Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A new fixed-time stability of neural network to solve split convex feasibility problems

Jinlan Zheng[1,2], Rulan Gan[3*], Xingxing Ju[4] and Xiaoqing Ou[5]

*Correspondence:
ganrl0411@163.com
[3]College of Computer and Information Science, Southwest University, 400715 Chongqing, China
Full list of author information is available at the end of the article

Abstract

In this paper, we propose a novel neural network that achieves stability within the fixed time (NFxNN) based on projection to solve the split convex feasibility problems. Under the bounded linear regularity assumption, the NFxNN admits a solution of the split convex feasibility problem. We introduce the relationships between NFxNN and the corresponding neural networks. Additionally, we also prove the fixed-time stability of the NFxNN. The convergence time of the NFxNN is independent of the initial states. The effectiveness and superiority of the NFxNN are also demonstrated by numerical experiments compared with the other methods.

Keywords: Neural network; Finite-time stability; Fixed-time stability; Split convex feasibility problems; Bounded linear regularity

1 Introduction

Since the split convex feasibility problem (SCFP) was introduced in [1] in 1994, the SCFP was widely used in retrieval problems [1], image restoration [2], signal processing [3], IMRT [4], systems biology [5]. Many numerical algorithms have been proposed to solve the split convex feasibility problem (see, e.g., [5–11]). The SCFP is as follows:

$$\text{find } x \in C, y \in Q \quad \text{such that} \quad Ax = y, \tag{1}$$

where $C \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^m$ are two non-empty, closed and convex sets, and $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear mapping. Among them, some classical and semi-alternating CQ algorithms were proposed to solve the split convex feasible problems in [3, 7]. In fact, the CQ algorithm is a projected gradient method, and its constrained minimization problem is as follows:

$$\min_{x \in C} \frac{1}{2} \|Ax - P_Q Ax\|^2. \tag{2}$$

As we know, projection method only is related to the orthogonal projection that is easy to compute, and it does not require matrix inversion. The CQ algorithm is an efficient and popular method to solve SCFP. The split convex feasibility problem can be converted into the CQ algorithm with the following iterative format:

$$x_{k+1} = P_C\bigl[x_k - \lambda A^T(I - P_Q A x_k\bigr], \quad k = 0, 1, 2, \ldots, \tag{3}$$

where P_C, P_Q are the orthogonal projections onto C, Q, respectively, λ is suitable positive constant, A^T is the transpose of A, and I is the identical operator,

In addition, a wide variety of continuous-time algorithms attracted increasing attention in areas such as optimization, artificial intelligence, image restoration, and optimal control (see, e.g., [12–15]). The dynamic system method was first proposed in [12] to solve the linear programming problem of the electronic analog computer. Subsequently, a recurrent neural network was presented to deal with the nonlinear projection formulation in [13]. To further solve the optimization problem, a new continuous dynamical system of a gradient-projection type was proposed and studied in [14]. Some neural network models were introduced to solve projection formulation in [15]. The continuation of the split convex feasibility problem has recently been studied. A projection dynamical system was proposed to solve the split equality problem in [16]. Continuous CQ algorithms were proposed, and an exponential convergence was obtained under the bounded linear regularity property in [17]. Although neural networks and their variants have been widely adopted, discrete algorithms generally focus on the analysis of convergence and complexity, and continuous algorithms mainly focus on the convergence rate, and theoretical research mainly deals with the asymptotic or exponential stability (see, e.g., [17–21]). Among the existing works on the CQ algorithm and variations for the SCFP, the convergence and convergence rate of algorithms have received great attention (see, e.g., [5, 22]), but a few papers deal with convergence time.

In practical production, more attention is paid to the convergence time of the method. As discussed in [23], the design of finite-time convergence based on asymptotic or exponential stability was of more practical significance than the classical asymptotic and exponential stability. Nevertheless, the finite-time terminating neural network method is highly dependent on the initial conditions. To overcome this drawback, the notion of convergence within a fixed time was introduced in [24]. Recently, some neural networks, which achieve convergence within a finite time or fixed time, were proposed to solve some mathematical models, such as ℓ_1-minization problem, absolute value equation, and mixed variational inequalities (see, e.g., [25–31]). First, a novel proximal dynamical system, which achieves the fixed-time stability, was presented to deal with mixed variational inequality problems in [26]. For different problems, a new proximal neurodynamic network that is fixed-time converging (FXPNN) was proposed to solve equilibrium problems (EPs) in [28]. Recently, a dynamical system was released to deal with the split feasibility problem within the fixed time in [30]. To further decrease the convergence time of the dynamical system, a novel dynamical system that can update automatically, dynamical stepsize was established to deal with pseudo-monotone mixed variational inequalities in [31]. As far as we know, few papers exist for solving split convex feasible problems in fixed-time stability. We want to combine continuous algorithms with convergence time to make the algorithm more intuitive and efficient.

Motivated by the above works, we investigate a novel fixed-time stability neural network (NFxNN) to deal with the SCFP. Further, the fixed-time stability of the NFxNN is obtained under the bounded linear regularity assumption. Under mild conditions, we prove that the equilibrium point of the NFxNN globally converges to a solution of the SCFP. Besides, the convergence time of the NFxNN is uniformly bounded and is independent of any initial condition. Numerical examples present the efficiency and competitiveness of the NFxNN compared with the neural network systems.

Compared with the existing related work, the contributions of this paper can be summarized as follows:

(i) To solve the SCFP, we propose a new fixed-time neural network model. The stability of the neural network is proved. It is proved that the NFxNN converges to the solution within a fixed time. The convergence time is uniformly bounded and is independent of the initial condition.

(ii) The convergence results are compared with the relative methods (such as [17, 30]). We provide an upper bound formula for the convergence time independent of the initial condition. Compared with other methods, the NFxNN is faster to solve the SCFP. Examples in the paper show that the convergence time of the NFxNN is slightly faster than the fixed-time system in [30] and is faster than the neural network in [17].

(iii) The relations between the parameters of the NFxNN and the fixed-time stability are discussed through the combination of theory and simulation. In this paper, the influence of parameters γ, α_1, α_2 on the NFxNN is presented. This provides a reference for us to choose the appropriate parameters when solving the SCFP with this model.

The rest of this article is structured as follows: Sect. 2 reviews useful notions and definitions. A new fixed-time stability neural network model and the consistent discretization of the NFxNN are presented, and its global convergence is proved in Sect. 3. Numerical examples also confirm the effectiveness of the method in Sect. 4. Section 5 gives a conclusion.

2 Preliminaries

In this paper, let $\mathbb{R}^n$ be the n-dimensional Euclidean space with the inner products $\langle\cdot,\cdot\rangle$ and the Euclidean norm $\|\cdot\|$. Θ denotes the solution set of the split convex feasibility problem. For the sets $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^m$, the family of k-times continuously differentiable functions from U to V is denoted by $C^k(U,V)$, i.e., $C^k(U,V) := \{f : f : U \to V$ is k-times continuously differentiable mapping$\}$. The supremum of a set S, denoted as $\sup(S)$, is the smallest real number M such that M is greater than or equal to every element in S.

2.1 Definitions and notions

We first review some common definitions and facts.

Definition 1 [32] Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a mapping. F is called

(i) μ-Lipschitz continuous on $\mathbb{R}^n$ if there is a constant $\mu > 0$ that satisfies

$$\mu\|x-y\| \geq \big\|F(x)-F(y)\big\|, \quad \forall x,y \in \mathbb{R}^n,$$

and F is said to be nonexpansive if it is 1-Lipschitz continuous.

(ii) β strongly monotone on $\mathbb{R}^n$ if there is a constant $\beta > 0$ to satisfy

$$\langle F(x) - F(y), x - y \rangle \geq \beta \| F(x) - F(y) \|^2, \quad \forall x, y \in \mathbb{R}^n.$$

F is said to be firmly nonexpansive if it is 1-inverse strongly monotone.

(iii) ϱ-averaged mapping if it is reformulated into

$$F(x) = (1 - \varrho)I + \varrho T,$$

where $\varrho \in (0, 1)$, I is the identity mapping, and $T : \mathbb{R}^n \to \mathbb{R}^n$ denotes a nonexpansive mapping.

(iv) Firmly nonexpansive iff

$$\|x - y\|^2 - \| (I - F)x - (I - F)y \|^2 \geq \| F(x) - F(y) \|^2, \quad \forall x, y \in \mathbb{R}^n.$$

Property 1 We give the nonexpansive projection mapping:

(i) Both P_C and $I - P_C$ are firmly nonexpansive, iff

$$\langle x - P_C(x), y - P_C(x) \rangle \leq 0, \quad \forall x \in \mathbb{R}^n, y \in C,$$

and

$$\langle P_C(x) - P_C(y), x - y \rangle \geq \| P_C(x) - P_C(y) \|^2, \quad \forall x, y \in \mathbb{R}^n.$$

(ii) A mapping is said to be nonexpansive if it is averaged or firmly nonexpansive.

Definition 2 [17] Suppose that the solution set Θ of the SCFP is non-empty, the SCFP is called to satisfy the bounded linear regularity property if, for any $r > 0$, such that $\overline{\mathbb{B}(0, r)} \cap \Theta \neq \emptyset$, there is a $\varpi_r > 0$ to satisfy

$$\operatorname{dis}(x, \Theta) \leq \varpi_r \operatorname{dis}(Ax, Q), \quad \forall x \in \overline{\mathbb{B}(0, r)} \cap C, \tag{4}$$

where Θ is the solution set of the SCFP, and $\operatorname{dis}(\cdot, \cdot)$ is the distance function, $\mathbb{B}(x, r)$ and $\overline{\mathbb{B}(x, r)}$ denote the open and closed ball with center at x and radius r, respectively.

We consider the following differential equation:

$$\dot{y} = f(y), \tag{5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a vector valued function, and $f(\mathbf{0}) = \mathbf{0}$. Suppose that the solution of differential equation (5) exists. Without loss of generality, we suppose that the origin is the unique equilibrium point of equation (5).

Definition 3 [33] The origin is called to be a finite-time stability equilibrium of equation (5) if equation (5) is Lyapunov stable and finite-time convergent, i.e., for any $y(0) \in \mathcal{O} \setminus \{\mathbf{0}\}$, $\lim_{t \to T} y(t) = \mathbf{0}$, where $\mathcal{O}$ is some open neighborhood of the origin and $T = T(y(0)) < \infty$. The origin is called to be a globally finite-time stability equilibrium of equation (5) whenever $\mathcal{O} = \mathbb{R}^n$.

Definition 4 [24] The origin is called to be a fixed-time stability equilibrium of equation (5) if it is globally finite-time stability, and the setting time $T(y(0))$ is uniformly bounded, i.e., there exists $\bar{T} < \infty$ such that $\sup_{x(0)\in\mathbb{R}^n} T(y(0)) \leq \bar{T}$.

Lemma 1 [33] *Suppose that there is a positive definite function* $V \in C^1(D,\mathbb{R})$ *to satisfy*

$$\dot{V}(y) + cV(y)^{\alpha} \leq 0, \quad \forall y \in \mathcal{V} \setminus \{\mathbf{0}\}, \tag{6}$$

where $D \subseteq \mathbb{R}^n$ *is a neighborhood of the origin, an open neighborhood* $\mathcal{V} \subseteq D$ *of the origin,* $c > 0$ *and* $\alpha \in (0,1)$ *are constants. Then, the origin is a finite-time stability equilibrium point of equation* (5) *with the setting time* $T \leq \frac{V(y(0))^{1-\alpha}}{c(1-\alpha)}$.

Lemma 2 [24] *Suppose that there exists a positive definite function* $V \in C^1(\mathbb{R}^n,\mathbb{R})$ *to satisfy*

$$\dot{V}(y) \leq -mV(y)^{a} - nV(y)^{b}, \quad \forall y \in V\backslash\{0\}, \tag{7}$$

where $m, n > 0$, $0 < a < 1$ *and* $b > 1$. *Then, equation* (5) *is a fixed-time stability with settling time* $T \leq \frac{1}{m(1-a)} + \frac{1}{n(b-1)}$.

2.2 Neural network approach

In this subsection, we present a projection-based neural network approach to solve the SCFP (1). We give some decision methods for solving the SCFP (1).

It follows that $x \in C$ is a solution of the SCFP (1) iff it is a solution of the following:

$$\text{Find } x \in C \quad \text{such that} \quad y = Ax \in Q.$$

Based on CQ algorithm (3), the paper [17] proposed a continuous neural network (NN) to deal with the SCFP (1). For simplicity, let us use the following auxiliary functions:

$$U = I - \gamma A^T(I - P_Q)A, \tag{8}$$

where $x \in \mathbb{R}^n$ and $\gamma \in (0, \frac{2}{\|A\|^2})$. The neural network (NN) method is as follows:

$$\dot{x} = -\lambda\{x - P_C \circ Ux\}, \tag{9}$$

where the function U is defined by (8) for $\gamma \in (0, \frac{2}{\|A\|^2})$ and $\lambda > 0$. As a convenience, we let $\Psi(x) = x - P_C \circ Ux$.

Lemma 3 [34] *Assume that the solution set* Θ *of the SCFP is non-empty. The following statements are the same*:

(i) x^* *solves the SCFP.*

(ii) $P_C \circ Ux^* = x^*$, *i.e.,* $\Theta = \{x|\Psi(x) = 0\}$.

(iii) $x^* \in C$ *and* $Ux^* = x^*$.

(iv) $x^* \in C$ *and* $Ax^* = P_Q Ax^*$.

2.3 Blanket assumptions

Assumption 1 The SCFP is satisfied with bound linear condition with a constant ϖ.

Assumption 2 The parameter $\gamma \in (0, \frac{2}{L})$ with $L = \|A\|^2$.

Assumption 3 If $\varpi^2 L < 1$, we take $\gamma \in (0, \frac{1-\sqrt{1-\varpi^2 L}}{L})$ or $\gamma \in (\frac{1+\sqrt{1-\varpi^2 L}}{L}, \frac{2}{L})$.

Assumption 4 There exists $x \in \mathbb{R}^n$, such that $\Psi(x) = 0$.

Remark 1 Assumption 2 guarantees that the NFxNN (10) has a unique solution. Assumptions 1 and 3 guarantee that the NFxNN (10) achieves stability within the fixed time. Assumption 4 guarantees a consistent discretization of the NFxNN (10).

Lemma 4 [3] *If Assumption 2 holds, then $U : \mathbb{R}^n \to \mathbb{R}^n$ is an averaged mapping.*

Lemma 5 *Assumption 2 holds. Then,*

$$\left\| \Psi(x) - \Psi(y) \right\| \le 2\|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

Proof It is proved in [17, Theorem 1, p. 2993]. □

3 Fixed-time stability of the NFxNN and its convergence time estimation

A novel neural network is introduced to deal with the split convex feasibility problem. The fixed-time stability of the proposed neural network is obtained, and the upper bound forum of its convergence time is uniformly bounded. Moreover, it is irrelevant to the initial states.

3.1 A fixed-time converging neural network to deal with the SCFP

Inspired by [17, 31], we now present a new fixed-time converging neural network (NFxNN) to solve the SCFP (1):

$$\dot{x} = -\rho(x)(x - P_C \circ Ux), \tag{10}$$

where $\lambda > 0$, $\gamma \in (0, \frac{2}{\|A\|^2})$, U is defined by (8), and

$$\rho(x) := \begin{cases} \rho_1 \frac{1}{\|\Psi(x)\|^{1-\alpha_1}} + \rho_2 \frac{1}{\|\Psi(x)\|^{1-\alpha_2}} + \rho_3 \frac{1}{\|\Psi(x)\|}, & \text{if } x \in \mathbb{R}^n \setminus \mathrm{E}(\Psi), \\ 0, & \text{else.} \end{cases}$$

Where $\rho_1, \rho_2, \rho_3 > 0$, $\alpha_1 \in (0,1)$, $\alpha_2 > 1$ are tunable parameters, $\Psi(x) = x - P_C \circ Ux$, and $\mathrm{E}(\Psi) = \{x \in \mathbb{R}^n | \Psi(x) = \mathbf{0}\}$ is the equilibrium points set of Ψ.

Carefully observe NFxNN (10), which can also be expressed as:

$$\dot{x} = -\omega(x)\rho_1 \left(\left\| \Psi(x) \right\|^{\alpha_1} \left(1 + \frac{\rho_2}{\rho_1} \left\| \Psi(x) \right\|^{\alpha_2 - \alpha_1} \right) + \frac{\rho_3}{\rho_1} \right), \tag{11}$$

where $-\omega(x) = \frac{\Psi(x)}{\|\Psi(x)\|} = \frac{x - P_C \circ Ux}{\|\Psi(x)\|}$. From the perspective of algorithm design, $\omega(x)$ can be deemed as a search direction, and $\alpha_t := \rho_1(\|\Psi(x)\|^{\alpha_1}(1 + \frac{\rho_2}{\rho_1}\|\Psi(x)\|^{\alpha_2-\alpha_1}) + \frac{\rho_3}{\rho_1})$ can be deemed as stepsize at the current iterate $x = x(t)$ in NFxNN (10). The stepsize α_t can be self-update with respect to t whenever ρ_1, ρ_2, ρ_3 and α_1, α_2 are given. If $\rho_3 = 1$, the NFxNN (10) is reduced to

$$\dot{x} = -\omega(x)\rho_1\left(\left\|\Psi(x)\right\|^{\alpha_1}\left(1 + \frac{\rho_2}{\rho_1}\left\|\Psi(x)\right\|^{\alpha_2-\alpha_1}\right) + \frac{1}{\rho_1}\right).$$

If $\rho_1 = \rho_2 = 0$ in the NFxNN (10), the NFxNN (10) is reduced to neural network (NN) in [17]

$$\dot{x} = -\rho_3\omega(x). \tag{12}$$

If $\rho_2 = \rho_3 = 0$, the NFxNN (10) is reduced to the following finite-time converging neural network (FiNN):

$$\dot{x} = -\rho_1\omega(x)(x - P_C \circ Ux)^{\alpha_1}. \tag{13}$$

If $\rho_3 = 0$ in, specially the tunable parameter $\alpha = 1$ in [30], the NFxNN (10) degenerates into the fixed-time dynamical system (FxDS) in [30].

$$\dot{x} = -\omega(x)\big(\rho_1(x - P_C \circ Ux)^{\alpha_1} + \rho_2(x - P_C \circ Ux)^{\alpha_2}\big). \tag{14}$$

Besides, the stepsize α_t in NFxNN (10) when $\rho_3 > 0$ is larger than that when $\rho_3 = 0$. Generally, the larger stepsize can speed up the convergence rate of the numerical algorithms. It will be showed in Sect. 5. So, the NFxNN (10) is distinct of the fixed-time dynamical system in [30].

3.2 Convergence analysis of the NFxNN with the fixed time

We establish the globally convergence of the NFxNN. Futhermore, its convergence time estimation is obtained under the bounded linear regularity assumption and the Lipschitz continuity assumption. For proving the convergence of the NFxNN (10), we give the following auxiliary lemmas.

Lemma 6 *$x^* \in \Theta$ is a solution of the NN (9) iff it is a solution of the NFxNN (10).*

Proof If $x^* \in \Theta$ is a solution of the NFxNN (10), then

$$\begin{aligned}
&\rho\big(x^*\big)\big(x^* - P_C \circ Ux^*\big) = 0 \\
&\quad\Rightarrow\quad \rho_1\frac{\Psi(x^*)}{\|\Psi(x^*)\|^{1-\alpha_1}} + \rho_2\frac{\Psi(x^*)}{\|\Psi(x^*)\|^{1-\alpha_2}} + \rho_3\frac{\Psi(x^*)}{\|\Psi(x^*)\|} = 0 \quad \text{or} \quad \rho\big(x^*\big) = 0 \\
&\quad\Rightarrow\quad \Psi\big(x^*\big)\left(\rho_1\frac{1}{\|\Psi(x^*)\|^{1-\alpha_1}} + \rho_2\frac{1}{\|\Psi(x^*)\|^{1-\alpha_2}} + \rho_3\frac{1}{\|\Psi(x^*)\|}\right) = 0 \quad \text{or} \quad x^* \in E(\Psi) \\
&\quad\Rightarrow\quad \Psi\big(x^*\big) = 0 \quad \text{or} \quad x^* \in E(\Psi),
\end{aligned}$$

it means that x^* is a solution of the equation (9).

On the other hand, if x^* is a solution of the equation (9), $\Psi(x^*) = 0$, we get

$$\rho\big(x^*\big)\big(P_C \circ Ux^* - x^*\big) = 0,$$

which gets that x^* is a solution of the NFxNN (10). So far, we have completed the proof. □

Remark 2 The significant idea of the projection neural approach is to establish the equivalence between optimization problems and Brouwer's fixed-point problems. The SCFP (1) is equivalent to the solution of equation (9). Due to Lemma 6, the NFxNN (10) can solve the SCFP (1).

Lemma 7 *Let $x(t)$ be the trajectory of system* (10). *Where $x(t_0) = x_0 \in C$ and $x^* \in \Theta$, which satisfies $x(t) \to x^*$. Let $x^* \in \Theta$ be a solution of the SCFP. Let Assumptions* 1, 2, *and* 3 *hold. Then, $\|P_C \circ Ux - P_\Theta x\| \leq \tau\|x - P_\Theta x\|$, where $0 < \tau = \sqrt{1 - \frac{\gamma(2-\gamma L)}{\varpi^2}} < 1$, and U is defined by* (8).

Proof Due to nonexpansive property of projection mapping, we get

$$\big\|P_C \circ Ux - P_C \circ U(P_\Theta x)\big\|^2 \leq \big\|Ux - U(P_\Theta x)\big\|^2. \tag{15}$$

Recalling the definition of U in (8), we have

$$\begin{aligned}
&\big\|Ux - U(P_\Theta x)\big\|^2 \\
&\quad = \big\|x - P_\Theta x - \gamma A^T(I - P_Q)Ax + \gamma A^T(I - P_Q)AP_\Theta x\big\|^2 \\
&\quad = \|x - P_\Theta x\|^2 + \big\|\gamma A^T(I - P_Q)Ax - \gamma A^T(I - P_Q)A(P_\Theta x)\big\|^2 \\
&\qquad - 2\big\langle x - P_\Theta x, \gamma A^T(I - P_Q)Ax - \gamma A^T(I - P_Q)A(P_\Theta x)\big\rangle.
\end{aligned}$$

We now estimate the last two terms of the above formula. Using (iii) of Lemma 3 and the L definition of Assumption 2, we have that

$$\big\|\gamma A^T(I - P_Q)Ax - \gamma A^T(I - P_Q)A(P_\Theta x)\big\|^2 \leq \gamma^2 L\big\|(I - P_Q)Ax\big\|^2.$$

From (*i*) of Property 1 and (*iv*) of Lemma 3, we get

$$\begin{aligned}
&-2\big\langle x - P_\Theta x, \gamma A^T(I - P_Q)Ax - \gamma A^T(I - P_Q)A(P_\Theta x)\big\rangle \\
&\quad = -2\gamma\big\langle Ax - A(P_\Theta x), (I - P_Q)Ax - (I - P_Q)A(P_\Theta x)\big\rangle \\
&\quad \leq -2\gamma\big\|(I - P_Q)Ax\big\|^2.
\end{aligned}$$

Substitute the above two inequalities into (15), we get

$$\big\|P_C \circ Ux - P_C \circ U(P_\Theta x)\big\|^2 \leq \|x - P_\Theta x\|^2 - \gamma(2 - \gamma L)\big\|(I - P_Q)Ax\big\|^2.$$

Sine $x(t) \in C$ for all $t \geq t_0$ was proved in [17, Lemma 3.4, p. 2998], and the bounded linear regularity property holds, there is a $\varpi > 0$ and $\bar{t} \geq t_0$ such that

$$\operatorname{dist}(x, \Theta) = \|x - P_\Theta x\| \leq \varpi\big\|(I - P_Q)Ax\big\|, \quad \forall t \geq \bar{t}.$$

So,

$$-\gamma(2-\gamma L)\left\|(I-P_Q)Ax\right\|^2 \le -\frac{\gamma(2-\gamma L)}{\varpi^2}\|x-P_\Theta x\|^2.$$

If $\varpi^2 L \ge 1$, then $0 < \frac{\gamma(2-\gamma L)}{\varpi^2} \le 1$. If $\varpi^2 L < 1$, we take $\gamma \in (0, \frac{1-\sqrt{1-\varpi^2 L}}{L})$ or $\gamma \in (\frac{1+\sqrt{1-\varpi^2 L}}{L}, \frac{2}{L})$, then $0 < \frac{\gamma(2-\gamma L)}{\varpi^2} < 1$. We can get that

$$\left\|Ux - U(P_\Theta x)\right\|^2 \le \left(1 - \frac{\gamma(2-\gamma L)}{\varpi^2}\right)\|x-P_\Theta x\|^2. \tag{16}$$

By Lemma 3, we have $P_C \circ U(P_\Theta x) = P_\Theta x$. It follows from (15) and (16) that

$$\|P_C \circ Ux - P_\Theta x\| \le \tau\|x-P_\Theta x\|,$$

with $\tau = \sqrt{1-\frac{\gamma(2-\gamma L)}{\varpi^2}} \in (0,1)$. It completes the proof. □

Theorem 1 *Let $x(t)$ be the trajectory of system* (10). *Where $x(t_0) = x_0 \in C$ and $x^* \in \Theta$ satisfy $x(t) \to x^*$. Let $x^* \in \Theta$ be a solution of the SCFP. Assumptions* 1, 2, *and* 3 *hold. Then the solution of the SCFP is a globally fixed-time equilibrium point of the NFxNN, and the convergence time is given as*

$$T(x_0) \le \frac{1}{N_1(1-p_1)} + \frac{1}{N_2(p_2-1)},$$

where x_0 is an initial point of the NFxNN (10), *and $p_1 = \frac{1+\alpha_1}{2}$, $p_2 = \frac{1+\alpha_2}{2}$, $N_1 = 2^{\frac{1+\alpha_1}{2}}\rho_1 \times \frac{1-\tau}{(1+\tau)^{1-\alpha_1}}$, $N_2 = 2^{\frac{1+\alpha_2}{2}}(1-\tau)^{\alpha_2}(\rho_2 + \rho_3\frac{1}{\|\Psi(x)\|^{\alpha_2}})$ for $x \in \mathbb{R}^n$ and $\tau = \sqrt{1-\frac{\gamma(2-\gamma L)}{\varpi^2}} \in (0,1)$.*

Proof It follows from the global version of the *Picard-Lindelöf* theorem in [35, Theorem 2.2, p. 38] that the NFxNN (10) can achieve a unique solution under the Assumptions 1, 2, and 3. Since x^* is a solution of the NFxNN (10), we deduce by the global version of the *Picard-Lindelöf* theorem and Lemma 6 that $x^* \in \mathbb{R}^n$ is a unique solution of the NFxNN (10). To simplify the proof, we derive from Lemma 7 that

$$\begin{aligned}
\left\langle x-P_\Theta x, \omega(x)\right\rangle &= \frac{1}{\|\Psi(x)\|}\langle x-P_\Theta x, x-P_\Theta x + P_\Theta x - P_C \circ Ux\rangle \\
&= \frac{1}{\|\Psi(x)\|}\left(\|x-P_\Theta x\|^2 - \langle x-P_\Theta x, P_C \circ Ux - P_\Theta x\rangle\right) \\
&\ge (1-\tau)\frac{1}{\|\Psi(x)\|}\|x-P_\Theta x\|^2,
\end{aligned} \tag{17}$$

where the inequality can be obtained by Lemma 7, i.e.,

$$\left\langle x-P_\Theta x, \Psi(x)\right\rangle \ge (1-\tau)\|x-P_\Theta x\|^2.$$

The solution of the NFxNN (10) exists uniquely for any initial state according to [26, Proposition 2]. Now, consider a candidate Lyapunov function:

$$V\left(x(t)\right) = \frac{1}{2}\operatorname{dist}^2\left(x(t),\Theta\right) = \frac{1}{2}\left\|x(t) - P_\Theta x(t)\right\|^2. \tag{18}$$

The time-derivation of the candidate Lyapunov function V along the solution of (10), we get

$$\begin{aligned}
\dot{V}(x) &= (x - P_\Theta x)^\top \dot{x} \\
&= -\big(\rho_1 \big\| \Psi(x) \big\|^{\alpha_1} + \rho_2 \big\| \Psi(x) \big\|^{\alpha_2} + \rho_3\big)\big\langle x - P_\Theta x, \omega(x)\big\rangle \\
&\leq -\rho_1(1-\tau)\frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|^{\alpha_1}} - \rho_2(1-\tau)\frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|^{\alpha_2}} - \rho_3(1-\tau)\frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|} \\
&\leq -\rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}}\|x - P_\Theta x\|^{1+\alpha_1} \\
&\quad - (1-\tau)^{\alpha_2}\big(\rho_2 + \rho_3\big\|\Psi(x)\big\|^{\alpha_2}\big)\|x - P_\Theta x\|^{1+\alpha_2},
\end{aligned} \tag{19}$$

where the first inequality is established by inequality (17), and the second by Lemma 7. Then,

$$\dot{V} \leq -M_1\|x - P_\Theta x\|^{1+\alpha_1} - M_2\|x - P_\Theta x\|^{1+\alpha_2}, \tag{20}$$

where $M_1 = \rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}}$ and $M_2 = (1-\tau)^{\alpha_2}(\rho_2 + \rho_3\|\Psi(x)\|^{\alpha_2})$.

It can be seen from (18) and (21) that

$$\begin{aligned}
\dot{V}(x) &\leq -M_1\|x - P_\Theta x\|^{1+\alpha_1} - M_2\|x - P_\Theta x\|^{1+\alpha_2} \\
&= -M_1\big(2V(x)\big)^{\frac{1+\alpha_1}{2}} - M_2\big(2V(x)\big)^{\frac{1+\alpha_2}{2}} \\
&\leq -\big(N_1 V(x)^{p_1} + N_2 V(x)^{p_2}\big),
\end{aligned} \tag{21}$$

where $p_i = \frac{1+\alpha_i}{2}$, $i = (1,2)$, $N_i = M_i 2^{p_i}$, $i = (1,2)$. And $N_1 = M_1 2^{p_1} > 0$, $p_1 = \frac{1+\alpha_1}{2} \in (0, 0.5)$ since $\alpha_1 \in (0,1)$, $N_2 = M_2^{p_2} > 0$, $p_2 = \frac{1+\alpha_2}{2} \in (1,\infty)$ since $\alpha_2 \in (1,\infty)$. It then follows from Lemma 2, (ii) and (iv) of Lemma 3, and the intersection of two convex sets is still convex, that the equilibrium point x^* of the NFxNN (10) is globally fixed-time stability, and the settling time satisfies

$$T\big(x(0)\big) \leq \frac{1}{N_1(1-p_1)} + \frac{1}{N_2(p_2-1)}, \tag{22}$$

where $x(0)$ is an initial condition of the NFxNN (10) and $N_1 = 2^{\frac{1+\alpha_1}{2}} \rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}}$ and $N_2 = 2^{\frac{1+\alpha_2}{2}}(1-\tau)^{\alpha_2}(\rho_2 + \rho_3 \frac{1}{\|\Psi(x)\|^{\alpha_2}})$. Here, $\|\Psi(x)\|^{\alpha_2} > 0$, for $x \in \mathbb{R}^n$, and since $\rho_3 > 0$, $\rho_3 \frac{1}{\|\Psi(x)\|^{\alpha_2}} > 0$. We can find that the NFxNN (10) reduces the upper bound of the convergence time, and the method is also independent of the initial state. The proof is completed. □

Theorem 2 *Let $x(t)$ be the trajectory of the FiNN (13). Where $x(t_0) = x_0 \in C$, and $x^* \in \Theta$ satisfies $x(t) \to x^*$. Assumptions 1, 2, and 3 hold. For each $\alpha_1 \in (0,1)$, $\rho_1 > 0$, the FiNN (13) is finite-time stability with the convergence time*

$$T \leq \frac{V(x(0))^{1-\beta}}{c(1-\beta)},$$

where $c = 2^{\frac{1+\alpha_1}{2}} \rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}} > 0$, $\beta = \frac{1+\alpha_1}{2} \in (0.5,1)$, $0 \leq \tau = \sqrt{1 - \frac{\gamma(2-\gamma L)}{\varpi^2}} < 1$ and $V(x(t)) = \frac{1}{2}\operatorname{dist}^2(x(t),\Theta)$.

Proof Follow the proof of Theorem 1. Then, the time-derivation of the candidate Lyapunov function V along the solution of (13), one has

$$\begin{aligned}
\dot{V}(x) &= (x-\bar{x})^\top \dot{x} \\
&= \left\langle x - P_\Theta x, \rho_1 \frac{P_C \circ Ux - x}{\|\Psi(x)\|^{1-\alpha_1}} \right\rangle \\
&= \left\langle x - P_\Theta x, \rho_1 \frac{P_C \circ Ux - P_\Theta x + P_\Theta x - x}{\|\Psi(x)\|^{1-\alpha_1}} \right\rangle \\
&= -\rho_1 \frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|^{1-\alpha_1}} + \left\langle x - P_\Theta x, \rho_1 \frac{P_C \circ Ux - P_\Theta x}{\|\Psi(x)\|^{1-\alpha_1}} \right\rangle \\
&\leq -\rho_1 \frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|^{1-\alpha_1}} + \rho_1 \frac{\|x - P_\Theta x\| \|P_C \circ Ux - P_\Theta x\|}{\|\Psi(x)\|^{1-\alpha_1}} \\
&\leq -\rho_1 (1-\tau) \frac{\|x - P_\Theta x\|^2}{\|\Psi(x)\|^{1-\alpha_1}},
\end{aligned} \tag{23}$$

where the first inequality is established by the Cauchy-Schwarz inequality, and the second holds by Lemma 7, $\Psi(x) = P_C \circ Ux - x$ and $\psi(x) = P_C \circ Ux$. Using Lemma 7, there exists $0 \leq \tau = \sqrt{1 - \frac{\gamma(2-\gamma L)}{\varpi^2}} < 1$ such that

$$\begin{aligned}
&(1-\tau)\|x - P_\Theta x)\| \leq \|x - P_\Theta x\| - \left\|\psi(x) - P_\Theta x\right\| \leq \left\|x - \psi(x)\right\|, \quad \forall x \in \mathbb{R}^n, \\
&\left\|x - \psi(x)\right\| \leq \|x - P_\Theta x\| + \left\|\psi(x) - P_\Theta x\right\| \leq (1+\tau)\|x - P_\Theta x)\|, \quad \forall x \in \mathbb{R}^n.
\end{aligned} \tag{24}$$

Combining (23) with (24), we have

$$\begin{aligned}
\dot{V}(x) &\leq -\rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}} \|x - P_\Theta x\|^{1+\alpha_1} \\
&= -2^{\frac{1+\alpha_1}{2}} \rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}} V^{\frac{1+\alpha_1}{2}} \\
&= -cV^\beta,
\end{aligned}$$

where $c = 2^{\frac{1+\alpha_1}{2}} \rho_1 \frac{1-\tau}{(1+\tau)^{1-\alpha_1}} > 0$, $\beta = \frac{1+\alpha_1}{2} \in (0.5, 1)$. Therefore, it follows from Lemma 1 that for each $\alpha_1 \in (0,1)$, the FiNN (13) is globally finite-time stability with the convergence time

$$T = T\bigl(x(0)\bigr) \leq \frac{V(x(0))^{1-\beta}}{c(1-\beta)},$$

which means that x^* is a finite-time stability solution of the FiNN (13). □

Remark 3 (i) Theorem 1 presents the upper bound of the convergence time of the NFxNN (10), which is less, and the accuracy is higher. Besides, the the upper bound of the convergence time of the NFxNN (10) is less than that in [30].

(ii) We can see from Fig. 1 that the convergence time of the NFxNN (10) is about 0.01 second, and the convergence time of the FiNN (13) and the FxDS (14) is about 0.025 second. The convergence time of the NN (12) is about 0.5 second. It can thus be seen that the convergence time of the NFxNN (10) is less than that of the NN (12), the FiNN model

(13), and the FxDS model (14), respectively. The convergence responses of the errors are different in the NN (12), FiNN (13), FxDS (14), and NFxNN (10). The NFxNN (10) can reach it quickly under the equal error condition, and NN (12) is much slower than them.

3.3 Consistent discretization of the NFxNN

Continuous-time neural networks provide an effective way to accelerate the design of schemes for solving the split convex feasibility problems. Typically, discrete-time implementations use iterative methods to solve problems. We note that the fixed-time convergence of continuous-time neural networks may not hold in discrete cases. Uniform discretization refers to the discretization scheme that still has the convergence result of the continuous-time neural network under discrete conditions. In this section, we give the uniform discretization of the NFxNN (10). We next give the result of the forward-Euler discretization of the NFxNN (10):

$$\frac{x_{n+1}-x_n}{\eta} = -\rho(x_n)(x_n - P_C \circ Ux_n)), \tag{25}$$

where $\eta > 0$ is the time-step and $\rho(\cdot)$ is given by (10), with $\rho_1, \rho_2, \rho_3 > 0$, and $x_k := x(t)$ with $t = \eta n$ and $\dot{x} = \frac{dx}{dt}$.

Theorem 3 *Let x^* be a trajectory of the SCFP. All assumptions of Theorem 1 hold. Then, for every $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$ and $\gamma \in (0, \frac{2}{L})$, satisfying Assumptions 2, 3, and 4, there exist $\delta \in (0.5, 1)$, N_1, N_2 and $\eta^* > 0$ satisfied for any $\eta \in (0, \eta^*]$, the sequence x_n generated by (25) holds:*

$$\|x_n - x^*\| < \begin{cases} \sqrt{2}(\sqrt{\frac{N_1}{N_2}}\tan(\frac{\pi}{2} - \sqrt{N_1N_2}\delta\eta k))^{\frac{1}{2-2\delta}} + \varepsilon, & n \le n^*; \\ \varepsilon, & otherwise, \end{cases}$$

where $n^* = \lceil \frac{\pi}{2\sqrt{N_1N_2}\eta\delta} \rceil$, $\lceil x \rceil = [x] + 1$. $[x]$ denotes the largest integer not exceeding x, and x_n is a solution of (25) staring from the point x_0, and $x^* \in \mathbb{R}^n$ is the solution of (10). α_1, α_2, N_1, N_2 is given by Theorem 1.

Proof Since $P_C \circ Ux$ is Lipschitz continuous, $\Psi(x)$ is continuous. The remainder of proof is like Theorem 4 in [26]. So, it is omitted. □

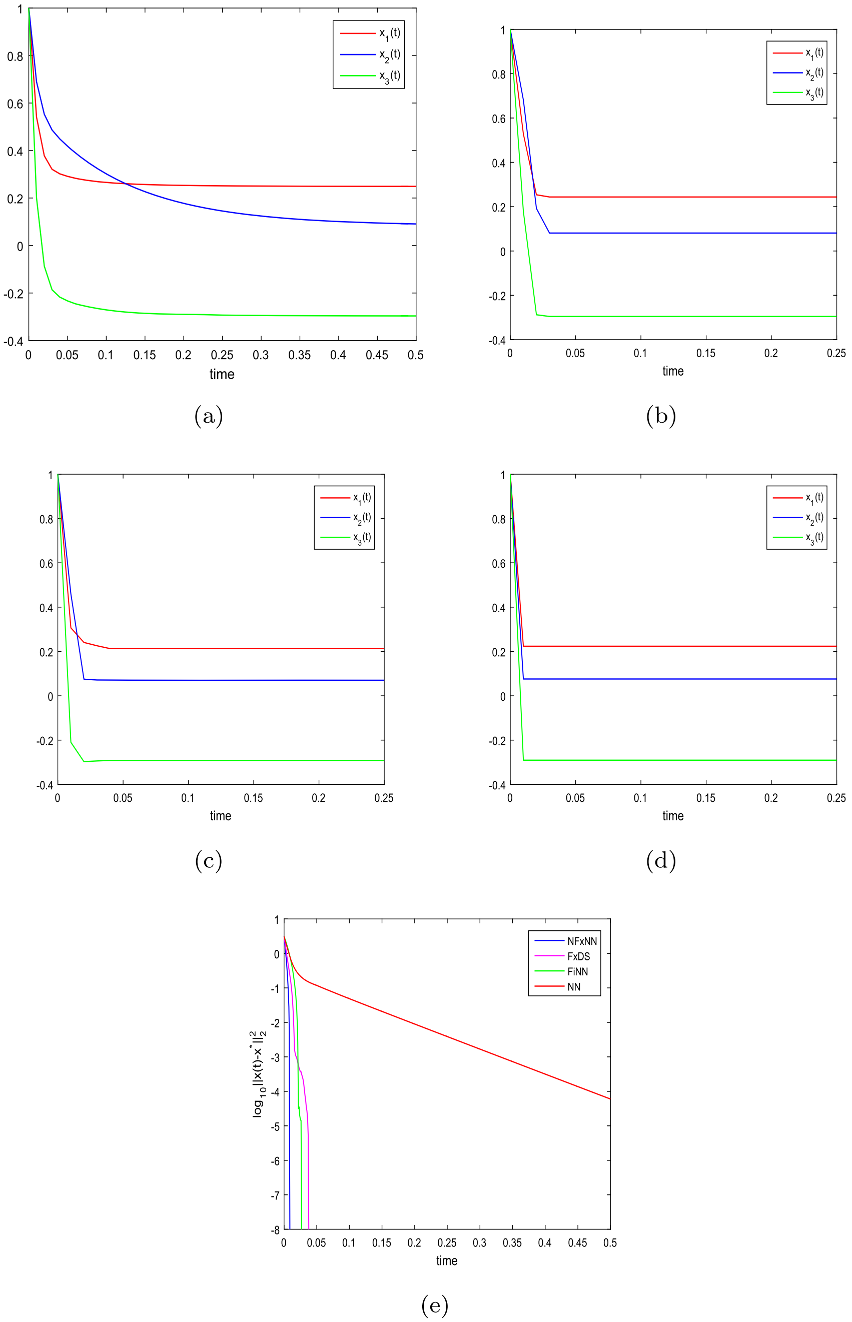
4 Numerical experiments

In this section, we aim to illustrate the analytical results with some simulations. We demonstrate the validity of the proposed approaches to deal with the split convex feasibility problem by several numerical examples. All programs are edited by Matlab 2018. All programs are executed on a personal computer. Unless otherwise stated, let $\alpha = 0.2$, $\alpha_1 = 0.2$, $\alpha_2 = 1.2$, $\rho_1 = \rho_2 = 100$ and $\gamma = \frac{1}{L}$.

Example 1 [6, 17] Consider the SCFP (1), where

$$A = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 5 \\ 2 & 0 & 2 \end{bmatrix}, \qquad B = I, \qquad C = \{x \in \mathbb{R}^3 | x_1 + x_2^2 + 2x_3 \le 0\} \quad \text{and}$$

$$Q = \{y \in \mathbb{R}^3 | y_1^2 + y_2 - y_3 \le 0\}.$$



(a)

(b)

(c)

(d)

(e)

Fig. 1 (a) The trajectory of the NN (12) in [17] with initial condition $x_0 = [1,1,1]^T$; (b) The trajectory of the FiNN (13) with initial condition $x_0 = [1,1,1]^T$; (c) The trajectory of the FxDS (14) with initial condition $x_0 = [1,1,1]^T$; (d) The trajectory of the NFxNN (10) with initial condition $x_0 = [1,1,1]^T$; (e) Convergence responses of the errors $\log_{10}\|x(t) - x^*\|_2^2$ for the neural network (12) (red), the FiNN (13) (green), the FxDS (14) (purple) and the NFxNN (10) (blue)

All requirements in Theorem 1 can be met. Numerical simulations of the NN system (12) in [17] achieve the equilibrium point $x^* = [0.2488, 0.0831, -0.2969]^T \in \Theta$. Numerical simulations of the FiNN system (13) obtain the solution $x^* = [0.2437, 0.0810, -0.2954]^T \in \Theta$. Numerical simulations of the FxDS system (14) converge to the point $x^* = [0.2129, 0.0703,$

$-0.2914]^T \in \Theta$. Numerical simulations of the NFxNN system (10) achieve the solution $x^* = [0.2237, 0.0758, -0.2905]^T \in \Theta$. Figure 1 presents the convergence trajectories of the NN (12) in [17] , the FiNN model (13), the NFxNN (10) for initial conditions $x_0 = [1, 1, 1]^T$. We can see from Fig. 1 that the convergence time of the NFxNN (10) is about 0.01 second, and the convergence time of the FiNN (13) and the FxDS (14) is about 0.025 second. The convergence time of the NN (12) is about 0.5 second. It can thus be seen that the convergence time of the NFxNN (10) is less than that of the NN (12), the FiNN model (13), and the FxDS model (14), respectively. The convergence responses of the errors are different in the NN (12), FiNN (13), FxDS (14), and NFxNN (10). The NFxNN (10) can reach it quickly under the equal error condition, and NN (12) is much slower than them.

Example 2 [17] We consider SCFP (1): the matrix $A = (a_{ij})_{2\times2}$ and $a_{ij} \in (0, 1)$ are generated randomly, and $C = \{u \in \mathbb{R}^2 | c(u) \leq 0\}$ and $Q = \{v \in \mathbb{R}^2 | q(v) \leq 0\}$ are the non-empty closed convex set with

$$\begin{aligned} &c(u) = -u_1 + u_2^2, \\ &q(v) = v_1 + v_2^2; \end{aligned}$$

for all $u = (u_1, u_2)^T \in \mathbb{R}^2$ and $v = (v_1, v_2)^T \in \mathbb{R}^2$. We take

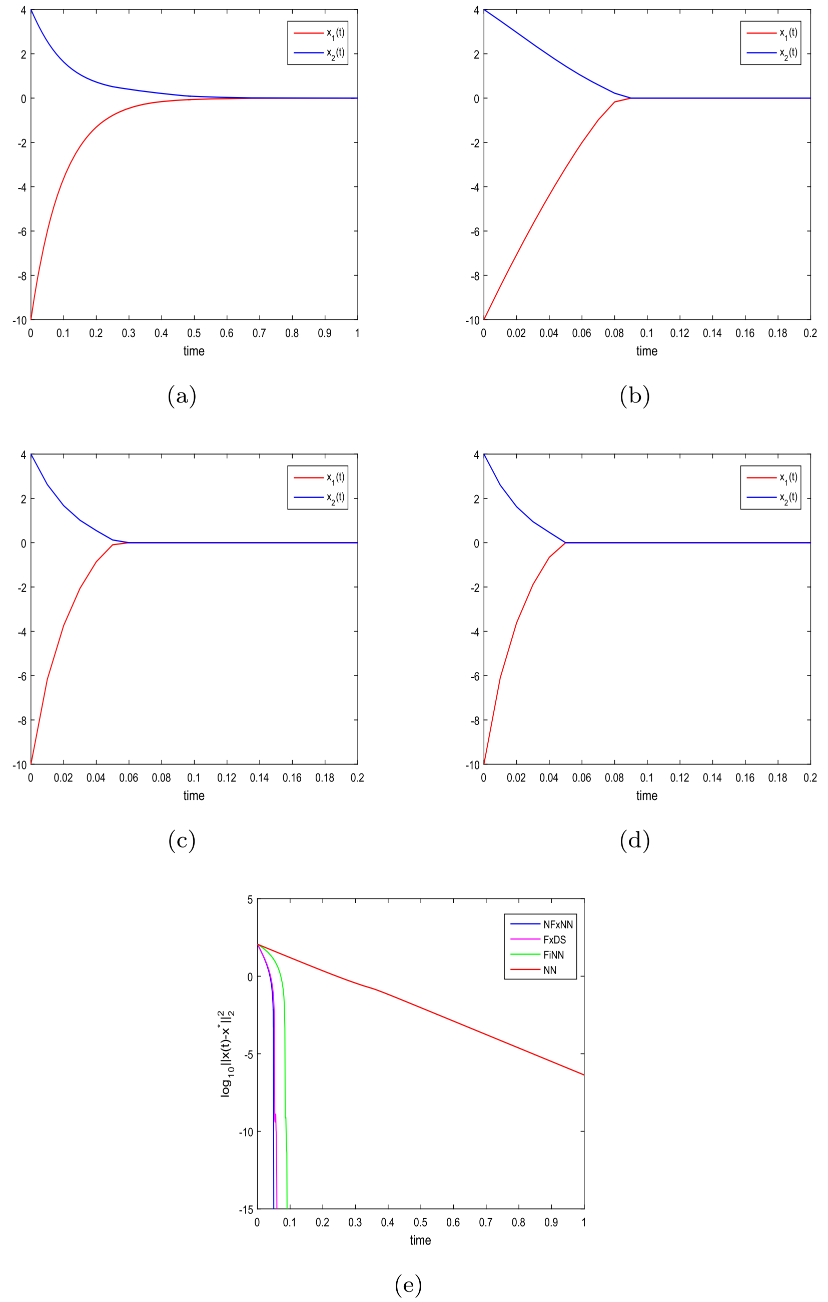
$$A = \begin{bmatrix} 0.9572 & 0.8013 \\ 0.4854 & 0.1419 \end{bmatrix} \quad \text{and} \quad x_0 = \begin{bmatrix} -10 \\ 4 \end{bmatrix},$$

generated randomly. All conditions of Theorem 1 are satisfied. The trajectory $x(t)$ of the NN in [17] achieves the point $x^* = [-0.0003956, 0.0005248]^T \in \Theta$. Numerical simulations of the FiNN system (13) converge to the equilibrium point $x^* = [0.00001986, -0.00003241]^T \in \Theta$. Numerical simulations of the FxDS (14) obtain the solution $x^* = [0.00001994, -0.00002881]^T \in \Theta$. Numerical simulations of the NFxNN (10) converge to the point $x^* = [0.00001994, -0.00002881]^T \in \Theta$.

Figure 2 demonstrates the convergence trajectories of the NN (12) in [17], the FiNN (13), the FxDS (14), and the NFxNN (10) for initial conditions x_0. We can observe from Fig. 2 that the convergence time of the NFxNN (10) is about 0.05 second. The NFxNN (10) takes about 0.06 second, and the convergence time of the NN (12) is much more than 0.5 second. The FiNN (13) takes about 0.09 second. We can observe the difference from the convergence responses of the errors in Fig. 2 (d). It also presents that the convergence rate of the NFxNN (10) is faster than that of the NN (12) and slightly faster than that of the FiNN (13) and FxDS (14). So, the convergence time of the NFxNN (10) is less than that of the NN (12), FiNN (13), and FxDS (14). From the error image, we can also see that the NN (12) achieves less accuracy than the FiNN (13), FxDS (14), and NFxNN (10).

Example 3 [18] Consider linear equation problem (LEP), find $x \in \mathbb{R}^n$ to satisfy

$$Bx = b.$$



(a) (b) (c) (d) (e)

Fig. 2 (a) The trajectory of the NN (12) in [17] with initial condition $x_0 = [-10, 4]^T$; (b) The trajectory of the FiNN (13) with initial condition $x_0 = [-10, 4]^T$; (c) The trajectory of the the FxDS (14) with initial condition $x_0 = [-10, 4]^T$; (d) The trajectory of the NFxNN (10) with initial condition $x_0 = [-10, 4]^T$; (e) Convergence responses of the errors $\log_{10} \|x(t) - x^*\|_2^2$ for the neural network (12) (red), the FiNN (13) (green), the FxDS (14) (purple) and the NFxNN (10) (blue)

When $C = \mathbb{R}^n$ and $Q = \{b\}$, the LEP can transform into a split feasible problem. In this case, the algorithm degrades into the following form:

$$x_{n+1} = x_n + \gamma B^T(b - Bx_n), \quad n = 0, 1, 2, \ldots,$$

for x_0 arbitrary. So, system (9) degrades into the following dynamical system:

$$\dot{x} = \gamma B^T(b - Bx), \quad n = 0, 1, 2, \ldots.$$

We consider LEP with

$$B = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 5 \\ 2 & 1 & 2 \end{bmatrix}, \qquad b = [1, 4, 7]^T.$$

By means of algebraic calculation, it is easy to verify that $x^* = [9, -1, -6]^T$ is a unique solution. All numerical simulations achieve the unique solution $x^* = [9, -1, -6]^T$. Figure 3 displays that the trajectories of dynamical system (12), (13), (14), and (10) with initial point $x_0 = [-3, 5, 2]^T$. And all globally converge to x^*.

Figure 3 presents the convergence trajectories of the NFxNN (10), FxDS (14), FiNN (13), and NN (12) in [17] for solving the split convex feasibility problem.

We can observe from Fig. 3 (a), (b), (c) the convergence times of the NFxNN (10), FxDS (14), FiNN (13), and NN (12). The NFxNN (10) takes about 0.1 second. The FxDS (14) and FiNN (13) are about 0.2 second, and the NN (12) is much more than 2 second. So, the convergence speed of the NFxNN (10) is faster than that of the FxDS (14), FiNN (13), and NN (12). The difference on the convergence error of the NFxNN (10), FxDS (14), FiNN (13), and NN (12) is also shown from the convergence errors in Fig. 3 (d).

Example 4 We use the continuous-time method to solve the problem of recovering noisy sparse signals from a limited number of samples. The problem was considered in [8]. Let $x \in \mathbb{R}^n$ be a $K - sparse$ signal and $K \ll n$. The sampling matrix $B \in \mathbb{R}^{m\times n}$ is stimulated by the standard Gaussian distribution and vector $b = Bx + e$ with e being an additional noisy. The main work is to recover the signal x from the data b. The recovering noisy sparse signals can be converted into the *LASSO* problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Bx - b\|_2^2$$
$$\text{s.t. } \|x\|_1 \leq t,$$

where $t > 0$ is a given constant. We take $C = \{x : \|x\|_1 \leq t\}$ and $Q = \{b\}$, the above *LASSO* problem is converted into SCFP (1). So, we can deal with the problem by dynamical system (12), (13), (14), and (10). In the process of using the system to solve (such as (12), (13), (14) and (10)), we take the following set of parameters: the initial point $x_0 \in \mathbb{R}^n$ of system (12), (13), (14), (10) and the matrix $B \in \mathbb{R}^{m\times n}$ are generated randomly. The true signal $x \in \mathbb{R}^n$ with K non-zero elements is generated from uniform distribution in the interval [–2,2] and vector $b = Ax$. Let $m = n = 100$, $t = K = 10$, $\gamma = \frac{1}{\|A\|^2}$ and $\lambda = 5$.

We compare the NFxNN (10) with NN (12) in [17], the FiNN (13), and FxDS (14). The convergence trajectories of the NFxNN (10), the NN (12) in [17], the FiNN (13), the FxDS (14) are shown in Fig. 4. We can observe from Fig. 4 (a), (b), (c), (d) the convergence time of the NFxNN (10), FxDS (14), FiNN (12), and NN (9). The NFxNN (10) is about 0.02 second. The FxDS (14) takes about 0.025 second. The FiNN (12) is about 0.09 second. The NN (9)

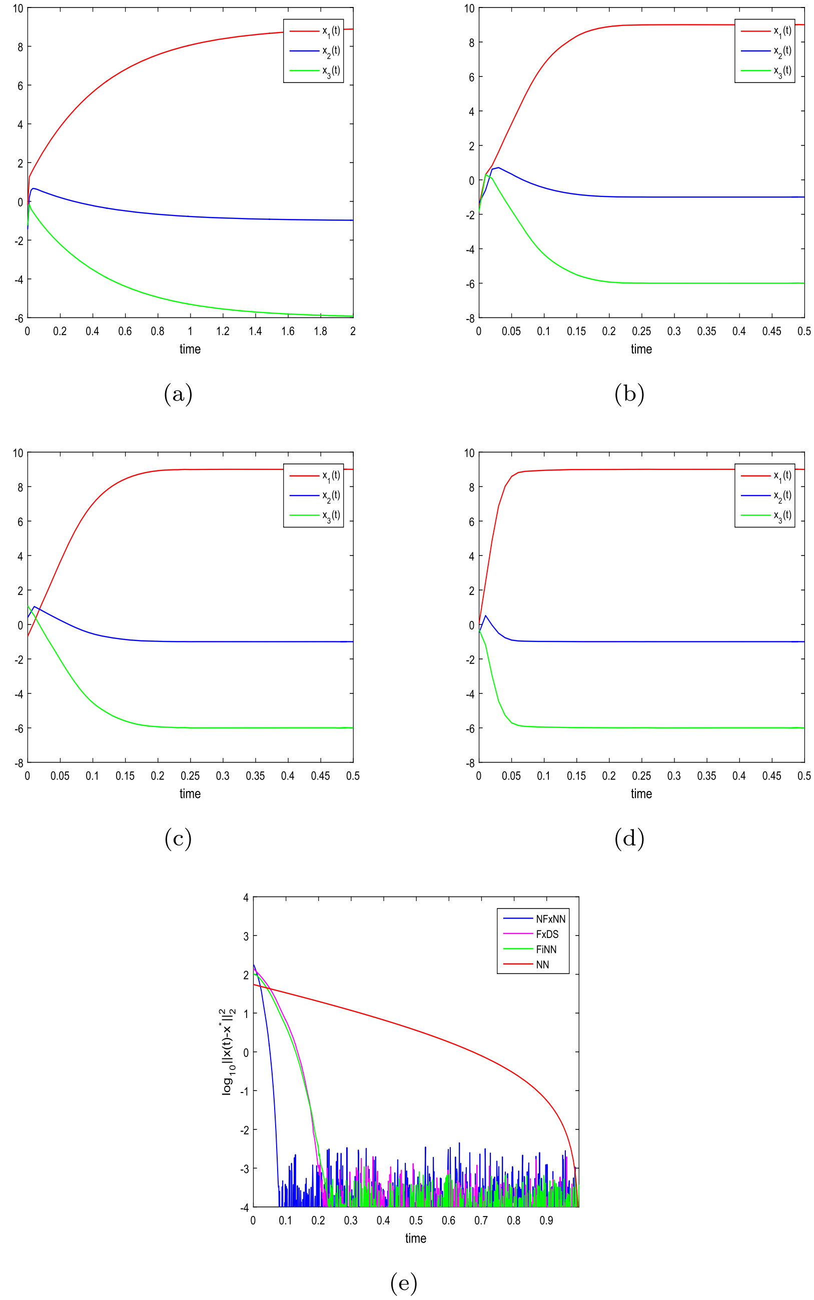
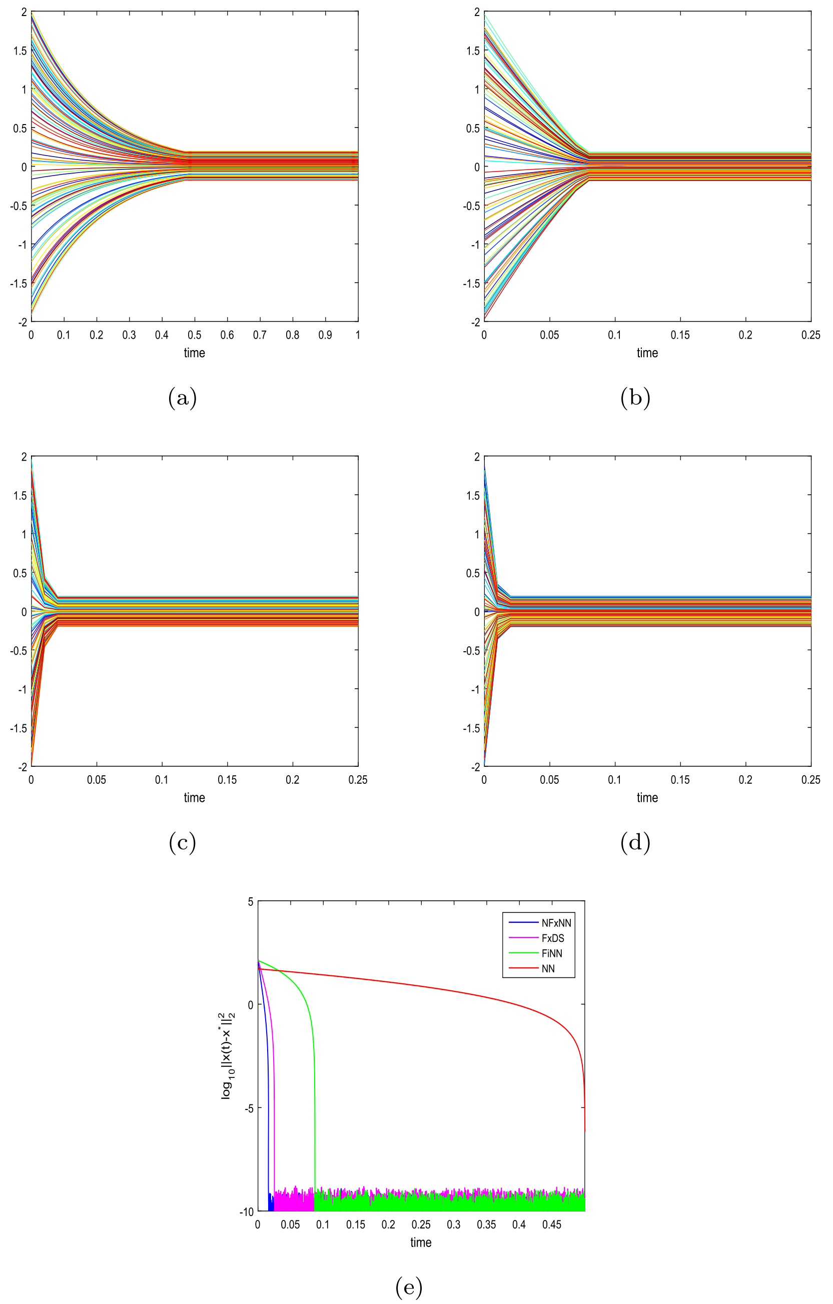


Fig. 3 (a) The trajectory of the NN (12) in [17] with initial condition $x_0 = [-3, 5, 2]^T$; (b) The trajectory of the FiNN (13) with initial condition $x_0 = [-3, 5, 2]^T$; (c) The trajectory of the FxDS (14) with initial condition $x_0 = [-3, 5, 2]^T$; (d) The trajectory of the NFxNN (10) with initial condition $x_0 = [-3, 5, 2]^T$; (e) Convergence errors $\log_{10} \|x(t) - x^*\|_2^2$ for the neural network (12) (red), the FiNN (13) (green), the FxDS (14) (purple) and the NFxNN (10) (blue)

is much more than 0.45 second. So, the convergence rate of the NFxNN (10) is quicker than that of the FxDS (14), FiNN (13), and NN (12). The difference in the convergence rates of the NFxNN (10), FxDS (14), FiNN (13), and NN (12) is also shown from the convergence responses of the errors in Fig. 3 (d). Figure 4 means that the convergence rate of the NFxNN (10) is faster than that of NN (12) in [17] and the FxDS (14).



(a)

(b)

(c)

(d)

(e)

Fig. 4 (a) The trajectory of the NN (12) in [17]; (b) The trajectory of the FiNN (13); (c) The trajectory of the FxDS (14); (d) The trajectory of the NFxNN (10); (e) Convergence responses of the errors $\log_{10}\|x(t)-x^*\|_2^2$ for the NN (12) (red), the FiNN (13) (green), the FxDS (14) (purple) and the NFxNN (10) (blue)

Taking Example 4 as an example, we explore the influence of parameters on the NFxNN method. Figure 5 illustrates the influence of parameters γ, α_1, and α_2 on the NFxNN (10), i.e., it means that the convergence time of the NFxNN (10) is obtained much less as γ increase. When α_1 tends to 0, the NFxNN (10) can achieve the solution faster. The NFxNN (10) can converge to the solution faster while α_2 tends to 1.

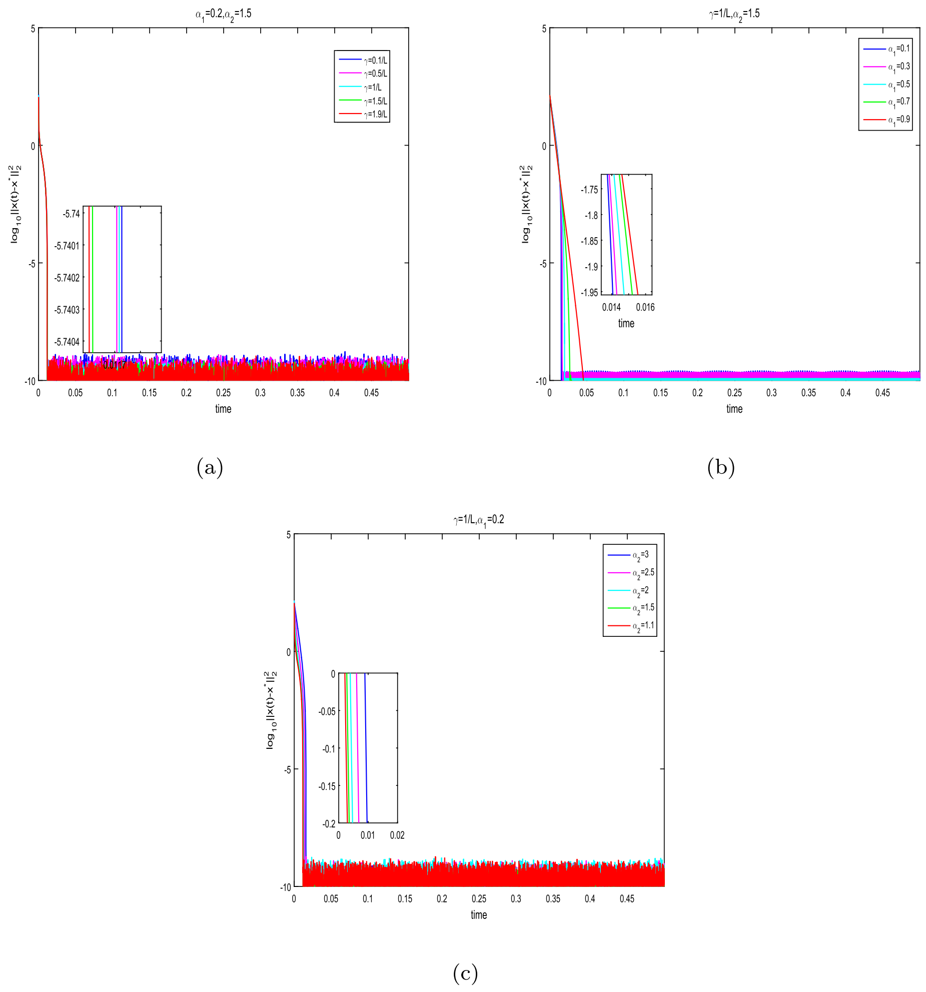


Fig. 5 (a) The NFxNN (10) method plot of $\log_{10}\|x(t)-x^*\|_2^2$ vs. t (time) for various values of γ; ; (b) The NFxNN (10) method plot of $\log_{10}\|x(t)-x^*\|_2^2$ vs. t (time) for various values of α_1; (c) The NFxNN (10) method plot of $\log_{10}\|x(t)-x^*\|_2^2$ vs. t (time) for various values of α_2

5 Conclusions

The NFxNN for solving the split convex feasibility problem is proposed by using the projection method. Under the bound linear regularity assumption, the NFxNN obtains a solution of the SCFP. At the same time, we derive the relationship between the method and the corresponding neural network. We obtain the results of the global fixed-time stability of the NFxNN, i.e., the convergence time of the NFxNN is independent of the initial conditions, and the convergence time is uniformly bounded. The results obtained are different from the relevant neural network methods (such as [17, 30]). Numerical examples also present the superiority and availability of the method.

Acknowledgements
This research was partially supported by the Natural Science Foundation of Chongqing (No. cstc2021jcyj-msxmX0925), and the Education Committee Project Research Foundation of Chongqing (No. KJQN202201802). Jinlan Zheng and Rulan Gan are corresponding authors.

Funding
Not applicable.

Availability of data and materials

All data generated or analysed during this study are included in this published article.

Declarations

Competing interests

The authors declare no competing interests.

Author contributions

JZ: Investigation, Methodology, Writing - original draft. RG: Conceptualization, Software, review & editing. XJ: Methodology & editing. XO: Methodology. All authors read and approved the final manuscript.

Author details

[1]School of Mathematics and Statistics, Southwest University, 400715 Chongqing, China. [2]School of Mathematics and Statistics Northeast Normal University, 130024 Changchun, China. [3]College of Computer and Information Science, Southwest University, 400715 Chongqing, China. [4]College of Electronics and Information Engineering, Sichuan University, Chengdu, China. [5]College of Management, Chongqing College of Humanities, Science & Technology, 401524 Chongqing, China.

Received: 8 March 2023 Accepted: 9 October 2023 Published online: 27 October 2023

References

1. Censor, Y., Elfving, T.: A multiprojection algorithm using Bregman projections in a product space. Numer. Algorithms **8**(2), 221–239 (1994)
2. He, H., Ling, C., Xu, H.-K.: An implementable splitting algorithm for the ℓ_1-norm regularized split feasibility problem. J. Sci. Comput. **67**(1), 281–298 (2016)
3. Byrne, C.: A unified treatment of some iterative algorithms in signal processing and image reconstruction. Inverse Probl. **20**(1), 103–120 (2003)
4. Censor, Y., Bortfeld, T., Martin, B., Trofimov, A.: A unified approach for inversion problems in intensity-modulated radiation therapy. Phys. Med. Biol. **51**(10), 2353 (2006)
5. Wang, J., Hu, Y., Li, C., Yao, J.-C.: Linear convergence of CQ algorithms and applications in gene regulatory network inference. Inverse Probl. **33**(5), 055017 (2017)
6. Bnouhachem, A., Noor, M.A., Khalfaoui, M., Zhaohan, S.: On descent-projection method for solving the split feasibility problems. J. Glob. Optim. **54**(3), 627–639 (2012)
7. Moudafi, A., et al.: Alternating CQ-algorithm for convex feasibility and split fixed-point problems. J. Nonlinear Convex Anal. **15**(4), 809–818 (2014)
8. Gibali, A., Mai, D.T., et al.: A new relaxed CQ algorithm for solving split feasibility problems in Hilbert spaces and its applications. J. Ind. Manag. Optim. **15**(2), 963 (2019)
9. Dang, Y.-Z., Sun, J., Zhang, S.: Double projection algorithms for solving the split feasibility problems. J. Ind. Manag. Optim. **15**(4), 2023 (2019)
10. Dang, Y.-Z., Xue, Z.-H., Gao, Y., Li, J.-X.: Fast self-adaptive regularization iterative algorithm for solving split feasibility problem. J. Ind. Manag. Optim. **16**(4), 1555 (2020)
11. Moudafi, A.: A semi-alternating algorithm for solving nonconvex split equality problems. Numer. Funct. Anal. Optim., 1–10 (2021)
12. Pyne, I.B.: Linear programming on an electronic analogue computer, transactions of the American Institute of Electrical Engineers, Part I. Commun. Electron. **75**(2), 139–143 (1956)
13. Xia, Y., Leung, H., Wang, J.: A projection neural network and its application to constrained optimization problems. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **49**(4), 447–458 (2002)
14. Attouch, H., Bolte, J., Redont, P., Teboulle, M.: Singular Riemannian barrier methods and gradient-projection dynamical systems for constrained optimization. Optimization **53**(5–6), 435–454 (2004)
15. Effati, S., Ghomashi, A., Nazemi, A.: Application of projection neural network in solving convex programming problems. Appl. Math. Comput. **188**(2), 1103–1114 (2007)
16. Tan, Z., Hu, R., Fang, Y.: A new method for solving split equality problems via projection dynamical systems. Numer. Algorithms **86**(4), 1705–1719 (2021)
17. Tan, Z.-Z., Hu, R., Zhu, M., Fang, Y.-P.: A dynamical system method for solving the split convex feasibility problem. J. Ind. Manag. Optim. **17**(6), 2989 (2021)
18. Xia, Y., Wang, J.: A recurrent neural network for solving linear projection equations. Neural Netw. **13**(3), 337–350 (2000)
19. Hu, X., Wang, J.: Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. IEEE Trans. Neural Netw. **17**(6), 1487–1499 (2006)
20. Liu, Q., Wang, J.: A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints. IEEE Trans. Neural Netw. Learn. Syst. **24**(5), 812–824 (2013)
21. Eshaghnezhad, M., Effati, S., Mansoori, A.: A neurodynamic model to solve nonlinear pseudo-monotone projection equation and its applications. IEEE Trans. Cybern. **47**(10), 3050–3062 (2016)
22. Qu, B., Wang, C., Xiu, N.: Analysis on Newton projection method for the split feasibility problem. Comput. Optim. Appl. **67**(1), 175–199 (2017)
23. Zhou, D., Sun, S., Teo, K.L.: Guidance laws with finite time convergence. J. Guid. Control Dyn. **32**(6), 1838–1846 (2009)
24. Polyakov, A.: Nonlinear feedback design for fixed-time stabilization of linear control systems. IEEE Trans. Autom. Control **57**(8), 2106–2110 (2011)
25. He, X., Wen, H., Huang, T.: A fixed-time projection neural network for solving ℓ_1-minimization problem. IEEE Trans. Neural Netw. Learn. Syst. **33**(12), 7818–7828 (2021)

26. Garg, K., Baranwal, M., Gupta, R., Benosman, M.: Fixed-time stable proximal dynamical system for solving MVIPs. IEEE Trans. Autom. Control (2022). https://doi.org/10.1109/TAC.2022.3214795
27. Ju, X., Li, C., Han, X., He, X.: Neurodynamic network for absolute value equations: a fixed-time convergence technique. IEEE Trans. Circuits Syst. II, Express Briefs **69**(3), 1807–1811 (2021)
28. Ju, X., Li, C., Che, H., He, X., Feng, G.: A proximal neurodynamic network with fixed-time convergence for equilibrium problems and its applications. IEEE Trans. Neural Netw. Learn. Syst. (2022). https://doi.org/10.1109/TNNLS.2022.3144148
29. Zheng, J., Chen, J., Ju, X.: Fixed-time stability of projection neurodynamic network for solving pseudomonotone variational inequalities. Neurocomputing **505**, 402–412 (2022)
30. Liu, K., Che, H., Li, M.: A dynamical system with fixed-time convergence for solving the split feasibility problem (2022). https://doi.org/10.21203/rs.3.rs-2033411/v1
31. Ju, X., Li, C., Dai, Y.-H., Chen, J.: A new dynamical system with self-adaptive dynamical stepsize for pseudomonotone mixed variational inequalities. Optimization, 1–30 (2022)
32. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, vol. 408. Springer, Berlin (2011)
33. Bhat, S.P., Bernstein, D.S.: Finite-time stability of continuous autonomous systems. SIAM J. Control Optim. **38**(3), 751–766 (2000)
34. Byrne, C.: Iterative oblique projection onto convex sets and the split feasibility problem. Inverse Probl. **18**(2), 441 (2002)
35. Teschl, G.: Ordinary Differential Equations and Dynamical Systems, vol. 140. American Mathematical Soc. (2012)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.