of A, which is important from the numerical point of view. Under further conditions on the mappings A and/or B, we prove the R-linear rate of convergence of the new algorithm. The numerical results show that the new algorithm is competing to the same class of algorithms.

References

[1] H. Attouch, L.M. Briceño-Arias and P.L. Combettes, A parallel splitting method for coupled monotone inclusions, *SIAM J. Control Optim.* 48 (2010) 3246–3270.

[2] H.H. Bauschke and P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[3] K. Bredies, A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space, *Inverse Problems* 25 (2008): 015005.

[4] P.L. Combettes, Solving monotone inclusions via compositions of nonexpansive averaged, *Optimization* 53 (2004) 475–504.

[5] P.L. Combettes and B.C. Vũ, Variable metric forward-backward splitting with applications to monotone inclusions in duality, *Optimization* 63 (2014) 1289–1318.

[6] H.G. Chen, Forward-Backward Splitting Techniques: Theory and Applications, Ph.D. Thesis, Department of Applied Mathematics, University of Washington, Seattle, WA, 1994.

[7] D. David and W.T. Yin, Convergence rate analysis of several splitting schemes, in *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer International Publishing, 2016, pp. 115–163.

[8] X. Ding and D. R. Han, A modification of the forward-backward splitting method for maximal monotone mappings, *Numerical Algebra Control and Optimization* 3 (2013) 295–307.

[9] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag, New York, 2003.

[10] D. Gabay, Applications of the method of multipliers to variational inequalities, in *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*, M. Fortin R. Glowinski (eds.), Elsevier, Amsterdam, 1983, pp. 299–331.

[11] E.G. Golshtein and N.V. Tretyakov, Modified Lagrangians in convex programming and their generalizations, in *Point-to-Set Maps and Mathematical Programming*, P. Huard and A. Auslender (eds.), Springer Berlin Heidelberg, North Holland, 1979, pp. 86–97.

[12] B.S. He, X.M. Yuan and J. Zhang, Comparison of two kinds of prediction-correction methods for monotone variational inequalities, *Comput. Optim. Appl.* 27 (2004) 247–267.

[13] K.N. Kumar, B. Sivaneasan, P.H. Cheah, P.L. So and D.Z.W. Wang, V2G capacity estimation using dynamic ev scheduling, *IEEE Trans. on Smart Grid* 5 (2014) 1051–1060.

[14] H.X. Liu and D.Z.W. Wang, Global optimization method for network design problem with stochastic user equilibrium, *Transport. Res. B-meth.* 72 (2015) 20–39.

[15] P.L. Lions, Une méthode itérative de resolution d'une inéquation variationnelle, *Israel J. Math.* 31 (1978) 204–208.

[16] P.L. Lions and B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.* 16 (1979) 964–979.

[17] B. Martinet, Régularisation d'inéquations variationelles par approximations successives, *Rev. Francaise Inf. Rech, Oper.* 4 (1970) 154–158.

[18] B. Mercier, *Inéquations Variationnelles de la Mécanique*, No. 80.01 in Publications Mathématiques d'Orsay, France Orsay, 1980.

[19] G.B. Passty, Ergodic convergence to a zero of the sum of monotone operators in Hilbert space, *J. Math. Anal. Appl.* 72 (1979) 383–390.

[20] R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optim.* 14 (1976) 877–898.

[21] P. Tseng, A modified forward-backward splitting method for maximal monotone mappings, *SIAM J. Control Optim.* 38 (2000) 431–446.

Manuscript received 29 April 2015
revised 23 December 2015
accepted for publication 22 February 2016

Zehui Jia
School of Mathematics and Statistics
Nanjing University of Information Science and Technology
Nanjing 210044, China

School of Mathematical Sciences
Key Laboratory for NSLSCS of Jiangsu Province Nanjing Normal University
Nanjing 210023, China
E-mail address: jiazehui90@126.com

Xingju Cai
School of Mathematical Sciences, Key Laboratory for NSLSCS of Jiangsu Province
Nanjing Normal University, Nanjing 210023, China
E-mail address: caixingju@njnu.edu.cn