EFFICIENT OPTIMIZATION FOR EQUILIBRIUM PROBLEMS APPLIED TO DATA CLASSIFICATION

WATCHARAPORN YAJAI

A Dissertation Submitted to University of Phayao
in Partial Fulfillment of the Requirements
for the Doctor of Philosophy in Mathematics
February 2025
Copyright 2025 by University of Phayao

Thesis

Title

Efficient optimization algorithms for equilibrium problems applied to data classification

Submitted by Watcharaporn Yajai

Approved in partial fulfillment of the requirements for the

Doctor of Philosophy in Mathematics

University of Phayao

Suthep Suantai ....Chairman

(Professor Dr.Suthep Suantai)

Rabian Wangkeeree ....Committee

(Professor Dr.Rabian Wangkeeree)

W. Cholamjiak ....Committee

(Associate Professor Dr.Watcharaporn Cholamjiak)

Tanakit Thianwan ....Committee

(Associate Professor Dr.Tanakit Thianwan)

P. Cholamjiak ....Committee

(Associate Professor Dr.Prasit Cholamjiak)

D. Yambangwai ....Committee

(Associate Professor Dr.Damrongsak Yambangwai)

Approved by

Sitthisak Pinmongkhonkul ....

(Associate Professor Dr.Sitthisak Pinmongkhonkul)

Dean of School of Science

February 2025

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my thesis advisors, Associate Professor Dr. Watcharaporn Cholamjiak and Associate Professor Dr. Prasit Cholamjiak, for their invaluable guidance, motivation, and immense knowledge. Their unwavering support throughout my Ph.D. study and related research has been instrumental to my success. I am deeply grateful for the inspiration, scientific ideas, and the encouragement they provided, allowing me to explore and implement my own ideas. Their assistance, warm relationships, and significant care played a crucial role in the completion of this study.

I am sincerely thankful to Assistant Professor Dr. Yann Savoye for his care, guidance, and academic support during my research period at Liverpool John Moores University, England. I extend my profound gratitude to Professor Dr. Choonkil Park and Professor Dr. Yeol Je Cho, along with their families, for their exceptional care and support during my research time in Korea. Their kindness and academic assistance have left a lasting impact on my professional growth.

I deeply appreciate my dissertation and thesis committees: Professor Dr. Suthep Suantai, Professor Dr. Rabian Wangkeeree, Associate Professor Dr. Tanakit Thianwan, and Associate Professor Dr. Damrongsak Yambangwai, for their invaluable mentorship and support throughout my Ph.D. journey.

Lastly, I am deeply grateful to my family, friends, and colleagues for their constant support and encouragement, which drove my accomplishments. I also thank myself for the resilience and confidence to persevere through this challenging journey.

Watcharaporn Yajai

เรื่อง: อัลกอริทึมการหาค่าเหมาะที่สุดที่มีประสิทธิภาพสำหรับปัญหาค่าสมดุลประยุกต์ใช้กับการจำแนกข้อมูล

ผู้วิจัย: วัชราภรณ์ ยาใจ วิทยานิพนธ์: ปร.ด. (คณิตศาสตร์) มหาวิทยาลัยพะเยา 2568

ประธานที่ปรึกษา: รองศาสตราจารย์ ดร.วัชรภรณ์ ช่อลำเจียก

กรรมการที่ปรึกษา: รองศาสตราจารย์ ดร.ธนกฤต เทียนหวาน รองศาสตราจารย์ ดร.ประสิทธิ์ ช่อลำเจียก

คำสำคัญ: ปัญหาค่าสมดุล ปริภูมิฮิลเบิร์ต เทคนิคเฉื่อย การจำแนกข้อมูล

บทคัดย่อ

ปัญหาค่าสมดุลนั้น เป็นแนวคิดที่กว้างขวาง และมีความครอบคลุม สามารถประยุกต์ใช้ได้กับปัญหาหลายประเภท เช่น ปัญหาอสมการการแปรผัน ปัญหาจุดอานม้า ปัญหาการหาค่าต่ำที่สุดแบบมีเงื่อนไข ปัญหาความสมบูรณ์ และปัญหาค่าสมดุลของแนช ปัญหาค่าสมดุลเป็นแนวคิดพื้นฐานที่มีความสำคัญในหลากหลายสาขาวิชา เช่น ฟิสิกส์ เคมี ชีววิทยา และเศรษฐศาสตร์ โดยสะท้อนถึงสภาวะที่แรงตรงข้ามอยู่ร่วมกันได้ เพื่อรักษาความเสถียร และความสมดุลในระบบนั้นๆ นอกจากนี้ยังมีบทบาทสำคัญในการแก้ปัญหาในโลกแห่งความเป็นจริง เช่น การกู้คืนสัญญาณ การประมวลผลภาพ และการเรียนรู้ของเครื่อง จุดมุ่งหมายของวิทยานิพนธ์นี้ คือ การสร้างอัลกอริทึมที่มีประสิทธิภาพ 5 แบบสำหรับแก้ปัญหาค่าสมดุล ดังนี้ (1) วิธีการปรับปรุงแบบเอ็กซ์ตร้าเกรเดียนต์เชิงหนืดที่เสริมด้วยเทคนิคเฉื่อยสำหรับการแก้ปัญหาค่าสมดุลเพื่อประยุกต์ใช้กับการจำแนกโรคเบาหวานในวิธีการเรียนรู้ของเครื่อง (2) อัลกอริทึมการปรับปรุงแบบซับเกรเดียนต์เอ็กซ์ตร้าเกรเดียนต์เชิงหนืดที่เสริมด้วยเทคนิคเฉื่อยสำหรับการแก้ปัญหาค่าสมดุลที่ไม่เป็นโมโนโทนเพื่อประยุกต์ใช้ในการจำแนกโรคหัวใจและหลอดเลือด (3) อัลกอริทึมมานน์แบบฉายที่เสริมด้วยเทคนิคเฉื่อยสำหรับการแก้ปัญหาค่าสมดุลแยกส่วนเพื่อประยุกต์ใช้ในการจำแนกโรคพาร์กินสัน (4) อัลกอริทึมแบบหนืดที่เสริมด้วยเทคนิคเฉื่อยผ่อนปรนสองขั้นรูปแบบใหม่สำหรับการแก้ปัญหาค่าสมดุลแยกส่วนเพื่อประยุกต์ใช้ในการจำแนกโรคกระดูกพรุน (5) อัลกอริทึมมานน์ที่เสริมด้วยเทคนิคเฉื่อยสองขั้นสำหรับการแก้ปัญหาค่าสมดุลแยกส่วนเพื่อประยุกต์ใช้ในการตรวจคัดกรองมะเร็งเต้านม โดยทฤษฎีบทการลู่เข้าของอัลกอริทึมที่เสนอทั้งหมดได้ถูกพิสูจน์ภายใต้เงื่อนไขที่เหมาะสมในปริภูมิฮิลเบิร์ต

Title: EFFICIENT OPTIMIZATION ALGORITHMS FOR EQUILIBRIUM PROBLEM APPLIED TO DATA CLASSIFICATION

Author: Watcharaporn Yajai, Thesis: Ph.D. (Mathematics), University of Phayao, 2025

Advisor: Associate Professor Dr.Watcharaporn Cholamjiak

Co-advisor: Associate Professor Dr.Tanakit Thianwan, Associate Professor Dr.Prasit Cholamjiak

Keywords: equilibrium problem, Hilbert space, inertial technique, data classification

ABSTRACT

Equilibrium problems is very general in the sense that it includes, as special cases, the variational inequality problem, saddle point problems, constrained minimization, complementarity problem, as well as the Nash equilibrium problem. Equilibrium is a fundamental concept across disciplines such as physics, chemistry, biology, and economics, representing a state of balance where opposing forces coexist in harmony to maintain stability. It also plays a critical role in solving real-world problems, including signal recovery, image processing, and machine learning. Our aim in this thesis is to construct five new efficient algorithms for solving equilibrium problems as follows: (i) A modified inertial viscosity extragradient type method for equilibrium problems application to classification of diabetes mellitus: Machine learning methods (ii) A modified viscosity type inertial subgradient extragradient algorithm for nonmonotone equilibrium problems and application to cardiovascular disease detection (iii) An inertial projective Mann algorithm for solving split equilibrium problems classification to Parkinson's disease (iv) A new double relaxed inertial viscosity type algorithm for solving split equilibrium problems application to osteoporosis detection (v) A double inertial Mann algorithm for split equilibrium problems application to breast cancer screening. Under some suitable conditions in Hilbert spaces, the convergence theorems of the proposed algorithms are proved.

LIST OF CONTENTS

มหาวิทยาลัยพะเยา
UNIVERSITY OF PHAYAO

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

Figures **Page**

CHAPTER 1

INTRODUCTION

Equilibrium problems (EP) was first invented by Ky Fan (1972) [43], which is also often called the Ky Fan inequality. This general problem was named the "equilibrium problems" by Blum and Oettli (1994) [18], who stressed this unifying feature and provided a thorough investigation of its theoretical properties. At that time, this format did not receive much attention, and a few mathematicians were interested. One is Nikaido and Isoda (1955) [94] characterized Nash equilibria as the solutions of (EP) for an appropriate auxiliary bifunction but they did not consider the problem itself. Gwinner (1983) [57] introduced it just as a tool to develop a unified treatment of penalization techniques for optimization and variational inequalities. Antipin (1990) [6] formulated an inverse optimization problem as a noncooperative game and therefore in the (EP) format via the Nikaido-Isoda bifunction and provided a solution method for the general problem in [4, 6].

Indeed, equilibrium problems started to gain real interest only after the publication of the seminal paper of Blum and Oettli (1994) [18]. Actually, the possibility to exploit results and algorithms developed for one class of problems in another framework was not a novelty at all: this kind of bridge already finds roots in the analytical development of variational inequalities through the connection with optimization via complementarity problems. Anyway, a large number of applications has been described successfully via the concept of equilibrium solution and therefore many researchers devoted their efforts to studying (EP). In fact, nowadays there is a good theory for equilibria and a rapidly increasing number of algorithms for finding them.

This problem is very general in the sense that it includes, as special cases, the variational inequality problem, fixed point problem, saddle point problems, constrained minimization, complementarity problem, optimization problem as well as the Nash equilibrium problem, see, e.g., in Blum and Oettli (1994) [18], Muu and Oettli (1992) [90]. Equilibrium problems theory provides us with a unified, natural, innovative and general framework to study a wide class of problems arising in physics, chemistry, engineering, finance, economics, network analysis, transportation, elasticity, optimization, image restoration, signal recovery. For instance, it may refer to physical or mechanical structures, chemical processes, the distribution of traffic over computer and telecommunication networks or over public roads (see, for in- stance, [16, 34, 44, 91, 96, 109, 120]). In economics it often refers to production competition or the dynamics of offer and demand [7], exploiting the mathematical model of noncooperative games and the corresponding equilibrium concept by Nash [92, 93]. This theory has witnessed an explosive growth in theoretical advances and applications across all disciplines of pure and applied sciences. In recent years, several numerical techniques including projection, resolvent and auxiliary principle have been developed and analyzed for solving variational inequalities, which can be solved by the problem (EP), see [2], [4], [5], [7], [9]-[18], [49]-[54], [72], [93]. Many solving methods have been proposed for approximating a solution of problem (EP), for example, the proximal point methods (Konnov 2003 [71]; Moudafi 1999 [86]), the extragradient methods (or the proximal-like methods; Anh 2013 [3]; Flam and Antipin 1997 [45]; Hieu 2017d; Quoc et al. 2008 [99]; Korpelevich 1976 [72]; Tran et al 2008 [119]), the hybrid methods (Hieu et al. 2016 [60]; Hieu 2017b [61], c; Vuong et al. 2012), the projected subgradient methods (Hieu 2017a [62], 2018; Santos and Scheimberg 2011 [104]), halpern subgradient extragradient method (Kraikaew and Saejung [73]; Hieu [63]), viscosity-type method with the extragradient (Muangchoo [89]), inertial extragradient method (Shehu et al. [107]).

The aim of this work is to design new efficient optimization algorithms for solving equilibrium problems. We prove the convergence theorems under some suitable conditions in Hilbert spaces. Finally, we apply the proposed algorithm to solve data classification in machine learning.

มหาวิทยาลัยพะเยา UNIVERSITY OF PHAYAO

CHAPTER 2

REVIEW OF RELATED LITERATURE AND RESEARCH

Let H be a real Hilbert space and C be a nonempty closed and convex subset of a real Hilbert space H. We denote $\mathbb{R}$ for a real number set. The equilibrium problem (EP) is to find an element $x^* \in C$ introduced by Ky Fan [43] such that

$$f(x^*, y) \geqslant 0, \ \forall y \in C, \tag{2.1.1}$$

where $f : C \times C \to \mathbb{R}$ is a bifunction and satisfying $f(z, z) = 0$ for all $z \in H$, and $EP(f, C)$ is denoted for a solution set of EP(2.1.1). EP(2.1.1) generalizes many various problems in optimization problems such as variational inequalities problems, fixed point problems, Nash equilibrium problems, linear programming problems, among others.

Many real-world problems can be solved via formulation in the EP(2.1.1). To address such problems, various algorithms have been proposed. One of the famous algorithms, introduced by Korpelevich [72], is the Extragradient Method (EM), solving the variational inequality problem which is a special case of equilibrium problems. Korpelevich proved the weak convergence of the generated sequence under the assumptions of Lipschitz continuity and monotonicity. This method was generated by $x_0 \in H$ and,

$$\begin{cases} y_k = \underset{y \in C}{\operatorname{argmin}}\{\lambda f(x_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \\ x_{k+1} = \underset{y \in C}{\operatorname{argmin}}\{\lambda f(y_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \ \forall k \geqslant 0, \end{cases}$$

where $\lambda \subset (0, \frac{1}{L})$ with L is the Lipschitz-type constants. The EM has two major limitations. The first is that the step-size is depends on the Lipschitz constant and the second is that, calculation of two projections are involved. These limitations affect the computational efficiency of the method. In other to avoid these setbacks, many methods have been studied in the last few decades (see in [27, 28, 53]).

Building upon the idea of the extragradient method, Tran et al. [119] introduced the Two-Step Extragradient Method (TSEM). This method extends to the equilibrium problems when the bifunction f is pseudomonotone and satisfies Lipschitz-type continuous conditions with positive constants c_1 and c_2. This method was generated by $x_0 \in C$ and,

$$\begin{cases} y_k = \underset{y \in C}{\operatorname{argmin}}\{\lambda f(x_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \\ x_{k+1} = \underset{y \in C}{\operatorname{argmin}}\{\lambda f(y_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \ \forall k \geqslant 0, \end{cases}$$

where $0 < \lambda < \min\{\frac{1}{2c_1}, \frac{1}{2c_2}\}, c_1, c_2 = \frac{L}{2}$, and λ is some constant depending on the Lipschitz constant of the involved bifunction. The TSEM is an extension of the EM to increase the ability to solve more complex problems, especially in cases where the EM is non-monotonicity bifunction. The EM was able to replace the strong monotonicity assumption on f by a weaker assumption called pseudo-monotonicity, that is the TSEM.

One of the drawbacks of the Extragradient Method and the Two-Step Extragradient Method is the necessity of two projections onto the set C in each iterate. It is not easy to compute when the structure of the set C is complicated. Therefore, Censor et al. [27] modified the EM to the Subgradient Extragradient Method (SEM), replacing two projections onto C with one projection onto C and one onto a half-space and allows a clear computation. This method is defined by

$x_0 \in H$ and

$$\begin{cases} y_k &= \underset{y\in C}{\operatorname{argmin}}\{\lambda f(x_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \\ T_k &= \{v \in H : \langle x_k - \lambda t_k - y_k, v - y_k\rangle \leqslant 0\},\ t_k \in \partial_2 f(x_k, y_k), \\ x_{k+1} &= \underset{y\in T_k}{\operatorname{argmin}}\{\lambda f(y_k, y) + \frac{1}{2}\|x_k - y\|^2\},\ \forall k \geqslant 0, \end{cases}$$

where $\lambda \subset (0, \frac{1}{L})$ with L-Lipschitz constant and f is monotone bifunction.

One of the famous strongly convergent algorithm is the Halpern iteration [58]. Using the idea of this algorithm, Hieu [63] proposed the Halpern Subgradient Extragradient Method (HSEM) which was modified from the HSEM of Kraikaew and Saejung [73] for variational inequalities. This method is defined by $u, x_0 \in H$ and

$$\begin{cases} y_k &= \underset{y\in C}{\operatorname{argmin}}\{\lambda f(x_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \\ T_k &= \{v \in H : \langle x_k - \lambda t_k - y_k, v - y_k\rangle \leqslant 0\},\ t_k \in \partial_2 f(x_k, y_k), \\ z_k &= \underset{y\in T_k}{\operatorname{argmin}}\{\lambda f(y_k, y) + \frac{1}{2}\|x_k - y\|^2\}, \\ x_{k+1} &= \alpha_k u + (1-\alpha_k) z_k,\ \forall k \geqslant 0, \end{cases} \tag{2.1.2}$$

where λ is still some constant depending on the interval that makes the bifunction f satisfies the Lipschitz condition and $\{\alpha_k\} \subset (0, 1)$ which satisfies the principle conditions

$$\lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=1}^{+\infty} \alpha_k = +\infty.$$

Another famous strongly convergent algorithm is the Viscosity Method [86], Muangchoo [89] modified HSEM to Viscosity-Type Subgradient Extragradient Method (VSEM), which was generated by $x_1 \in H$. This method is defined by

$$\begin{cases} y_k &= \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(x_k, y) + \frac{1}{2}\|x_k - y\|^2\},\\ T_k &= \{v \in H : \langle x_k - \lambda_k t_k - y_k, v - y_k\rangle \leqslant 0\},\ t_k \in \partial_2 f(x_k, y_k),\\ z_k &= \underset{y\in T_k}{\operatorname{argmin}}\{\mu\lambda_k f(y_k, y) + \frac{1}{2}\|x_k - y\|^2\},\\ x_{k+1} &= \alpha_k V(x_k) + (1-\alpha_k)z_k, \end{cases} \tag{2.1.3}$$

where $\mu \in (0, \sigma) \subset \left(0, \min\left\{1, \frac{1}{2c_1}, \frac{1}{2c_2}\right\}\right)$, V is a contraction function on H with contraction constant $\alpha \in (0, 1)$, $(\|V(x) - V(y)\| \leqslant \alpha\|x - y\|,\ \forall x, y \in H)$, $\{\alpha_k\}$ satisfies the principle conditions $\lim\limits_{k\to\infty} \alpha_k = 0$ and $\sum\limits_{k=1}^{+\infty} \alpha_k = +\infty$, and the step-sizes $\{\lambda_k\}$ is developed by updating the step-sizes method without knowing the Lipschitz-type constants of the bifunction f which satisfies the following:

$$\lambda_{k+1} = \begin{cases} \min\{\sigma, \frac{\mu f(y_k, z_k)}{\mathcal{S}_k}\}, & \text{if } \frac{\mu f(y_k, z_k)}{\mathcal{S}_k} > 0,\\ \lambda_0, & \text{otherwise,} \end{cases}$$

where $\mathcal{S}_k = f(x_k, z_k) - f(x_k, y_k) - c_1\|x_k - y_k\|^2 - c_2\|z_k - y_k\|^2 + 1$.

One technique to speed up the convergence of an algorithm is the well-known inertial technique, which appeared in the heavy ball method introduced by Polyak [98] in 1964. The algorithm was generated by $x_0, x_1 \in H$, $r_k > 0$ and

$$x_{k+1} = x_k + \theta_k(x_k - x_{k-1}) - r_k \nabla F(x_k),\ \forall k \in \mathbb{N}, \tag{2.1.4}$$

where $F : H \to H$ is differentiable and $\{\theta_k\} \subset [0, 1)$ is the extrapolation coefficient of the inertial step $\theta_k(x_k - x_{k-1})$. Later, the inertial technique was used to modify the algorithm to speed up the convergence of the algorithms by many mathematicians [8, 20, 80, 97]. The inertial techniques have been proposed for solving the equilibrium problems, see in [64, 87].

Finding technique to speed up the convergence of the algorithm is the way that many mathematicians are interested. One of that is an inertial which was first introduced by Polyak [98]. Very recently, Shehu et al. [107] modified the inertial technique with the Halpern-type algorithm and subgradient extragradient method for obtaining strong convergence to a solution of $EP(f, C)$ such that f is pseudomonotone. This method is defined by $u \in H$ and

$$\begin{cases} w_k &= \alpha_k u + (1-\alpha_k)x_k + \delta_k(x_k - x_{k-1}), \\ y_k &= \underset{y\in C}{\operatorname{argmin}} \left\{ \lambda_k f(w_k, y) + \frac{1}{2}\|w_k - y\|^2 \right\}, \\ T_k &= \{v \in H : \langle (w_k - \lambda_k t_k) - y_k, v - y_k \rangle\}, \quad t_k \in \partial_2 f(w_k, y_k), \\ z_k &= \underset{y\in T_k}{\operatorname{argmin}} \left\{ \lambda f(y_k, y) + \frac{1}{2}\|w_k - y\|^2 \right\}, \\ x_{k+1} &= \tau w_k + (1-\tau) z_k, \end{cases} \tag{2.1.5}$$

where the inertial parameter $\{\delta_k\} \subset [0, \frac{1}{3})$, $\tau \in (0, \frac{1}{2}]$, the update step-size $\{\lambda_k\}$ satisfies the following:

$$\lambda_{k+1} = \begin{cases} \min\{\frac{\mu}{2}\frac{\|w_k - y_k\|^2 + \|z_k - y_k\|^2}{\mathcal{P}_k}, \lambda_k\}, & \text{if } \mathcal{P}_k > 0, \\ \lambda_k, & \text{otherwise,} \end{cases}$$

where $\mathcal{P}_k = f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k)$ and $\{\alpha_k\}$ still satisfies the principle conditions $\lim_{k\to\infty} \alpha_k = 0$ and $\sum_{k=1}^{+\infty} \alpha_k = +\infty$.

Nowadays, mathematicians have created many methods. In 2024, Yao et al. [124] presented two computationally efficient proximal-type algorithms to solve equilibrium problems with pseudo-monotone bifunction in Hilbert spaces. The first algorithm converges weakly and obtains a linear rate of convergence. The second method was designed as a viscosity version of the first algorithm and obtained strong convergence results. This method is defined by $\lambda_1 > 0,\ w_0, y_0 \in$

$H,\ x_1 = x_0 \in C$ as follows:

$$\begin{cases} y_k &= x_k + \eta(w_{k-1} - x_k), \\ z_k &= y_k + \delta_k(y_k - y_{k-1}), \\ x_{k+1} &= \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(x_k, y) + \frac{1}{2}\|y - z_k\|^2\}, \end{cases}$$

where $\sum_{k=1}^{\infty} t^k < \infty, \mu \in (0,1), \theta \in (0,1), -1 < \delta_k \leqslant \delta_{k+1} \leqslant 0$ and $\eta \in [\frac{(\sqrt{5}-1)}{2}, 1)$ with stepsizes

$$\lambda_{k+1} = \begin{cases} \lambda_k + t_k, & \text{if } \mathcal{O}_k \leqslant 0, \\ \min\left\{\frac{\mu\theta}{4}\frac{(\|x_k - x_{k+1}\|^2 + \|x_k - x_{k-1}\|^2)}{\mathcal{O}_k}, \lambda_k + t_k\right\}, & \text{otherwise,} \end{cases}$$

where $\mathcal{O}_k = f(x_{k-1}, x_{k+1}) - f(x_{k-1}, x_k) - f(x_k, x_{k+1})$.

Due to the numerous applications of the theory of the EP(2.1.1), many authors have extended and generalized it in various directions. For instance, the split equilibrium problem (SEP), the SEP enables us to split the solution between two different subsets of spaces. We assume that H_1 and H_2 are real Hilbert spaces. Let C and Q be nonempty subsets of H_1 and H_2, respectively. Given $f : C \times C \to \mathbb{R}$ and $g : Q \times Q \to \mathbb{R}$ be two bifunctions. Suppose $A : H_1 \to H_2$ be a bounded linear operator. In 2011, the spilt equilibrium problems (SEP) introduced by Moudafi [88] is to find $x^* \in C$ such that

$$f(x^*, y) \geqslant 0,\ \forall y \in C \tag{2.1.6}$$

and such that

$$\mu^* = Ax^* \in Q \text{ solves } g(\mu^*, z) \geqslant 0,\ \forall z \in Q. \tag{2.1.7}$$

We denote problem (2.1.6) is the classical equilibrium problem by solution set

$EP(f)$. Denote problem (2.1.7) is solution set by $EP(g)$. The solution set of the split equilibrium problem (2.1.6) and (2.1.7) denoted by $\omega = \{x \in EP(f) : Ax \in EP(g)\}$.

To solve the split equilibrium problems, He [59] proposed the following proximal point method, when f and g are monotone. He showed that the generated sequence converges weakly to a solution of the SEP under some certain conditions on the control parameters. This method was defined as follows: select $x_0 \in C$ and

$$\begin{cases} y_k \in C \text{ such that } (y_k, y) + \frac{1}{r_k}\langle y - y_k, y_k - x_k\rangle \geqslant 0, \ \forall y \in C, \\ u_k \in Q \text{ such that } g(u_k, z) + \frac{1}{r_k}\langle z - u_k, u_k - Ay_k\rangle \geqslant 0, \ \forall z \in Q, \\ x_{k+1} = P_C(y_k + \beta A^T(u_k - Ay_k), \ \forall k \geqslant 0, \end{cases}$$

where $\beta \subset (0, \frac{1}{\|A\|^2})$, $\{r_k\} \subset (0, +\infty)$ with $\liminf_{k\to\infty} r_k > 0$, and P_C s a projection operator from H_1 into C. He obtained the converges weakly to a solution of the split equilibrium problems. Here, the algorithm of He [59] will be called the PPA Algorithm.

In recent years, many authors have made several efforts to develop implementable iterative methods for solving all these problems. In 2016, Suantai et al. [114] propose an iterative algorithm for solving common solution of fixed point problem of nonspreading multi-valued mapping and the SEP as follows:

$$\begin{cases} x_1 \in C, \\ z_k = T^f_{r_k}(x_k - \beta A^T(I - T^g_{r_k})Ax_k), \\ x_{k+1} = \alpha_k x_k + (1 - \alpha_k)Wz_k, \ \forall k \geqslant 1, \end{cases} \tag{2.1.8}$$

where I is an identity mapping, W is a $\frac{1}{2}-$nonspreading multivalued mapping ($W : C \to K(C)$ be a k-nonspeading multi-valued mapping such that $k \in (0, \frac{1}{2}]$), $I - T^g_{r_k}$ is 1-inverse strongly monotone, $\beta \subset (0, \frac{1}{L})$ such that L is the spectal radius of A^TA and A^T is the adjoint of A, $\{r_k\} \subset (0,\infty)$, and $\{\alpha_k\} \subset (0,1)$. Under the conditions this algorithm get converges weakly to an element of the common solution of split equilibrium problem and fixed point problem.

Recently, Dang [35] propose algorithm to solving spilt equilibrium problem which combines the proximal method, the projection method, and the diagonal subgradient method to converges strongly to solution of SEP, where f is a pseudomonotone bifunction. This algorithm is designed as follows: select $x_0 \in C$, $w_k \in \partial_{\epsilon_k} f(x_k,\cdot)(x_k)$ and

$$\begin{cases} y_k = \max\{\rho_k, \|w_k\|\},\ d_k = \frac{\mu_k}{y_k}, \\ z_k = P_C(x_k - d_k w_k), \\ x_{k+1} = P_C(z_k - \beta_k A^T(I - T^g_{r_k})Az_k), \end{cases}$$

where $\partial_\epsilon f(x,\cdot)(x)$ is ϵ-diagonal subdifferential of f at $x \in C$, $\rho_k \geqslant \rho > 0, \mu_k > 0, \epsilon_k > 0, r_k \geqslant r > 0$, and $0 < a \leqslant \beta_k \leqslant b < \frac{2}{\|A\|^2}$.

In this dissertation, we study two problems: (1) equilibrium problem in which the bifunction is pseudomonotone and (2) split equilibrium problem in which the bifunction is monotone. We focus on using the step size and Lipschitz condition. This flexibility makes it particularly suitable for data classification tasks in machine learning, especially when the dataset has a finite number of features.

CHAPTER 3

PRELIMINARIES

In this section, we provide some basic concepts, definitions, and lemmas that will be used in the following sections.

3.1 Fundamentals

Definition 3.1.1 [1]**(Normed space)** Let X be a vector space over field $\mathbb{S}$ ($\mathbb{R}$ or $\mathbb{C}$) and $\|\cdot\| : X \to [0, \infty)$ be a function. Then $\|\cdot\|$ is said to be a norm if the following properties hold: for all $x, y \in X$ and $\alpha \in \mathbb{S}$,

1. $\|x\| \geqslant 0$;
2. $\|x\| = 0 \Leftrightarrow x = 0$;
3. $\|\alpha x\| = |\alpha|\|x\|$;
4. $\|x + y\| \leqslant \|x\| + \|y\|$.

$\|x\|$ is called the norm of x. $(X, \|\cdot\|)$ denotes the normed space just defined.

Example 3.1.2 $\mathbb{R}^n$ is a normed space with the following norms:

$$\begin{aligned}
\|x\|_1 &= \sum_{k=1}^{n} |x_k| \text{ for all } x = (x_1, x_2, .., x_n) \in \mathbb{R}^n; \\
\|x\|_p &= \left(\sum_{k=1}^{n} |x_k|^p\right)^{1/p} \text{ for all } x = (x_1, x_2, .., x_n) \in \mathbb{R}^n \text{ and } p \in (1, \infty); \\
\|x\|_\infty &= \max_{1 \leq k \leq n} |x_k| \text{ for all } x = (x_1, x_2, .., x_n) \in \mathbb{R}^n.
\end{aligned}$$

Example 3.1.3 Let $X = \ell_1$, the linear space whose elements consist of all absolutely convergent sequences $(x_1, x_2, ..., x_k, ...)$ of scalars ($\mathbb{R}$ or $\mathbb{C}$),

$$\ell_1 = \{x : x = (x_1, x_2, ..., x_k, ...) \text{ and } \sum_{k=1}^{\infty} |x_k| < \infty\}.$$

Then ℓ_1 is a normed space with the norm defined by $\|x\|_1 = \sum_{k=1}^{\infty} |x_k|$.

Example 3.1.4 Let $X = \ell_p$ $(1 < p < \infty)$ be the linear space whose elements consist of all presumable sequences $(x_1, x_2, ..., x_k, ...)$ of scalars ($\mathbb{R}$ or $\mathbb{C}$),

$$\ell_p = \{x : x = (x_1, x_2, ..., x_k, ...) \text{ and } \sum_{k=1}^{\infty} |x_k|^p < \infty\}.$$

Then ℓ_p is a normed space with the norm defined by $\|x\|_p = (\sum_{k=1}^{\infty} |x_k|^p)^{1/p}$.

Example 3.1.5 Let $X = \ell_\infty$ be the linear space whose elements consist of all bounded sequences $(x_1, x_2, ..., x_k, ...)$ of scalars ($\mathbb{R}$ or $\mathbb{C}$),

$$\ell_\infty = \{x : x = (x_1, x_2, ..., x_k, ...) \text{ and } \{x_k\}_{k=1}^{\infty} \text{ is bounded}\}.$$

Then ℓ_∞ is a normed space with the norm defined by $\|x\|_\infty = \sup_{k \in \mathbb{N}} |x_k|$.

Example 3.1.6 Let $X = L_2[a, b]$ be the linear space of all continuous real-valued functions on $[a, b]$ forms a normed space X with norm defined by

$$\|x\| = \left(\int_a^b x(t)^2 dt \right)^{\frac{1}{2}}.$$

Definition 3.1.7 [1]**(Cauchy sequence)** A sequence $\{x_k\}$ in a normed space X is said to be Cauchy if $\lim_{m,k \to \infty} \|x_m - x_k\| = 0$, i.e., for $\varepsilon > 0$, there exists an integer $n_0 \in \mathbb{N}$ such that $\|x_m - x_k\| < \varepsilon$ for all $m, k \geqslant n_0$.

Definition 3.1.8 [1]**(Convergent sequence)** A sequence $\{x_k\}$ in a normed space X is said to be convergent to x if $\lim_{k \to \infty} \|x_k - x\| = 0$. In this case, we write $x_k \to x$ or $\lim_{k \to \infty} x_k = x$.

Definition 3.1.9 [115]**(Strong convergence)** Let H be an inner product space and let $x \in H$. A sequence $\{x_k\}$ in H is said to be converges strongly to x, denoted by $x_k \to x$, if $\|x_k - x\| \to 0$.

Definition 3.1.10 [74]**(Weak convergence)** A sequence $\{x_k\}$ in a normed space X is said to be weakly convergent if there is an $x \in X$ such that for every f in the dual space X',

$$\lim_{k\to\infty} f(x_k) = f(x).$$

Definition 3.1.11 [1]**(Completeness)** The space X is said to be complete if every Cauchy sequence in X converges strongly.

Example 3.1.12 The Euclidean space $\mathbb{R}^n$ is complete with

$$d(x,y) = \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_n-y_n)^2}$$

where $x=(x_1,x_2,...,x_n)$, $y=(y_1,y_2,...,y_n)\in\mathbb{R}^n$.

Example 3.1.13 The sequence space ℓ_∞ is complete.

Example 3.1.14 The sequence space ℓ_p is complete.

Definition 3.1.15 [1]**(Inner product space)** Let X be a vector space over field $\mathbb{S}$ ($\mathbb{R}$ or $\mathbb{C}$) and $\langle .,. \rangle : X\times X \to \mathbb{S}$ be a function. Then $\langle .,. \rangle$ is said to be an inner product if the following properties hold: for all $x,y\in X$ and $\alpha\in\mathbb{S}$,

1. $\langle x,x\rangle \geqslant 0$;
2. $\langle x,x\rangle = 0 \Leftrightarrow x=0$;
3. $\langle \alpha x,y\rangle = \alpha\langle x,y\rangle$;
4. $\langle x,y\rangle = \overline{\langle y,x\rangle}$;
5. $\langle x+y,z\rangle = \langle x,z\rangle + \langle y,z\rangle$.

$\langle x,y\rangle$ is called the inner product of x and y, and $\overline{\langle y,x\rangle}$ is conjugate symmetry of $\langle x,y\rangle$. $(X,\|\cdot\|)$ denotes the inner product space just defined.

Definition 3.1.16 [1]**(Hilbert space)** An inner product space H is said to be a Hilbert space if it is complete, i.e., every Cauchy sequence is strongly convergent sequence in H.

Example 3.1.17 The Euclidean space $\mathbb{R}^n$ is a Hilbert space with inner product defined by

$$\langle x, y\rangle = \sum_{k=1}^{n} x_k y_k,$$

where $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n) \in \mathbb{R}^n$.

Example 3.1.18 The space l_2 is a Hilbert space with inner product defined by

$$\langle x, y\rangle = \sum_{k=1}^{\infty} x_k \overline{y_k},$$

where $x, y \in l_2$.

Example 3.1.19 The space $L_2[a, b]$ is a Hilbert space with inner product defined by

$$\langle x, y\rangle = \int_a^b x(t)y(t)dt,$$

where $a, b \in [-\infty, +\infty]$ and $a < b$.

Proposition 3.1.20 [1] Let X be an inner product space. Then the function $\|.\| : X \to [0, +\infty)$ defined by

$$\|x\| = \sqrt{\langle x, x\rangle}, \quad x \in X$$

is a norm on X.

Proposition 3.1.21 [24]**(The Cauchy-Schwarz inequality)** Let X be an inner product space. The following inequality holds for all $x, y \in X$:

$$|\langle x, y\rangle| \leqslant \|x\|\|y\|.$$

Proposition 3.1.22 [24]**(Properties of the inner product)** The following equalities hold: for all $x, y \in H$ and $\alpha \in [0, 1]$,

1. $\|x + y\|^2 = \|x\|^2 + \|y\|^2 + 2\langle x, y\rangle$,

2. $\|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2\langle x, y\rangle$,
3. $\langle x+y, x-y\rangle = \|x\|^2 - \|y\|^2$,
4. $\|x+y\|^2 + \|x-y\|^2 = 2(\|x\|^2 + \|y\|^2)$,
5. $\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2$.

Definition 3.1.23 [11]**(Bounded sequence)** Let H be an inner product space. A sequence $\{x_k\}$ in H is said to be bounded if there is $M > 0$ such that for all $k \in \mathbb{N}$,

$$\|x_k\| \leq M.$$

Definition 3.1.24 [1]**(Bounded linear operator)** Let X and Y be normed spaces and $T : X \to Y$ be a linear operator. The operator T is said to be bounded if there is a real number $M > 0$ such that for all $x \in X$,

$$\|Tx\| \leqslant M\|x\|.$$

Definition 3.1.25 [1]**(Convex subsets)** Let H be a Hilbert space. A subset $C \subseteq H$ is said to be convex, if $(1-\lambda)x + \lambda y \in C$ for all $x, y \in C$ and for all $\lambda \in [0, 1]$.

Definition 3.1.26 [1]**(Closed set)** Let H be an inner product space. A subset C of H is said to be closed if for each a sequence $\{x_k\}$ in C with $x_k \to x$ implies that $x \in C$.

Definition 3.1.27 [12]**(Weak convergence in a Hilbert space)** A sequence $\{x_k\}$ in a Hilbert space H is said to converge weakly to a point x in H if

$$\langle x_k, y\rangle \to \langle x, y\rangle$$

for all $y \in H$ and denote that $x_k \rightharpoonup x$.

Proposition 3.1.28 [24] Let H be a Hilbert space. Then every bounded sequence $\{x_k\}$ in H, there exists a weakly convergent subsequence $\{x_{k_m}\}$ of $\{x_k\}$.

Definition 3.1.29 [24] Let H be a Hilbert space. Let $T : H \to H$ be an operator. Then 1. The operator T is called L-Lipschitz continuous with $L > 0$ if

$$\|Tx - Ty\| \leqslant L\|x - y\|, \quad \forall x, y \in H.$$

If $L = 1$, then T is called nonexpansive.

If $L \in [0, 1)$, then T is called a contraction mapping.

2. The operator T is called monotone if

$$\langle Tx - Ty, x - y\rangle \geqslant 0, \quad \forall x, y \in H.$$

3. The operator T is called firmly nonexpansive if

$$\|Tx - Ty\|^2 \leqslant \|x - y\|^2 - \|(I - T)x - (I - T)y\|^2,$$

or equivalently

$$\langle Tx - Ty, x - y\rangle \geqslant \|Tx - Ty\|^2, \quad \forall x, y \in C.$$

4. d-cocoercive or d-inverse strongly monotone if dT is firmly nonexpansive when $d > 0$.

Definition 3.1.30 [24] Let C be a nonempty subset of H and $x \in H$. If there exists a point $x^* \in C$ such that

$$\|x^* - x\| \leqslant \|y - x\|, \ \forall y \in C,$$

then x^* is called a metric projection of x on C, denoted by $P_C x$. It $P_C x$ exists and is unique for all x, then the function P_C of H onto C is called the metric projection.

Theorem 3.1.31 [47] Let C be a nonempty closed convex subset of H. Then, for any $x \in H$ there exists a metric projection $P_C x$ onto C and it is unique.

Proposition 3.1.32 [115] Let C be a nonempty convex subset of H and let $x \in H$, $x^* \in C$. Then,

$$x^* = P_C x \Leftrightarrow \langle x - x^*, y - x^* \rangle \leqslant 0, \ \forall y \in C.$$

Definition 3.1.33 [12]**(Proximal operator)** Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function, The proximal operator of f is defined by

$$\text{prox}_f(y) = \underset{x}{\text{argmin}} \Big(f(x) + \frac{1}{2}\|x - y\|_2^2 \Big),$$

and the proximal operator of the scalar function γf, where $\gamma > 0$, which can be expressed as

$$\text{prox}_{\gamma f}(y) = \underset{x}{\text{argmin}} \Big(f(x) + \frac{1}{2\gamma}\|x - y\|_2^2 \Big),$$

then $\text{prox}_{\gamma f}$ is call the proximal operator of f with parameter γ.

Definition 3.1.34 [112]**(The sum rule)** If f and g are both differentiable, then

$$\frac{\partial}{\partial x}[f(x, y) + g(x, y)] = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial x},$$
$$\frac{\partial}{\partial y}[f(x, y) + g(x, y)] = \frac{\partial f}{\partial y} + \frac{\partial g}{\partial y}.$$

Definition 3.1.35 [30] Let H be a real Hilbert space, and $f : H \to \mathbb{R} \cup \{+\infty\}$. The effective domain of f is defined by $dom(f) = \{a \in H : f(a) < +\infty\}$. Then, f is called proper if $dom(f) \neq \emptyset$.

Definition 3.1.36 [30] Let H be a real Hilbert space, and let $f : H \to \mathbb{R} \cup \{-\infty, +\infty\}$. The following statements are equivalent:

(i) f is lower semicontinuous;

(ii) f is closed;

(iii) for any $\delta \in \mathbb{R}$, the level set $lev(f, \delta) = \{a \in H : f(a) \leqslant \delta\}$ is closed.

Let us begin with some concepts of monotonicity of a bifunction [18, 90]. Let C be a nonempty, closed and convex subset of H. A bifunction $f : H \times H \to \mathbb{R}$ is said to be:

(i) strongly monotone on C if there exists a constant $\gamma > 0$ such that

$$f(x, y) + f(y, x) \leqslant -\gamma \|x - y\|^2, \ \forall x, y \in C;$$

(ii) mototone on C if $f(x, y) \leqslant -f(y, x), \forall x, y \in C$;

(iii) pseudomonotone on C if $f(x, y) \geqslant 0 \Longrightarrow f(y, x) \leqslant 0, \forall x, y \in C$;

(iv) satisfying Lipschitz-type condition on C if there exist two positive constants c_1, c_2 such that

$$c_1\|x - y\|^2 + c_2\|y - z\|^2 \geqslant f(x, z) - f(x, y) - f(y, z), \forall x, y, z \in C.$$

For every $x \in H$, the metric projection $P_C x$ of x onto C is the nearest point of x in C , that is, $P_C x = \arg\min\{\|y - x\| : y \in C\}$.

A differentiable function f is convex if and only if there holds the inequality

$$f(z) \geqslant f(x) + \langle \nabla f(x), z - x \rangle, \ \forall z \in H.$$

Definition 3.1.37 [24]**(Subdifferential)** Let $f : H \to \mathbb{R}$ be convex. The subset

$$\partial f(x) = \{g \in H : \langle g, y - x \rangle \leqslant f(y) - f(x) \text{ for all } y \in H\}$$

is called a subdifferential of f at $x \in H$. The function f is said to be subdifferentiable at x if $\partial f(x) \neq \emptyset$. An element of the subdifferential $\partial f(x)$ is called a subgradient of f at x.

Definition 3.1.38 [12] Let H be a real Hilbert space and let $f : H \to \mathbb{R}$, function f is said to be lower semi-continuous at x if $x_k \to x$, then $f(x) \leqslant \liminf_{k\to\infty} f(x_k)$.

Definition 3.1.39 [13]**(Fermat's rule)** Let $f : H \to (-\infty, +\infty]$ be proper. Then

$$\text{argmin} f = \{x \in H : 0 \in \partial f(x)\}.$$

Definition 3.1.40 [42] Let C be a convex subset of a real Hilbert space H and $g : C \to \mathbb{R}$ be subdifferentiable on C. Then x^* is a solution to the following convex problem:

$$\min\{g(x) : x \in C\}$$

if and only if $0 \in \partial g(x^*) + N_C(x^*)$, where $\partial g(\cdot)$ denotes the subdifferential of g and $N_C(x^*)$ is the mormal cone of C at $x^* \in C$.

For each $x, z \in H$, by $\partial_2 f(z, x)$, we denote the subdifferential of convex function $f(z, .)$ at x, i.e.

$$\partial_2 f(z, x) = \{u \in H : f(z, y) \geqslant \langle u, y - x\rangle + f(z, x),\ \forall y \in H\}.$$

In particular, for $z \in C$,

$$\partial_2 f(z, z) = \{u \in H : f(z, y) \geqslant \langle u, y - z\rangle,\ \forall y \in H\}.$$

The normal cone N_C to C at a point $x \in C$ is defined by $N_C(x) = \{w \in H : \langle y - x, w\rangle \geqslant 0,\ \forall y \in C\}$.

Assumption 3.1.41 [18] Let $f : C \times C \to \mathbb{R}$ be a bifunction satisfying the following assumptions:

(1) f is monotone, i.e., $f(x, y) + f(y, x) \leqslant 0$ for all $x \in C$;

(2) $f(x, x) = 0$ for all $x \in C$;

(3) for each $x \in C, y \to f(x, y)$ is convex and lower semi-continuous;

(4) for each $x, y, z \in C$, $\limsup_{k\to 0^+} f(kz + (1-k)x, y) \leqslant f(x, y)$.

Proposition 3.1.42 [125] *Let $A : H \to H$ be an d-inverse strongly monotone mapping, then*

(i) If β is any constant in $(0, 2d]$, then the mapping $I - \beta A$ is nonexpansive, where I is the identity mapping on H.

(ii) A is an $\frac{1}{d}$-Lipschitz continuous and monotone mapping.

A mapping $A : C \to H_1$ is called $\rho-$inverse strongly monotone if there exists $\rho > 0$ such that $\langle x - \delta, Ax - A\delta\rangle \geqslant \rho\|Ax - A\delta\|^2$, $\forall x, \delta \in C$.

3.2 Lemmas

Lemma 3.2.1 [12] *For each $x \in H$ and $\lambda > 0$,*

$$\frac{1}{\lambda}\langle x - prox_{\lambda g}(x), y - prox_{\lambda g}(x)\rangle \leqslant g(y) - g(prox_{\lambda g}(x)), \ \forall y \in C,$$

where $prox_{\lambda g}(x) = \arg\min\{\lambda g(y) + \frac{1}{2}\|x - y\|^2 : y \in C\}$.

Lemma 3.2.2 [106, 107] *Let $S \subseteq \mathbb{R}$ be a nonempty, closed, and convex subset of a real Hilbert space H. Let $u \in H$ be arbitrarity given, $z = P_S u$, and $\Omega = \{x \in H : \langle x - u, x - z\rangle \leqslant 0\}$. Then $\Omega \cap S = \{z\}$.*

Lemma 3.2.3 [122] *Let $\{a_k\}$ and $\{c_k\}$ be nonnegative sequences of real numbers such that $\sum_{k=1}^{\infty} c_k < \infty$, and let $\{b_k\}$ be a sequence of real numbers such that $\limsup\limits_{k\to\infty} b_k \leqslant 0$. If for any $k \in \mathbb{N}$ such that*

$$a_{k+1} \leqslant (1 - \gamma_k)a_k + \gamma_k b_k + c_k,$$

where $\{\gamma_k\}$ is a sequence in $(0, 1)$ such that $\sum\limits_{k=1}^{\infty} \gamma_k = \infty$, then $\lim\limits_{k\to\infty} a_k = 0$.

Lemma 3.2.4 [95] *Let* $\{a_k\}$, $\{b_k\}$ *and* $\{c_k\}$ *be positive sequences, such that*

$$a_{k+1} \leqslant (1+c_k)a_k + b_k, \ \ k \geqslant 1.$$

If $\sum_{k=1}^{\infty} c_k < +\infty$ *and* $\sum_{k=1}^{\infty} b_k < +\infty$*; then,* $\lim_{k\to+\infty} a_k$ *exists.*

Lemma 3.2.5 [113] *Let* X *be a Banach space which satisfies Opial's condition and* $\{x_k\}$ *be a sequence in* X*. Let* $y, z \in X$ *be such that* $\lim_{k\to\infty} \|x_k - y\|$ *and* $\lim_{k\to\infty} \|x_k - z\|$ *exist. If* $\{x_{k_n}\}$ *and* $\{x_{l_n}\}$ *are subsequences of* $\{x_k\}$ *which converge weakly to* y *and* z*, respectively, then* $y = z$*.*

Lemma 3.2.6 [32] *Let* $f : C \times C \to \mathbb{R}$ *satisfy Assumption 3.1.41. Denote a mapping* $T_r^f : H_1 \to C$*, for each* $r > 0$ *and* $e \in H_1$*, as follows:*

$$T_r^f(e) = \{x \in C : f(x,y) + \frac{1}{r}\langle y - x, x - e\rangle \geqslant 0, \ \ \forall y \in C\}.$$

Then the following hold:

(1) T_r^f *is nonempty and single-valued;*

(2) T_r^f *is firmly nonexpansive, i.e.,* $\exists\, e, v \in H_1$,

$$\|T_r^f e - T_r^f v\|^2 \leqslant \langle T_r^f e - T_r^f v, e - v\rangle;$$

(3) $EP(f, C) = F(T_r^f)$*;*

(4) $EP(f, C)$ *is convex and closed.*

Further, let $g : Q \times Q \to \mathbb{R}$ *satisfy Assumption 3.1.41. Define a mapping* $T_s^g : H_2 \to Q$*, for each* $s > 0$ *and* $b \in H_2$*, as follows:*

$$T_s^g(b) = \{h \in Q : g(h,p) + \frac{1}{s}\langle p - h, h - b\rangle \geqslant 0, \ \ \forall p \in Q\}.$$

Then we easily observe that:

(1) T_s^g *is nonempty and single-valued;*

(2) T_s^g *is firmly nonexpansive;*

(3) $F(T_s^g) = EP(g, Q)$;

(4) $EP(g, Q)$ *is closed and convex.*

Lemma 3.2.7 [116] *Let* $\{a_k\}, \{b_k\}$ *be sequences of nonnegative real numbers such that*

$$a_{k+1} \leqslant a_k + b_k, \ \ k \in \mathbb{N}.$$

If $\sum_{k=1}^{\infty} b_k < \infty$, *then* $\lim_{k\to\infty} a_k$ *exists.*

Lemma 3.2.8 [126] *Let* $f : C \times C \to \mathbb{R}$ *be an equilibrium function, and let* T_r^f *be defined as in Lemma 3.2.6 for* $r > 0$. *Let* $x, y \in H$ *and* $r_1, r_2 > 0$, *then*

$$\|T_{r_2}^f(y) - T_{r_1}^f(x)\| \leqslant \|y - x\| + \left|\frac{r_2 - r_1}{r_2}\right| \|T_{r_2}^f(y) - x\|.$$

Lemma 3.2.9 [30] *Let* H_1, H_2 *be two real Hilbert spaces and* $C \subset H_1, Q \subset H_2$ *be nonempty, closed, and convex subset of* H_1 *and* H_2, *respectively, and let* $A : H_1 \to H_2$ *be a bounded linear operator. Let* $f : C \times C \to \mathbb{R}, g : Q \times Q \to \mathbb{R}$ *be bifunction satisfying Assumption 3.1.41, and let* T_r^f *be defined as in Lemma 3.2.6. Then, the following hold: (i)* $x^* \in \omega$ *if only if* $x^* = T_r^f(I - \beta A^T(I - T_r^g)A)x^*$ *for all* $\beta \in (0, \frac{1}{L})$, *where* L *is the spectral radius of* $A^T A$ *and* A^T *is the adjoint of* A; *(ii) if* $0 < \beta < \frac{2}{L}$, *then* $T_r^f(I - \beta A^T(I - T_r^g)A)$ *is nonexpansive.*

Lemma 3.2.10 [123] *Let* $\{a_k\} \subset [0,\infty), \{b_k\} \subset [0,\infty)$ *and* $\{c_k\} \subset [0,1)$ *be sequences of real numbers such that*

$$a_{k+1} \leqslant (1-c_k)a_k + b_k, \ k \in \mathbb{N},$$
$$\sum_{k=1}^{\infty} c_k = \infty \text{ and } \sum_{k=1}^{\infty} b_k < \infty$$

Then $\lim_{k\to\infty} a_k = 0$.

CHAPTER 4

MAIN RESULTS

This chapter is to present our results for equilibrium problems. We have 3 section; pseudomonotone equilibrium problems, split equilibrium problems, numerical example. Also, we provide some applications in data classification including the numerical experiments for supporting our main theorems.

4.1 Pseudomonotone equilibrium problems

The convergence of algorithms will be given under the conditions that

Condition 4.1.1 (A1) f is pseudomonotone on C with $int(C) \neq \emptyset$ or $f(x, \cdot)$ is continuous at some $z \in C$ for every $x \in C$;

(A2) f satisfies Lipschitz-type condition on H with two constants c_1 and c_2;

(A3) $f(\cdot, y)$ is sequentially weakly upper semicontinuous on C for each fixed point $y \in C$, i.e. if $\{x_k\} \subset C$ is a sequence converging weakly to $x \in C$, then $f(x, y) \geqslant \limsup_{k\to\infty} f(x_k, y)$;

(A4) for $x \in H$, $f(x, \cdot)$ is convex and lower semicontinuous, subdifferentiable on H;

(A5) $V : H \to H$ is contraction with contraction constant α.

4.1.1 A modified inertial viscosity extragradient type method for equilibrium problems application to classification of diabetes mellitus: machine learning methods

Algorithm 4.1.2 (Modified viscosity type inertial extragradient algorithm for EP)

Initialization: Select $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$, $\tau \in (0, \frac{1}{2}]$, $\{\alpha_k\} \subset (0,1)$, and $\{\theta_k\} \subset [0, \frac{1}{3})$. **Iterative step:** Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = \alpha_k V(x_k) + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. Calculate:

$$z_k = \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. Calculate the next iteration via:

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

Replace k by $k+1$ and return to **Step1.**

For the rest of this thesis, we assume the following condition.

Condition 4.1.3 (i) $\{\alpha_k\} \subset (0,1]$ is non-increasing with $\lim_{k\to\infty} \alpha_k = 0$ and $\sum_{k=1}^{\infty} \alpha_k = \infty$;

(ii) $0 \leqslant \theta_k \leqslant \theta_{k+1} \leqslant \theta < \frac{1}{3}$ and $\lim_{k\to\infty} \frac{\theta_k}{\alpha_k}\|x_k - x_{k-1}\| = 0$;

(iii) $EP(f, C) \neq \emptyset$.

Before we prove the strong convergence result, we need some lemmas below.

Lemma 4.1.4 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 4.1.2. Then there exists* $N > 0$ *such that*

$$\|x_{k+1} - u\|^2 \leqslant \|w_k - u\|^2 - \|x_{k+1} - w_k\|^2, \ \forall u \in EP(f, C), \ \ k \geqslant N.$$

Proof. By the definition of y_k, and Lemma 3.2.1, we have

$$\frac{1}{\lambda_k}\langle w_k - y_k, y - y_k\rangle \leqslant f(w_k, y) - f(w_k, y_k), \ \forall y \in C. \tag{4.1.1}$$

Putting $y = z_k$ into (4.1.1), we obtain

$$\frac{1}{\lambda_k}\langle y_k - w_k, y_k - z_k\rangle \leqslant f(w_k, z_k) - f(w_k, y_k). \tag{4.1.2}$$

By the definition of z_k, we have

$$\frac{1}{\lambda_k}\langle w_k - z_k, y - z_k\rangle \leqslant f(y_k, y) - f(y_k, z_k), \ \forall y \in C. \tag{4.1.3}$$

(4.1.2) and (4.1.3) imply that

$$\begin{aligned}\frac{2}{\lambda_k}\langle w_k - z_k, y - z_k\rangle + \frac{2}{\lambda_k}\langle y_k - w_k, y_k - z_k\rangle \\ \leqslant 2f(y_k, y) + 2(f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k)).\end{aligned} \tag{4.1.4}$$

If $f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k) > 0$, then

$$f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k) \leqslant c_1\|w_k - y_k\|^2 + c_2\|z_k - y_k\|^2. \tag{4.1.5}$$

Observe that (4.1.5) is also satisfied if $f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k) \leqslant 0$. By

(4.1.4) and (4.1.5), we have

$$\langle w_k - z_k, y - z_k\rangle + \langle y_k - w_k, y_k - z_k\rangle \\ \leqslant \lambda_k f(y_k, y) + \lambda_k c_1\|w_k - y_k\|^2 + \lambda_k c_2\|z_k - y_k\|^2. \quad (4.1.6)$$

Note that

$$\langle w_k - z_k, z_k - y\rangle = \frac{1}{2}(\|w_k - y\|^2 - \|w_k - z_k\|^2 - \|z_k - y\|^2) \quad (4.1.7)$$

and

$$\langle w_k - y_k, z_k - y_k\rangle = \frac{1}{2}(\|w_k - y_k\|^2 + \|z_k - y_k\|^2 - \|w_k - z_k\|^2). \quad (4.1.8)$$

Using (4.1.7) and (4.1.8) in (4.1.6), we obtain, for all $y \in C$,

$$\|z_k - y\|^2 \leqslant \|w_k - y\|^2 - (1 - 2\lambda_k c_1)\|w_k - y_k\|^2 \\ - (1 - 2\lambda_k c_2)\|z_k - y_k\|^2 + 2\lambda_k f(y_k, y). \quad (4.1.9)$$

Taking $y = u \in EP(f, C) \subset C$, one has $f(u, y_k) \geqslant 0, \forall k$. By (A1), we obtain $f(y_k, u) \leqslant 0,\ \forall k$. Hence, we obtain from (4.1.9) that

$$\|z_k - u\|^2 \leqslant \|w_k - u\|^2 - (1 - 2\lambda_k c_1)\|w_k - y_k\|^2 - (1 - 2\lambda_k c_2)\|z_k - y_k\|^2. \quad (4.1.10)$$

It follows from $\lambda_k \in (0, \frac{1}{2\max\{c_1, c_2\}})$ and (4.1.10), we have

$$\|z_k - u\| \leqslant \|w_k - u\|.$$

On the other hand, we have

$$\|x_{k+1} - u\|^2 = (1 - \tau)\|w_k - u\|^2 + \tau\|z_k - u\|^2 - (1 - \tau)\tau\|z_k - w_k\|^2. \quad (4.1.11)$$

Substituting (4.1.10) into (4.1.11), we obtain

$$\begin{aligned}\|x_{k+1}-u\|^2 \leq \|w_k-u\|^2 - \tau\|w_k-u\|^2 + \tau\|w_k-u\|^2 - \tau(1-2\lambda_k c_1)\|w_k-y_k\|^2 \\ - \tau(1-2\lambda_k c_2)\|z_k-y_k\|^2 - (1-\tau)\tau\|z_k-w_k\|^2. \qquad (4.1.12)\end{aligned}$$

Moreover, we have $z_k - w_k = \frac{1}{\tau}(x_{k+1}-w_k)$, which together with (4.1.12) gives

$$\begin{aligned}\|x_{k+1}-u\|^2 &\leqslant \|w_k-u\|^2 - \tau(1-2\lambda_k c_1)\|w_k-y_k\|^2 - \tau(1-2\lambda_k c_2)\|z_k-y_k\|^2 \\ &\quad - (1-\tau)\tau\frac{1}{\tau^2}\|x_{k+1}-w_k\|^2 \\ &\leqslant \|w_k-u\|^2 - \frac{1-\tau}{\tau}\|x_{k+1}-w_k\|^2 \\ &\leqslant \|w_k-u\|^2 - x\|x_{k+1}-w_k\|^2, \ \forall k \geqslant N, \qquad (4.1.13)\end{aligned}$$

where $x = \frac{1-\tau}{\tau}$. □

Lemma 4.1.5 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 4.1.2. Then, for all* $u \in EP(f, C)$,

$$\begin{aligned}&-2\alpha_k\langle x_k-u, x_k-V(x_k)\rangle \\ &\quad \geqslant \|x_{k+1}-u\|^2 - \|x_k-u\|^2 + 2\theta_{k+1}\|x_{k+1}-x_k\|^2 - 2\theta_k\|x_k-x_{k-1}\|^2 \\ &\quad + \alpha_{k+1}\|V(x_k)-x_{k+1}\|^2 - \alpha_k\|x_k-V(x_k)\|^2 - \theta_k\|x_k-u\|^2 \\ &\quad + \theta_{k-1}\|x_{k-1}-u\|^2 + (1-3\theta_{k+1}-\alpha_k)\|x_k-x_{k+1}\|^2. \qquad (4.1.14)\end{aligned}$$

Proof. By Lemma 3.2.4, we have

$$\|x_{k+1}-u\|^2 \leqslant \|w_k-u\|^2 - \|x_{k+1}-w_k\|^2. \qquad (4.1.15)$$

Moreover, from the definition of w_k, we obtain that

$$\|w_k-u\|^2 = \|x_k-u\|^2 + \|\theta_k(x_k-x_{k-1}) - \alpha_k(x_k-V(x_k))\|^2$$

$$
\begin{aligned}
&+ 2\langle x_k - u, \theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\rangle \\
&= \|x_k - u\|^2 + \|\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\|^2 \\
&+ 2\theta_k\langle x_k - u, x_k - x_{k-1}\rangle - 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle. \qquad (4.1.16)
\end{aligned}
$$

Replacing u by x_{k+1} in (4.1.16), we have

$$
\begin{aligned}
&\|w_k - x_{k+1}\|^2 \\
&= \|x_k - x_{k+1}\|^2 + \|\alpha_k(x_k - V(x_k)) - \theta_k(x_k - x_{k-1})\|^2 \\
&+ 2\theta_k\langle x_k - x_{k+1}, x_k - x_{k-1}\rangle - 2\alpha_k\langle x_k - x_{k+1}, x_k - V(x_k)\rangle. \qquad (4.1.17)
\end{aligned}
$$

Substituting (4.1.16) and (4.1.17) into (4.1.15), we have

$$
\begin{aligned}
&\|x_{k+1} - u\|^2 \\
&\leqslant \|x_k - u\|^2 + \|\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\|^2 + 2\theta_k\langle x_k - u, x_k - x_{k-1}\rangle \\
&\quad - 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle - \|x_k - x_{k+1}\|^2 - 2\theta_k\langle x_k - x_{k+1}, x_k - x_{k-1}\rangle \\
&\quad + 2\alpha_k\langle x_k - x_{k+1}, x_k - V(x_k)\rangle - \|\alpha_k(x_k - V(x_k)) - \theta_k(x_k - x_{k-1})\|^2 \\
&= \|x_k - u\|^2 - \|x_k - x_{k+1}\|^2 + 2\theta_k\langle x_k - u, x_k - x_{k-1}\rangle - 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\quad + 2\alpha_k\langle x_k - x_{k+1}, x_k - V(x_k)\rangle + \theta_k\|x_k - x_{k+1}\|^2 + \theta_k\|x_k - x_{k-1}\|^2 \\
&\quad - \theta_k\|x_k - x_{k+1} + (x_k - x_{k-1})\|^2.
\end{aligned}
$$

Therefore, we obtain

$$
\begin{aligned}
&\|x_{k+1} - u\|^2 - \|x_k - u\|^2 - \theta_k\|x_k - x_{k-1}\|^2 + \|x_k - x_{k+1}\|^2 - \theta_k\|x_k - x_{k+1}\|^2 \\
&\leqslant 2\theta_k\langle x_k - u, x_k - x_{k-1}\rangle - 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle + 2\alpha_k\langle x_k - x_{k+1}, x_k - V(x_k)\rangle \\
&= -2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle - \theta_k\|x_{k-1} - u\|^2 + \theta_k\|x_k - u\|^2 + \theta_k\|x_k - x_{k-1}\|^2 \\
&\quad - \alpha_k\|V(x_k) - x_{k+1}\|^2 + \alpha_k\|x_{k+1} - x_k\|^2 + \alpha_k\|x_k - V(x_k)\|^2.
\end{aligned}
$$

It follows that

$$
\begin{aligned}
&- 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 - \theta_k\|x_k - x_{k-1}\|^2 + \|x_k - x_{k+1}\|^2 \\
&\quad - \theta_k\|x_k - x_{k+1}\|^2 + \theta_k\|x_{k-1} - u\|^2 - \theta_k\|x_k - u\|^2 - \theta_k\|x_k - x_{k-1}\|^2 \\
&\quad + \alpha_k\|V(x_k) - x_{k+1}\|^2 - \alpha_k\|x_{k+1} - x_k\|^2 - \alpha_k\|x_k - V(x_k)\|^2 \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&\quad + \theta_k(\|x_{k-1} - u\|^2 - \|x_k - u\|^2) + \alpha_k(\|V(x_k) - x_{k+1}\|^2 - \|x_k - V(x_k)\|^2) \\
&\quad + (1 - \theta_k - 2\theta_{k+1} - \alpha_k)\|x_{k+1} - x_k\|^2.
\end{aligned}
$$

Since θ_k is non-decreasing and α_k is non-increasing, we then obtain

$$
\begin{aligned}
&- 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&\quad + \alpha_{k+1}\|V(x_k) - x_{k+1}\|^2 - \alpha_k\|x_k - V(x_k)\|^2 - \theta_k\|x_k - u\|^2 \\
&\quad + \theta_{k-1}\|x_{k-1} - u\|^2 + (1 - 3\theta_{k+1} - \alpha_k)\|x_k - x_{k+1}\|^2.
\end{aligned}
$$

□

Lemma 4.1.6 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then* $\{x_k\}$ *generated by Algorithm 4.1.2 is bounded.*

Proof. From (4.1.13) and Condition 4.1.3 (ii), there exists $K > 0$ such that

$$
\begin{aligned}
\|x_{k+1} - u\| &\leqslant \|w_k - u\| \\
&= \|\alpha_k V(x_k) + (1 - \alpha_k)x_k + \theta_k(x_k - x_{k-1}) - u\| \\
&\leqslant \alpha_k\|V(x_k) - u\| + (1 - \alpha_k)\|x_k - u\| + \theta_k\|x_k - x_{k-1}\| \\
&= \alpha_k\|V(x_k) - u\| + (1 - \alpha_k)\|x_k - u\| + \alpha_k\frac{\theta_k}{\alpha_k}\|x_k - x_{k-1}\|
\end{aligned}
$$

$$
\begin{aligned}
&\leqslant \alpha_k\|V(x_k)-u\| + (1-\alpha_k)\|x_k-u\| + \alpha_k K \\
&\leqslant \alpha_k(\|V(x_k)-V(u)\| + \|V(u)-u\|) + (1-\alpha_k)\|x_k-u\| + \alpha_k K \\
&\leqslant (1-\alpha_k(1-\alpha))\|x_k-u\| + \alpha_k(1-\alpha)(\frac{\|V(u)-u\|+K}{1-\alpha}) \\
&\leqslant max\{\|x_k-u\|, \frac{\|V(u)-u\|+K}{1-\alpha}\}.
\end{aligned}
$$

This implies that $\|x_{k+1}-u\| \leqslant max\{\|x_1-u\|, \frac{\|V(u)-u\|+K}{1-\alpha}\}$. This shows that $\{x_k\}$ is bounded. □

Lemma 4.1.7 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 4.1.2. For each* $k \geqslant 1$*, define*

$$
u_k = \|x_k-u\|^2 - \theta_{k-1}\|x_{k-1}-u\|^2 + 2\theta_k\|x_k-x_{k-1}\|^2 + \alpha_k\|x_k-V(x_k)\|^2.
$$

Then $u_k \geqslant 0$.

Proof. Since $\{\theta_k\}$ is non-decreasing with $0 \leqslant \theta_k < \frac{1}{3}$, and $2\langle x,y\rangle = \|x\|^2 + \|y\|^2 - \|x-y\|^2$ for all $x, y \in H$, we have

$$
\begin{aligned}
u_k = &\|x_k-u\|^2 - \theta_{k-1}[\|x_{k-1}-x_k\|^2 + \|x_k-u\|^2 + 2\langle x_{k-1}-x_k, x_k-u\rangle] \\
&+ 2\theta_k\|x_k-x_{k-1}\|^2 + \alpha_k\|x_k-V(x_k)\|^2 \\
= &\|x_k-u\|^2 - \theta_{k-1}[2\|x_{k-1}-x_k\|^2 + 2\|x_k-u\|^2 - \|x_{k-1}-2x_k+u\|^2] \\
&+ 2\theta_k\|x_k-x_{k-1}\|^2 + \alpha_k\|x_k-V(x_k)\|^2 \\
\geqslant &\|x_k-u\|^2 - 2\theta_k\|x_{k-1}-x_k\|^2 - \frac{2}{3}\|x_k-u\|^2 + \theta_{k-1}\|x_{k-1}-2x_k+u\|^2 \\
&+ 2\theta_k\|x_k-x_{k-1}\|^2 + \alpha_k\|x_k-V(x_k)\|^2 \\
\geqslant &\frac{1}{3}\|x_k-u\|^2 + \alpha_k\|x_k-V(x_k)\|^2 \\
\geqslant &0.
\end{aligned}
$$

This completes the proof. □

Lemma 4.1.8 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 4.1.2. Suppose*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0,$$

and

$$\lim_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k \|x_k - u\|^2) = 0.$$

Then $\{x_k\}$ *converges strongly to* $u \in EP(f, C)$.

Proof. By our assumptions, we have

$$\begin{aligned} 0 &= \lim_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k \|x_k - u\|^2) \\ &= \lim_{k\to\infty} [(\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\|)(\|x_{k+1} - u\| - \sqrt{\theta_k}\|x_k - u\|)]. \end{aligned} \quad (4.1.18)$$

In the case

$$\lim_{k\to\infty} (\|x_{i+1} - u\| + \sqrt{\theta_i}\|x_i - u\|) = 0,$$

this implies that $\{x_k\}$ converges strongly to u immediately. Assume this limit does not hold. Then there is a subset $N^* \subseteq N$ and a constant $\rho > 0$ such that

$$\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\| \geqslant \rho, \ \forall k \in N^*. \quad (4.1.19)$$

Using (4.1.18) and $\theta_k \leqslant \theta < 1$. For $k \in N^*$, it then follows that

$$\begin{aligned} 0 &= \lim_{k\to\infty} (\|x_{k+1} - u\| - \sqrt{\theta_k}\|x_k - u\|) \\ &\geqslant \limsup_{k\to\infty} (\|x_k - u\| - \|x_{k+1} - x_k\| - \sqrt{\theta_k}\|x_k - u\|) \end{aligned}$$

$$
\begin{aligned}
&\geqslant \limsup_{k\to\infty}((1-\sqrt{\theta})\|x_k-u\|-\|x_{k+1}-x_k\|)\\
&=(1-\sqrt{\theta})\limsup_{k\to\infty}\|x_k-u\|-\lim_{k\to\infty}\|x_{k+1}-x_k\|\\
&=(1-\sqrt{\theta})\limsup_{k\to\infty}\|x_k-u\|.
\end{aligned}
$$

Consequently, we have $\limsup_{k\to\infty}\|x_k-u\|\leqslant 0$. Since $\liminf_{k\to\infty}\|x_k-u\|\geqslant 0$ obviously holds, it follows that $\lim_{k\to\infty}\|x_k-u\|=0$. This implies (by (4.1.19))

$$
\begin{aligned}
\|x_{k+1}-x_k\| &\geqslant \|x_{k+1}-u\|-\|x_k-u\|\\
&=\|x_{k+1}-u\|+\sqrt{\theta_k}\|x_k-u\|-(1+\sqrt{\theta_k})\|x_k-u\|\\
&\geqslant \frac{\rho}{2},
\end{aligned}
$$

for all $k\in N^*$ sufficiently large, a contradiction to the assumption that $\lim_{k\to\infty}\|x_{k+1}-x_k\|=0$. This completes the proof. □

We now give the following strong convergence result of Algorithm 4.1.2.

Theorem 4.1.9 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then* $\{x_k\}$ *generated by Algorithm 4.1.2 strongly converges to the solution* $u=P_{EP(f,C)}V(u)$.

Proof. From Lemma 4.1.7 and (4.1.14), we have

$$
\begin{aligned}
&-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle\\
&\geqslant u_{k+1}-u_k-\alpha_{k+1}\|x_{k+1}-V(x_{k+1})\|^2+\alpha_{k+1}\|V(x_k)-x_{k+1}\|^2\\
&+(1-3\theta_{k+1}-\alpha_k)\|x_k-x_{k+1}\|^2. \qquad (4.1.20)
\end{aligned}
$$

Since $P_{EP(f,C)}V$ is a contraction, by the Banach fixed point theorem, there exists a unique $u=P_{EP(f,C)}V(u)$. It follows from Lemma 4.1.4 that

$$
\begin{aligned}
&\|x_{k+1}-u\|^2\\
&\leqslant \|w_k-u\|^2\\
&= \|\alpha_k(V(x_k)-u)+(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2\\
&\leqslant \|(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2+2\langle\alpha_k(V(x_k)-u),w_k-u\rangle\\
&= \|(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2+2\alpha_k\langle V(x_k)-V(u),w_k-u\rangle\\
&\quad +2\alpha_k\langle V(u)-u,w_k-u\rangle\\
&\leqslant \|(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2+2\alpha_k\langle V(u)-u,w_k-u\rangle\\
&\quad +2\alpha_k\alpha\|x_k-u\|\|w_k-u\|\\
&\leqslant \|(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2+2\alpha_k\langle V(u)-u,w_k-u\rangle\\
&\quad +\alpha_k\alpha(\|x_k-u\|^2+\|w_k-u\|^2)\\
&\leqslant \frac{1}{1-\alpha_k\alpha}\Big(\|(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\|^2+\alpha_k\alpha\|x_k-u\|^2\\
&\quad +2\alpha_k\langle V(u)-u,w_k-u\rangle\Big)\\
&\leqslant \frac{1}{1-\alpha_k\alpha}\Big(\|(1-\alpha_k)(x_k-u)\|^2+2\langle\theta_k(x_k-x_{k-1}),(1-\alpha_k)(x_k-u)\\
&\quad +\theta_k(x_k-x_{k-1})\rangle+\alpha_k\alpha\|x_k-u\|^2+2\alpha_k\langle V(u)-u,w_k-u\rangle\Big)\\
&= \frac{(1-\alpha_k)^2+\alpha_k\alpha}{1-\alpha_k\alpha}\|x_k-u\|^2+\frac{1}{1-\alpha_k\alpha}\Big(2\langle\theta_k(x_k-x_{k-1}),\\
&\quad (1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\rangle+2\alpha_k\langle V(u)-u,w_k-u\rangle\Big)\\
&= \big(1-(\frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}-\frac{(\alpha_k)^2}{1-\alpha_k\alpha})\big)\|x_k-u\|^2+\frac{1}{1-\alpha_k\alpha}\Big(2\langle\theta_k(x_k-x_{k-1}),\\
&\quad (1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\rangle+2\alpha_k\langle V(u)-u,w_k-u\rangle\Big)\\
&\leqslant \big(1-\frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}\big)\|x_k-u\|^2+\frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}\Big(\frac{\alpha_k}{2(1-\alpha)}\|x_k-u\|^2\\
&\quad +\frac{1}{\alpha_k(1-\alpha)}\langle\theta_k(x_k-x_{k-1}),(1-\alpha_k)(x_k-u)+\theta_k(x_k-x_{k-1})\rangle\\
&\quad +\frac{1}{1-\alpha}\langle V(u)-u,w_k-u\rangle\Big). \qquad (4.1.21)
\end{aligned}
$$

We will consider into 2 cases.

Case 1. In the case of $u_{k+1} \leqslant u_k + t_k$ for all $k \geqslant k_0$ for some $k_0 \in \mathbb{N}$, $t_k \geqslant 0$ and $\sum_{k=1}^{\infty} t_k < +\infty$. Then $u_k \geqslant 0$, $\forall k \geqslant 1$ by Lemma 3.2.4, we have $\lim_{k\to\infty} u_k = \lim_{k\to\infty} u_{k+1}$ exists. Since $\{x_k\}$ is bounded by Lemma 4.1.6, there exists $M_1 > 0$ such that $2|\langle x_k - u, x_k - V(x_k)\rangle| \leqslant M_1$ and $M_2 > 0$ such that $\|x_{k+1} - V(x_{k+1})\|^2 + \|V(x_k) - x_{k+1}\|^2 \leqslant M_2$. Since $0 \leqslant \theta_k \leqslant \theta_{k+1} \leqslant \theta < \frac{1}{3}$ and $\lim_{k\to\infty} \alpha_k = 0$, there exist $N \in \mathbb{N}$ and $\gamma_1 > 0$ such that $1 - 3\theta_{k+1} - \alpha_k \geqslant \gamma_1$ for all $k \geqslant N$. Therefore, for $k \geqslant N$, we obtain from (4.1.20) that

$$\gamma_1\|x_{k+1} - x_k\|^2 \leqslant \alpha_k M_1 + \alpha_{k+1} M_2 + u_k - u_{k+1} \to 0, \tag{4.1.22}$$

as $k \to \infty$. Hence $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. For $u \in EP(f, C)$, we have

$$\begin{aligned}
&\|w_k - u\|^2 \\
&= \|\alpha_k V(x_k) + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}) - u\|^2 \\
&\leqslant \|\alpha_k V(x_k) + (1-\alpha_k)x_k - u\|^2 + 2\langle \theta_k(x_k - x_{k-1}), w_k - u\rangle \\
&\leqslant \alpha_k\|V(x_k) - u\|^2 + (1-\alpha_k)\|x_k - u\|^2 + 2\theta_k\|x_k - x_{k-1}\|\|w_k - u\| \\
&\leqslant \alpha_k\|V(x_k) - u\|^2 + (1-\alpha_k)\|x_k - u\|^2 + 2\frac{\theta_k}{\alpha_k}\|x_k - x_{k-1}\|\|w_k - u\| \\
&\leqslant \alpha_k\|V(x_k) - u\|^2 + \|x_k - u\|^2 + 2\frac{\theta_k}{\alpha_k}\|x_k - x_{k-1}\|\|w_k - u\|,
\end{aligned}$$

and from (4.1.12), we have

$$\begin{aligned}
\|x_{k+1} - u\|^2 &= \|w_k - u\|^2 - \tau(1 - 2\lambda_k c_1)\|w_k - y_k\|^2 - \tau(1 - 2\lambda_k c_2)\|z_k - y_k\|^2 \\
&\quad - (1-\tau)\tau\frac{1}{\tau^2}\|x_{k+1} - w_k\|^2 \\
&\leqslant \alpha_k\|V(x_k) - u\|^2 + \|x_k - u\|^2 + 2\frac{\theta_k}{\alpha_k}\|x_k - x_{k-1}\|\|w_k - u\| \\
&\quad - \tau(1 - 2\lambda_k c_1)\|w_k - y_k\|^2 - \tau(1 - 2\lambda_k c_2)\|z_k - y_k\|^2 \\
&\quad - \frac{1-\tau}{\tau}\|x_{k+1} - w_k\|^2.
\end{aligned}$$

This implies that

$$\begin{aligned}&\tau(1-2\lambda_k c_1)\|w_k-y_k\|^2+\tau(1-2\lambda_k c_2)\|z_k-y_k\|^2+\frac{1-\tau}{\tau}\|x_{k+1}-w_k\|^2\\&\leqslant \alpha_k\|V(x_k)-u\|^2+\|x_k-u\|^2+2\frac{\theta_k}{\alpha_k}\|x_k-x_{k-1}\|\|w_k-u\|-\|x_{k+1}-u\|^2.\end{aligned}$$

By our condition and (4.1.22), we obtain

$$\lim_{k\to\infty}\|w_k-y_k\|=\lim_{k\to\infty}\|z_k-y_k\|=\lim_{k\to\infty}\|x_{k+1}-w_k\|=0. \tag{4.1.23}$$

Since $\{x_k\}$ is bounded, that is, there exists a subsequence $\{x_{k_i}\}$ of $\{x_k\}$ such that $x_{k_i}\rightharpoonup x^*$ for some $x^*\in H$. From (4.1.22) and (4.1.23), we get $w_{k_i}\rightharpoonup x^*$ and $y_{k_i}\rightharpoonup x^*$ as $i\to\infty$. By the definition of z_k and (4.1.5), we have

$$\begin{aligned}&\lambda_{k_i}f(y_{k_i},y)\\&\geqslant \lambda_{k_i}f(y_{k_i},z_{k_i})+\langle w_{k_i}-z_{k_i},y-z_{k_i}\rangle\\&\geqslant \lambda_{k_i}f(w_{k_i},z_{k_i})-\lambda_{k_i}f(w_{k_i},y_{k_i})-c_1\|w_{k_i}-y_{k_i}\|^2-c_2\|z_{k_i}-y_{k_i}\|^2\\&\quad+\langle w_{k_i}-z_{k_i},y-z_{k_i}\rangle\\&\geqslant \langle y_{k_i}-w_{k_i},y_{k_i}-z_{k_i}\rangle+\langle w_{k_i}-z_{k_i},y-z_{k_i}\rangle-c_1\|w_{k_i}-y_{k_i}\|^2-c_2\|z_{k_i}-y_{k_i}\|^2.\end{aligned}$$

It follows from $\{z_{k_i}\}$ is bounded, $0<\lambda_{k_i}\leqslant\lambda<\frac{1}{2\max\{c_1,c_2\}}$ and Condition 4.1.1 (A3) that $0\leqslant\limsup_{k\to\infty}f(y_{k_i},y)\leqslant f(x^*,y)$ for all $y\in H$. This implies that $f(x^*,y)\geqslant 0$ for all $y\in C$. This shows that $x^*\in EP(f,C)$. Then, we have

$$\begin{aligned}\limsup_{k\to\infty}\langle V(u)-u,w_k-u\rangle&=\lim_{k\to\infty}\langle V(u)-u,w_{k_i}-u\rangle\\&=\langle V(u)-u,x^*-u\rangle\leqslant 0,\end{aligned} \tag{4.1.24}$$

by $u=P_{EP(f,C)}V(u)$. Applying (4.1.24) to the inequality (4.1.21) with Lemma 3.2.3, we can conclude that $x_k\to u=P_{EP(f,C)}V(u)$ as $k\to\infty$.

Case 2. In another case of $\{u_k\}$, we let $\phi : \mathbb{N} \to \mathbb{N}$ be the map defined for all $k \geqslant k_0$ (for some $k_0 \in \mathbb{N}$ large enough) by

$$\phi(k) = \max\{i \in \mathbb{N} : i \leqslant k, u_k + t_k \leqslant u_{k+1}\}.$$

Clearly, $\phi(k)$ is a non-decreasing sequence such that $\phi(k) \to \infty$ for $k \to \infty$ and $u_{\phi(k)} + t_{\phi(k)} \leqslant u_{\phi(k)+1}$ for all $k \geqslant k_0$. Hence, similar to the proof of Case 1, we therefore obtain from (4.1.22) that

$$\gamma_1 \|x_{\phi(k)+1} - x_{\phi(k)}\|^2 \leqslant \alpha_{\phi(k)} M_1 + \alpha_{\phi(k)+1} M_2 + u_{\phi(k)} - u_{\phi(k)+1} \to 0$$

for some constant $M_1, M_2 > 0$. Thus

$$\lim_{k\to\infty} \|x_{\phi(k)+1} - x_{\phi(k)}\| = 0. \tag{4.1.25}$$

By the same proof of Case 1, one also derive

$$\lim_{k\to\infty} \|x_{\phi(k)+1} - w_{\phi(k)}\| = \lim_{k\to\infty} \|w_{\phi(k)} - x_{\phi(k)}\| = \lim_{k\to\infty} \|x_{\phi(k)} - z_{\phi(k)}\| = 0. \tag{4.1.26}$$

Again observe that for $j \geqslant 0$ by (4.1.20), we have $u_{j+1} < u_j + t_j$ when $x_j \notin \Omega = \{x \in H : \langle x - x_0, x - u\rangle \leqslant 0\}$ (note that this Ω is the same set as in Lemma 3.2.2). Hence $x_{\phi(k)} \in \Omega$ for all $k \geqslant k_0$ since $u_{\phi(k)} + t_{\phi(k)} \leqslant u_{\phi(k)+1}$. Sine $\{x_{\phi(k)}\}$ is bounded, there exist subsequence $\{x_{\phi(k)}\}$ of $\{x_{\phi(k)}\}$ which converges weakly to some $x^* \in H$. As Ω is a closed and convex set, it is then weakly closed and so $x^* \in \Omega$. Using (4.1.26), one can see as in Case 1 that $z_{\phi(k)} \rightharpoonup x^*$ and $x^* \in EP(f, C)$. Consequently, we have $x^* \in \Omega \cap EP(f, C)$. In view of Lemma 3.2.2, the intersection $\Omega \cap EP(f, C)$ contains u as its only element. We therefore have $x^* = u$. Furthermore,

$$\|x_{\phi(k)} - u\|^2 = \langle x_{\phi(k)} - V(x_k), x_{\phi(k)} - u\rangle - \langle u - V(x_k), x_{\phi(k)} - u\rangle$$
$$\leqslant -\langle u - V(x_k), x_{\phi(k)} - u\rangle$$

due to $x_{\phi(k)} \in \Omega$. This gives

$$\limsup_{k\to\infty} \|x_{\phi(k)} - u\| \leqslant 0.$$

Hence

$$\lim_{k\to\infty} \|x_{\phi(k)} - u\| = 0. \tag{4.1.27}$$

By definition, $u_{\phi(k)+1}$, we have

$$\begin{aligned} u_{\phi(k)+1} &= \|x_{\phi(k)+1} - u\|^2 - \theta_{\phi(k)}\|x_{\phi(k)} - u\|^2 + 2\theta_{\phi(k)+1}\|x_{\phi(k)+1} - x_{\phi(k)}\|^2 \\ &\quad + \alpha_{\phi(k)+1}\|x_{\phi(k)+1} - V_{\phi(k)+1}\|^2 \\ &\leqslant (\|x_{\phi(k)+1} - x_{\phi(k)}\| + \|x_{\phi(k)} - u\|)^2 - \theta_{\phi(k)}\|x_{\phi(k)} - u\|^2 \\ &\quad + 2\theta_{\phi(k)+1}\|x_{\phi(k)+1} - x_{\phi(k)}\|^2 + \alpha_{\phi(k)+1}\|x_{\phi(k)+1} - V_{\phi(k)+1}\|^2. \end{aligned}$$

By our Condition 4.1.3 (i), (4.1.25) and (4.1.27), we obtain $\lim_{k\to\infty} u_{\phi(k)+1} = 0$. We next show that we actually have $\lim\limits_{k\to\infty} u_k = 0$. To this end, first observe that, for $k \geqslant k_0$, one has $u_k + t_k \leqslant u_{\phi(k)+1}$ if $k \neq \phi(k)$. It follows that for all $k \geqslant k_0$, we have $u_k \leqslant \max\{u_{\phi(k)}, u_{\phi(k)+1}\} = u_{\phi(k)+1} \to 0$, since $\lim_{k\to\infty} t_k = 0$, hence $\limsup\limits_{k\to\infty} u_k \leqslant 0$. On the other hand, Lemma 4.1.7 implies that $\liminf\limits_{k\to\infty} u_k \geqslant 0$. Hence, we obtain $\lim\limits_{k\to\infty} u_k = 0$. Consequently, the boundedness of $\{x_k\}$, $\lim\limits_{k\to\infty} \alpha_k = 0$, and (4.1.20) show that $\|x_k - x_{k+1}\| \to 0$, as $k \to \infty$. Hence the definition of u_k yields $(\|x_{k+1} - u\|^2 - \theta_k\|x_k - u\|^2) \to 0$, as $k \to \infty$. By using Lemma 4.1.8, we obtain the desired conclusion immediately. □

Setting $V(x) = x_0$, $\forall x \in H$, then we obtain the following modified Halpern inertial extragradient algorithm for EPs:

Algorithm 4.1.10 (Modified Halpern inertial extragradient algorithm for EP)

Initialization: Select $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$, $\tau \in (0, \frac{1}{2}], \{\alpha_k\} \subset (0,1)$, and $\{\theta_k\} \subset [0, \frac{1}{3})$. **Iterative step:** Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = \alpha_k x_0 + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. Calculate:

$$z_k = \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. Calculate the next iteration via:

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

Replace k by $k+1$ and return to **Step1.**

From Algorithm 4.1.2, the convergence depends on the parameter $\{\lambda_k\}$ with the condition $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$. So, the step size $\{\lambda_k\}$ can be considered in many ways. Applying step size concept of Shehu et al. [107], we then obtain the following modified viscosity type inertial extragradient stepsize algorithm for EPs:

Algorithm 4.1.11 (Modified viscosity type inertial extragradient step-size algorithm for EP)

Initialization: Select $\lambda_k \in (0, \frac{1}{2\max\{c_1,c_2\}}), \mu \in (0,1), \tau \in (0, \frac{1}{2}], \{\alpha_k\} \subset (0,1)$, and $\{\theta_k\} \subset [0, \frac{1}{3})$. **Iterative step:** Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = \alpha_k V(x_k) + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \operatorname*{argmin}_{y\in C}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. Calculate:

$$z_k = \operatorname*{argmin}_{y\in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. Calculate the next iteration via:

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

and

$$\lambda_{k+1} = \begin{cases} \min\{\frac{\mu}{2}\frac{\|w_k-y_k\|^2+\|z_k-y_k\|^2}{f(w_k,z_k)-f(w_k,y_k)-f(y_k,z_k)}, \lambda_k\}, & \text{if } f(w_k,z_k)-f(w_k,y_k)-f(y_k,z_k) > 0, \\ \lambda_k, & \text{Otherwise.} \end{cases}$$

Replace k by $k+1$ and return to **Step1.**

Remark 4.1.12 (i) Since $V(x) = x_0,\ \forall x \in H$ is a contraction, thus the modified Halpern inertial extragradient algorithm 4.1.10 converges strongly to $x^* = P_{EP(f,C)}x_0$ with Condition 4.1.1 and Condition 4.1.3;
(ii) Since the step size $\{\lambda_k\}$ in Algorithm 4.1.11 is a monotonically decreasing sequence with lower bound $\min\{\lambda_1, \frac{1}{2\max\{c_1,c_2\}}\}$ [107], thus Algorithm 4.1.11 converges strongly to the solution $x_0 = P_{EP(f,C)}V(x_0)$ by Theorem 4.1.9.

4.1.2 A modified viscosity type inertial subgradient extragradient algorithm for nonmonotone equilibrium problems and application to cardiovascular disease detection

Algorithm 4.1.13 (Modified viscosity type inertial subgradient extragradient algorithm - MVISE)
Initialization: Select $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}, \tau \in (0, \frac{1}{2}], \{\theta_k\} \subset [0, \frac{1}{3}), \{\alpha_k\} \subset (0,1)$ **Iterative step:** Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:
Step1. Compute:

$$\begin{aligned} w_k &= x_k + \theta_k(x_k - x_{k-1}), \\ y_k &= \alpha_k V(x_k) + (1-\alpha_k)w_k, \end{aligned}$$

and

$$z_k = \underset{y\in C}{\operatorname{argmin}}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - y_k\|^2\}.$$

Step 2. Choose $o_k \in \partial_2 f(y_k, \cdot)(z_k)$ such that there exists $s_k \in N_C(z_k)$ satisfying

$$s_k = y_k - \lambda_k o_k - z_k,$$

and construct a half-space

$$\Gamma_k = \{e \in H : \langle y_k - \lambda_k o_k - z_k, e - z_k\rangle \leqslant 0\}.$$

Compute

$$e_k = \underset{y \in \Gamma_k}{\operatorname{argmin}}\{\lambda_k f(z_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 3. Calculate:

$$x_{k+1} = (1-\tau)y_k + \tau e_k.$$

Replace k by $k+1$ and return to **Step1.**

Next, we give some useful lemmas for proving our main results.

Lemma 4.1.14 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be a sequence generated by Algorithm 4.1.13. Then, for all* $u \in EP(g, C)$*, there exists* $N > 0$ *such that*

$$\|x_{k+1} - u\|^2 \leqslant \|y_k - u\|^2 - \|x_{k+1} - y_k\|^2, \ k \geqslant N.$$

Proof. By Lemma 3.2.1, and the definition of z_k, we get

$$\frac{1}{\lambda_k}\langle y_k - z_k, y - z_k\rangle \leqslant f(y_k, y) - f(y_k, z_k), \ \forall y \in C. \tag{4.1.28}$$

Putting $y = e_k$ in (4.1.28), we obtain

$$\frac{1}{\lambda_k}\langle z_k - y_k, z_k - e_k\rangle \leqslant f(y_k, e_k) - f(y_k, z_k). \tag{4.1.29}$$

By the definition of e_k, we get

$$\frac{1}{\lambda_k}\langle y_k - e_k, y - e_k\rangle \leqslant f(z_k, y) - f(z_k, e_k), \ \forall y \in \Gamma_k. \tag{4.1.30}$$

(4.1.29) and (4.1.30) imply that

$$\begin{aligned}&\frac{2}{\lambda_k}\langle y_k - e_k, y - e_k\rangle + \frac{2}{\lambda_k}\langle z_k - y_k, z_k - e_k\rangle \\ &\leqslant 2f(z_k, y) + 2(f(y_k, e_k) - f(y_k, z_k) - f(z_k, e_k)).\end{aligned} \tag{4.1.31}$$

If $f(y_k, e_k) - f(y_k, z_k) - f(z_k, e_k) > 0$, then

$$f(y_k, e_k) - f(y_k, z_k) - f(z_k, e_k) \leqslant c_1\|y_k - z_k\|^2 + c_2\|e_k - z_k\|^2. \tag{4.1.32}$$

If $f(y_k, e_k) - f(y_k, z_k) - f(z_k, e_k) \leqslant 0$, then (4.1.32) is obviously true. It follows from (4.1.31) and (4.1.32) that

$$\begin{aligned}&\langle y_k - e_k, y - e_k\rangle + \langle z_k - y_k, z_k - e_k\rangle \\ &\leqslant \lambda_k f(z_k, y) + \lambda_k c_1\|y_k - z_k\|^2 + \lambda_k c_2\|e_k - z_k\|^2.\end{aligned} \tag{4.1.33}$$

Note that

$$\langle y_k - e_k, e_k - y\rangle = \frac{1}{2}(\|y_k - y\|^2 - \|y_k - e_k\|^2 - \|e_k - y\|^2) \tag{4.1.34}$$

and

$$\langle y_k - z_k, e_k - z_k\rangle = \frac{1}{2}(\|y_k - z_k\|^2 + \|e_k - z_k\|^2 - \|y_k - e_k\|^2). \tag{4.1.35}$$

By (4.1.33), (4.1.34), and (4.1.35), we have, $\forall y \in C$,

$$\|e_k - y\|^2 \leqslant 2\lambda_k f(z_k, y) - (1 - 2\lambda_k c_1)\|y_k - z_k\|^2$$

$$-(1-2\lambda_k c_2)\|e_k - z_k\|^2 + \|y_k - y\|^2. \tag{4.1.36}$$

With $y = u \in EP(f, C) \subset C$, one has $f(u, z_k) \geqslant 0, \forall k$. By Condition 4.1.1 (A1), we gain $f(z_k, u) \leqslant 0,\ \forall k$. Thus, we gain from (4.1.36) that

$$\|e_k - u\|^2 \leqslant \|y_k - u\|^2 - (1-2\lambda_k c_1)\|y_k - z_k\|^2 - (1-2\lambda_k c_2)\|e_k - z_k\|^2. \tag{4.1.37}$$

It follows from $\lambda_k \in (0, \frac{1}{2\max\{c_1, c_2\}})$ and (4.1.37) that we have

$$\|e_k - u\| \leqslant \|y_k - u\|.$$

On the other hand, we have

$$\|x_{k+1} - u\|^2 = (1-\tau)\|y_k - u\|^2 + \tau\|e_k - u\|^2 - (1-\tau)\tau\|e_k - y_k\|^2. \tag{4.1.38}$$

By (4.1.37) and (4.1.38), we have

$$\begin{aligned}\|x_{k+1} - u\|^2 \leqslant \|y_k - u\|^2 &- \tau\|y_k - u\|^2 + \tau\|y_k - u\|^2 \\ &- \tau(1-2\lambda_k c_1)\|y_k - z_k\|^2 - \tau(1-2\lambda_k c_2)\|e_k - z_k\|^2 \\ &- (1-\tau)\tau\|e_k - y_k\|^2. \end{aligned} \tag{4.1.39}$$

Moreover, we have $e_k - y_k = \frac{1}{\tau}(x_{k+1} - y_k)$, which, together with (4.1.39), gives

$$\begin{aligned}\|x_{k+1} - u\|^2 &\leqslant \|y_k - u\|^2 - \tau(1-2\lambda_k c_1)\|y_k - z_k\|^2 \\ &\quad - \tau(1-2\lambda_k c_2)\|e_k - z_k\|^2 - (1-\tau)\tau\frac{1}{\tau^2}\|x_{k+1} - y_k\|^2 \\ &\leqslant \|y_k - u\|^2 - \frac{1-\tau}{\tau}\|x_{k+1} - y_k\|^2 \\ &\leqslant \|y_k - u\|^2 - \epsilon\|x_{k+1} - y_k\|^2,\ \forall k \geqslant N, \end{aligned} \tag{4.1.40}$$

where $\epsilon = \frac{1-\tau}{\tau}$. □

Lemma 4.1.15 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be a sequence generated by Algorithm 4.1.13. Then, for all* $u \in EP(g, C)$,

$$\begin{aligned}
&- 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&\quad + (1 - \alpha_k)\theta_{k-1}\|x_{k-1} - u\|^2 - \theta_k\|x_k - u\|^2 + \alpha_{k+1}\|V(x_k) - x_{k+1}\|^2 \\
&\quad - \alpha_k\|x_k - V(x_k)\|^2 + (1 - 3\theta_{k+1} - \alpha_k)\|x_k - x_{k+1}\|^2. \qquad (4.1.41)
\end{aligned}$$

Proof. By Lemma 4.1.14, we get

$$\|x_{k+1} - u\|^2 \leqslant \|y_k - u\|^2 - \|x_{k+1} - y_k\|^2. \qquad (4.1.42)$$

In addition, from the definition of y_k, we gain that

$$\begin{aligned}
&\|y_k - u\|^2 \\
&= \|\alpha_k V(x_k) + (1 - \alpha_k)w_k - u\|^2 \\
&= \|\alpha_k V(x_k) + (1 - \alpha_k)x_k + (1 - \alpha_k)\theta_k(x_k - x_{k-1}) - u\|^2 \\
&= \|x_k - u\|^2 + \|(1 - \alpha_k)\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\|^2 \\
&\quad + 2\langle x_k - u, (1 - \alpha_k)\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\rangle \\
&= \|x_k - u\|^2 + \|(1 - \alpha_k)\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\|^2 \\
&\quad + 2(1 - \alpha_k)\theta_k\langle x_k - u, x_k - x_{k-1}\rangle - 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle. \qquad (4.1.43)
\end{aligned}$$

Replacing u by x_{k+1} in (4.1.43), we have

$$\begin{aligned}
\|y_k - x_{k+1}\|^2 = \|x_k - x_{k+1}\|^2 &+ 2(1 - \alpha_k)\theta_k\langle x_k - x_{k+1}, x_k - x_{k-1}\rangle \\
&- 2\alpha_k\langle x_k - x_{k+1}, x_k - V(x_k)\rangle \\
&+ \|(1 - \alpha_k)\theta_k(x_k - x_{k-1}) - \alpha_k(x_k - V(x_k))\|^2. \qquad (4.1.44)
\end{aligned}$$

Substituting (4.1.43) and (4.1.44) into (4.1.42), we have

$$\begin{aligned}
&\|x_{k+1}-u\|^2\\
&\leqslant \|x_k-u\|^2+\|(1-\alpha_k)\theta_k(x_k-x_{k-1})-\alpha_k(x_k-V(x_k))\|^2\\
&\quad +2(1-\alpha_k)\theta_k\langle x_k-u,x_k-x_{k-1}\rangle-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle-\|x_k-x_{k+1}\|^2\\
&\quad -2(1-\alpha_k)\theta_k\langle x_k-x_{k+1},x_k-x_{k-1}\rangle+2\alpha_k\langle x_k-x_{k+1},x_k-V(x_k)\rangle\\
&\quad -\|(1-\alpha_k)\theta_k(x_k-x_{k-1})-\alpha_k(x_k-V(x_k))\|^2\\
&=\|x_k-u\|^2+2(1-\alpha_k)\theta_k\langle x_k-u,x_k-x_{k-1}\rangle-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle\\
&\quad -\|x_k-x_{k+1}\|^2+2\alpha_k\langle x_k-x_{k+1},x_k-V(x_k)\rangle+(1-\alpha_k)\theta_k\|x_k-x_{k+1}\|^2\\
&\quad +(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|^2-(1-\alpha_k)\theta_k\|x_k-x_{k+1}+(x_k-x_{k-1})\|^2.
\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}
&\|x_{k+1}-u\|^2-\|x_k-u\|^2-(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|^2+\|x_k-x_{k+1}\|^2\\
&-(1-\alpha_k)\theta_k\|x_k-x_{k+1}\|^2\\
&\leqslant 2(1-\alpha_k)\theta_k\langle x_k-u,x_k-x_{k-1}\rangle-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle\\
&\quad +2\alpha_k\langle x_k-x_{k+1},x_k-V(x_k)\rangle\\
&=-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle-(1-\alpha_k)\theta_k\|x_{k-1}-u\|^2+(1-\alpha_k)\theta_k\|x_k-u\|^2\\
&\quad +(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|^2-\alpha_k\|V(x_k)-x_{k+1}\|^2+\alpha_k\|x_k-x_{k+1}\|^2\\
&\quad +\alpha_k\|x_k-V(x_k)\|^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
&-2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle\\
&\geqslant \|x_{k+1}-u\|^2-\|x_k-u\|^2-(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|^2+\|x_k-x_{k+1}\|^2\\
&\quad -(1-\alpha_k)\theta_k\|x_k-x_{k+1}\|^2+(1-\alpha_k)\theta_k\|x_{k-1}-u\|^2-(1-\alpha_k)\theta_k\|x_k-u\|^2\\
&\quad -(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|^2+\alpha_k\|V(x_k)-x_{k+1}\|^2
\end{aligned}$$

$$
\begin{aligned}
&- \alpha_k\|x_k - x_{k+1}\|^2 - \alpha_k\|x_k - V(x_k)\|^2 \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2(1-\alpha_k)\theta_k\|x_k - x_{k-1}\|^2 \\
&+ (1-\alpha_k)\theta_k\|x_{k-1} - u\|^2 - (1-\alpha_k)\theta_k\|x_k - u\|^2 + \alpha_k\|V(x_k) - x_{k+1}\|^2 \\
&- \alpha_k\|x_k - V(x_k)\|^2 + (1 - (1-\alpha_k)\theta_k - 2\theta_{k+1} - \alpha_k)\|x_{k+1} - x_k\|^2 \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&+ (1-\alpha_k)\theta_k\|x_{k-1} - u\|^2 - \theta_k\|x_k - u\|^2 + \alpha_k\|V(x_k) - x_{k+1}\|^2 \\
&- \alpha_k\|x_k - V(x_k)\|^2 + (1 - \theta_k - 2\theta_{k+1} - \alpha_k)\|x_{k+1} - x_k\|^2.
\end{aligned}
$$

As $\{\alpha_k\}$ is non-increasing and $\{\theta_k\}$ is non-decreasing, we then gain

$$
\begin{aligned}
&- 2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&+ (1-\alpha_k)\theta_{k-1}\|x_{k-1} - u\|^2 - \theta_k\|x_k - u\|^2 + \alpha_{k+1}\|V(x_k) - x_{k+1}\|^2 \\
&- \alpha_k\|x_k - V(x_k)\|^2 + (1 - 3\theta_{k+1} - \alpha_k)\|x_k - x_{k+1}\|^2.
\end{aligned}
$$

□

Lemma 4.1.16 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then, $\{x_k\}$ generated by Algorithm 4.1.13 is bounded.*

Proof. By (4.1.40) and Condition 4.1.3 (ii), we can find $K > 0$ such that

$$
\begin{aligned}
&\|x_{k+1} - u\| \\
&\leqslant \|y_k - u\| \\
&= \|\alpha_k V(x_k) + (1-\alpha_k)w_k - u\| \\
&= \|\alpha_k V(x_k) + (1-\alpha_k)x_k + (1-\alpha_k)\theta_k(x_k - x_{k-1}) - u\| \\
&\leqslant \alpha_k\|V(x_k) - u\| + (1-\alpha_k)\|x_k - u\| + (1-\alpha_k)\theta_k\|x_k - x_{k-1}\| \\
&= \alpha_k(\|V(x_k) - V(u)\| + \|V(u) - u\|) + (1-\alpha_k)\|x_k - u\|
\end{aligned}
$$

$$
\begin{aligned}
&+ (1-\alpha_k)\theta_k\|x_k - x_{k-1}\| \\
&\leqslant (\alpha_k\alpha + (1-\alpha_k))\|x_k - u\| + \alpha_k\|V(u) - u\| + \alpha_k\frac{(1-\alpha_k)\theta_k}{\alpha_k}\|x_k - x_{k-1}\| \\
&\leqslant (1-\alpha_k(1-\alpha))\|x_k - u\| + \alpha_k\|V(u) - u\| + \alpha_k K \\
&\leqslant (1-\alpha_k(1-\alpha))\|x_k - u\| + \alpha_k(1-\alpha)(\frac{\|V(u) - u\| + K}{1-\alpha}) \\
&\quad \vdots \\
&\leqslant \max\{\|x_1 - u\|, \frac{\|V(u) - u\| + K}{1-\alpha}\}.
\end{aligned}
$$

This implies that $\{x_k\}$ is bounded. □

Lemma 4.1.17 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be a sequence generated by Algorithm 4.1.13. For each* $k \geqslant 1$*, define*

$$
u_k = \|x_k - u\|^2 - \theta_{k-1}\|x_{k-1} - u\|^2 + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2.
$$

Then $u^k \geqslant 0, \ \forall k \geq 1$.

Proof. By the non-decreasingness of $\{\theta_k\}$ with $0 \leqslant \theta_k < \frac{1}{3}$, and $2\langle x, y\rangle = \|x\|^2 + \|y\|^2 - \|x - y\|^2$ for all $x, y \in H$, we obtain

$$
\begin{aligned}
u^k &= \|x_k - u\|^2 - \theta_{k-1}[\|x_{k-1} - x_k\|^2 + \|x_k - u\|^2 + 2\langle x_{k-1} - x_k, x_k - u\rangle] \\
&\quad + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2 \\
&= \|x_k - u\|^2 - \theta_{k-1}[2\|x_{k-1} - x_k\|^2 + 2\|x_k - u\|^2 - \|x_{k-1} - 2x_k + u\|^2] \\
&\quad + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2 \\
&\geqslant \|x_k - u\|^2 - 2\theta_k\|x_{k-1} - x_k\|^2 - \frac{2}{3}\|x_k - u\|^2 + \theta_{k-1}\|x_{k-1} - 2x_k + u\|^2 \\
&\quad + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2 \\
&\geqslant \frac{1}{3}\|x_k - u\|^2 + \alpha_k\|x_k - V(x_k)\|^2 \\
&\geqslant 0.
\end{aligned}
$$

□

Lemma 4.1.18 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let $\{x_k\}$ be a sequence generated by Algorithm 4.1.13. Suppose*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0,$$

and

$$\lim_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k \|x_k - u\|^2) = 0.$$

Then $\{x_k\}$ converges strongly to $u \in EP(f, C)$.

Proof. By the assumptions, we obtain

$$\lim_{k\to\infty} [(\|x_{k+1} - u\| - \sqrt{\theta_k}\|x_k - u\|)(\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\|)] = 0. \quad (4.1.45)$$

In the case

$$\lim_{k\to\infty} (\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\|) = 0,$$

we obtain that $\{x_k\}$ converges strongly to u. Suppose that $\lim_{k\to\infty}(\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\|) \neq 0$. Then there are a set of natural numbers N^* and $\rho > 0$ such that

$$\|x_{k+1} - u\| + \sqrt{\theta_k}\|x_k - u\| \geqslant \rho, \ \forall k \in N^*. \quad (4.1.46)$$

By (4.1.45) and the assumption $\theta_k \leqslant \theta < 1$, for $k \in N^*$ we have

$$\begin{aligned} 0 &= \lim_{k\to\infty} (\|x_{k+1} - u\| - \sqrt{\theta_k}\|x_k - u\|) \\ &\geqslant \limsup_{k\to\infty}(\|x_k - u\| - \|x_{k+1} - x_k\| - \sqrt{\theta_k}\|x_k - u\|) \end{aligned}$$

$$\geqslant \limsup_{k\to\infty}((1-\sqrt{\theta})\|x_k-u\|-\|x_{k+1}-x_k\|)$$
$$=(1-\sqrt{\theta})\limsup_{k\to\infty}\|x_k-u\|-\lim_{k\to\infty}\|x_{k+1}-x_k\|$$
$$=(1-\sqrt{\theta})\limsup_{k\to\infty}\|x_k-u\|.$$

This implies that $\limsup_{k\to\infty}\|x_k-u\|\leqslant 0$. It is obvious that $\liminf_{k\to\infty}\|x_k-u\|\geqslant 0$, we have $\lim_{k\to\infty}\|x_k-u\|=0$. It follows by (4.1.46) that

$$\begin{aligned}\|x_{k+1}-x_k\| &\geqslant \|x_{k+1}-u\|-\|x_k-u\|\\ &=\|x_{k+1}-u\|+\sqrt{\theta_k}\|x_k-u\|-(1+\sqrt{\theta_k})\|x_k-u\|\\ &\geqslant \frac{\rho}{2},\end{aligned}$$

for all $k\in N^*$ sufficiently large, which is a contradiction to $\lim_{k\to\infty}\|x_{k+1}-x_k\|=0$. Thus, the proof is Lemma 5.5.15 is completed. □

Theorem 4.1.19 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Then,* $\{x_k\}$ *generated by Algorithm 4.1.13 strongly converges to the solution* $u=P_{EP(f,C)}V(u)$.

Proof. From (4.1.41) and Lemma 5.5.14, we get

$$\begin{aligned}&u_{k+1}-u_k-\theta_{k-1}\|x_{k-1}-u\|^2+(1-\alpha_k)\theta_{k-1}\|x_{k-1}-u\|^2\\ &-\alpha_{k+1}\|x_{k+1}-V(x_{k+1})\|^2+\alpha_{k+1}\|V(x_k)-x_{k+1}\|^2\\ &+(1-3\theta_{k+1}-\alpha_k)\|x_k-x_{k+1}\|^2\\ &\leqslant -2\alpha_k\langle x_k-u,x_k-V(x_k)\rangle.\end{aligned} \tag{4.1.47}$$

From the Banach fixed point theorem, there exists uniquely $u=P_{EP(f,C)}V(u)$. By Lemma 4.1.14, we have

$$
\begin{aligned}
&\|x_{k+1}-u\|^2\\
&\leqslant \|y_k-u\|^2\\
&= \|\alpha_k(V(x_k)-u)+(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2\\
&\leqslant \|(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2\\
&\quad +2\langle\alpha_k(V(x_k)-u),y_k-u\rangle\\
&= \|(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2\\
&\quad +2\alpha_k\langle V(x_k)-V(u),y_k-u\rangle+2\alpha_k\langle V(u)-u,y_k-u\rangle\\
&\leqslant \|(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2+2\alpha_k\langle V(u)-u,y_k-u\rangle\\
&\quad +2\alpha_k\alpha\|x_k-u\|\|y_k-u\|\\
&\leqslant \|(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2+2\alpha_k\langle V(u)-u,y_k-u\rangle\\
&\quad +\alpha_k\alpha(\|x_k-u\|^2+\|y_k-u\|^2)\\
&\leqslant \frac{1}{1-\alpha_k\alpha}\Big(\|(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\|^2+\alpha_k\alpha\|x_k-u\|^2\\
&\quad +2\alpha_k\langle V(u)-u,y_k-u\rangle\Big)\\
&\leqslant \frac{1}{1-\alpha_k\alpha}\Big(\|(1-\alpha_k)(x_k-u)\|^2+\alpha_k\alpha\|x_k-u\|^2\\
&\quad +2\alpha_k\langle V(u)-u,y_k-u\rangle\\
&\quad +2\langle(1-\alpha_k)\theta_k(x_k-x_{k-1}),(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\rangle\Big)\\
&= \frac{(1-\alpha_k)^2+\alpha_k\alpha}{1-\alpha_k\alpha}\|x_k-u\|^2+\frac{1}{1-\alpha_k\alpha}\Big(2\alpha_k\langle V(u)-u,y_k-u\rangle\\
&\quad +2\langle(1-\alpha_k)\theta_k(x_k-x_{k-1}),(1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\rangle\Big)\\
&= \Big(1-(\frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}-\frac{(\alpha_k)^2}{1-\alpha_k\alpha})\Big)\|x_k-u\|^2\\
&\quad +\frac{1}{1-\alpha_k\alpha}\Big(2\langle(1-\alpha_k)\theta_k(x_k-x_{k-1}),\\
&\quad (1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\rangle+2\alpha_k\langle V(u)-u,y_k-u\rangle\Big)
\end{aligned}
$$

$$\begin{aligned}
&\leqslant \left(1-\frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}\right)\|x_k-u\|^2 \\
&\quad + \frac{2\alpha_k(1-\alpha)}{1-\alpha_k\alpha}\left(\frac{\alpha_k}{2(1-\alpha)}\|x_k-u\|^2+\frac{1}{\alpha_k(1-\alpha)}\langle(1-\alpha_k)\theta_k(x_k-x_{k-1}),\right. \\
&\quad (1-\alpha_k)(x_k-u)+(1-\alpha_k)\theta_k(x_k-x_{k-1})\rangle \\
&\quad \left. + \frac{1}{1-\alpha}\langle V(u)-u, y_k-u\rangle\right). \qquad (4.1.48)
\end{aligned}$$

Next, we shall reconsider into 2 cases. Case 1: If $u_{k+1} \leqslant u_k + t_k$ for all $k \geqslant k_0$ for some $k_0 \in \mathbb{N}$, $t_k \geqslant 0$, and $\sum_{k=1}^{\infty} t_k < +\infty$, by Lemma 3.2.4 we have that $\lim_{k\to\infty} u_k = \lim_{k\to\infty} u_{k+1}$ exists. Since $\{x_k\}$ is bounded by Lemma 5.5.13, we can find $M_1 > 0$ such that $2|\langle x_k-u, x_k-V(x_k)\rangle|+\theta_{k-1}\|x_{k-1}-u\|^2 \leqslant M_1$ and $M_2 > 0$ such that $\|x_{k+1}-V(x_{k+1})\|^2 \leqslant M_2$. Since $0 \leqslant \theta_k \leqslant \theta_{k+1} \leqslant \theta < \frac{1}{3}$ and $\lim_{k\to\infty} \alpha_k = 0$, there exist $N \in \mathbb{N}$ and $\gamma_1 > 0$ such that $1-3\theta_{k+1}-\alpha_k > \gamma_1$ for all $k \geqslant N$. As a result, for $k \geqslant N$, we gain from (4.1.47) that

$$\gamma_1\|x_{k+1}-x_k\|^2 \leqslant \alpha_k M_1 + \alpha_{k+1}M_2 + u_k - u_{k+1} \to 0, \qquad (4.1.49)$$

as $k \to \infty$. Thus, $\lim_{k\to\infty}\|x_{k+1}-x_k\| = 0$. For $u \in EP(f,C)$, we have

$$\begin{aligned}
&\|y_k-u\|^2 \\
&= \|\alpha_k f(x_k)+(1-\alpha_k)x_k+(1-\alpha_k)\theta_k(x_k-x_{k-1})-u\|^2 \\
&\leqslant \|\alpha_k V(x_k)+(1-\alpha_k)x_k-u\|^2+2\langle(1-\alpha_k)\theta_k(x_k-x_{k-1}), y_k-u\rangle \\
&\leqslant \alpha_k\|V(x_k)-u\|^2+(1-\alpha_k)\|x_k-u\|^2+2(1-\alpha_k)\theta_k\|x_k-x_{k-1}\|\|y_k-u\| \\
&\leqslant \alpha_k\|V(x_k)-u\|^2+(1-\alpha_k)\|x_k-u\|^2+\frac{2(1-\alpha_k)\theta_k}{\alpha_k}\|x_k-x_{k-1}\|\|y_k-u\| \\
&\leqslant \alpha_k\|V(x_k)-u\|^2+\|x_k-u\|^2+\frac{2(1-\alpha_k)\theta_k}{\alpha_k}\|x_k-x_{k-1}\|\|y_k-u\|,
\end{aligned}$$

and from (4.1.39) we have

$$\begin{aligned}\|x_{k+1}-u\|^2 &= \|y_k-u\|^2-\tau(1-2\lambda_k c_1)\|y_k-\gamma_k\|^2-\tau(1-2\lambda_k c_2)\|e_k-z_k\|^2\\&\quad-(1-\tau)\tau\frac{1}{\tau^2}\|x_{k+1}-y_k\|^2\\&\leqslant \alpha_k\|V(x_k)-u\|^2+\|x_k-u\|^2+\frac{2(1-\alpha_k)\theta_k}{\alpha_k}\|x_k-x_{k-1}\|\|y_k-u\|\\&\quad-\tau(1-2\lambda_k c_1)\|y_k-z_k\|^2-\tau(1-2\lambda_k c_2)\|e_k-z_k\|^2\\&\quad-\frac{1-\tau}{\tau}\|x_{k+1}-y_k\|^2.\end{aligned}$$

This implies that

$$\begin{aligned}&\tau(1-2\lambda_k c_1)\|y_k-z_k\|^2+\tau(1-2\lambda_k c_2)\|e_k-z_k\|^2+\frac{1-\tau}{\tau}\|x_{k+1}-y_k\|^2\\&\leqslant \alpha_k\|V(x_k)-u\|^2+\|x_k-u\|^2+\frac{2(1-\alpha_k)\theta_k}{\alpha_k}\|x_k-x_{k-1}\|\|y_k-u\|\\&\quad-\|x_{k+1}-u\|^2.\end{aligned}$$

Then, it follows from (4.1.49) and our condition that

$$\lim_{k\to\infty}\|y_k-z_k\|=\lim_{k\to\infty}\|e^k-z_k\|=\lim_{k\to\infty}\|x_{k+1}-y_k\|=0. \tag{4.1.50}$$

Since $\{x_k\}$ is bounded, we can find a subsequence $\{x_{k_i}\}$ of $\{x_k\}$ such that $x_{k_i}\rightharpoonup x^*$ for some $x^*\in H$. From (4.1.49) and (4.1.50), we gain $y_{k_i}\rightharpoonup x^*$ and $z_{k_i}\rightharpoonup x^*$ as $i\to\infty$. By (4.1.32) and the definition of e_k, we obtain

$$\begin{aligned}&\lambda_{k_i}f(z_{k_i},y)\\&\geqslant \lambda_{k_i}f(z_{k_i},e_{k_i})+\langle y_{k_i}-e_{k_i},y-e_{k_i}\rangle\\&\geqslant \lambda_{k_i}f(y_{k_i},e_{k_i})-\lambda_{k_i}f(y_{k_i},z_{k_i})-c_1\|y_{k_i}-z_{k_i}\|^2-c_2\|e_{k_i}-z_{k_i}\|^2\\&\quad+\langle y_{k_i}-e_{k_i},y-e_{k_i}\rangle\\&\geqslant \langle z_{k_i}-y_{k_i},z_{k_i}-e_{k_i}\rangle+\langle y_{k_i}-e_{k_i},y-e_{k_i}\rangle-c_1\|y_{k_i}-z_{k_i}\|^2-c_2\|e_{k_i}-z_{k_i}\|^2.\end{aligned}$$

It follows from this that $\{e_{k_i}\}$ is bounded, and from $0 < \lambda_{k_i} \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$ and Condition 4.1.1 (A3) that $0 \leqslant \limsup_{i\to\infty} f(z_{k_i}, y) \leqslant f(x^*, y)\ \forall y \in H$. These yield that $f(x^*, y) \geqslant 0\ \forall y \in C$, which implies that $x^* \in EP(f, C)$. Therefore, we gain

$$\begin{aligned}\limsup_{k\to\infty}\langle V(u) - u, y_k - u\rangle &= \lim_{i\to\infty}\langle V(u) - u, y_{k_i} - u\rangle \\ &= \langle V(u) - u, x^* - u\rangle \\ &\leqslant 0, \end{aligned} \tag{4.1.51}$$

by $u = P_{EP(f,C)}V(u)$. Using (4.1.51) and (4.1.48), by Lemma 3.2.3 we are able to summarize that $x_k \to u = P_{EP(f,C)}V(u)$ as $k \to \infty$.

Case 2: In another case of $\{u_k\}$, we give $\phi : \mathbb{N} \to \mathbb{N}$ as the map defined for all $k \geqslant k_0$ (for some $k_0 \in \mathbb{N}$ large enough) by

$$\phi(k) = \max\{i \in \mathbb{N} : i \leqslant k, u_k + t_k \leqslant u_{k+1}\}.$$

By the same proof of Case 1 and using (4.1.49), we have

$$\gamma_1\|x_{\phi(k)+1} - x_{\phi(k)}\|^2 \leqslant \alpha_{\phi(k)}M_1 + \alpha_{\phi(k)+1}M_2 + u_{\phi(k)} - x_{\phi(k)+1} \to 0$$

for some constants $M_1, M_2 > 0$. So,

$$\lim_{k\to\infty}\|x_{\phi(k)+1} - x_{\phi(k)}\| = 0. \tag{4.1.52}$$

By the identical proof used in the proof of Case 1, we obtain

$$\lim_{k\to\infty}\|x_{\phi(k)+1} - y_{\phi(k)}\| = \lim_{k\to\infty}\|y_{\phi(k)} - x_{\phi(k)}\| = \lim_{k\to\infty}\|x_{\phi(k)} - e_{\phi(k)}\| = 0. \tag{4.1.53}$$

For $j \geqslant 0$ by (4.1.47), we have $u_{j+1} < u_j + t_j$ when $x_j \notin \Omega = \{x \in H : \langle x - V(x), x - u\rangle \leqslant 0\}$. Hence, $x_{\phi(k)} \in \Omega$ for all $k \geqslant k_0$ since $u_{\phi(k)} + t_{\phi(k)} \leqslant$

$u_{\phi(k)+1}$. Since $\{x_{\phi(k)}\}$ is bounded, let $\{x_{\phi(k)}\}$ be a subsequence of $\{x_{\phi(k)}\}$ such that $x_{\phi(k)} \rightharpoonup x^*$, $x^* \in H$. Since Ω is a closed and convex set, so $x^* \in \Omega$. Using (4.1.53), we have $e_{\phi(k)} \rightharpoonup x^*$. By the same proof in Case 1, we obtain $x_{\phi(k)} \rightharpoonup u = P_{EP(f,C)}V(u)$. Furthermore,

$$\begin{aligned}\|x_{\phi(k)} - u\|^2 &= \langle x_{\phi(k)} - V(x_k), x_{\phi(k)} - u\rangle - \langle u - V(x_k), x_{\phi(k)} - u\rangle \\ &\leqslant -\langle u - V(x_k), x_{\phi(k)} - u\rangle,\end{aligned}$$

due to $x_{\phi(k)} \in \Omega$. This gives

$$\limsup_{k\to\infty} \|x_{\phi(k)} - u\| \leqslant 0. \tag{4.1.54}$$

Accordingly,

$$\lim_{k\to\infty} \|x_{\phi(k)} - u\| = 0.$$

By definition, we have

$$\begin{aligned}u_{\phi(k)+1} &= \|x_{\phi(k)+1} - u\|^2 - \theta_{\phi(k)}\|x_{\phi(k)} - u\|^2 + 2\theta_{\phi(k)+1}\|x_{\phi(k)+1} - x_{\phi(k)}\|^2 \\ &\quad + \alpha_{\phi(k)+1}\|x_{\phi(k)+1} - V(x_{\phi(k)+1})\|^2 \\ &\leqslant (\|x_{\phi(k)+1} - x_{\phi(k)}\| + \|x_{\phi(k)} - u\|)^2 - \theta_{\phi(k)}\|x_{\phi(k)} - u\|^2 \\ &\quad + 2\theta_{\phi(k)+1}\|x_{\phi(k)+1} - x_{\phi(k)}\|^2 + \alpha_{\phi(k)+1}\|x_{\phi(k)+1} - V(x_{\phi(k)+1})\|^2.\end{aligned}$$

By our Condition 4.1.3 (i), (4.1.52), and (4.1.54), we gain $\lim_{k\to\infty} u_{\phi(k)+1} = 0$. We shall show that $\lim\limits_{k\to\infty} u_k = 0$. For $k \geqslant k_0$, one has $u_k + t_k \leqslant u_{\phi(k)+1}$ if $k \neq \phi(k)$, since $\lim_{k\to\infty} t_k = 0$, hence $\limsup\limits_{k\to\infty} u_k \leqslant 0$. By Lemma 5.5.14 we get $\liminf\limits_{k\to\infty} u_k \geqslant 0$. Hence, we obtain $\lim\limits_{k\to\infty} u_k = 0$. It follows from this that $\{x_k\}$ is bounded, and from $\lim\limits_{k\to\infty} \alpha_k = 0$ and equation (4.1.47) that $\lim\limits_{k\to\infty} \|x_k - x_{k+1}\| = 0$. From the definition of u_k, we have $\lim\limits_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k\|x_k - u\|^2) = 0$. By

Lemma 5.5.15, $x_k \to u = P_{EP(f,C)}V(u)$. Theorem 4.1.19 is completed. □

Setting $V(x) = x_0$, $\forall x \in H$, then we obtain the following modified Halpern inertial subgradient extragradient algorithm for EPs:

Algorithm 4.1.20 (Modified Halpern inertial subgradient extragradient algorithm - MHISE)

Initialization: Select $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}, \tau \in (0, \frac{1}{2}], \{\theta_k\} \subset [0, \frac{1}{3}), \{\alpha_k\} \subset (0,1)$ **Iterative step:** Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$\begin{aligned} w_k &= x_k + \theta_k(x_k - x_{k-1}), \\ y_k &= \alpha_k x_0 + (1-\alpha_k)w_k, \end{aligned}$$

and

$$z_k = \operatorname*{argmin}_{y \in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 2. Choose $o_k \in \partial_2 f(y_k, \cdot)(z_k)$ such that there exists $s_k \in N_C(z_k)$ satisfying

$$s_k = y_k - \lambda_k o_k - z_k,$$

and construct a half-space

$$\Gamma_k = \{e \in H : \langle y_k - \lambda_k o_k - z_k, e - z_k\rangle \leqslant 0\}.$$

Compute

$$e_k = \operatorname*{argmin}_{y \in \Gamma_k}\{\lambda_k f(z_k, y) + \frac{1}{2}\|y - y_k\|^2\}.$$

Step 3. Calculate:

$$x_{k+1} = (1-\tau)y_k + \tau e_k.$$

Replace k by $k+1$ and return to **Step1.**

Remark 4.1.21 (i) The advantage of Algorithm 4.1.13 lies in its step size $\{\lambda_k\}$, which can be chosen within the interval $(0, \frac{1}{2\max\{c_1,c_2\}})$. This flexibility makes it particularly suitable for data classification tasks in machine learning, especially when the dataset has a finite number of features.
(ii) The algorithm by Yao et al. [124] features a fixed format for the step size $\{\lambda_k\}$, with a lower bound given by $\min\left\{\frac{\theta\mu}{4\max\{c_1,c_2\}}, \lambda_0\right\}$. This approach is practical without knowing the Lipschitz constants and can be applied to image restoration and signal recovery tasks involving sparse matrices.

4.2 Split equilibrium problems

Let $C, E \subset H_1$, $Q \subset H_2$ be nonempty, closed and convex subsets of real Hilbert space H_1 and H_2, respectively. Assume that $A : H_1 \to H_2$ is a bounded linear operator. Let $f : C\times C \to \mathbb{R}$, $g : Q\times Q \to \mathbb{R}$ be bifunction satisfying Assumption 3.1.41 and g is upper semi-continuous in the first argument. Let L be the spectral radius of A^TA and A^T be the adjoint of A. Assume that $V : H \to H$ is contraction with contraction constant α. Denote that $\omega = \{x^* \in EP(f) \text{ and } Ax^* \in EP(g)\}$ is the solution of SEP.

Condition 4.2.1 Assume that the following conditions hold:
(C1) $\{\beta_k\} \subset (0, \frac{1}{L})$;
(C2) $\sum_{k=1}^{\infty} \theta_k\|x_k - x_{k-1}\| < +\infty$;
(C3) $0 < \liminf_{k\to\infty} \alpha_k \leqslant \limsup_{k\to\infty} \alpha_k < 1$;
(C4) $0 < \liminf_{k\to\infty} \beta_k \leqslant \limsup_{k\to\infty} \beta_k < \dfrac{1}{L}$.

4.2.1 An inertial projective Mann algorithm for solving split equilibrium problems classification to Parkinson's disease

Algorithm 4.2.2 (An inertial projective Mann algorithm)

Initialization: Select $\{\alpha_k\} \subset (0,1), \{\theta_k\} \subset [0, \frac{1}{3}), \{r_k\} \subset (0, \infty), \{\beta_k\} \subset (0, \frac{1}{L})$

Iterative step: Let $x_0, x_1 \in C$ arbitrarily and start $k = 0$. Calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k,$$

Step 2. Calculate:

$$x_{k+1} = P_E(\alpha_k w_k + (1 - \alpha_k)y_k).$$

Replace k by $k+1$ and return to **Step1.**

Theorem 4.2.3 *Suppose that Condition 4.2.1 hold. Then the sequence $\{x^k\}$ generated by Algorithm 4.2.2 converges weakly to $x^* \in \omega \cap E$.*

Proof. Since $T^g_{r_k}$ is firmly nonexpansive and $I - T^g_{r_k}$ is 1-inverse strongly monotone, $A^T(I - T^g_{r_k})A$ is an $\frac{1}{L}-$inverse strongly and monotone mapping. Indeed,

$$\|A^T(I - T^g_{r_k})Ax - A^T(I - T^g_{r_k})Ay\|^2$$
$$= \langle A^T(I - T^g_{r_k})(Ax - Ay), A^T(I - T^g_{r_k})(Ax - Ay)\rangle$$

$$= \langle (I - T^g_{r_k})(Ax - Ay), AA^T(I - T^g_{r_k})(Ax - Ay)\rangle$$
$$\leqslant L\langle (I - T^g_{r_k})(Ax - Ay), (I - T^g_{r_k})(Ax - Ay)\rangle$$
$$= L\|(I - T^g_{r_k})(Ax - Ay)\|^2$$
$$\leqslant L\langle Ax - Ay, (I - T^g_{r_k})(Ax - Ay)\rangle$$
$$= L\langle x - y, A^T(I - T^g_{r_k})Ax - A^T(I - T^g_{r_k})Ay\rangle$$

for all $x, y \in H_1$. This implies that $A^T(I - T^g_{r_k})A$ is an $\frac{1}{L}-$inverse strongly and monotone mapping. As $\{\beta_k\} \subset (0, \frac{1}{L})$, $I - \beta_k A^T(I - T^g_{r_k})A$ is nonexpansive by Proposition 3.1.42.

Step 1. Show that $\{x^k\}$ is bounded.

Let $x^* \in \omega \cap E$. Then $x^* = T^f_{r_k}x^*$ and $(I - \beta_k A^T(I - T^g_{r_k})A)x^* = x^*$ by Lemma 3.2.9. Since $T^f_{r_k}$ and P_E are nonexpansive in a Hilbert space, we have

$$\begin{aligned}\|y_{k+1} - x^*\| &= \|P_E(\alpha_k w_k + (1 - \alpha_k)y_k) - x^*\| \\ &\leqslant \alpha_k\|w_k - x^*\| + (1 - \alpha_k)\|y_k - x^*\| \\ &\leqslant \|w_k - x^*\| \\ &\leqslant \|x_k - x^*\| + \theta_k\|x_k - x_{k-1}\|.\end{aligned}$$

From Lemma 3.2.7 and Condition 4.2.1 (C2), then we have $\lim_{k\to\infty} \|x^k - x^*\|$ exists. This implies that $\{x^k\}$ is bounded. By the definition of $\{w_k\}$, $\{w_k\}$ is also bounded.

Step 2. Show that $\lim_{k\to\infty} \|y_k - w_k\| = 0$. Since $A^T(I - T^g_{r_k})A$ is $\frac{1}{L}-$Lipchitz continuous monotone, $I - \beta_k A^T(I - T^g_{r_k})A$ is nonexpansive and $T^f_{r_k}$ is firmly nonexpansive, we have

$$\begin{aligned}\|y_{k+1} - x^*\|^2 &= \|P_E(\alpha_k w_k + (1 - \alpha_k)y_k) - x^*\|^2 \\ &\leqslant \|\alpha_k(w_k - x^*) + (1 - \alpha_k)(y_k - x^*)\|^2\end{aligned}$$

$$\leqslant \alpha_k\|w_k - x^*\|^2 + (1-\alpha_k)\|y_k - x^*\|^2 - \alpha_k(1-\alpha_k)\|w_k - y_k\|^2$$
$$\leqslant \|w_k - x^*\|^2 - \alpha_k(1-\alpha_k)\|w_k - y_k\|^2$$
$$\leqslant \|x_k + \theta_k(x_k - x_{k-1}) - x^*\|^2 - \alpha_k(1-\alpha_k)\|w_k - y_k\|^2$$
$$\leqslant \|x_k - x^*\|^2 + 2\langle \theta_k(x_k - x_{k-1}), w_k - x^*\rangle - \alpha_k(1-\alpha_k)\|w_k - y_k\|^2$$
$$\leqslant \|x_k - x^*\|^2 + 2\theta_k\|x_k - x_{k-1}\|\|w_k - x^*\| - \alpha_k(1-\alpha_k)\|w_k - y_k\|^2.$$

This implies that

$$\alpha_k(1-\alpha_k)\|w_k - y_k\|^2 \leqslant \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + 2\theta_k\|x_k - x_{k-1}\|\|w_k - x^*\|$$

by Condition 4.2.1 (C2)-(C3) and $\lim_{k\to\infty}\|x_k - x^*\|$ exists, we have

$$\lim_{k\to\infty}\|w_k - y_k\| = 0. \tag{4.2.1}$$

Step 3. Show that $\lim_{k\to\infty}\|Aw_k - T^g_{r_k}Aw_k\| = 0$. Again by $T^f_{r_k}$ is firmly nonexpnsive, we have

$$\|y_k - x^*\|^2$$
$$= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - x^*\|^2$$
$$= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_k}x^*\|^2$$
$$\leqslant \|w_k - \beta_k A^T(I - T^g_{r_k})Aw_k - x^*\|^2$$
$$= \|w_k - x^*\|^2 + \beta_k^2\|A^T(I - T^g_{r_k})Aw_k\|^2 + 2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})Aw_k\rangle$$
$$\leqslant \|w_k - x^*\|^2 + \beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, AA^T(I - T^g_{r_k})Aw_k\rangle$$
$$+ 2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})Aw_k\rangle. \tag{4.2.2}$$

On the other hand, we have

$$\beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, AA^T(I - T^g_{r_k})Aw_k\rangle \leqslant L\beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, Aw_k - T^g_{r_k}Aw_k\rangle$$
$$= L\beta_k^2\|Aw_k - T^g_{r_k}Aw_k\|^2 \qquad (4.2.3)$$

and

$$\begin{aligned}
&2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})Aw_k\rangle \\
&= 2\beta_k\langle A(x^* - w_k), Aw_k - T^g_{r_k}Aw_k\rangle \\
&= 2\beta_k\langle A(x^* - w_k) + (Aw_k - T^g_{r_k}Aw_k) - (Aw_k - T^g_{r_k}Aw_k), Aw_k - T^g_{r_k}Aw_k\rangle \\
&= 2\beta_k\{\langle Ax^* - T^g_{r_k}Aw_k, Aw_k - T^g_{r_k}Aw_k\rangle - \|Aw_k - T^g_{r_k}Aw_k\|^2\} \\
&\leqslant 2\beta_k\{\frac{1}{2}\|Aw_k - T^g_{r_k}Aw_k\|^2 - \|Aw_k - T^g_{r_k}Aw_k\|^2\} \\
&< -\beta_k\|Aw_k - T^g_{r_k}Aw_k\|^2. \qquad (4.2.4)
\end{aligned}$$

Using (4.2.2), (4.2.3) and (4.2.4), we have

$$\begin{aligned}
\|y_k - x^*\|^2 &\leqslant \|w_k - x^*\|^2 + L\beta_k^2\|Aw_k - T^g_{r_k}Aw_k\|^2 - \beta_k\|Aw_k - T^g_{r_k}Aw_k\|^2 \\
&= \|w_k - x^*\|^2 + \beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2. \qquad (4.2.5)
\end{aligned}$$

It follows that

$$\begin{aligned}
&\|y_{k+1} - x^*\|^2 \\
&= \|P_E(\alpha_k w_k + (1 - \alpha_k)y_k) - x^*\|^2 \\
&\leqslant \alpha_k\|w_k - x^*\|^2 + (1 - \alpha_k)\|y_k - x^*\| \\
&= \alpha_k\|w_k - x^*\|^2 + (1 - \alpha_k)(\|w_k - x^*\|^2 + \beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2) \\
&\leqslant \|w_k - x^*\|^2 + (1 - \alpha_k)\beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2 \\
&\leqslant \|x^k - x^*\|^2 + 2\langle\theta_k(x_k - x_{k-1}), w_k - x^*\rangle + (1 - \alpha_k)\beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2.
\end{aligned}$$

Therefore, we have

$$(1-\alpha_k)\beta_k(1-L\beta_k)\|Aw_k - T^g_{r_k}Aw_k\|^2 \leqslant \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + 2\langle \theta_k(x_k - x_{k-1}), w_k - x^*\rangle$$

Since Condition 4.2.1 (C3)-(C4) and $\lim_{k\to\infty}\|x_k - x^*\|$ exists, we obtain

$$\lim_{k\to\infty}\|Aw_k - T^g_{r_k}Aw_k\| = 0. \qquad (4.2.6)$$

Step 4. Show that $\|y_k - x_k\| \to 0$ as $k \to \infty$.

We have

$$\begin{aligned}\|w_k - x_k\| &= \|x_k + \theta_k(x_k - x_{k-1}) - x_k\| \\ &\leqslant \theta_k\|x_k - x_{k-1}\| \ \to 0 \qquad (4.2.7)\end{aligned}$$

as $k \to \infty$.

By combining (4.2.1) and (4.2.7), then we have

$$\|y_k - x_k\| \leqslant \|y_k - w_k\| + \|w_k - x_k\| \ \to 0 \qquad (4.2.8)$$

as $k \to \infty$.

Step 5. Let $\varrho_W(x_k) = \{x \in H : x_{k_i} \rightharpoonup x, \{x_{k_i}\} \subset \{x_k\}\}$. From the reflexiveness of H, we have $\varrho_W(x_k) \neq \emptyset$. Let $l \in \varrho_W(x_k)$, there exists a subsequence $\{x_{k_i}\} \subset \{x_k\}$ converging weakly to l. Since $\{x_k\}$ is a sequence in E for all $k \geqslant 2$ and E is a closed convex set, $l \in E$. From (4.2.8), it follows that $y_{k_i} \rightharpoonup l$ as $i \to \infty$.

Next we show that $l \in EP(f)$. From $y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k$, we have

$$f(y_k, y) + \frac{1}{r_k}\langle y - y_k, y_k - w_k - \beta_k A^T(I - T^g_{r_k})Aw_k\rangle \geqslant 0$$

for all $y \in C$, which implies that

$$f(y_k, y) + \frac{1}{r_k}\langle y - y_k, y_k - w_k\rangle - \frac{1}{r_k}\langle y - y_k, \beta_k A^T(I - T^g_{r_k})Aw_k\rangle \geqslant 0$$

for all $y \in C$. By Assumption 3.1.41 (1), we have

$$\frac{1}{r_{k_i}}\langle y - y_{k_i}, y_{k_i} - w_{k_i}\rangle - \frac{1}{r_{k_i}}\langle y - y_{k_i}, \beta_{k_i} A^T(I - T^g_{r_{k_i}})Aw_{k_i}\rangle \geqslant f(y, y_{k_i})$$

for all $y \in C$. From $\liminf_{k\to\infty} r_k > 0$, from Assumption 3.1.41 (3), (4.2.1), and (4.2.6), we gain

$$f(y, l) \leqslant 0$$

for all $y \in C$. For each $0 < c \leqslant 1$ and $y \in C$, define $y^c = cy + (1-c)l$. As $y \in C$ and $l \in C, y^c \in C$, and therefore $f(y^c, l) \leqslant 0$. By Assumption 3.1.41 (2), we have

$$0 = f(y^c, y^c) \leqslant cf(y^c, y) + (1-c)f(y^c, l) \leqslant cf(y^c, y)$$

and hence $f(y^c, y) \geqslant 0$. By Assumption 3.1.41 (4), $f(l, y) \geqslant 0$ for all $y \in C$ and therefore $l \in EP(f)$. As A is a bounded linear operator, $Ax_{k_i} \rightharpoonup Al$. Then from (4.2.6), we have

$$T^g_{r_{k_i}} Ax_{k_i} \rightharpoonup Al \tag{4.2.9}$$

as $i \to \infty$. By the definition of $T^g_{r_{k_i}} Ax_{k_i}$, we have

$$g(T^g_{r_{k_i}} Ax_{k_i}, y) + \frac{1}{r_{k_i}}\langle y - T^g_{r_{k_i}} Ax_{k_i}, T^g_{r_{k_i}} Ax_{k_i} - Ax_{k_i}\rangle \geqslant 0$$

for all $y \in C$. By in the first argument of g is upper semi-continuous, it follows from (4.2.9) that

$$g(Al, y) \geqslant 0$$

for all $y \in C$. This shows that $Al \in EP(g)$. Thus $l \in \omega \cap E$.

Step 6. We will show that $\{x_k\}$ and $\{y_k\}$ converge weakly to an element of $\omega \cap E$. It is sufficient to show that $\varrho_W(x_k)$ is singleton. Let $x^*, l \in \varrho_W(x_k)$ and $\{x_{k_n}\}, \{x_{k_m}\} \subset \{x_k\}$ be such that $x_{k_n} \rightharpoonup x^*$ and $x_{k_m} \rightharpoonup l$. From (4.2.8), we also have $y_{k_n} \rightharpoonup x^*$ and $y_{k_m} \rightharpoonup l$. Using Lemma 3.2.5, we gain $x^* = l$. This completes the proof. □

4.2.2 A new double relaxed inertial viscosity-type algorithm for solving split equilibrium problems application to osteoporosis detection

Algorithm 4.2.4 (Double relaxed inertial viscosity-type algorithm)

Initialization: Select $\{\theta_k\}, \{\delta_k\} \subset (-\infty, \infty), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty), \{\alpha_k\} \subset (0, 1)$. **Iterative step:** Let $x_{-2}, x_{-1}, x_0 \in C$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = x_k + \theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. Calculate:

$$x_{k+1} = \alpha_k V(x_k) + (1-\alpha_k)y_k.$$

Replace k by $k+1$ and return to **Step1.**

Condition 4.2.5 Assume that the following conditions hold:

(C1) $\lim\limits_{k\to\infty} \alpha_k = 0; \sum\limits_{k=1}^{\infty} \alpha_k = \infty; \sum\limits_{k=0}^{\infty} |\alpha_{k+1} - \alpha_k| < \infty;$

(C2) $\liminf\limits_{k\to\infty} r_k > 0; \sum\limits_{k=0}^{\infty} |r_{k+1} - r_k| < \infty;$

(C3) $0 < \liminf\limits_{k\to\infty} \beta_k \leqslant \limsup\limits_{k\to\infty} \beta_k < \frac{1}{L};$

(C4) $\lim\limits_{k\to\infty} \frac{|\theta_k|}{\alpha_k}\|x_k - x_{k-1}\| = 0; \lim\limits_{k\to\infty} \frac{|\delta_k|}{\alpha_k}\|x_{k-1} - x_{k-2}\| = 0.$

Theorem 4.2.6 *Suppose that Condition 4.2.5 hold. Let $\{x_k\}$ be a sequence defined by Algorithm 4.2.4. Then the sequence $\{x_k\}$ converges strongly to $x^* = P_\omega V(x^*)$, where $\omega = \{x^* \in EP(f)$ and $Ax^* \in EP(g)\}$.*

Proof. Since $T^g_{r_k}$ is firmly nonexpansive and $I - T^g_{r_k}$ is 1-inverse strongly monotone, $A^T(I - T^g_{r_k})A$ is an $\frac{1}{L}-$inverse strongly and monotone mapping. Indeed,

$$\begin{aligned}
&\|A^T(I - T^g_{r_k})Ax - A^T(I - T^g_{r_k})Ay\|^2 \\
&= \langle A^T(I - T^g_{r_k})(Ax - Ay), A^T(I - T^g_{r_k})(Ax - Ay)\rangle \\
&= \langle (I - T^g_{r_k})(Ax - Ay), AA^T(I - T^g_{r_k})(Ax - Ay)\rangle \\
&\leqslant L\langle (I - T^g_{r_k})(Ax - Ay), (I - T^g_{r_k})(Ax - Ay)\rangle \\
&= L\|(I - T^g_{r_k})(Ax - Ay)\|^2 \\
&\leqslant L\langle Ax - Ay, (I - T^g_{r_k})(Ax - Ay)\rangle \\
&= L\langle x - y, A^T(I - T^g_{r_k})Ax - A^T(I - T^g_{r_k})Ay\rangle
\end{aligned}$$

for all $x, y \in H_1$. This implies that $A^T(I - T^g_{r_k})A$ is an $\frac{1}{L}-$inverse strongly and monotone mapping. Since $\beta_k \in (0, \frac{1}{L})$, $I - \beta_k A^T(I - T^g_{r_k})A$ is nonexpansive by

Proposition 3.1.42.

Step 1. We will show that $\{x_k\}$ is bounded.

Let $x^* \in \omega$. Then $x^* = T^f_{r_k} x^*$ and $(I - \beta_k A^T(I - T^g_{r_k})A)x^* = x^*$ by Lemma 3.2.9.

$$\begin{aligned} \|y_k - x^*\| &= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x^*\| \\ &\leqslant \|w_k - x^*\| \end{aligned} \tag{4.2.10}$$

for all $k \in \mathbb{N}$. Then from (4.2.10), we have

$$\begin{aligned} &\|x_{k+1} - x^*\| \\ &= \|\alpha_k V(x_k) + (1 - \alpha_k)y_k - x^*\| \\ &\leqslant \alpha_k\|V(x_k) - x^*\| + (1 - \alpha_k)\|y_k - x^*\| \\ &\leqslant \alpha_k\|V(x_k) - x^*\| + (1 - \alpha_k)\|w_k - x^*\| \\ &\leqslant \alpha_k\|V(x_k) - x^*\| + (1 - \alpha_k)\|x_k - x^*\| + \frac{\alpha_k}{\alpha_k}(1 - \alpha_k)|\theta_k|\|x_k - x_{k-1}\| \\ &\quad + \frac{\alpha_k}{\alpha_k}(1 - \alpha_k)|\delta_k|\|x_{k-1} - x_{k-2}\| \\ &\leqslant \alpha_k\|V(x_k) - x^*\| + (1 - \alpha_k)\|x_k - x^*\| \\ &\quad + \alpha_k\left\{\frac{|\theta_k|}{\alpha_k}\|x_k - x_{k-1}\| + \frac{|\delta_k|}{\alpha_k}\|x_{k-1} - x_{k-2}\|\right\}. \end{aligned}$$

By the Condition 4.2.5 (C4), let $M = \sup_{k \geqslant 0}\left\{\frac{|\theta_k|}{\alpha_k}\|x_k - x_{k-1}\| + \frac{|\delta_k|}{\alpha_k}\|x_{k-1} - x_{k-2}\|\right\}$, then we have

$$\begin{aligned} &\|x_{k+1} - x_k\| \\ &\leqslant \alpha_k\|V(x_k) - x^*\| + (1 - \alpha_k)\|x_k - x^*\| + \alpha_k M \\ &\leqslant \alpha_k(\|V(x_k) - V(x^*)\| + \|V(x^*) - x^*\|) + (1 - \alpha_k)\|x_k - x^*\| + \alpha_k M \\ &\leqslant \alpha_k\rho\|x_k - x^*\| + \alpha_k\|V(x^*) - x^*\| + (1 - \alpha_k)\|x_k - x^*\| + \alpha_k M \\ &\leqslant (1 - \alpha_k(1 - \rho))\|x_k - x^*\| + \alpha_k(1 - \rho)\left(\frac{\|V(x^*) - x^*\| + M}{(1 - \rho)}\right) \end{aligned}$$

$$\vdots$$
$$\leqslant \max\left\{\|x_1 - x^*\|, \frac{\|V(x^*) - x^*\| + M}{(1-\rho)}\right\}.$$

This implies that $\{x_k\}$ is bounded, and also $\{w_k\}$ and $\{y_k\}$ are bounded.

Step 2. We will show that $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$.

$$\begin{aligned}
&\|x_{k+1} - x_k\| \\
&= \|\alpha_k V(x_k) + (1-\alpha_k)y_k - \alpha_{k-1}V(x_{k-1}) - (1-\alpha_{k-1})y_{k-1}\| \\
&= \|\alpha_k V(x_k) - \alpha_k V(x_{k-1}) + \alpha_k V(x_{k-1}) - \alpha_{k-1}V(x_{k-1}) + (1-\alpha_k)y_k \\
&\quad - (1-\alpha_k)y_{k-1} + (1-\alpha_k)y_{k-1} - (1-\alpha_{k-1})y_{k-1}\| \\
&= \|\alpha_k V(x_k) - \alpha_k V(x_{k-1}) + \alpha_k V(x_{k-1}) - \alpha_{k-1}V(x_{k-1}) + (1-\alpha_k)y_k \\
&\quad - (1-\alpha_k)y_{k-1} + y_{k-1} - \alpha_k y_{k-1} - y_{k-1} + \alpha_{k-1}y_{k-1}\| \\
&\leqslant \alpha_k\rho\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K + (1-\alpha_k)\|y_k - y_{k-1}\|
\end{aligned}$$

where $K = \sup\{\|V(x_k)\| + \|y_k\| : k \in \mathbb{N}\}$. From $y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k$, $y_{k-1} = T^f_{r_{k-1}}(I - \beta_{k-1}A^T(I - T^g_{r_{k-1}})A)w_{k-1}$, and Lemma 3.2.8, we have

$$\begin{aligned}
&\|y_k - y_{k-1}\| \\
&= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_{k-1}}(I - \beta_{k-1}A^T(I - T^g_{r_{k-1}})A)w_{k-1}\| \\
&\leqslant \|w_k - \beta_k A^T(I - T^g_{r_k})Aw_k - w_{k-1} + \beta_{k-1}A^T(I - T^g_{r_{k-1}})Aw_{k-1}\| \\
&\quad + \left|\frac{r_k - r_{k-1}}{r_k}\right| \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - w_k + \beta_k A^T(I - T^g_{r_k})Aw_k\| \\
&= \|w_k - w_{k-1} + \beta_k A^T[(T^g_{r_k} - I)Aw_k - (T^g_{r_{k-1}} - I)Aw_{k-1}] \\
&\quad + \left|1 - \frac{r_{k-1}}{r_k}\right| \|T^f_{r_k}(w_k - \beta_k A^T(I - T^g_{r_k})Aw_k) - w_k - \beta_k A^T(T^g_{r_k} - I)Aw_k\| \\
&\leqslant \|w_k - w_{k-1} - \beta_k A^T A(w_k - w_{k-1})\| + \beta_k\|A^T\|\|T^g_{r_k}Aw_k - T^g_{r_{k-1}}Aw_{k-1}\| \\
&\quad + \left|1 - \frac{r_{k-1}}{r_k}\right|\gamma_{k-1}
\end{aligned}$$

$$\leqslant \left\{ \|w_k - w_{k-1}\|^2 - 2\|w_k - w_{k-1}\|\|\beta_k A^T A(w_k - w_{k-1})\| + \|\beta_k A^T A(w_k - w_{k-1})\|^2 \right\}^{\frac{1}{2}}$$

$$+ \quad \beta_k\|A\|\left\{ \|Aw_k - Aw_{k-1}\| + \left|\frac{r_k - r_{k-1}}{r_k}\right| \|T^g_{r_k} Aw_k - w_k\| \right\} + \left|1 - \frac{r_{k-1}}{r_k}\right|\gamma_{k-1}$$

$$\leqslant \left(1 - 2\beta_k\|A\|^2 + \beta_k^2\|A\|^4\right)^{\frac{1}{2}} \|w_k - w_{k-1}\| + \beta_k\|A\|^2\|w_k - w_{k-1}\|$$

$$\left|1 - \frac{r_{k-1}}{r_k}\right|\beta_k\|A\|\|T^g_{r_k} Aw_k - w_k\| + \left|1 - \frac{r_{k-1}}{r_k}\right|\gamma_{k-1}$$

$$= (1 - \beta_k\|A\|^2)\|w_k - w_{k-1}\| + \beta_k\|A\|^2\|w_k - w_{k-1}\| + \left|1 - \frac{r_{k-1}}{r_k}\right|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right)$$

$$= \|w_k - w_{k-1}\| + \left|1 - \frac{r_{k-1}}{r_k}\right|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right)$$

where $\gamma_{k-1} = \|T^f_{r_k}(w_k - \beta_k A^T(I - T^g_{r_k})Aw_k) - w_k - \beta_k A^T(T^g_{r_k} - I)Aw_k\|$ and $\epsilon_{k-1} = \|T^g_{r_k} Aw_k - w_k\|$.

From (4.2.10) and without loss of generality, let us assume that there exists a real number b such that $r_k > b > 0$ for all $k \in \mathbb{N}$. Then we have

$$\|x_{k+1} - x_k\|$$

$$\leqslant \alpha_k\rho\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K + (1 - \alpha_k)\|y_k - y_{k-1}\|$$

$$\leqslant \alpha_k\rho\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K$$

$$+ (1 - \alpha_k)\left\{ \|w_k - w_{k-1}\| + \left|1 - \frac{r_{k-1}}{r_k}\right|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right) \right\}$$

$$\leqslant \alpha_k\rho\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K$$

$$+ (1 - \alpha_k)\Big\{ \|x_k - x_{k-1}\| + \|\theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}) - \theta_{k-1}(x_{k-1} - x_{k-2})$$

$$- \delta_{k-1}(x_{k-2} - x_{k-3})\Big\} + (1 - \alpha_k)\left|1 - \frac{r_{k-1}}{r_k}\right|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right)$$

$$\leqslant (1 - \alpha_k + \alpha_k\rho)\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K + (1 - \alpha_k)|\theta_k|\|x_k - x_{k-1}\|$$

$$+ (1 - \alpha_k)|\delta_k|\|x_{k-1} - x_{k-2}\| + (1 - \alpha_k)|\theta_{k-1}|\|x_{k-1} - x_{k-2}\|$$

$$+ (1 - \alpha_k)|\delta_{k-1}|\|x_{k-2} - x_{k-3}\| + (1 - \alpha_k)\frac{1}{r_k}|r_k - r_{k-1}|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right)$$

$$\leqslant (1 - (1 - \rho)\alpha_k)\|x_k - x_{k-1}\| + |\alpha_k - \alpha_{k-1}|K + (1 - \alpha_k)|\theta_k|\|x_k - x_{k-1}\|$$

$$+ (1-\alpha_k)|\delta_k|\|x_{k-1} - x_{k-2}\| + (1-\alpha_k)|\theta_{k-1}|\|x_{k-1} - x_{k-2}\|$$
$$+ (1-\alpha_k)|\delta_{k-1}|\|x_{k-2} - x_{k-3}\| + (1-\alpha_k)\frac{1}{b}|r_k - r_{k-1}|\left(\beta_k\|A\|\epsilon_{k-1} + \gamma_{k-1}\right).$$

Using Lemma 3.2.10, we have $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$.

Step 3. We will show that $\lim_{k\to\infty} \|y_k - w_k\| = 0$. For $x^* \in \omega$, we have

$$\begin{aligned}
&\|y_k - x^*\|^2 \\
&= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x^*\|^2 \\
&\leqslant \langle T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x^*, w_k - x^*\rangle \\
&= \langle y_k - x^*, w_k - x^*\rangle \\
&= \frac{1}{2}\left(\|y_k - x^*\|^2 - \|y_k - w_k\|^2 + \|w_k - x^*\|^2\right)
\end{aligned}$$

and hence

$$\|y_k - x^*\|^2 \leqslant \|w_k - x^*\|^2 - \|y_k - w_k\|^2. \tag{4.2.11}$$

Since $A^T(I - T^g_{r_k})A$ is $\frac{1}{L}-$Lipschitz contonuous monotone, $I - \beta_k A^T(I - T^g_{r_k})A$ is nonexpansive, $T^g_{r_k}$ is firmly nonexpansive, by (4.2.11),

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|\alpha_k V(x_k) + (1-\alpha_k)y_k - x^*\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|y_k - x^*\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|w_k - x^*\|^2 - \|y_k - w_k\|^2.
\end{aligned}$$

From (4.2.10), we have

$$\begin{aligned}
&\|w_k - y_k\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|w_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 - \|x_{k+1} - x^*\|^2
\end{aligned}$$

$$
\begin{aligned}
&+ \|x_k - x^*\|^2 + 2\langle \theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}), w_k - x^*\rangle \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \|x_k - x^*\|^2 \\
&\quad + 2\Big(|\theta_k|\|x_k - x_{k-1}\| + |\delta_k|\|x_{k-1} - x_{k-2}\|\Big)\|w_k - x^*\| \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 - \|x_{k+1} - x^*\|^2 + \|x_k - x^*\|^2 \\
&\quad + 2\Big(\frac{|\theta_k|}{\alpha_k}\|x_k - x_{k-1}\| + \frac{|\delta_k|}{\alpha_k}\|x_{k-1} - x_{k-2}\|\Big)\|w_k - x^*\|.
\end{aligned}
$$

By the Condition 4.2.5 (C1) and (C4), we obtain $\lim_{k\to\infty}\|y_k - w_k\| = 0$.

Step 4. We will show that $\lim_{k\to\infty}\|Aw_k - T^g_{r_k}Aw_k\| = 0$.

Again since $T^f_{r_k}$ is firmly nonexpansive, we have

$$
\begin{aligned}
&\|y_k - x^*\|^2 \\
&= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - x^*\|^2 \\
&\leqslant \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k - T^f_{r_k}x^*\|^2 \\
&\leqslant \|w_k - \beta_k A^T(I - T^g_{r_k})Aw_k - x^*\|^2 \\
&\leqslant \|w_k - x^*\|^2 + \beta_k^2\|A^T(I - T^g_{r_k})A)w_k\|^2 + 2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})A)w_k\rangle \\
&\leqslant \|w_k - x^*\|^2 + \beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, AA^T(I - T^g_{r_k})Aw_k\rangle \\
&\quad + 2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})Aw_k\rangle. \qquad (4.2.12)
\end{aligned}
$$

On the other hand, we have

$$
\begin{aligned}
&\beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, AA^T(I - T^g_{r_k})Aw_k\rangle \\
&\leqslant L\beta_k^2\langle Aw_k - T^g_{r_k}Aw_k, Aw_k - T^g_{r_k}Aw_k\rangle \\
&= L\beta_k^2\|Aw_k - T^g_{r_k}Aw_k\|^2 \qquad (4.2.13)
\end{aligned}
$$

and

$$
2\beta_k\langle x^* - w_k, A^T(I - T^g_{r_k})Aw_k\rangle
$$

$$
\begin{aligned}
&= 2\beta_k\langle A(x^* - w_k), Aw_k - T^g_{r_k}Aw_k\rangle \\
&= 2\beta_k\langle A(x^* - w_k) + (Aw_k - T^g_{r_k}Aw_k) - (Aw_k - T^g_{r_k}Aw_k), Aw_k - T^g_{r_k}Aw_k\rangle \\
&= 2\beta_k\Big\{\langle Ax^* - T^g_{r_k}Aw_k, Aw_k - T^g_{r_k}Aw_k\rangle - \|Aw_k - T^g_{r_k}Aw_k\|^2\Big\} \\
&\leqslant 2\beta_k\Big\{\frac{1}{2}\|Aw_k - T^g_{r_k}Aw_k\|^2 - \|Aw_k - T^g_{r_k}Aw_k\|^2\Big\} \\
&< -\beta_k\|Aw_k - T^g_{r_k}Aw_k\|^2. \qquad (4.2.14)
\end{aligned}
$$

Using (4.2.12), (4.2.13) and (4.2.14), we have

$$
\begin{aligned}
&\|y_k - x^*\|^2 \\
&\leqslant \|w_k - x^*\|^2 + L\beta_k^2\|Aw_k - T^g_{r_k}Aw_k\|^2 - \beta_k\|Aw_k - T^g_{r_k}Aw_k\|^2 \\
&\leqslant \|w_k - x^*\|^2 + \beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2.
\end{aligned}
$$

This implies that

$$
\begin{aligned}
&\|x_{k+1} - x^*\|^2 \\
&= \|\alpha_k V(x_k) + (1-\alpha_k)y_k - x^*\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|y_k - x^*\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|w_k - x^*\|^2 + \beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2 \\
&\leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|x_k - x^*\|^2 + \beta_k(L\beta_k - 1)\|Aw_k - T^g_{r_k}Aw_k\|^2
\end{aligned}
$$

and

$$
\beta_k(1 - L\beta_k)\|Aw_k - T^g_{r_k}Aw_k\|^2 \leqslant \alpha_k\|V(x_k) - x^*\|^2 + \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2
$$

and so we obtain

$$
\lim_{k\to\infty}\|Aw_k - T^g_{r_k}Aw_k\| = 0.
$$

Step 5. We will show that $\limsup_{k\to\infty}\langle V(x^*) - x^*, w_k - x^*\rangle \leqslant 0$ where $x^* = P_\omega V(x^*)$. To show this inequality, we choose a subsequence $\{w_{k_i}\}$ of $\{w_k\}$ such that $\lim_{i\to\infty}\langle V(x^*) - x^*, w_{k_i} - x^*\rangle = \limsup_{k\to\infty}\langle V(x^*) - x^*, w_k - x^*\rangle$. Since $\{y_{k_i}\}$ is bounded, there exists a subsequence $\{y_{k_{i_j}}\}$ of $\{y_{k_i}\}$ which converges weakly to η. Without loss of generality in Hilbert space, we can assume that $y_{k_i} \rightharpoonup \eta$. From $\|w_k - y_k\| \to 0$, we obtain $w_{k_i} \rightharpoonup \eta$.

Let us show $\eta \in \omega$. By $y_{k_i} = T^f_{r_{k_i}}(I - \beta_{k_i}A^T(I - T^g_{r_{k_i}})A)w_{k_i}$, we have

$$f(y_{k_i}, y) + \frac{1}{r_{k_i}}\langle y - y_{k_i}, y_{k_i} - (I - \beta_{k_i}A^T(I - T^g_{r_{k_i}})A)w_{k_i}\rangle \geqslant 0, \forall y \in C,$$

$$f(y_{k_i}, y) + \frac{1}{r_{k_i}}\langle y - y_{k_i}, y_{k_i} - w_{k_i}\rangle - \frac{1}{r_{k_i}}\langle y - y_{k_i}, \beta_{k_i}A^T(T^g_{r_{k_i}} - I)Aw_{k_i}\rangle \geqslant 0, \forall y \in C.$$

From Assumption 3.1.41 (A2), we also have

$$-\frac{1}{r_{k_i}}\langle y - y_{k_i}, \beta_{k_i}A^T(T^g_{r_{k_i}} - I)Aw_{k_i}\rangle + \frac{1}{r_{k_i}}\langle y - y_{k_i}, y_{k_i} - w_{k_i} \geqslant f(y, y_k)$$

and

$$-\frac{1}{r_{k_i}}\langle y - y_{k_i}, \beta_{k_i}A^T(T^g_{r_{k_i}} - I)Aw_{k_i}\rangle + \langle y - y_{k_i}, \frac{y_{k_i} - w_{k_i}}{r_{k_i}} \geqslant f(y, y_{k_i}).$$

Since $\|y_k - w_k\| \to 0$ and $\|Aw_k - T^g_{r_k}Aw_k\| \to 0$, from Assumption 3.1.41 (A4), we have

$$0 \geqslant f(y, \eta), \text{for all } y \in C.$$

For c with $0 < c \leqslant 1$ and $y \in C$, let $y^c = cy + (1 - c)\eta$. Since $y \in C$ and $\eta \in C$, we have $y^c \in C$ and hence $f(y^c, \eta) \leqslant 0$. So, from Assumption 3.1.41 (A1) and (A4) we have

$$0 = f(y^c, y^c)$$

$$\leqslant cf(y^c, y) + (1-c)f(y^c, \eta)$$
$$\leqslant cf(y^c, y)$$

and hence $0 \leqslant f(y^c, y)$. From Assumption 3.1.41 (A3), we have $0 \leqslant f(\eta, y)$ for all $y \in C$ and hence $\eta \in EP(f)$. Since A is a bounded linear operator, $Aw_{k_i} \rightharpoonup A\eta$. Thus $\lim_{k\to\infty} \|Aw_k - T^g_{r_k} Aw_k\| = 0$, and so

$$T^g_{r_{k_i}} Aw_{k_i} \rightharpoonup A\eta \tag{4.2.15}$$

as $i \to \infty$. By the definition of $T^g_{r_{k_i}} Aw_{k_i}$, we have

$$g(T^g_{r_{k_i}} Aw_{k_i}, y) + \frac{1}{r_{k_i}} \langle y - T^g_{r_{k_i}} Aw_{k_i}, T^g_{r_{k_i}} Aw_{k_i} - Aw_{k_i} \rangle \geqslant 0$$

for all $y \in C$. Since the first argument of g is upper semi-continuous, it follows from (4.2.15) that $g(A\eta, y) \geqslant 0$ for all $y \in Q$. This shows that $A\eta \in EP(g)$. Thus $\eta \in \omega$. Therefore $\eta \in \omega$. Since $x^* = P_\omega V(x^*)$, we have

$$\begin{aligned}\limsup_{k\to\infty} \langle V(x^*) - x^*, w_k - x^* \rangle &= \lim_{i\to\infty} \langle V(x^*) - x^*, w_{k_i} - x^* \rangle \\ &= \langle V(x^*) - x^*, \eta - x^* \rangle \leqslant 0.\end{aligned}$$

From $x_{k+1} - x^* = \alpha_k(V(x_k) - x^*) + (1-\alpha_k)(y_k - x^*)$, we have

$$(1-\alpha_k)^2 \|y_k - x^*\|^2 \geqslant \|x_{k+1} - x^*\|^2 - 2\alpha_k \langle V(x_k) - x^*, x_{k+1} - x^* \rangle.$$

So, we have

$$\begin{aligned}&\|x_{k+1} - x^*\|^2 \\ &\leqslant (1-\alpha_k)^2 \|y_k - x^*\|^2 + 2\alpha_k \langle V(x_k) - x^*, x_{k+1} - x^* \rangle \\ &\leqslant (1-\alpha_k)^2 \|y_k - x^*\|^2 + 2\alpha_k \langle V(x_k) - V(x^*), x_{k+1} - x^* \rangle\end{aligned}$$

$$
\begin{aligned}
&+ 2\alpha_k\langle V(x^*) - x^*, x_{k+1} - x^*\rangle \\
&\leqslant (1-\alpha_k)^2\Big(\|x_k - x^* + \theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2})\|^2\Big) \\
&\quad + 2\alpha_k\rho\|x_k - x^*\|\|x_{k+1} - x^*\| + 2\alpha_k\langle V(x^*) - x^*, x_{k+1} - x^*\rangle \\
&\leqslant (1-\alpha_k)^2\Big(\|x_k - x^*\|^2 + 2\langle\theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}), w_k - x^*\rangle\Big) \\
&\quad + 2\alpha_k\rho\|x_k - x^*\|\|x_{k+1} - x^*\| + 2\alpha_k\langle V(x^*) - x^*, x_{k+1} - x^*\rangle \\
&\leqslant (1-\alpha_k)^2\|x_k - x^*\|^2 + \alpha_k\rho\{\|x_k - x^*\|^2 + \|x_{k+1} - x^*\|^2\} \\
&\quad + 2(1-\alpha_k)^2\Big(|\theta_k|\|x_k - x_{k-1}\| + |\delta_k|\|x_{k-1} - x_{k-2}\|\Big)\Big(\|w_k - x^*\|\Big) \\
&\quad + 2\alpha_k\langle V(x^*) - x^*, x_{k+1} - x^*\rangle.
\end{aligned}
$$

This implies that

$$
\begin{aligned}
&\|x_{k+1} - x^*\|^2 \\
&\leqslant \left(\frac{(1-\alpha_k)^2 + \alpha_k\rho}{(1-\alpha_k\rho)}\right)\|x_k - x^*\|^2 + \frac{2\alpha_k}{(1-\alpha_k\rho)}\langle V(x^*) - x^*, x_{k+1} - x^*\rangle \\
&\quad + \frac{2(1-\alpha_k)^2}{(1-\alpha_k\rho)}\Big(|\theta_k|\|x_k - x_{k-1}\| + |\delta_k|\|x_{k-1} - x_{k-2}\|\Big)\Big(\|w_k - x^*\|\Big) \\
&= \frac{(1 - 2\alpha_k + \alpha_k\rho)}{(1-\alpha_k\rho)}\|x_k - x^*\|^2 + \frac{\alpha^{k^2}}{(1-\alpha_k\rho)}\|x_k - x^*\|^2 \\
&\quad + \frac{2\alpha_k}{(1-\alpha_k\rho)}\langle V(x^*) - x^*, x_{k+1} - x^*\rangle \\
&\quad + \frac{2(1-\alpha_k)^2}{(1-\alpha_k\rho)}\Big(|\theta_k|\|x_k - x_{k-1}\| + |\delta_k|\|x_{k-1} - x_{k-2}\|\Big)\Big(\|w_k - x^*\|\Big) \\
&\leqslant \left(1 - \frac{2(1-\rho)\alpha_k}{(1-\alpha_k\rho)}\right)\|x_k - x^*\|^2 + \frac{2(1-\rho)\alpha_k}{(1-\alpha_k\rho)}\left\{\frac{1}{1-\rho}\langle V(x^*) - x^*, x_{k+1} - x^*\rangle\right. \\
&\quad \left. + \frac{\alpha_k M}{2(1-\rho)} + \frac{(1-\alpha_k)^2}{(1-\rho)\alpha_k}\Big(|\theta_k|\|x_k - x_{k-1}\| + |\delta_k|\|x_{k-1} - x_{k-2}\|\Big)\Big(\|w_k - x^*\|\Big)\right\}
\end{aligned}
$$

where $M = \sup\{\|x_k - x^*\|^2 : k \in \mathbb{N}\}$. Put $\mu_k = \frac{2(1-\rho)\alpha_k}{(1-\alpha_k\rho)}$. Then, we have $\sum_{k=1}^{\infty}\mu_k = \infty$ and $\lim_{k\to\infty}\mu_k = 0$. Thus $\lim_{k\to\infty}\|x_k - x^*\| = 0$. So, we conclude that $\{x_k\}$ converges strongly to $x^* \in \omega$, where $x^* = P_\omega V(x^*)$. □

From Algorithm 4.2.4, we see that if $\theta_k = 0$, then the algorithm can be reduced to new relaxed inertial viscosity-type algorithms. If $\theta_k = 0, \delta_k = 0$ in Algorithm 4.2.4, the algorithm can be reduced to standard viscosity-type algorithms.

Algorithm 4.2.7 (Relaxed inertial viscosity-type algorithm)

Initialization: Select $\{\delta_k\} \subset (-\infty, \infty), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty), \{\alpha_k\} \subset (0, 1)$. **Iterative step:** Let $x_{-2}, x_{-1}, x_0 \in C$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = x_k + \delta_k(x_{k-1} - x_{k-2}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. Calculate:

$$x_{k+1} = \alpha_k V(x_k) + (1 - \alpha_k)y_k.$$

Replace k by $k + 1$ and return to **Step1.**

Algorithm 4.2.8 (Standard viscosity-type algorithm)

Initialization: Select $\{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty)$, and $\{\alpha_k\} \subset (0, 1)$.

Iterative step: Let $x_0 \in C$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = x_k,$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. Calculate:

$$x_{k+1} = \alpha_k V(x_k) + (1 - \alpha_k)y_k.$$

Replace k by $k + 1$ and return to **Step1.**

4.2.3 A double inertial Mann algorithm for split equilibrium problems application to breast cancer screening

Algorithm 4.2.9 (Double relaxed inertial Mann algorithm)

Initialization: Select $\{\alpha_k\} \subset (0, 1), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty), \{\theta_k\}, \{\delta_k\} \subset (-\infty, \infty)$. **Iterative step:** Let $x_0, y_{-1}, y_0 \in H_1$ arbitrarily and calculate x_{k+1} as follows:

Step1. Compute:

$$w_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x_k.$$

Step 2. Calculate:

$$y_{k+1} = (1 - \alpha_k)x_k + \alpha_k w_k.$$

Step 3. Calculate:

$$x_{k+1} = y_{k+1} + \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}).$$

Replace k by $k+1$ and return to **Step1.**

Condition 4.2.10 Assume that the following conditions hold:

(C1) $0 < \liminf_{k\to\infty} \alpha_k \leqslant \limsup_{k\to\infty} \alpha_k < 1$;

(C2) $\liminf_{k\to\infty} r_k > 0$;

(C3) $0 < \liminf_{k\to\infty} \beta_k \leqslant \limsup_{k\to\infty} \beta_k < \frac{1}{L}$;

(C4) $\sum_{k=0}^{\infty} \theta_k \|y_{k+1} - y_k\| < \infty$;

(C5) $\sum_{k=0}^{\infty} \delta_k \|y_k - y_{k-1}\| < \infty$.

Theorem 4.2.11 *Suppose that Condition 4.2.10 hold. Let* $\{x_k\}$ *be a sequence defined by Algorithm 4.2.9. Then the sequence* $\{x_k\}$ *converges weakly to* $x^* \in \omega$.

Proof. First, we will show $\{x_k\}$ is bounded. Let $x^* \in \omega$. Then $x^* = T^f_{r_k} x^*$ and $(I - \beta_k A^T(I - T^g_{r_k})A)x^* = x^*$ by Lemma 3.2.9. Since $T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)$ is nonexpansive, we have

$$\begin{aligned}\|w_k - x^*\| &= \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x_k - T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x^*\| \\ &\leqslant \|x_k - x^*\|. \end{aligned} \tag{4.2.16}$$

From inequality (4.2.16),

$$\begin{aligned}&\|x_{k+1} - x^*\| \\ &= \|y_{k+1} + \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}) - x^*\| \\ &\leqslant \|y_{k+1} - x^*\| + |\theta_k|\|y_{k+1} - y_k\| + |\delta_k|\|y_k - y_{k-1}\| \\ &= \|(1-\alpha_k)x_k + \alpha_k w_k - x^*\| + |\theta_k|\|y_{k+1} - y_k\| + |\delta_k|\|y_k - y_{k-1}\| \\ &\leqslant (1-\alpha_k)\|x_k - x^*\| + \alpha_k\|w_k - x^*\| + |\theta_k|\|y_{k+1} - y_k\| + |\delta_k|\|y_k - y_{k-1}\| \\ &\leqslant (1-\alpha_k)\|x_k - x^*\| + \alpha_k\|x_k - x^*\| + |\theta_k|\|y_{k+1} - y_k\| + |\delta_k|\|y_k - y_{k-1}\| \\ &\leqslant \|x_k - x^*\| + |\theta_k|\|y_{k+1} - y_k\| + |\delta_k|\|y_k - y_{k-1}\|.\end{aligned}$$

By Condition 4.2.10 (C4)-(C5), it follows from Lemma 3.2.7, that $\lim_{k\to\infty} \|x_k - x^*\|$ exists. This implies that $\{x_k\}$ is bounded. By the definition of $\{w_k\}$ and $\{y_{k+1}\}$, $\{w_k\}$ and $\{y_{k+1}\}$ are also bounded.

Next, we will show that $\lim_{k\to\infty} \|Ax_k - T^g_{r_k}Ax_k\| = 0$ and $\lim_{k\to\infty} \|w_k - x_k\| = 0$.

$$\begin{aligned}\beta_k^2\|A^T(I - T^g_{r_k})Ax_k\|^2 &\leqslant \beta_k^2\langle Ax_k - T^g_{r_k}Ax_k, AA^T(I - T^g_{r_k})Ax_k\rangle \\ &\leqslant L\beta_k^2\langle Ax_k - T^g_{r_k}Ax_k, Ax_k - T^g_{r_k}Ax_k\rangle \\ &= L\beta_k^2\|Ax_k - T^g_{r_k}Ax_k\|^2 \qquad (4.2.17)\end{aligned}$$

and

$$\begin{aligned}&2\beta_k\langle x^* - x_k, A^T(I - T^g_{r_k})Ax_k\rangle \\ &= 2\beta_k\langle A(x^* - x_k), Ax_k - T^g_{r_k}Ax_k\rangle \\ &= 2\beta_k\langle A(x^* - x_k) + (Ax_k - T^g_{r_k}Ax_k) - (Ax_k - T^g_{r_k}Ax_k), Ax_k - T^g_{r_k}Ax_k\rangle \\ &= 2\beta_k\Big\{\langle Ax^* - T^g_{r_k}Ax_k, Ax_k - T^g_{r_k}Ax_k\rangle - \|Ax_k - T^g_{r_k}Ax_k\|^2\Big\} \\ &\leqslant 2\beta_k\Big\{\frac{1}{2}\|Ax_k - T^g_{r_k}Ax_k\|^2 - \|Ax_k - T^g_{r_k}Ax_k\|^2\Big\} \\ &< -\beta_k\|Ax_k - T^g_{r_k}Ax_k\|^2. \qquad (4.2.18)\end{aligned}$$

From $T^g_{r_k}$ is firmly nonexpansive, it follows from inequality (4.2.17) and (4.2.18), that

$$\begin{aligned}&\|x_{k+1} - x^*\|^2 \\ &= \|y_{k+1} + \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}) - x^*\|^2 \\ &\leqslant \|y_{k+1} - x^*\|^2 \\ &\quad + 2\langle\theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), y_{k+1} + \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}) - x^*\rangle \\ &\leqslant (1 - \alpha_k)\|x_k - x^*\|^2 + \alpha_k\|w_k - x^*\|^2 - (1 - \alpha_k)\alpha_k\|x_k - w_k\|^2 \\ &\quad + 2\langle\theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle\end{aligned}$$

$$
\begin{aligned}
&\leqslant (1-\alpha_k)\|x_k - x^*\|^2 + \alpha_k \|T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x_k - T^f_{r_k}x^*\|^2 \\
&\quad + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle \\
&\leqslant (1-\alpha_k)\|x_k - x^*\|^2 + \|x_k - \beta_k A^T(I - T^g_{r_k})Ax_k - x^*\|^2 \\
&\quad + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle \\
&= (1-\alpha_k)\|x_k - x^*\|^2 + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle \\
&\quad + \left\{ \|x_k - x^*\|^2 + \beta_k^2 \|A^T(I - T^g_{r_k})Ax_k\|^2 + 2\beta_k\langle x^* - x_k, A^T(I - T^g_{r_k})Ax_k\rangle \right\} \\
&\leqslant \|x_k - x^*\|^2 + \beta_k^2 \|A^T(I - T^g_{r_k})Ax_k\|^2 + 2\beta_k\langle x^* - x_k, A^T(I - T^g_{r_k})Ax_k\rangle \\
&\quad + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle \\
&= \|x_k - x^*\|^2 + L\beta_k^2\|Ax_k - T^g_{r_k}Ax_k\|^2 - \beta_k\|Ax_k - T^g_{r_k}Ax_k\|^2 \\
&\quad + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle,
\end{aligned}
$$

this implies that

$$
\begin{aligned}
&\beta_k(1 - L\beta_k)\|Ax_k - T^g_{r_k}Ax_k\|^2 \\
&\leqslant \|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 + 2\langle \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}), x_{k+1} - x^*\rangle.
\end{aligned} \tag{4.2.19}
$$

By $\lim_{k\to\infty}\|x_k - x^*\|$ exists, (4.2.19) and Conditions 4.2.10 (C3)-(C5), we have

$$
\lim_{k\to\infty}\|Ax_k - T^g_{r_k}Ax_k\| = 0. \tag{4.2.20}
$$

Using (4.2.16), we have

$$
\begin{aligned}
&\|y_{k+1} - x^*\|^2 \\
&= \|(1-\alpha_k)x_k + \alpha_k w_k - x^*\|^2 \\
&\leqslant (1-\alpha_k)\|x_k - x^*\|^2 + \alpha_k\|w_k - x^*\|^2 - \alpha_k(1-\alpha_k)\|w_k - x_k\|^2.
\end{aligned}
$$

and

$$\begin{aligned}
&\alpha_k(1-\alpha_k)\|w_k - x_k\|^2 \\
&\leqslant (1-\alpha_k)\|x_k - x^*\|^2 + \alpha_k\|w_k - x^*\|^2 - \|y_{k+1} - x^*\|^2 \\
&\leqslant (1-\alpha_k)\|x_k - x^*\|^2 + \alpha_k\|x_k - x^*\|^2 - \|y_{k+1} - x^*\|^2 \\
&\leqslant \|x_k - x^*\|^2 - \|y_{k+1} - x^*\|^2.
\end{aligned}$$

By the condition $0 < \liminf_{k\to\infty} \alpha_k \leqslant \limsup_{k\to\infty} \alpha_k < 1$, we obtain

$$\lim_{k\to\infty} \|w_k - x_k\| = 0. \tag{4.2.21}$$

So,

$$\begin{aligned}
\|y_{k+1} - x_k\| &= \|(1-\alpha_k)x_k + \alpha_k w_k - x_k\| \\
&\leqslant (1-\alpha_k)\|x_k - x_k\| + \alpha_k\|w_k - x_k\| \\
&\leqslant \|w_k - x_k\| \ \to \ 0
\end{aligned}$$

as $k \to \infty$. Next, let $\varrho_W(x_k) = \{x \in H : x_{k_i} \rightharpoonup x, \{x_{k_i}\} \subset \{x_k\}\}$. From the reflexiveness of H, we have $\varrho_W(x_k) \neq \emptyset$. Let $l \in \varrho_W(x_k)$, there exists a subsequence $\{x_{k_i}\} \subset \{x_k\}$ converging weakly to l. From (4.2.21), it follows that $w_{k_i} \rightharpoonup l$ as $i \to \infty$.

Next show that $l \in EP(f)$. From $w_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x_k$, we have

$$f(w_k, y) + \frac{1}{r_k}\langle y - w_k, w_k - x_k - \beta_k A^T(I - T^g_{r_k})Ax_k\rangle \geqslant 0$$

for all $y \in C$, which implies that

$$f(w_k, y) + \frac{1}{r_k}\langle y - w_k, w_k - x_k\rangle - \frac{1}{r_k}\langle y - w_k, \beta_k A^T(I - T^g_{r_k})Ax_k\rangle \geqslant 0$$

for all $y \in C$. By Assumption 3.1.41 (A2), we have

$$\frac{1}{r_{k_i}}\langle y - w_{k_i}, w_{k_i} - x_{k_i}\rangle - \frac{1}{r_{k_i}}\langle y - w_{k_i}, \beta_{k_i} A^T(I - T^g_{r_{k_i}})Ax_{k_i}\rangle \geqslant f(y, w_{k_i})$$

for all $y \in C$. From $\liminf_{k\to\infty} r_k > 0$, from Assumption 3.1.41 (A4), (4.2.20), and (4.2.21), we gain

$$f(y, l) \leqslant 0$$

for all $y \in C$. For each $0 < c \leqslant 1$ and $y \in C$, define $y^c = cy + (1-c)l$. As $y \in C$ and $l \in C, y^c \in C$, and therefore $f(y^c, l) \leqslant 0$. By Assumption 3.1.41(A1), we have

$$0 = f(y^c, y^c) \leqslant cf(y^c, y) + (1-c)f(y^c, l) \leqslant cf(y^c, y)$$

and hence $f(y^c, y) \geqslant 0$. By Assumption 3.1.41 (A3), $f(l, y) \geqslant 0$ for all $y \in C$ and therefore $l \in EP(f)$. As A is a bounded linear operator, $Ax_{k_i} \rightharpoonup Al$. Then it follows from (4.2.20), that

$$T^g_{r_{k_i}} Ax_{k_i} \rightharpoonup Al \tag{4.2.22}$$

as $i \to \infty$. By the definition of $T^g_{r_{k_i}} Ax_{k_i}$, we have

$$g(T^g_{r_{k_i}} Ax_{k_i}, y) + \frac{1}{r_{k_i}}\langle y - T^g_{r_{k_i}} Ax_{k_i}, T^g_{r_{k_i}} Ax_{k_i} - Ax_{k_i}\rangle \geqslant 0$$

for all $y \in C$. By in the first argument of g is upper semi-continuous, it follows from (4.2.22) that

$$g(Al, y) \geqslant 0$$

for all $y \in C$. This shows that $Al \in EP(g)$. Thus $l \in \omega$.

Step 6. We will show that $\{x_k\}$ and $\{w_k\}$ converge weakly to an element of ω. It is sufficient to show that the set of weak sequential cluster point of $\{x_k\}$ $(\varrho_W(x_k))$ is singleton. Let $x^*, l \in \varrho_W(x_k)$ and $\{x_{k_n}\}, \{x_{k_m}\} \subset \{x_k\}$ be such that $x_{k_n} \rightharpoonup x^*$ and $x_{k_m} \rightharpoonup l$. From (4.2.21), we also have $w_{k_n} \rightharpoonup x^*$ and $w_{k_m} \rightharpoonup l$. By Step 5, we obtain $x^*, l \in \omega$. Using Lemma 3.2.5, we gain $x^* = l$. This completes the proof. $\square$

4.3 Numerical example

We now give an example in infinitely dimensional spaces $L_2[0,1] = \{x(t) : \int_0^1 x(t)dt < \infty\}$, where such that $\|\cdot\|$ is L_2-norm defined by $\|x\| = \sqrt{\int_0^1 x(t)^2 dt}$ to support Theorem 4.1.9.

Example 4.3.1 Let $V : L_2[0,1] \to L_2[0,1]$ be defined by $V(x(t)) = \frac{x(t)}{2}$ where $x(t) \in L_2[0,1]$. We can choose $x_0(t) = \frac{sin(t)}{2}$ and $x_1(t) = sin(t)$. The stopping criterion is defined by $\|x_k - x_{k-1}\| < 10^{-2}$.

We set the following parameters for each algorithm, as seen in Table 1.

Table 1: Chosen parameters of each algorithm

	Algorithm 4.1.2	Algorithm 4.1.10	Algorithm 4.1.11
λ_k	0.1	0.1	-
λ_1	-	-	0.12
θ_k	0.29	0.29	0.29
α_k	$\frac{1}{100k+1}$	$\frac{1}{100k+1}$	$\frac{1}{100k+1}$
τ_k	0.15	0.1	0.15
μ	-	-	0.2

Next, we compare the performance of Algorithm 4.1.2, Algorithm 4.1.10, and Algorithm 4.1.11. We obtain the results as seen in Table 2.

Table 2: The performance of each algorithm

	Algorithm 4.1.2	Algorithm 4.1.10	Algorithm 4.1.11
CPU Time	1.2626	1.2010	177.9459
Iter. No.	2	2	2

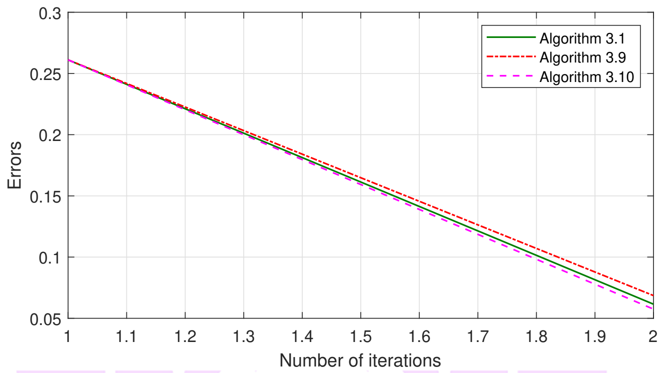

Figure 1: The Cauchy error and number of iterations

From Figure 1, we see that the performance of Algorithm 4.1.11 (Algorithm 3.10) is better than Algorithm 4.1.2 (Algorithm 3.1) and Algorithm 4.1.10 (Algorithm 3.9).

CHAPTER 5

APPLICATIONS

Data Classification is the process of categorizing data to distinguish between types or groups, playing a crucial role in analysis and decision-making across various domains such as disease diagnosis, fraud detection, and targeted marketing. Commonly used methods include statistical techniques such as Logistic Regression [33] and Linear Discriminant Analysis (LDA) [66], along with machine learning algorithms like Decision Trees [100], Support Vector Machines (SVM) [19], and Random Forest [23]. Additionally, deep learning approaches such as Neural Networks [85] and Convolutional Neural Networks (CNNs) [48] efficiently handle complex data like images and text. These methods collectively enable efficient and accurate data management, effectively addressing diverse needs in the digital age. Data classification is the process of categorizing data into predefined groups, which is essential in applications like medical diagnosis and pattern recognition. Extreme Learning Machine (ELM), a type of feedforward neural network with a single hidden layer, is highly efficient for this task due to its fast learning speed and strong generalization. By utilizing randomly assigned weights and biases, ELM eliminates the need for iterative tuning, making it ideal for large-scale and complex classification problems.

We focus on extreme learning machine (ELM) proposed by Huang et al. [65] for applying our algorithms to solve data classification problems. It is defined as follows: Let $E := \{(\mathbf{x}_n, \mathbf{t}_n) : \mathbf{x}_n \in \mathbb{R}^q, \mathbf{t}_n \in \mathbb{R}^m, n = 1, 2, ..., P\}$ be a training set of P distinct samples where $\mathbf{x}_n$ is an input training data and $\mathbf{t}_n$ is a training target. The output function of ELM for single-hidden layer feed forward neural networks (SLFNs) with M hidden nodes and activation function U is

$$\mathbf{O}_n = \sum_{i=1}^{M} a_i U(w_i \mathbf{x}_n + b_i),$$

where w_i and b_i are parameters of weight and finally the bias, respectively and a_i is the optimal output weight at the i-th hidden node. The hidden layer output matrix H is defined as follows:

$$H = \begin{bmatrix} U(w_1\mathbf{x}_1 + b_1) & \dots & U(w_M\mathbf{x}_1 + b_M) \\ \vdots & \ddots & \vdots \\ U(w_1\mathbf{x}_P + b_1) & \dots & U(w_M\mathbf{x}_P + b_M) \end{bmatrix}$$

The goal of ELM is to find optimal output weight $a = [a_1^T, ..., a_M^T]^T$ such that $Ha = T$, where $T = [\mathbf{t}_1^T, ..., \mathbf{t}_P^T]^T$ is the training data. In some cases, finding $a = H^{\dagger}T$, where $H^{\dagger}$ is the *Moore-Penrose generalized inverse* of H. When $H^{\dagger}$ is difficult to calculate. The difficulty in calculating the Moore-Penrose inverse of a matrix H arises when the matrix is not full rank or has a low rank. In such cases, the matrix does not have a unique inverse, making the computation of the Moore-Penrose inverse more complex. Additionally, for large matrices with incomplete data or complex structures, calculating the Moore-Penrose inverse often requires advanced numerical methods or approximations, which can be computationally expensive and resource-intensive. So we can solve $Ha = T$ by the following least square problems as follows:

$$\min_{a\in\mathbb{R}^M} \{\|Ha - T\|_2^2\}. \tag{5.0.1}$$

In this work, the performance of machine learning techniques for all classes is accurately measured. The accuracy is calculated by adding the total number of correct predictions to the total number of predictions. The performance parameter calculation of precision and recall are measured. The formulation of three measures [117] are defined as follow:

$$\text{Precision(Pre)} = \frac{TP}{TP + FP} \times 100\%.$$

$$\text{Recall(Rec)} = \frac{TP}{TP + FN} \times 100\%.$$

$$\text{Accuracy(Acc)} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%,$$

where a confusion matrix for original and predicted classes are shown in terms of TP, TN, FP, and FN are the True Positive, True Negative, False Positives, and False Negatives, respectively. Similarly, P and N are the Positive and Negative population of Malignant and Benign cases, respectively. And we define F-Measure as the harmonic mean value between recall and precision as follow:

$$\text{F-Measure} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \times 100\%.$$

5.1 Diabetes mellitus

According to the International Diabetes Federation (IDF), there are approximately 463 million people with diabetes worldwide, and it is estimated that by 2045 there will be 629 million people with diabetes. In Thailand, the incidence of diabetes is continuously increasing. There are about 300,000 new cases per year, and 3.2 million people with diabetes are registered in the Ministry of Public Health's registration system. They are causing huge losses in health care costs. Only one disease of diabetes causes the average cost of treatment costs up to 47,596 million baht per year. This has led to an ongoing campaign about the dangers of the disease. Furthermore, diabetes mellitus makes additional noncommunicable diseases that present a high risk for the patient, as they easily contact and are susceptible to infectious diseases such as COVID-19 [121]. Because it is a chronic disease that cannot be cured. There is a chance that the risk of complications spreading to the extent of the loss of vital organs of the body. By the International Diabetes Federation and the World Health Organization (WHO) has designated November 14 of each year as World Diabetes Day to recognize the importance of this disease.

In this research, we used the PIMA Indians diabetes dataset which was downloaded from Kaggle (https://www.kaggle.com/uciml/pima-indians-diabetes database) and is available publicly on UCI repository for training processing by our proposed algorithm. The dataset contains 768 pregnant female patients which 500 were non-diabetics and 268 were diabetics. There were 9 variables present inside the dataset; eight variables contain information about patients, and the 9th variable is the class predicting the patients as diabetic and nondiabetic. This dataset contains the various attributes that are Number of times pregnant; Plasma glucose concentration at 2 Hours in an oral glucose tolerance test (GTIT); Diastolic Blood Pressure (mm Hg); Triceps skin fold thickness (mm); 2-Hour Serum insulin (lh/ml); Body mass index [weight in kg/(Height in m)]; Diabetes pedigree function; Age (years); Binary value indicating non-diabetic /diabetic. For the implementation of machine learning algorithms, 614 were used as a training dataset and 154 were used as a testing training dataset by using 5-fold cross-validation [75]. For benchmarking classifier, we consider the following various methods which have been proposed to classify diabetes:

Table 3: Classification accuracy of different methods with literature

Authors	Methods	Accuracy (%)
Li [77]	Ensemble of SVM, ANN, and NB	58.3
Brahim-Belhouari and Bermak [22]	NB, SVM, DT	76.30
Smith et al.[110]	Neural ADAP algorithm	76
Quinlan [101]	C4.5 Decision trees	71.10
Bozkurt et al.[21]	Artificial neural network	76.0
Sahan et al.[102]	Artificial immune System	75.87
Smith et al.[110]	Ensemble of MLP and NB	64.1
Chatrati et al.[29]	Linear discriminant analysis	72
Deng and Kasabov [37]	Self-organizing maps	78.40
Deng and Kasabov [37]	Self-organizing maps	78.40
Choubey et al. [31]	Ensemble of RF and XB	78.9
Saxena et al. [105]	Feature selection of KNN, RF, DT, MLP	79.8
Our Algorithm 4.1.2	Extreme learning machine	**80.03**

In this section, we process some experiments on the classification problem. This problem can be seen as the following convex minimization problem:

$$\min_{a \in C}\{\|Ha - T\|_2^2\},$$

where $C = \{a \in \mathbb{R}^M : \|a\|_1 \leq \lambda\}$. This problem is called the least absolute shrinkage and selection operator (LASSO) [118]. Setting $f(a, \zeta) = \langle H^T(Ha - T), \zeta - a\rangle$ and $V(x) = cx$ where c is constant in $(0, 1)$.

The binary cross-entropy loss function along with sigmoid activation function for binary classification calculates the loss of an example by computing the following average:

$$Loss = -\frac{1}{K}\sum_{j=1}^{K} y_j \log \hat{y}_j + (1 - y_j)\log(1 - \hat{y}_j)$$

where $\hat{y}_j$ is the j-th scalar value in the model output, y_j is the corresponding target value, and K is the number of scalar values in the model output.

For starting our computation, we set the activation function as sigmoid, hidden nodes $M = 160$, regularization parameter $\lambda = 1 \times 10^{-5}$, $\theta_k = 0.3$, $\alpha_k = \frac{1}{k+1}$, $\tau = 0.5$, $\mu = 0.2$ for Algorithm 4.1.2, 4.1.10, and 4.1.11 and $c = 0.9999$ for Algorithm 4.1.2 and 4.1.11. The stopping criteria is the number of iteration 250. We obtain the results of the different parameters S when $\lambda_k = \frac{S}{max(eigenvalue(A^TA))}$ for Algorithm 4.1.2, 4.1.10 and the different parameters λ_1 for Algorithm 4.1.11 as seen in Table 4.

Table 4: Training and validation loss and training time of the different parameter λ_k and λ_1

			Loss	
	S, λ_1	Training Time	Training	Validation
Algorithm 4.1.2	0.2	0.4164	0.286963	0.275532
	0.4	0.4337	0.283279	0.273650
	0.6	0.4164	0.286963	0.275532
	0.9	0.4459	0.278714	0.272924
	0.99	0.4642	0.278144	0.272921
Algorithm 4.1.10	0.2	0.4283	0.291883	0.279878
	0.4	0.5293	0.288831	0.277365
	0.6	0.4246	0.286890	0.276099
	0.9	0.4247	0.284851	0.275079
	0.99	0.5096	0.284356	0.274879
Algorithm 4.1.11	0.2	1.3823	0.286963	0.275532
	0.4	1.5652	0.283279	0.273650
	0.6	1.4022	0.281060	0.273120
	0.9	1.9170	0.278714	0.272924
	0.99	1.3627	0.278144	0.272921

We can see that $\lambda_k = \lambda_1 = 0.99$ greatly improves the performance of Algorithm 4.1.2, Algorithm 4.1.10, and Algorithm 4.1.11. Therefore, we choose it as the default inertial parameter for next computation.

We next choose $\lambda_k = \frac{0.99}{max(eigenvalue(A^TA))}$, $\alpha_k = \frac{1}{k+1}$, $\tau = 0.5$ for Algorithm 4.1.2 and Algorithm 4.1.10 and $c = 0.9999$ for Algorithm 4.1.2 with $\lambda_1 = \frac{0.99}{max(eigenvalue(A^TA))}$, $\alpha_k = \frac{1}{k+1}$, $\tau = 0.5$, $c = 0.9999$, and $\mu = 0.2$ for Algorithm 4.1.11. The stopping criteria is the number of iteration 250. We consider the different initialization parameter θ where

$$\theta_k = \begin{cases} \frac{\theta}{k^2\|x_k - x_{k-1}\|} & if\ x_k \neq x_{k-1}\ and\ k > N, \\ \theta & \text{otherwise}, \end{cases}$$

where N is a number of iterations that we want to stop. We obtain the numerical results as seen in Table 5.

Table 5: Training and validation loss and training time of the different parameter θ

			Loss	
	θ	Training Time	Training	Validation
Algorithm 4.1.2	0.1	0.4608	0.279629	0.272965
	0.2	0.4515	0.278938	0.272931
	0.3	0.4523	0.278144	0.272921
	$\frac{1}{k}$	0.4591	0.280107	0.273004
	$\frac{1}{\|x_k - x_{k-1}\| + k^2}$	0.5003	0.280221	0.273015
Algorithm 4.1.10	0.1	0.4723	0.284808	0.274993
	0.2	0.4641	0.284587	0.274935
	0.3	0.4634	0.284356	0.274879
	$\frac{1}{k}$	0.5297	0.285004	0.275049
	$\frac{1}{\|x_k - x_{k-1}\| + k^2}$	0.4825	0.285019	0.275053
Algorithm 4.1.11	0.1	1.4071	0.279629	0.272965
	0.2	1.3505	0.278938	0.272931
	0.3	1.4819	0.278144	0.272921
	$\frac{1}{k}$	1.3276	0.280107	0.273004
	$\frac{1}{\|x_k - x_{k-1}\| + k^2}$	1.4228	0.280221	0.273015

We can see that $\theta = 0.3$ greatly improves the performance of Algorithm 4.1.2, Algorithm 4.1.10, and Algorithm 4.1.11. Therefore, we choose it as the default inertial parameter for next computation.

We next set $\lambda_k = \frac{0.99}{max(eigenvalue(A^T A))}$, $\theta = 0.3$, $\tau = 0.5$ for Algorithm 4.1.2 and Algorithm 4.1.10 and $c = 0.9999$ for Algorithm 4.1.2 with $\lambda_1 = \frac{0.99}{max(eigenvalue(A^T A))}$, $\theta_k = 0.3$, $\tau = 0.5$, $C = 0.9999$, and $\mu = 0.2$ for Algorithm 4.1.11. The stopping criteria is the number of iteration 250. We consider the different initialization parameter α_k. The numerical results are shown in Table 6.

Table 6: Training and validation loss and training time of the different parameter α_k

			Loss	
	α_k	Training Time	Training	Validation
Algorithm 4.1.2	$\frac{1}{k+1}$	0.4407	0.278144	0.272921
	$\frac{1}{10k+1}$	0.4054	0.278143	0.272921
	$\frac{1}{k^2+1}$	0.4938	0.278143	0.272921
	$\frac{1}{10k^2+1}$	0.4876	0.278143	0.272921
Algorithm 4.1.10	$\frac{1}{k+1}$	0.4163	0.284356	0.274879
	$\frac{1}{10k+1}$	0.4274	0.279201	0.273129
	$\frac{1}{k^2+1}$	0.5150	0.278294	0.272931
	$\frac{1}{10k^2+1}$	0.5960	0.278160	0.272922
Algorithm 4.1.11	$\frac{1}{k+1}$	1.4292	0.278144	0.272921
	$\frac{1}{10k+1}$	1.3803	0.278143	0.272921
	$\frac{1}{k^2+1}$	1.2452	0.278143	0.272921
	$\frac{1}{10k^2+1}$	1.4100	0.278143	0.272921

We can see that $\alpha_k = \frac{1}{10k+1}$ greatly improves the performance of Algorithm 4.1.2, $\alpha_k = \frac{1}{10k^2+1}$ greatly improves the performance of Algorithm 4.1.10, and $\alpha_k = \frac{1}{k^2+1}$ greatly improves the performance of Algorithm 4.1.11. Therefore, we choose it as the default inertial parameter for next computation.

We next calculate the numerical results by setting $\lambda_k = \frac{0.99}{max(eigenvalue(A^TA))}$, $\theta = 0.3$, $\alpha_k = \frac{1}{10k+1}$ and $c = 0.9999$ for Algorithm 4.1.2, $\lambda_k = \frac{0.99}{max(eigenvalue(A^TA))}$, $\theta = 0.3$, $\alpha_k = \frac{1}{10k^2+1}$ for Algorithm 4.1.10 and $\lambda_1 = \frac{0.99}{max(eigenvalue(A^TA))}$, $\theta = 0.3$, $c = 0.9999$, $\alpha_k = \frac{1}{k^2+1}$, and $\mu = 0.2$ for Algorithm 4.1.11. The stopping criteria is the number of iteration 250. We consider the different initialization parameter τ. The numerical results are shown in Table 7.

Table 7: Training and validation loss and training time of the different parameter τ

			Loss	
	τ	Training Time	Training	Validation
Algorithm 4.1.2	0.1	0.4278	0.300531	0.299144
	0.3	0.4509	0.299074	0.293717
	0.5	0.5239	0.278143	0.272921
	$\frac{k}{2k+1}$	0.4708	0.282187	0.274017
Algorithm 4.1.10	0.1	0.4592	0.300531	0.299144
	0.3	0.4900	0.299074	0.293717
	0.5	0.4261	0.278160	0.272922
	$\frac{k}{2k+1}$	0.5224	0.282191	0.274018
Algorithm 4.1.11	0.1	1.3401	0.300531	0.299144
	0.3	1.3771	0.299074	0.293717
	0.5	1.8681	0.278143	0.272921
	$\frac{k}{2k+1}$	1.4671	0.282187	0.274017

We can see that $\tau = 0.5$ greatly improves the performance of Algorithm 4.1.2, Algorithm 4.1.10, and Algorithm 4.1.11. Therefore, we choose it as the default inertial parameter for next computation.

We next calculate the numerical results by setting $\lambda_k = \frac{0.99}{max(eigenvalue(A^TA))}$, $\theta = 0.3$, $\tau = 0.5$ for Algorithm 4.1.2 and Algorithm 4.1.10 and $\alpha_k = \frac{1}{10k+1}$ for Algorithm 4.1.2 with $\alpha_k = \frac{1}{10k^2+1}$ for Algorithm 4.1.10 and $\lambda_1 = \frac{0.99}{max(eigenvalue(A^TA))}$, $\theta = 0.3$, $\alpha_k = \frac{1}{k^2+1}$, $\tau = 0.5$, and $\mu = 0.2$ for Algorithm 4.1.11. The stopping criteria is the number of iteration 250. We obtain the results of the different parameters c when $V(x) = cx$ for Algorithm 4.1.2 and Algorithm 4.1.11 as seen in Table 8.

Table 8: Training and validation loss and training time of the different parameter c

	c	Training Time	Loss: Training	Loss: Validation
Algorithm 4.1.2	0.3	0.4796	0.278902	0.273066
	0.5	0.4270	0.278695	0.273024
	0.7	0.4190	0.278480	0.272982
	0.9	0.4209	0.278257	0.272941
	0.9999	0.4844	0.278143	0.272921
Algorithm 4.1.11	0.3	1.5886	0.278251	0.272928
	0.5	1.6358	0.278222	0.272926
	0.7	1.3808	0.278191	0.272924
	0.9	1.5176	0.278159	0.272922
	0.9999	1.4598	0.278143	0.272921

From Tables 4-8, we choose the parameters for Algorithm 4.1.2 to compare the exist algorithms from the literature. The following Table 9 shows choosing the necessary parameters of each algorithm.

Table 9: Chosen parameters of each algorithm

	Algorithm in (2.1.2)	Algorithm in (2.1.3)	Algorithm in (2.1.5)	Algorithm 4.1.2	Algorithm 4.1.10	Algorithm 4.1.11
μ	-	0.3	0.3	-	-	0.2
λ_1	-	$\frac{0.5}{max(eig(A^TA))}$	$\frac{0.9999}{max(eig(A^TA))}$	-	-	$\frac{0.99}{max(eig(A^TA))}$
λ_k	$\frac{0.5}{max(eig(A^TA))}$	-	-	$\frac{0.99}{max(eig(A^TA))}$	$\frac{0.99}{max(eig(A^TA))}$	-
θ	-	-	0.3	0.3	0.3	0.3
α_k	$\frac{1}{100k+1}$	$\frac{1}{100k+1}$	$\frac{1}{2k+1}$	$\frac{1}{10k+1}$	$\frac{1}{10k^2+1}$	$\frac{1}{k^2+1}$
τ	-	-	0.5	0.5	0.5	0.5
C	-	-	-	0.9999	-	0.9999

For comparison, We set sigmoid as an activation function, hidden nodes $M = 160$ and regularization parameter $\lambda = 1 \times 10^{-5}$.

Table 10: The performance of each algorithm

Algorithm	**Iteration No.**	**Training Time**	**Pre**	**Rec**	**Acc** (%)
Algorithm of Hieu (2.1.2)	25	0.0537	80.97	97.50	80.03
Algorithm of Muangchoo (2.1.3)	25	0.3132	80.97	97.50	80.03
Algorithm of Shehu (2.1.5)	30	0.1182	80.97	97.50	80.03
Algorithm 4.1.2	18	0.0375	80.97	97.50	80.03
Algorithm 4.1.10	18	0.0401	80.97	97.50	80.03
Algorithm 4.1.11	18	0.1045	80.97	97.50	80.03

Table 10 shows that Algorithm 4.1.2 has the highest efficiency in precision, recall, and accuracy. It also has the lowest number of iterations. It has the highest probability of correctly classifying tumors compared to algorithms examinations. We present the training and validation loss with the accuracy of training to show that Algorithm 4.1.2 has no overfitting in the training dataset.

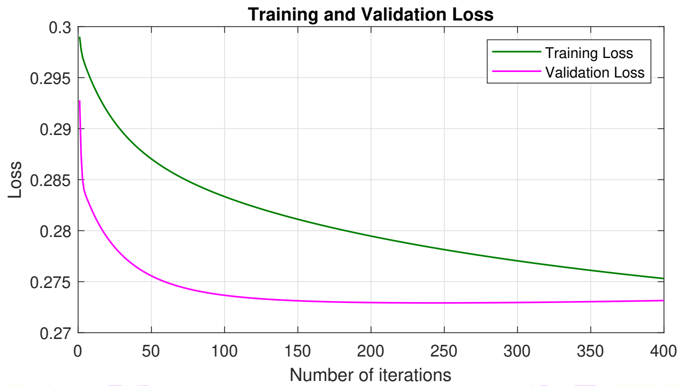

Figure 2: Training and validation loss plots of Algorithm 4.1.2

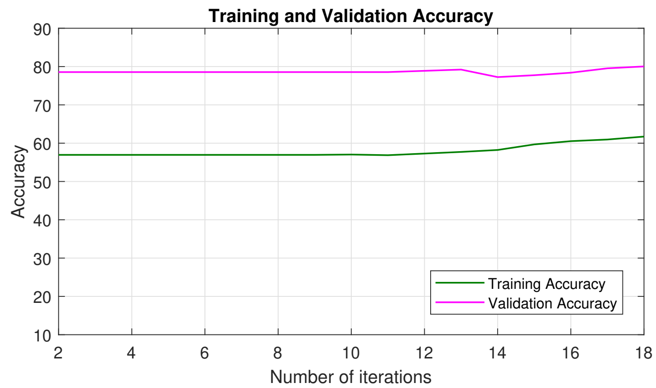

Figure 3: Training and validation accuracy plots of Algorithm 4.1.2

From Figures 2-3, we see that our Algorithm 4.1.2 has good fit model this means that our Algorithm 4.1.2 suitably learns the training dataset and generalizes well to classification the PIMA Indians diabetes dataset.

5.2 Cardiovascular disease

Cardiovascular Diseases or Heart Diseases are a broader name of conditions that cause abnormalities of the heart and blood vessels in various parts; there are many forms, such as Coronary heart disease, Cerebrovascular disease, Peripheral Arterial Disease, Rheumatic heart disease, Congenital heart disease, Abnormal Heart Rhythms, Deep vein thrombosis, and pulmonary embolism. The leading causes of cardiovascular disease are small lumps of fat gathering together to become Plaque, embedded in the blood vessel wall, which blocks transport to various body parts, such as the brain and heart. Cardiovascular diseases are one of the leading causes of death both in Thailand and abroad. In some cases, symptoms may not be present or thought to be a symptom of another disease. This causes the patient not to get the proper treatment and can lead to sudden death. Two factors cause heart disease: those that cannot be controlled (such as genetics, age, and gender) and those that can controlled by lifestyle behaviors (such as smoking, eating high-fat foods, not exercising, and as a result of other diseases for example, diabetes, and high blood pressure). Heart disease patients tend to get tired quickly when performing various activities, chest pain as if pressed by something, palpitations and shortness of breath, and swelling of the legs and feet due to heart failure.

Cardiovascular disease is classified as a group of non-communicable diseases (NCDs) that can be prevented by reducing risk factors for disease by changing health behaviors such as blood pressure, fat levels, high sugar levels, BMI, smoking, etc. Cardiovascular disease (CVDs) is the world's number one killer to-day. CVDs take the lives of nearly 18 million people every year, 31% of all global deaths. Triggering these diseases - which manifest primarily as heart attacks and strokes - are tobacco use, unhealthy diet, physical inactivity, and the harmful use of alcohol. These, in turn, show up in people as raised blood pressure, elevated blood glucose, and overweight and obesity, risks detrimental to good heart health. From general information, it is found that most patients are at high risk of developing coronary heart disease because they have aged over 60 years with an average body mass index of $23.9 \pm 4.6\ kg/m^2$, which is a condition of being overweight have atherosclerosis, heart disease with three coronary artery disease is associated with hyperlipidemia and high blood pressure.

In this research, we use the Cardiovascular Disease dataset which was downloaded from Kaggle (https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset). The dataset comprise of 70,000 records of patients data, 1 target and 11 features.

There are 3 types of input features:

- Objective: factual information;
- Examination: results of medical examination;
- Subjective: information given by the patient.

Table 11: Data description

Features	Types		
1. Age	Objective Feature	age	int (days)
2. Height	Objective Feature	height	int (cm)
3. Weight	Objective Feature	weight	float (kg)
4. Gender	Objective Feature	gender	categorical code
5. Systolic blood pressure	Examination Feature	ap_hi	int
6. Diastolic blood pressure	Examination Feature	ap_lo	int
7. Cholesterol	Examination Feature	cholesterol	1: normal, 2: above normal, 3: well above normal
8. Glucose	Examination Feature	gluc	1: normal, 2: above normal, 3: well above normal
9. Smoking	Subjective Feature	smoke	binary
10. Alcohol intake	Subjective Feature	alco	binary
11. Physical activity	Subjective Feature	active	binary
12. Presence or absence of cardiovascular disease	Target Variable	cardio	binary

We use the Data Cleaner in MATLAB and use smooth data technique for height, weight, systolic blood pressure, and diastolic blood pressure. The followings are the details of choosing smoothing factor of parameters.

Table 12: Choose smoothing method, smoothing parameter and smoothing factor

	Smoothing method	Smoothing parameter	Smoothing factor
Height	Moving mean	Smoothing factor	0.25
Weight	Moving mean	Smoothing factor	0.3
Systolic blood pressure (ap_hi)	Moving median	Smoothing factor	0.65
Diastolic blood pressure (ap_lo)	Moving median	Smoothing factor	0.85

Table 13: The visualizations of all data after preprocessing

Features	Minimum	Maximum	Mean	Median	Mode	Standard Deviation
Height	105.50	209.00	164.36	164.50	165.00	5.79
Weight	37.00	144.50	74.21	73.00	70.00	10.34
Systolic blood pressure	80.00	210.00	126.13	120.00	120.00	11.95
Diastolic blood pressure	60.00	110.00	81.60	80.00	80.00	4.95

Table 11-13, show overview of data.

Table 14: Accuracy, Precision, Recall and F-Measure of methods in literature and our algorithm

Methods	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
AdaBoost [15]	73.28	77.10	65.00	70.60
Artificial Neural Network [83]	65.10	63.70	71.80	66.90
Decision Tree [83]	73.10	71.90	76.80	73.90
k-Nearest Neighbors [83]	57.10	57.00	57.80	57.40
LightGBM [15]	74.11	76.30	68.90	72.40
Logistic Regression [83]	72.30	70.30	77.50	73.50
Naive Bayes [83]	59.10	55.50	90.80	68.90
Ontology [83]	75.70	79.30	82.30	80.70
Random Forest [15]	72.08	72.60	69.60	71.10
Support Vector Machine [83]	64.80	63.40	71.20	66.70
XGBoost [15]	74.07	76.30	68.80	72.40
Algorithm of Hieu [63]	74.87	67.99	94.00	78.91
Algorithm of Muangchoo [89]	74.84	67.96	94.03	78.90
Algorithm of Shehu [107]	74.93	68.06	94.00	78.95
Algorithm 4.1.2	77.25	71.36	91.06	80.02
Our Algorithm 4.1.13	**77.50**	**71.87**	**90.41**	**80.08**

From Table 14, we can see that our Algorithm 4.1.13 is high accuracy of another methods.

In this section, we process some experiments on the classification problem. This problem can be seen as the following the least square problems:

$$\min_{a\in\mathbb{R}^M}\{\|Ha-T\|_2^2\},$$

Set $f(a,\zeta)=\langle H^T(Ha-T),\zeta-a\rangle$ and $V(x)=cx$ where c is a constant in $(0,1)$.

The binary cross-entropy loss function along with sigmoid activation function for binary classification calculates the loss of an example by computing the following average:

$$Loss=-\frac{1}{K}\sum_{j=1}^{K}y_j\log\hat{y}_j+(1-y_j)\log(1-\hat{y}_j)$$

where $\hat{y}_j$ is the j-th scalar value in the model output, y_j is the corresponding target value, and K is the number of scalar values in the model output.

In our calculation we set the activation function as sigmiod, hidden node $M=90, N=200, \tau=0.5$, $\alpha_k=\frac{k}{1.2k}$, $\theta=0.33, \lambda=1\times10^{-5}$ for Algorithm 4.1.13 and Algorithm 4.1.20, $c=0.9999$ for Algorithm 4.1.13. We gain the outcomes of the different parameters S where $\lambda_k=\frac{S}{max(eigenvalue(A^TA))}$ for Algorithm 4.1.13 and Algorithm 4.1.20 as seen in Table 15.

When

$$\theta_k=\begin{cases}\frac{\theta}{k^2\|x_k-x_{k-1}\|} & if\ x_k\neq x_{k-1}\ and\ k>N,\\ \theta & \text{otherwise,}\end{cases}$$

we require to stop with N, where N is an iteration number.

Table 15: Training time and numerical results of both of training and validation loss when λ_k is different

	S	Training Time	Loss Training	Loss Validation
Algorithm 4.1.13	0.2	1.9910	0.2893	0.2895
	0.4	1.9914	0.2802	0.2807
	0.6	1.9977	0.2731	0.2737
	0.9	1.9954	0.2652	0.2660
	0.999	1.9997	0.2632	0.2640
Algorithm 4.1.20	0.2	1.9982	0.2943	0.2945
	0.4	1.9985	0.2885	0.2888
	0.6	1.9961	0.2835	0.2839
	0.9	1.9953	0.2771	0.2776
	0.999	1.9933	0.2752	0.2758

We can see that $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$ greatly improves the performance of Algorithm 4.1.13 and Algorithm 4.1.20.

We next select $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$, $\alpha_k = \frac{k}{1.2k}$, $\tau = 0.5$ for Algorithm 4.1.20 and $c = 0.9999$ for Algorithm 4.1.13 with $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$, $\alpha_k = \frac{k}{1.2k}$ and $\tau = 0.5$. With $N = 200$, we investigate the different initialization parameter θ as seen in Table 16.

Table 16: Training time and numerical results of both of training and validation loss when θ is different

	θ	Training Time	Loss	
			Training	Validation
Algorithm 4.1.13	0.1	1.9950	0.2639	0.2647
	0.15	2.0127	0.2638	0.2646
	0.2	1.9961	0.2636	0.2644
	0.3	1.9957	0.2633	0.2641
	0.33	1.9915	0.2632	0.2640
Algorithm 4.1.20	0.1	2.0010	0.2756	0.2762
	0.15	2.0045	0.2755	0.2761
	0.2	1.9990	0.2754	0.2760
	0.3	1.9945	0.2753	0.2758
	0.33	1.9962	0.2752	0.2758

We can see that $\theta = 0.33$ greatly improves the performance of Algorithm 4.1.13 and Algorithm 4.1.20.

We next set $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}$, $\theta = 0.33$, $\tau = 0.5$ for Algorithm 4.1.20 and $c = 0.9999$ for Algorithm 4.1.13 with $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}$, $\theta = 0.33$, $\tau = 0.5$. With $N = 200$, we investigate the different parameter α_k. The numeric outcomes are displayed in Table 17.

Table 17: Training time and numerical results of both of training and validation loss when α_k is different

			Loss	
	α_k	Training Time	Training	Validation
Algorithm 4.1.13	$\frac{k}{k+1}$	1.9948	0.2642	0.2649
	$\frac{k}{k+2}$	1.9891	0.2640	0.2648
	$\frac{k}{k+5}$	1.9950	0.2637	0.2645
	$\frac{k}{1.1k}$	1.9910	0.2637	0.2645
	$\frac{k}{1.2k}$	1.9969	0.2632	0.2640
Algorithm 4.1.20	$\frac{k}{k+1}$	1.9994	0.2770	0.2775
	$\frac{k}{k+2}$	2.0049	0.2768	0.2773
	$\frac{k}{k+5}$	1.9956	0.2761	0.2766
	$\frac{k}{1.1k}$	1.9922	0.2762	0.2767
	$\frac{k}{1.2k}$	1.9961	0.2752	0.2758

We can see that $\alpha_k = \frac{k}{1.2k}$ greatly reform the efficiency of Algorithm 4.1.13 and $\alpha_k = \frac{k}{1.2k}$ greatly reform the efficiency of Algorithm 4.1.20. As a result, we select it as the default inertial parameter for next calculation.

We next computation the numerical outcomes by setting $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$, $\theta = 0.33$, $\alpha_k = \frac{k}{1.2k}$ and $c = 0.9999$ for Algorithm 4.1.13, $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$, $\theta = 0.33$, $\alpha_k = \frac{k}{1.2k}$ for Algorithm 4.1.20. With $N = 200$, we investigate the different parameter τ. The numeric outcomes are dispalyed in Table 18.

Table 18: Training time and numerical results of both of training and validation loss when τ is different

	τ	Training Time	Loss	
			Training	Validation
Algorithm 4.1.13	0.1	1.9950	0.2894	0.2896
	0.2	1.9946	0.2803	0.2808
	0.3	1.9921	0.2732	0.2738
	0.4	1.9934	0.2676	0.2683
	0.5	1.9986	0.2632	0.2640
Algorithm 4.1.20	0.1	1.9967	0.2944	0.2945
	0.2	2.0020	0.2886	0.2889
	0.3	2.0053	0.2835	0.2839
	0.4	1.9927	0.2791	0.2796
	0.5	1.9995	0.2752	0.2758

We can see that $\tau = 0.5$ greatly improves the performance of Algorithm 4.1.13 and Algorithm 4.1.20. As a result, we select it as the default parameter for next calculation.

We calculated the numerical outcomes by setting $\lambda_k = \frac{0.999}{max(eigenvalue(A^TA))}$, $\theta = 0.33$, $\tau = 0.5$ and $\alpha_k = \frac{k}{1.2k}$ for Algorithm 4.1.13 and Algorithm 4.1.20. With $N = 200$, we gain the outcomes of the different parameters C when $V(x) = cx$ for Algorithm 4.1.13 as seen in Table 19.

Table 19: Training time and numerical results of both of training and validation loss when c is different

	C	Training Time	Loss	
			Training	Validation
Algorithm 4.1.13	0.3	2.009	0.3005	0.3005
	0.5	1.9867	0.3003	0.3003
	0.7	1.9919	0.2998	0.2998
	0.9	1.9978	0.2975	0.2975
	0.9999	1.9980	0.2632	0.2640

From Tables 15-19, we select the parameters for Algorithm 4.1.13 to collate the existent algorithms from the introduction.

Table 20: Selected parameters of each algorithm

	e	λ_1	λ_k	θ	α_k	τ	c	δ	μ	θ
Algorithm of Hieu [63]	-	-	$\frac{0.999}{2(max(eig(A^TA)))}$	-	$\frac{k}{1.2k}$	-	-	-	-	-
Algorithm of Muangchoo [89]	$\frac{0.999}{max(eig(A^TA))}$	$\frac{0.999}{2(max(eig(A^TA)))}$	-	-	$\frac{k}{1.2k}$	-	-	-	-	-
Algorithm of Shehu [107]	0.9	$\frac{0.999}{2(max(eig(A^TA)))}$	-	0.33	$\frac{k}{1.2k}$	0.5	-	-	-	-
Algorithm 4.1.2	-	-	$\frac{0.999}{2(max(eig(A^TA)))}$	0.33	$\frac{k}{1.2k}$	0.5	0.999	-	-	-
Algorithm 4.1.13	-	-	$\frac{0.999}{max(eig(A^TA))}$	0.33	$\frac{k}{1.2k}$	0.5	0.999	-	-	-
Algorithm 4.1.20	-	-	$\frac{0.999}{max(eig(A^TA))}$	0.33	$\frac{k}{1.2k}$	0.5	-	-	-	-
Algorithm of Yao et al. [124]	-	$\frac{0.999}{2(max(eig(A^TA)))}$	-	-0.1	-	-	-	$\frac{\sqrt{5}-1}{2}$	0.1	0.1

Table 20 displays select the essential parameters of each algorithm. For comparing, we set the activation function as sigmiod ($\frac{1}{1+e^{-x}}$), hidden node $M = 90$ and $\lambda = 1 \times 10^{-5}$.

Table 21: Efficiency comparative of existing algorithms and our algorithms

Algorithms	**Iteration count**	**Training Time**	**Pre**	**Rec**	**Acc**	**F-measure**
Algorithm of Hieu [63]	52	0.5316	67.99	94.00	74.87	78.91
Algorithm of Muangchoo [89]	52	3.1999	67.96	94.03	74.84	78.90
Algorithm of Shehu [107]	52	1.6957	68.06	94.00	74.93	78.95
Algorithm 4.1.2	52	0.5312	71.36	91.06	77.25	80.02
Algorithm 4.1.13	52	0.5341	71.87	90.41	77.50	80.08
Algorithm 4.1.20	52	0.5377	71.86	90.42	77.49	80.08
Algorithm of Yao et al. [124]	52	2.0370	67.53	94.30	74.47	78.70

Table 21 shows that the Algorithm 4.1.13 has the highest performance of precision accuracy and F-measure by the number of iterations is 52. It is the highest possibility of exactly classifying comparisons to algorithms investigations. Both of training-validation loss and accuracy plots show that the model for Cardiovascular

Disease dataset has a good fitting model when we use the Algorithm 4.1.13.

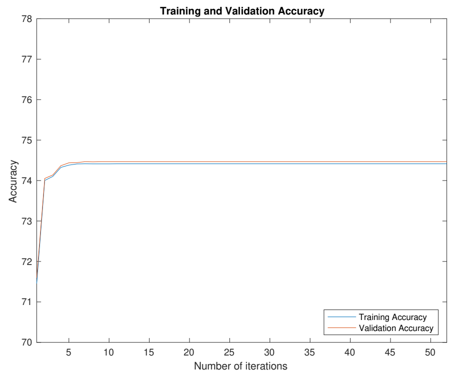

Figure 4: On the left is the accuracy plots of training and validation of algorithm 4.1.13 and on the right is the accuracy plots of training and validation of Algorithm Yao et al. [124]

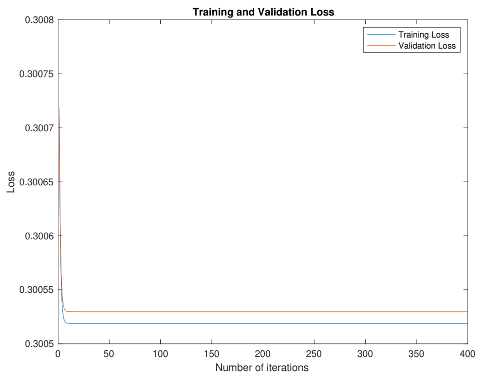

Figure 5: On the left is the loss plots of training and validation of algorithm 4.1.13 and on the right is the loss plots of training and validation of Algorithm Yao et al. [124]

From Figures 4-5, we see that our Algorithm 4.1.13 has a good fit model. This means that our Algorithm 4.1.13 appropriately learns the training dataset and generalizes well to classify the Cardiovascular disease dataset.

5.3 Parkinson's disease

Parkinson's disease is the most common age-related neurodegenerative disorder that primarily affects movement control by the loss of dopaminergic neurons in the substantia nigra pars compacta and by an accumulation of misfolded a-synuclein found in intra-cytoplasmic inclusions called Lewy bodies [17, 10]. While motor-related symptoms like bradykinesia (slow movement), tremors, and rigidity are well-known, non-motor symptoms, e.g., vocal or speech impairments, can also significantly impact the quality of life for individuals with PD. Vocal problems in individuals with PD are often called "Parkinson's speech disorder" or "hypokinetic dysarthria," which occur due to the degeneration of neurons in the brain's basal ganglia and related motor control pathways [36]. This motor speech disorder is characterized by a reduced range of motion and muscle weakness in the muscles responsible for speech production, leading to various issues affecting speech and voice, e.g., reduced volume (hypophonia), monotone speech, reduced articulation, imprecise consonants, stuttering or stammering, and decreased expressiveness [111]. A supportive method to diagnose or predict PD is clinically helpful, for example, focusing on the early speech dysfunction observed in people with PD [36], as the earliest stages of PD can be difficult to recognize, as reflected by the long delay of an average of 10 years that typically separates the timing of the diagnosis from the person's first noticeable symptom [50].

We use database in UCI (https://archive.ics.uci.edu/dataset/301/parkinson+speech+dataset +with+multiple+types+of+sound+recordings) that belongs to 20 PWP (6 female, 14 male) and 20 healthy individuals (10 female, 10 male) who appealed at the Department of Neurology in Cerrahpasa Faculty of Medicine, Istanbul University. From all subjects, multiple types of sound recordings (26 voice samples including sustained vowels, numbers, words and short sentences) are taken as follows, 1^{st} to 3^{rd} samples represent the sustained vowels /a/, /o/

and /u/, respectively, 4^{th} to 13^{th} samples represent numbers from 1 to 10, 14^{th} to 17^{th} samples represent short sentences, 18^{th} to 26^{th} samples represent individual words.

Table 22: Data description for Parkinson's disease dataset in Microsoft Excel

Column 1	Subject id	
Colum 2-27	Group: Frequency Parameters	Features
		1. Jitter (local)
		2. Jitter (local, absolute)
		3. Jitter (rap)
		4. Jitter (ppq5)
		5. Jitter (ddp)
	Group: Amplitude Parameters	Features
		6. Shimmer (local)
		7. Shimmer (local, dB)
		8. Shimmer (apq3)
		9. Shimmer (apq5)
		10. Shimmer (apq11)
		11. Shimmer (dda)
	Group: Harmonicity Parameters	Features
		12. Autocorrelation
		13. Noise to Harmonic
		14. Harmonic to Noise
	Group: Pitch Parameters	Features
		15. Median pitch
		16. Mean pitch
		17. Standard deviation
		18. Minimum pitch
		19. Maximum pitchh
	Group: Pulse Parameters	Features
		20. Number of pulses
		21. Number of periods
		22. Mean period
		23. Standard deviation of period
	Group: Voicing Parameters	Features
		24. Fraction of locally unvoiced frames
		25. Number of voice breaks
		26. Degree of voice breaks
Column 28	UPDRS	
Column 29	class information	

We have 1,040 observations with no missing data and 27 variables. Next, we use training data of 70% and testing data of 30%, and the dataset has two classes. In this thesis, we use Cross-validation, a crucial machine learning technique, to evaluate a model's performance by splitting the data into multiple subsets for training and testing. It helps prevent overfitting, ensures a more accurate assessment of the model's generalization ability, and allows for optimal model selection. We use every observation for training and testing, so we organized into groups in Figure 6.

Figure 6: Groups of dataset

For applying our algorithm to solve the least square problems (5.0.1), we set

$$f(x,y) = \begin{cases} 0, & \forall x, y \in C, \\ -1, & otherwise \end{cases}$$

and

$$g(x,y) = \begin{cases} 0, & \forall x, y \in Q, \\ -1, & otherwise. \end{cases}$$

Then, our algorithm can be reduced to solve split feasibility problem (SFP) which was introduced by Censor and Elfving [26] as follows: finding a point $x^* \in C$ such that

$$Ax^* \in Q \tag{5.3.1}$$

where C, Q be nonempty closed and convex subsets of $\mathbb{R}^B$ and $\mathbb{R}^D$, respectively. A is an $B \times D$ real matrix.

We create 3 model for our Algorithm of split feasibility problem (5.3.1) to obtain results in machine learning (5.0.1).

1. Least square model - (L)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = H_1$ and $Q = \{T\}$.

2. Least square on constraint set L_1 - (LL_1)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_1 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

3. Least square on constraint set L_2 - (LL_2)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_2^2 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

The binary cross-entropy loss function along with sigmoid activation function for binary classification calculates the loss of an example by computing the following average:

$$Loss = -\frac{1}{K}\sum_{j=1}^{K} y_j \log \hat{y}_j + (1-y_j)\log(1-\hat{y}_j)$$

where $\hat{y}_j$ is the j-th scalar value in the model output, y_j is the corresponding target value, and K is the number of scalar values in the model output.

In our calculation we beginning by set the activation function as sigmoid, we stop the number of iteration $N = 500$, hidden nodes $M = 150$, regularization parameter $\alpha_k = 0.5$, $\theta_k = 0.5$ for Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. We gain the outcomes of the different parameters S by $\lambda_k = \frac{S}{max(eigenvalue(A^TA))}$ for Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2 as seen in Table 23. When

$$\theta_k = \begin{cases} \frac{\theta}{k^2\|x_k - x_{k-1}\|} & if\ x_k \neq x_{k-1}\ and\ k > N, \\ \theta & \text{otherwise,} \end{cases}$$

Table 23: Training Time and numerical results of both of training and validation loss when λ_k is different

	S	Training Time	Loss Training	Loss Validation
Algorithm 4.2.2 L	0.15	0.4416	0.2721	0.2725
	0.3	0.4411	0.2575	0.2536
	0.45	0.4398	0.2477	0.2405
	0.6	0.4409	0.2406	0.2312
	0.85	0.4390	0.2327	0.2211
Algorithm 4.2.2 LL_1	0.15	0.4394	0.2721	0.2725
	0.3	0.4393	0.2575	0.2536
	0.45	0.4337	0.2477	0.2405
	0.6	0.4295	0.2406	0.2312
	0.85	0.4261	0.2327	0.2211
Algorithm 4.2.2 LL_2	0.15	0.6433	0.2721	0.2725
	0.3	0.6320	0.2575	0.2536
	0.45	0.6287	0.2477	0.2405
	0.6	0.6345	0.2406	0.2312
	0.85	0.5213	0.2327	0.2211

We can see that $\lambda_k = 0.85$ greatly improves the performance of Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. As a result, we select it as the default inertial parameter for next calculation.

Next, we will select $\lambda_k = \frac{0.85}{max(eigenvalue(A^TA))}$, $\alpha_k = 0.5$, $\theta = 0.5$ for Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. And we stop number of iterations $N = 500$. We gain the numeric outcomes as seen in Table 24.

Table 24: Training Time and numerical results of both of training and validation loss when α_k is different

			Loss	
	α_k	Training Time	Training	Validation
Algorithm 4.2.2 L	0.5	0.4273	0.2327	0.2211
	0.6	0.4378	0.2377	0.2274
	0.7	0.4403	0.2446	0.2364
	0.8	0.4338	0.2546	0.2496
	0.9	0.4739	0.2698	0.2695
Algorithm 4.2.2 LL_1	0.5	0.4420	0.2327	0.2211
	0.6	0.4841	0.2377	0.2274
	0.7	0.4521	0.2446	0.2364
	0.8	0.5063	0.2546	0.2496
	0.9	0.4437	0.2698	0.2695
Algorithm 4.2.2 LL_2	0.5	0.5122	0.2327	0.2211
	0.6	0.7036	0.2377	0.2274
	0.7	0.6256	0.2446	0.2364
	0.8	0.6271	0.2546	0.2496
	0.9	0.6177	0.2698	0.2695

We can see that $\alpha_k = 0.5$ greatly improves the performance of Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. As a result, we select it as the default inertial parameter for next calculation.

Next, we will select $\lambda_k = \frac{0.85}{max(eigenvalue(A^TA))}$, $\alpha_k = 0.5$, $\theta = 0.5$ for Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. And we stop number of iterations $N = 500$. We gain the numeric outcomes can see in Table 25.

Table 25: Training Time and numerical results of both of training and validation loss when θ is different

			Loss	
	θ	Training Time	Training	Validation
Algorithm 4.2.2 L	0.1	0.4282	0.2465	0.2389
	0.3	0.4355	0.2403	0.2309
	0.5	0.4244	0.2327	0.2211
	$\frac{1}{k}$	0.4342	0.2487	0.2419
	$\frac{1}{10k+2}$	0.4293	0.2490	0.2423
Algorithm 4.2.2 LL_1	0.1	0.4390	0.2465	0.2389
	0.3	0.4334	0.2403	0.2309
	0.5	0.4313	0.2327	0.2211
	$\frac{1}{k}$	0.4714	0.2487	0.2419
	$\frac{1}{10k+2}$	0.4352	0.2490	0.2423
Algorithm 4.2.2 LL_2	0.1	0.6094	0.2465	0.2389
	0.3	0.6133	0.2403	0.2309
	0.5	0.5149	0.2327	0.2211
	$\frac{1}{k}$	0.6099	0.2487	0.2419
	$\frac{1}{10k+2}$	0.5218	0.2490	0.2423

We can see that $\theta = 0.5$ greatly improves the performance of Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2. Table 26 shows accuracy of other medthods.

Table 26: Accuracy of another methods and our algorithm

Authors	**Methods**	**Accuracy** (%)
Sakar et al. [103]	KNN+SVM	55.00
Eskidere et al. [41]	Random Subspace Classifier Ensemble	74.17
Celik et al. [25]	LR	76.03
Celik et al. [25]	SVM	75.49
Benba et al. [14]	MFCC+SVM	82.50
Li et al. [78]	Ensemble learning algorithm	86.50
Algorithm 4.2.2 L		**90.38**
Algorithm 4.2.2 LL_1		**90.38**
Algorithm 4.2.2 LL_2		**90.38**

For comparing, We set activation function is sigmoid function and hidden nodes M = 150 in Table 27.

Table 27: Efficiency comparative of existing algorithms and our algorithms

	Iter	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Test box 1						
Algorithm 4.2.2 L	227	0.2046	79.57	94.87	86.55	85.26
Algorithm 4.2.2 LL_1	227	0.2062	79.57	94.87	86.55	85.26
Algorithm 4.2.2 LL_2	227	0.2419	79.57	94.87	86.55	85.26
Algorithm of Suantai L [114]	227	0.2039	79.10	89.74	84.08	83.01
Algorithm of Suantai LL_1 [114]	227	0.2042	79.10	89.74	84.08	83.01
Algorithm of Suantai LL_2 [114]	227	0.2337	79.10	89.74	84.08	83.01
Test box 2						
Algorithm 4.2.2 L	782	0.6618	86.21	96.15	90.91	90.38
Algorithm 4.2.2 LL_1	782	0.6879	86.21	96.15	90.91	90.38
Algorithm 4.2.2 LL_2	782	0.8087	86.21	96.15	90.91	90.38
Algorithm of Suantai L [114]	782	0.6741	83.62	94.87	88.89	83.01
Algorithm of Suantai LL_1 [114]	782	0.6659	83.62	94.87	88.89	83.01
Algorithm of Suantai LL_2 [114]	782	0.7850	83.62	94.87	88.89	83.01
Test box 3						
Algorithm 4.2.2 L	175	0.1691	85.80	96.79	90.96	90.38
Algorithm 4.2.2 LL_1	175	0.1637	85.80	96.79	90.96	90.38
Algorithm 4.2.2 LL_2	175	0.1925	85.80	96.79	90.96	90.38
Algorithm of Suantai L [114]	175	0.1617	86.50	90.38	88.40	88.14
Algorithm of Suantai LL_1 [114]	175	0.1627	86.50	90.38	88.40	88.14
Algorithm of Suantai LL_2 [114]	175	0.1873	86.50	90.38	88.40	88.14

	Iter	Time	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Test box 4						
Algorithm 4.2.2 L	487	0.4272	83.70	98.72	90.59	89.74
Algorithm 4.2.2 LL_1	487	0.4348	83.70	98.72	90.59	89.74
Algorithm 4.2.2 LL_2	487	0.5091	83.70	98.72	90.59	89.74
Algorithm of Suantai L [114]	487	0.4179	83.15	98.08	90.00	89.10
Algorithm of Suantai LL_1 [114]	487	0.4267	83.15	98.08	90.00	89.10
Algorithm of Suantai LL_2 [114]	487	0.5964	83.15	98.08	90.00	89.10
Test box 5						
Algorithm 4.2.2 L	464	0.4136	72.56	100.00	84.10	81.09
Algorithm 4.2.2 LL_1	464	0.4108	72.56	100.00	84.10	81.09
Algorithm 4.2.2 LL_2	464	0.4876	72.56	100.00	84.10	81.09
Algorithm of Suantai L [114]	464	0.4029	71.76	99.36	83.33	80.13
Algorithm of Suantai LL_1 [114]	464	0.4014	71.76	99.36	83.33	80.13
Algorithm of Suantai LL_2 [114]	464	0.4657	71.76	99.36	83.33	80.13
Test box 6						
Algorithm 4.2.2 L	571	0.5063	75.61	99.36	85.87	83.65
Algorithm 4.2.2 LL_1	571	0.5009	75.61	99.36	85.87	83.65
Algorithm 4.2.2 LL_2	571	0.5829	75.61	99.36	85.87	83.65
Algorithm of Suantai L [114]	571	0.4912	73.56	98.08	84.07	81.41
Algorithm of Suantai LL_1 [114]	571	0.4957	73.56	98.08	84.07	81.41
Algorithm of Suantai LL_2 [114]	571	0.5688	73.56	98.08	84.07	81.41
Test box 7						
Algorithm 4.2.2 L	350	0.3092	71.50	98.08	82.70	79.49
Algorithm 4.2.2 LL_1	350	0.3143	71.50	98.08	82.70	79.49
Algorithm 4.2.2 LL_2	350	0.3629	71.50	98.08	82.70	79.49
Algorithm of Suantai L [114]	350	0.3069	69.95	95.51	80.76	77.24
Algorithm of Suantai LL_1 [114]	350	0.3067	69.95	95.51	80.76	77.24
Algorithm of Suantai LL_2 [114]	350	0.3591	69.95	95.51	80.76	77.24
Test box 8						
Algorithm 4.2.2 L	1477	1.2572	83.71	95.51	89.22	88.46
Algorithm 4.2.2 LL_1	1477	1.2445	83.71	95.51	89.22	88.46
Algorithm 4.2.2 LL_2	1477	1.4572	83.71	95.51	89.22	88.46
Algorithm of Suantai L [114]	1477	1.2581	82.58	94.23	88.02	87.18
Algorithm of Suantai LL_1 [114]	1477	1.2520	82.58	94.23	88.02	87.18
Algorithm of Suantai LL_2 [114]	1477	1.4630	82.58	94.23	88.02	87.18

Table 27 shows that our Algorithm 4.2.2 L, Algorithm 4.2.2 LL_1 and Algorithm 4.2.2 LL_2 in Test box 3 are highly performance of accuracy, F1-score, and recall by the number of iteration is 175 and Algorithm 4.2.2 L use low of training time. It is the highest possibility of exactly classifying comparison to algorithms investigations. Both of training-validation loss and accuracy plots show that our algorithm has good fitting model for Parkinson's disease dataset.

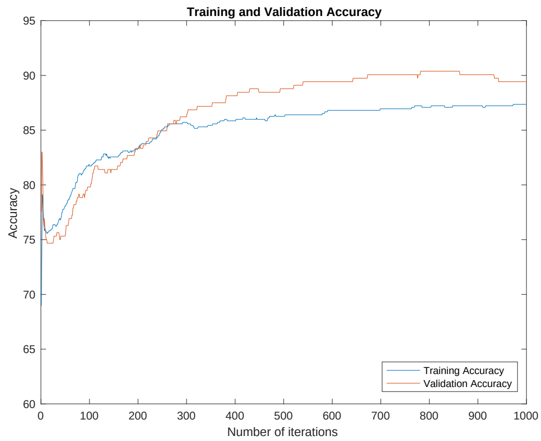

Figure 7: The accuracy plots of training and validation of Algorithm 4.2.2 L

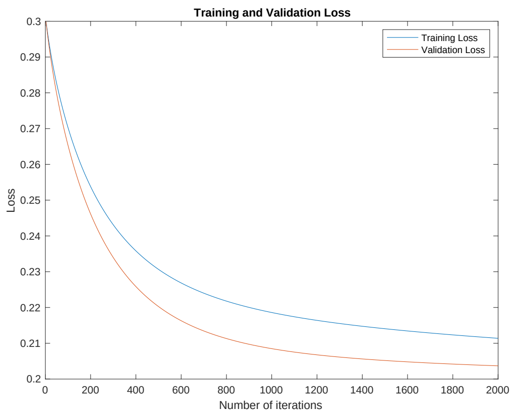

Figure 8: The loss plots of training and validation of Algorithm 4.2.2 L

From Figures 7-8, we see that Algorithm 4.2.2 L has a good fit model. This means that Algorithm 4.2.2 L appropriate learns the training dataset and generalizes well to classification the Parkinson's disease dataset.

5.4 Osteoporosis

Osteoporosis is becoming a significant public health problem in aging society. So, early detection is essential for preventing fracture. Apply this paper that uses six factors: gender, age, height, weight, smoking, and drinking alcohol. When people are aware that they are in a risk group, they can take care of themselves by exercising to strengthen their bones and muscles. They can consult a doctor to take medication, adjust their lifestyle, control weight, eat a balanced diet rich in calcium and vitamin D, take supplements, and avoid smoking and excessive alcohol consumption to promote bone density. In the case of fall history, early detection can help people recognize that they should prevent falls by early intervention for rehabilitation to improve mobility. The benefit of the country is that early detection can increase awareness and education for people to prevent fractures. The government can reduce the need for expensive treatments, surgeries, and long-term care, leading to significant healthcare savings.

The most impactful aspect of early detection is that it can increase the individual's awareness of their bone health, leading to better adherence to treatment plans and lifestyle changes that support bone strength. The long-term benefit of early detection and treatment of osteoporosis can lead to better long-term health outcomes, including a lower risk of severe fractures and the associated complications that can arise, ultimately extending life expectancy and improving life quality.

We use dataset in Harvard Dataverse (https://dataverse.harvard.edu/data set.xhtml?persistentId=doi:10.7910/DVN/UDZIJS), this data set is the original data of the original paper "Nonlinear Association between Serum Uric Acid levels and risk of Osteoporosis: A Retrospective Study", including clinical baseline data, and dual-energy X-ray measurement results. Dataset can called "Bone mineral density", that belongs to 40 variables and 1,537 observations.

Table 28: Data description for Bone mineral density dataset in Microsoft Excel

Column		Column	
1	Gender	21	P
2	Age	22	Mg
3	Height	23	Calsium
4	Weight	24	Calcitriol
5	BMI	25	Bisphosphonate
6	L1-4	26	Calcitonin
7	L1.4T	27	HTN
8	FN	28	COPD
9	FNT	29	DM
10	TL	30	Hyperlipidaemia
11	TLT	31	Hyperuricemia
12	ALT	32	AS
13	AST	33	VT
14	BUN	34	VD
15	CREA	35	OP
16	URIC	36	CAD
17	FBG	37	CKD
18	HDL-C	38	Fracture
19	LDL-C	39	Smoking
20	Ca	40	Drinking

Table 28 is overview of dataset that we use in machine learning and from Table 29 we use classification of Bone mineral density dataset by WHO.

Table 29: WHO definition of osteoporosis by BMD

Classification	T-score
Normal	$\geqslant -1.0$
Low bone mass (Osteopenia)	-1.0 to -2.5
Osteoporosis	$\leqslant -2.5$
Severe or established osteoporosis	$\leqslant -2.5$ with fragility fracture

We have 1,537 observations of dataset but we have found 36 missing data. Thus we clean data by Data Cleaner in MATLAB. So we use 1,501 observations and 6 variables (Gender, Age, Height, Weight, Smoking, and Drinking). Next we use training data at 80% and testing data at 20%. From Table 29 we have 4 classes of dataset. And we try to use every observations for traning and testing. We organize into groups by Figure 9.

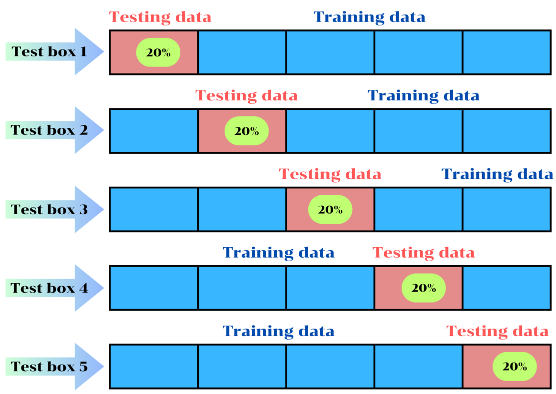

Figure 9: Groups of dataset

For applying our algorithm to solve the least sqare problems (5.0.1), we set

$$f(x,y) = \begin{cases} 0, & \forall x,y \in C, \\ -1, & otherwise \end{cases}$$

and

$$g(x,y) = \begin{cases} 0, & \forall x,y \in Q, \\ -1, & otherwise. \end{cases}$$

Then, our algorithm can be reduced to solve split feasibility problem (SFP) which was introduced by Censor and Elfving [26] as follows: finding a point $x^* \in C$ such that

$$Ax^* \in Q \tag{5.4.1}$$

where C, Q be nonempty closed and convex subsets of $\mathbb{R}^B$ and $\mathbb{R}^D$, respectively. A is an $B \times D$ real matrix.

We create 3 model for our Algorithm of split feasibility problem (5.4.1) to obtain results in machine learning (5.0.1).

1. Least square model - (L)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = H_1$ and $Q = \{T\}$.

2. Least square on constraint set L_1 - (LL_1)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_1 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

3. Least square on constraint set L_2 - (LL_2)

$$\min_{a \in C} \frac{1}{2} \|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_2^2 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

The multi-class cross-entropy loss function ($H(y, p)$), where W is the number of classes, y_k is the true label for class k, p_k is the predicted probability for class k, and the formula as follows:

$$H(y, p) = -\sum_{k=1}^{W} y_k \cdot \log(p_k)$$

We set the activation function as sigmoid, number of iteration for stop calculation is $N = 500$, hidden nodes $M = 300$, regularization parameter $\alpha_k = 0.9, \theta_k = 0.2$ and $\delta_k = 10$,

when

$$\theta_k = \begin{cases} \frac{1}{k^2\|x_k - x_{k-1}\|} & if\ x_k \neq x_{k-1}\ and\ k > N, \\ \theta & \text{otherwise,} \end{cases}$$

and

$$\delta_k = \begin{cases} \frac{1}{k^2\|x_{k-1} - x_{k-2}\|} & if\ x_{k-1} \neq x_{k-2}\ and\ k > N, \\ \delta & \text{otherwise,} \end{cases}$$

with $V(x) = cx$ such that $c = 0.9999$ for Algorithm 4.2.4, Algorithm 4.2.7, and Algorithm 4.2.8 with L, LL_1, LL_2 models. We gain the outcomes of the different parameters S by $\lambda_k = \frac{S}{max(eigenvalue(A^TA))}$ for Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2 as seen in Table 30.

Table 30: Training Time and numerical results of both of training and validation loss when λ_k is different

	S	Training Time	Loss Training	Loss Validation
Algorithm 4.2.4 L	0.2	2.4424	0.2956	0.2954
	0.5	2.3664	0.2941	0.2999
	0.7	2.7160	0.2790	0.2962
	0.9	2.3700	0.2783	0.2977
	0.999	2.3653	0.2746	0.2930
Algorithm 4.2.4 LL_1	0.2	2.4234	0.2956	0.2954
	0.5	2.4101	0.3080	0.3056
	0.7	2.8112	0.3016	0.3004
	0.9	2.3877	0.2907	0.2920
	0.999	2.3826	0.2870	0.2894
Algorithm 4.2.4 LL_2	0.2	2.5404	0.2956	0.2954
	0.5	2.9564	0.3080	0.3056
	0.7	2.5449	0.3016	0.3004
	0.9	2.5999	0.2907	0.2920
	0.999	2.5159	0.2870	0.2894

We can see that the least loss and training time of training and validation is obtained when $S = 0.999$ for each Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. Thus, we choose $S = 0.999$ for next experiment.

Next, we select $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}$ and set $\delta = 10$, $\alpha_k = 0.9, c = 0.9999$ for Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. We stop number of iterations $N = 500$ and hidden nodes $M = 300$. We gain the numeric outcomes as seen in Table 31.

Table 31: Training Time and numerical results of both of training and validation loss when θ is different

	θ	Training Time	Loss Training	Loss Validation
Algorithm 4.2.4 L	-0.2	2.3909	0.2896	0.2910
	-0.1	2.4000	0.2896	0.2910
	0	2.3760	0.2868	0.2847
	0.1	2.4346	0.2889	0.2905
	0.15	2.4328	0.2882	0.2901
	0.2	2.3888	0.2870	0.2894
Algorithm 4.2.4 LL_1	-0.2	2.4403	0.2896	0.2910
	-0.1	2.4384	0.2896	0.2910
	0	2.4037	0.2868	0.2847
	0.1	2.4249	0.2889	0.2905
	0.15	2.4234	0.2882	0.2901
	0.2	2.4228	0.2870	0.2894
Algorithm 4.2.4 LL_2	-0.2	2.5388	0.2896	0.2910
	-0.1	2.5296	0.2896	0.2910
	0	2.3708	0.2868	0.2847
	0.1	2.5492	0.2889	0.2905
	0.15	2.5894	0.2882	0.2901
	0.2	2.5288	0.2870	0.2894

We can see that the least loss and training time of training and validation is obtained when $\theta = 0.2$ for each Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. Thus, we choose $\theta = 0.2$ for next experiment.

Next, we select $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}, \theta = 0.2$ and set $\alpha_k = 0.9, c = 0.9999$ for Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. We stop number of iterations $N = 500$ and hidden nodes $M = 300$. We gain the numeric outcomes as seen in Table 32.

Table 32: Training Time and numerical results of both of training and validation loss when δ is different

			Loss	
	δ	Training Time	Training	Validation
Algorithm 4.2.4 L	-5	2.3982	0.2912	0.2921
	-1	2.3955	0.2909	0.2918
	0	2.4045	0.2910	0.2905
	1	2.5578	0.2907	0.2917
	5	2.4448	0.2901	0.2913
	10	2.3826	0.2870	0.2894
Algorithm 4.2.4 LL_1	-5	2.4599	0.2912	0.2921
	-1	2.4898	0.2909	0.2918
	0	2.3853	0.2910	0.2905
	1	2.4322	0.2907	0.2917
	5	2.3893	0.2901	0.2913
	10	2.3837	0.2870	0.2894
Algorithm 4.2.4 LL_2	-5	2.5302	0.2912	0.2921
	-1	2.5280	0.2909	0.2918
	0	2.5363	0.2910	0.2905
	1	2.5032	0.2907	0.2917
	5	2.5047	0.2901	0.2913
	10	2.5003	0.2870	0.2894

We can see that the least loss and training time of training and validation is obtained when $\delta = 10$ for each Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. Thus, we choose $\delta_k = 10$ for next experiment.

Next, we select $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}, \theta = 0.2, \delta = 10$ and set $c = 0.9999$ for Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. We stop number of iterations $N = 500$ and hidden nodes $M = 300$. We gain the numeric outcomes as seen in Table 33.

Table 33: Training Time and numerical results of both of training and validation loss when α_k is different

			Loss	
	α_k	Training Time	Training	Validation
	0.90	2.3770	0.2870	0.2894
	0.93	2.4233	0.2904	0.2915
Algorithm 4.2.4 L	0.95	2.4378	0.2912	0.2920
	0.97	2.4566	0.2919	0.2925
	0.99	2.4440	0.2926	0.2931
	0.90	2.4332	0.2870	0.2894
	0.93	2.4466	0.2904	0.2915
Algorithm 4.2.4 LL_1	0.95	2.4377	0.2912	0.2920
	0.97	2.4483	0.2919	0.2925
	0.99	2.4490	0.2926	0.2931
	0.90	2.5154	0.2870	0.2894
	0.93	2.5305	0.2904	0.2915
Algorithm 4.2.4 LL_2	0.95	2.5760	0.2912	0.2920
	0.97	2.5322	0.2919	0.2925
	0.99	2.5236	0.2926	0.2931

We can see that the least loss and training time of training and validation is obtained when $\alpha_k = 0.9$ for each Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. Thus, we choose $\alpha_k = 0.9$ for next experiment.

Next, we select $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}, \theta = 0.2, \delta = 10, \alpha_k = 0.9$ for Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. We stop number of iterations $N = 500$ and hidden nodes $M = 300$. We gain the numeric outcomes as seen in Table 34.

Table 34: Training Time and numerical results of both of training and validation loss when c is different

			Loss	
	c	Training Time	Training	Validation
Algorithm 4.2.4 L	0.89	2.4357	0.2952	0.2955
	0.91	2.4206	0.2950	0.2953
	0.95	2.4431	0.2939	0.2943
	0.97	2.4101	0.2931	0.2936
	0.9999	2.3783	0.2746	0.2930
Algorithm 4.2.4 LL_1	0.89	2.4794	0.2952	0.2955
	0.91	2.4471	0.2950	0.2953
	0.95	2.4192	0.2939	0.2943
	0.97	2.4392	0.2931	0.2936
	0.9999	2.4185	0.2874	0.2879
Algorithm 4.2.4 LL_2	0.89	2.6007	0.2952	0.2955
	0.91	2.5925	0.2950	0.2953
	0.95	2.5695	0.2939	0.2943
	0.97	2.5780	0.2931	0.2936
	0.9999	2.5399	0.2746	0.2930

We can see that the least loss and training time of training and validation is obtained when $c = 0.9999$ for each Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2. Table 30-34 show selecting every parameters, that is $\lambda_k = \frac{0.999}{max(eigenvalue(A^T A))}$, $\theta = 0.2$, $\delta = 10$, $\alpha_k = 0.9$ and $c = 0.9999$.

For comparing, we set number of iteration is 187, and hidden nodes $M = 300$ in Table 35.

Table 35: Efficiency comparative of existing algorithms and our algorithms

Test box 1	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	174	0.8478	NaN	75.00	NaN	79.97
Algorithm of Suantai LL_1 [114]	174	0.8346	NaN	75.00	NaN	79.97
Algorithm of Suantai LL_2 [114]	174	0.9004	NaN	75.00	NaN	79.97
Algorithm 4.2.4 L	174	0.8391	77.98	76.73	77.35	78.31
Algorithm 4.2.4 LL_1	174	0.8313	77.98	76.73	77.35	78.31
Algorithm 4.2.4 LL_2	174	0.8903	77.98	76.73	77.35	78.31
Algorithm 4.2.7 L	174	0.8386	89.99	75.39	82.04	80.13
Algorithm 4.2.7 LL_1	174	0.8522	89.99	75.39	82.04	80.13
Algorithm 4.2.7 LL_2	174	0.8924	89.99	75.39	82.04	80.13
Algorithm 4.2.8 L	174	0.8534	NaN	75.00	NaN	79.97
Algorithm 4.2.8 LL_1	174	0.8352	NaN	75.00	NaN	79.97
Algorithm 4.2.8 LL_2	174	0.8769	NaN	75.00	NaN	79.97
Test box 2	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	174	0.8378	NaN	75.00	NaN	79.97
Algorithm of Suantai LL_1 [114]	174	0.9761	NaN	75.00	NaN	79.97
Algorithm of Suantai LL_2 [114]	174	0.8822	NaN	75.00	NaN	79.97
Algorithm 4.2.4 L	174	0.8469	75.87	75.21	75.54	76.99
Algorithm 4.2.4 LL_1	174	0.8472	75.87	75.21	75.54	76.99
Algorithm 4.2.4 LL_2	174	0.9007	75.87	75.21	75.54	76.99
Algorithm 4.2.7 L	174	0.8325	64.96	74.88	69.57	79.80
Algorithm 4.2.7 LL_1	174	0.8857	64.96	74.88	69.57	79.80
Algorithm 4.2.7 LL_2	174	0.9006	64.96	74.88	69.57	79.80
Algorithm 4.2.8 L	174	0.8346	NaN	75.00	NaN	79.97
Algorithm 4.2.8 LL_1	174	0.8502	NaN	75.00	NaN	79.97
Algorithm 4.2.8 LL_2	174	0.8917	NaN	75.00	NaN	79.97
Test box 3	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	392	1.9425	NaN	75.00	NaN	80.07
Algorithm of Suantai LL_1 [114]	392	1.8938	NaN	75.00	NaN	80.07
Algorithm of Suantai LL_2 [114]	392	2.0126	NaN	75.00	NaN	80.07
Algorithm 4.2.4 L	392	1.8668	86.77	78.77	82.58	82.06
Algorithm 4.2.4 LL_1	392	1.9306	85.74	77.22	81.25	80.90
Algorithm 4.2.4 LL_2	392	2.0011	86.77	78.77	82.58	82.06
Algorithm 4.2.7 L	392	1.8760	NaN	75.00	NaN	80.07
Algorithm 4.2.7 LL_1	392	1.8696	NaN	75.00	NaN	80.07
Algorithm 4.2.7 LL_2	392	1.9938	NaN	75.00	NaN	80.07
Algorithm 4.2.8 L	392	1.8882	NaN	75.00	NaN	80.07
Algorithm 4.2.8 LL_1	392	1.8785	NaN	75.00	NaN	80.07
Algorithm 4.2.8 LL_2	392	1.9973	NaN	75.00	NaN	80.07

Test box 4	Iter	Time	Pre (%)	Rec (%)	F1 (%)	Acc (%)
Algorithm of Suantai L [114]	384	1.8451	NaN	75.00	NaN	80.17
Algorithm of Suantai LL_1 [114]	384	1.8382	NaN	75.00	NaN	80.17
Algorithm of Suantai LL_2 [114]	384	1.9368	NaN	75.00	NaN	80.17
Algorithm 4.2.4 L	384	1.8431	85.09	76.57	80.60	80.67
Algorithm 4.2.4 LL_1	384	1.8578	84.71	75.92	80.07	80.33
Algorithm 4.2.4 LL_2	384	1.9290	85.09	76.57	80.60	80.67
Algorithm 4.2.7 L	384	1.8329	NaN	75.00	NaN	80.17
Algorithm 4.2.7 LL_1	384	1.8517	NaN	75.00	NaN	80.17
Algorithm 4.2.7 LL_2	384	1.9435	NaN	75.00	NaN	80.17
Algorithm 4.2.8 L	384	1.8355	NaN	75.00	NaN	80.17
Algorithm 4.2.8 LL_1	384	1.8331	NaN	75.00	NaN	80.17
Algorithm 4.2.8 LL_2	384	1.9677	NaN	75.00	NaN	80.17
Test box 5	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	211	1.0199	NaN	75.00	NaN	80.10
Algorithm of Suantai LL_1 [114]	211	1.0082	NaN	75.00	NaN	80.10
Algorithm of Suantai LL_2 [114]	211	1.0669	NaN	75.00	NaN	80.10
Algorithm 4.2.4 L	211	1.0218	83.02	75.58	79.12	80.27
Algorithm 4.2.4 LL_1	211	1.0222	83.02	75.58	79.12	80.27
Algorithm 4.2.4 LL_2	211	1.0663	83.02	75.58	79.12	80.27
Algorithm 4.2.7 L	211	1.0125	90.03	75.09	81.88	80.10
Algorithm 4.2.7 LL_1	211	1.0224	90.03	75.09	81.88	80.10
Algorithm 4.2.7 LL_2	211	1.0633	90.03	75.09	81.88	80.10
Algorithm 4.2.8 L	211	1.0043	NaN	75.00	NaN	80.10
Algorithm 4.2.8 LL_1	211	1.0241	NaN	75.00	NaN	80.10
Algorithm 4.2.8 LL_2	211	1.0722	NaN	75.00	NaN	80.10

Table 35 shows that our Algorithm 4.2.4 L, Algorithm 4.2.4 LL_1 and Algorithm 4.2.4 LL_2 in test box 3 are highly performance of accuracy, F1-score, recall and precision by the number of iteration is 392 and all of matrics are better than Algorithm of Suantai [114] with $(L)-(LL_2)$ models. It is the highest possibility of exactly classifying comparison to algorithms investigations. Both of training-validation loss and accuracy plots show that our algorithm has good fitting model for Osteoporosis dataset. The NaN result for recall and precision of Suantai's algorithm [114] are obtained in the case of imbalance dataset which can be solved by many directions such as resampling, recrossvalidation, etc., in this research only deleting missing data is our data preprocessing.

Figure 10: The accuracy plots of training and validation of Algorithm 4.2.4 L

Figure 11: The loss plots of training and validation of Algorithm 4.2.4 L

From Figures 10-11, we saw that our Algorithm 4.2.4 L has a good fit model, which means that it appropriately learns the training dataset and generalizes well to classify the Osteoporosis dataset.

5.5 Breast cancer

In 2020, breast cancer was the most diagnosed cancer worldwide, more than lung and prostate cancer. The incidence of breast cancer is increasing throughout the world [79],[81]. In the United States, it is expected that 364,000 people will be diagnosed with this disease in 2040. The reason that it is becoming more common is the improvement in diagnosing this disease in developing countries. Moreover, women these day have different lifestyles from the past, such as delaying childbearing, not having children, exposure to hormonal therapy, obesity and alcohol drinking. The mortality rates have varied significantly from area to area, especially, in areas with low socio-economic status. This reflects a lack of resources or opportunities to access medical facilities for early diagnosis and timely treatment. Even in developed countries, discrepancies were also found between black and white women [84].

In Thailand, breast cancer was found to have the highest incidence among cancers in Thai women in 2012. Epidemiological studies indicate that the incidence is likely to continue increasing, from three to seven percent per year. Data from interviews and expert group meetings found that breast self-examination is the main method of breast cancer screening in Thailand [69]. According to the scantness of accessing standard screening with mammography and ultrasound. Although mammography and ultrasound are considered to be standard screening tests for diagnosing breast cancer in women aged 40 years and older [39],[40],[55],[79]. At the general policy level, Thailand has guidelines to encourage women to self-examine their breasts and be examined by a doctor/public health personnel if abnormalities are found [69]. Currently, computer-aided diagnosis (CAD) is being used increasingly for helping in the diagnosis of breast cancer [70],[108]. It is believed that this will help reduce the number of false negatives from obscure lesions or complex structure of the breast. To reassure the non-experienced radiologists in diagnosing breast cancer from mammography images [38],[67],[82]. It also decreases the rate of unnecessary biopsies [40],[56],[68]. However, the accuracy of using artificial intelligence (AI) or CAD in diagnosing breast cancer from mammography remains unclear. Particularly, if acting in place of a radiologist. More studies may need to be done to conclude this fact [46],[76].

This research aimed to study the accuracy of a new machine learning algorithm for predicting the interpretation of mammography images using real data from Phayao Hospital. The work has done by collecting BI-RADS data on patient age, mass shape, margin, and density of the mass (Table 36). Then analyzed the proposed algorithm in comparison with other algorithms for breast cancer screening, including the development of Data-preprocessing pipeline. Describe each data for using data classification as below:

1. BI-RADS assessment: 0 to 6 (ordinal)
2. Age: patient's age in years (integer)
3. Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
4. Margin: mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)
5. Density: mass density: high=1 iso=2 low=3 fat-containing=4 (ordinal)
6. Severity: benign=0 or malignant=1 (binominal).

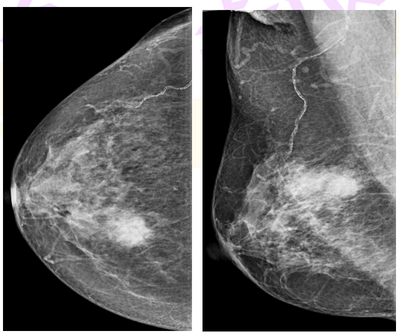

Figure 12: Mammography of 64-year-old woman

Table 36: Mammography data collection of 64-year-old woman

BI-RADS assessment	Age	Shape	Margin	Density	Severity	Pathology
4	64	3	1	1	1	invasive ductal carcinoma

From figure 12, example of mammography; right craniocaudal (CC) and mediolateral oblique (MLO) views of a 64-year-old woman, shows a lobulate-shaped, circumscribed high density mass at right inner-upper region. The BI-RADS assessment = 4. The pathological proven to be invasive ductal carcinoma.

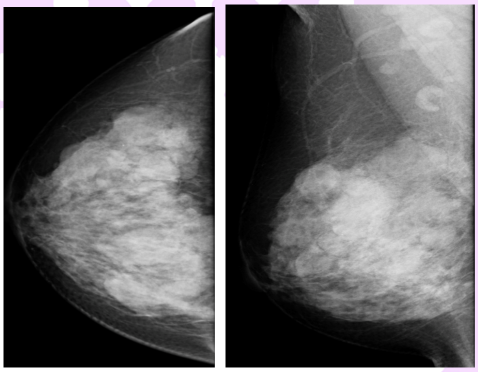

Figure 13: Mammography of 46-year-old woman

Table 37: Mammography data collection of 46-year-old woman

BI-RADS assessment	Age	Shape	Margin	Density	Severity	Pathology
2	46	1	1	1	0	N/A

From figure 13, example of mammography; right craniocaudal (CC) and mediolateral oblique (MLO) views of a 46-year-old woman, shows multiple round-shaped, circumscribed high density masses scattering in right breast. The BI-RADS assessment = 2. The ultrasonography proven to be multiple cysts.

The dataset from UCI is used to compare the efficiency of the proposed algorithm with our real dataset. For more efficiency, we thus consider three datasets as follows:

Data 1 = 829 observations and 6 classes of BI-RADS (0,2,3,4,5,6). This data set is the Discrimination of benign and malignant mammographic masses based on BI-RADS attributes and the patient's age that belongs to 961 observations and 6 variables (Age, Shape, Margin, Density, Severity are Features and BI-RADS is target) from UCI website (https://archive.ics. uci.edu/dataset/161/mammographic+mass). 132 missing data were deleted by Data Cleaner in MATLAB_R2023a.

Data 2 = 171 observations and 5 classes of BI-RADS (2,3,4,5,6). This datasets are corrected by the experts in Phayao Hospital.

Data 3 = 176 observations and 6 classes of BI-RADS (0,2,3,4,5,6). This datasets are combined from 171 observations of Phayao Hospital (Data 2) and 5 observations of 0 BI-RAD from UCI.

Next we use training data is 75% and testing data is 25%. And we try to use every observations for traning and testing, we organize into groups by Figure 14.

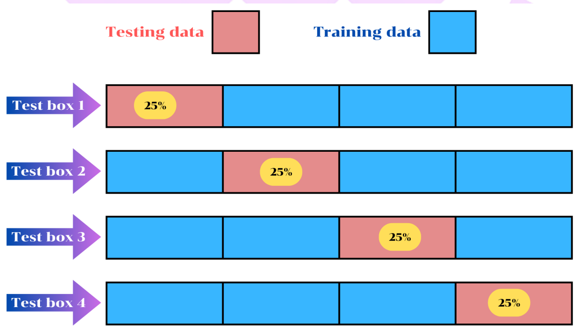

Figure 14: Groups of dataset

For applying our algorithm to solve the least sqare problems (5.0.1), we set

$$f(x,y) = \begin{cases} 0, & \forall x,y \in C, \\ -1, & otherwise \end{cases}$$

and

$$g(x,y) = \begin{cases} 0, & \forall x,y \in Q, \\ -1, & otherwise. \end{cases}$$

Then, our algorithm can be reduced to solve split feasibility problem (SFP) which was introduced by Censor and Elfving [26] as follows: finding a point $x^* \in C$ such that

$$Ax^* \in Q \tag{5.5.1}$$

where C, Q be nonempty closed and convex subsets of $\mathbb{R}^B$ and $\mathbb{R}^D$, respectively. A is an $B \times D$ real matrix.

We create 3 model for our Algorithm of split feasibility problem (5.5.1) to obtain results in machine learning (5.0.1).

1. Least square model - (L)

$$\min_{a \in C} \frac{1}{2}\|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = H_1$ and $Q = \{T\}$.

2. Least square on constraint set L_1 - (LL_1)

$$\min_{a \in C} \frac{1}{2}\|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_1 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

3. Least square on constraint set L_2 - (LL_2)

$$\min_{a\in C}\frac{1}{2}\|P_Q Ha - Ha\|_2^2$$

Setting $A = H, C = \{a \in H_1 : \|a\|_2^2 \leqslant \lambda\}$, where $\lambda > 0$ and $Q = \{T\}$.

The multi-class cross-entropy loss function ($H(y, p)$), where W is the number of classes, y_k is the true label for class k, p_k is the predicted probability for class k, and the formula as follows:

$$H(y,p) = -\sum_{k=1}^{W} y_k \cdot \log(p_k)$$

We use test box 2 for Data 1 and set the activation function as sigmoid, number of iteration for stop calculation is 2000, hidden nodes $M = 350$, regularization parameter $\alpha_k = 0.6, \delta_k = 0.9, \theta_k = 0$ for Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. We gain the outcomes of the different parameters S by $\lambda_k = \frac{S}{max(eigenvalue(A^T A))}$ for Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2 as seen in Table 38. When

$$\theta_k = \begin{cases} \frac{1}{k^2\|x_k - x_{k-1}\|} & if\ x_k \neq x_{k-1}\ and\ k > N, \\ \theta & \text{otherwise,} \end{cases}$$

and

$$\delta_k = \begin{cases} \frac{1}{k^2\|x_{k-1} - x_{k-2}\|} & if\ x_{k-1} \neq x_{k-2}\ and\ k > N, \\ \delta & \text{otherwise,} \end{cases}$$

Table 38: Training Time and numerical results of both of training and validation loss when λ_k is different

			Loss	
	S	Training Time	Training	Validation
Algorithm 4.2.9 L	0.1	0.0292	0.102302	0.100949
	0.65	0.0269	0.099504	0.098042
	0.7	0.0247	0.099391	0.097922
	0.75	0.0271	0.099285	0.097810
	0.8	0.0253	0.099187	0.097704
	0.9	0.0245	0.099010	0.097511
Algorithm 4.2.9 LL_1	0.1	0.0252	0.102302	0.100949
	0.65	0.0249	0.101715	0.100416
	0.7	0.0280	0.101715	0.100416
	0.75	0.0251	0.101715	0.100415
	0.8	0.0260	0.101715	0.100414
	0.9	0.0229	0.101715	0.100413
Algorithm 4.2.9 LL_2	0.1	0.0254	0.102302	0.100949
	0.65	0.0256	0.099504	0.098042
	0.7	0.0271	0.099391	0.097922
	0.75	0.0261	0.099285	0.097810
	0.8	0.0288	0.099187	0.097704
	0.9	0.0242	0.099010	0.097511

We can see that the least loss and training time of training and validation is obtained when $S = 0.9$ for each Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. Thus, we choose $S = 0.9$ for next experiment. Next, we select $\lambda_k = \frac{0.9}{max(eigenvalue(A^TA))}$ and set $\delta = 0.9$, $\theta = 0$ for Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. We gain the numeric outcomes as seen in Table 39.

Table 39: Training Time and numerical results of both of training and validation loss when α_k is different

			Loss	
	α_k	Training Time	Training	Validation
Algorithm 4.2.9 L	0.4	0.0302	0.099626	0.098169
	0.45	0.0276	0.099446	0.097981
	0.5	0.0261	0.099285	0.097810
	0.55	0.0269	0.099141	0.097654
	0.59	0.0287	0.099035	0.097539
	0.6	0.0257	0.099010	0.097511
Algorithm 4.2.9 LL_1	0.4	0.0266	0.101715	0.100416
	0.45	0.0300	0.101715	0.100416
	0.5	0.0274	0.101715	0.100415
	0.55	0.0230	0.101715	0.100414
	0.59	0.0308	0.101715	0.100413
	0.6	0.0242	0.101715	0.100413
Algorithm 4.2.9 LL_2	0.4	0.0275	0.099626	0.098169
	0.45	0.0276	0.099446	0.097981
	0.5	0.0259	0.099285	0.097810
	0.55	0.0324	0.099141	0.097654
	0.59	0.0352	0.099035	0.097539
	0.6	0.0248	0.099010	0.097511

We can see that the least loss and training time of training and validation is obtained when $\alpha_k = 0.6$ for each Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. Thus, we choose $\alpha_k = 0.6$ for next experiment. Next, we select $\lambda_k = \frac{0.9}{max(eigenvalue(A^TA))}, \alpha_k = 0.6$ and set $\theta = 0$ for Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. We gain the numeric outcomes as seen in Table 40.

Table 40: Training Time and numerical results of both of training and validation loss when δ is different

			Loss	
	δ	Training Time	Training	Validation
Algorithm 4.2.9 L	-0.9	0.0272	0.104091	0.103027
	-0.5	0.0259	0.103463	0.102291
	0	0.0271	0.102522	0.101202
	0.1	0.0271	0.102311	0.100962
	0.3	0.0258	0.101858	0.100457
	0.9	0.0246	0.099010	0.097511
Algorithm 4.2.9 LL_1	-0.9	0.0287	0.104091	0.103027
	-0.5	0.0293	0.103463	0.102291
	0	0.0237	0.102522	0.101202
	0.1	0.0265	0.102311	0.100962
	0.3	0.0279	0.101858	0.100457
	0.9	0.0235	0.101715	0.100413
Algorithm 4.2.9 LL_2	-0.9	0.0271	0.104091	0.103027
	-0.5	0.0279	0.103463	0.102291
	0	0.0262	0.102522	0.101202
	0.1	0.0329	0.102311	0.100962
	0.3	0.0326	0.101858	0.100457
	0.9	0.0255	0.099010	0.097511

We can see that the least loss and training time of training and validation is obtained when $\delta = 0.9$ for each Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. Thus, we choose $\delta = 0.9$ for next experiment. Next, we select $\lambda_k = \frac{0.9}{max(eigenvalue(A^TA))}, \alpha_k = 0.6$, and $\delta = 0.9$ for Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. We gain the numeric outcomes as seen in Table 41.

Table 41: Training Time and numerical results of both of training and validation loss when θ is different

	θ	Training Time	Loss	
			Training	Validation
Algorithm 4.2.9 L	-0.11	0.0250	0.130646	0.133783
	-0.04	0.0288	0.099520	0.098058
	-0.03	0.0260	0.099407	0.097939
	-0.02	0.0236	0.099285	0.097809
	-0.01	0.0265	0.099153	0.097667
	0	0.0225	0.099010	0.097511
Algorithm 4.2.9 LL_1	-0.11	0.0286	0.115907	0.116904
	-0.04	0.0286	0.101715	0.100416
	-0.03	0.0283	0.101715	0.100416
	-0.02	0.0313	0.101715	0.100415
	-0.01	0.0275	0.101715	0.100414
	0	0.0251	0.101715	0.100413
Algorithm 4.2.9 LL_2	-0.11	0.0292	0.130646	0.133783
	-0.04	0.0298	0.099520	0.098058
	-0.03	0.0253	0.099407	0.097939
	-0.02	0.0251	0.099285	0.097809
	-0.01	0.0256	0.099153	0.097667
	0	0.0234	0.099010	0.097511

We can see that the least loss and training time of training and validation is obtained when $\theta = 0$ for each Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2. From Table 38-41 shows select every parameters, that is $\lambda_k = \frac{0.9}{max(eigenvalue(A^TA))}$, $\alpha_k = 0.6$, $\delta = 0.9$ and $\theta = 0$.

Table 42: **Data 1 (829 observations)** - Efficiency comparative of existing algorithms and our algorithms

Test box 1	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	113	0.4003	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_1 [114]	113	0.4133	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_2 [114]	113	0.4969	NaN	83.33	NaN	85.51
Algorithm 4.2.9 L	113	0.3985	92.82	92.64	92.73	91.79
Algorithm 4.2.9 LL_1	113	0.4033	92.82	92.64	92.73	91.79
Algorithm 4.2.9 LL_2	113	0.4532	92.82	92.64	92.73	91.79
Test box 2	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	202	0.7060	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_1 [114]	202	0.7056	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_2 [114]	202	0.8289	NaN	83.33	NaN	85.51
Algorithm 4.2.9 L	202	0.7073	92.87	92.63	92.75	92.11
Algorithm 4.2.9 LL_1	202	0.7014	92.87	92.63	92.75	92.11
Algorithm 4.2.9 LL_2	202	0.7873	92.87	92.63	92.75	92.11
Test box 3	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	216	0.7521	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_1 [114]	216	0.7469	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_2 [114]	216	0.8517	NaN	83.33	NaN	85.51
Algorithm 4.2.9 L	216	0.7533	92.53	92.31	92.42	91.79
Algorithm 4.2.9 LL_1	216	0.7546	92.53	92.31	92.42	91.79
Algorithm 4.2.9 LL_2	216	0.8436	92.53	92.31	92.42	91.79
Test box 4	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	87	0.3186	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_1 [114]	87	0.3141	NaN	83.33	NaN	85.51
Algorithm of Suantai LL_2 [114]	87	0.3424	NaN	83.33	NaN	85.51
Algorithm 4.2.9 L	87	0.3083	90.08	89.85	89.96	89.21
Algorithm 4.2.9 LL_1	87	0.3107	90.08	89.85	89.96	89.21
Algorithm 4.2.9 LL_2	87	0.3592	90.08	89.85	89.96	89.21

From table 42 we can see that Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2 in test box 2 are high of accuracy, precision and F1-score.

Table 43: **Data 2 (171 observations)** - Efficiency comparative of existing algorithms and our algorithms

Test box 1	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	115	0.1460	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_1 [114]	115	0.1682	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_2 [114]	115	0.2392	NaN	80.00	NaN	79.53
Algorithm 4.2.9 L	115	0.1262	91.41	87.53	89.43	86.05
Algorithm 4.2.9 LL_1	115	0.1384	91.41	87.53	89.43	86.05
Algorithm 4.2.9 LL_2	115	0.1621	91.41	87.53	89.43	86.05
Test box 2	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	130	0.1302	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_1 [114]	130	0.1345	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_2 [114]	130	0.1583	NaN	80.00	NaN	79.53
Algorithm 4.2.9 L	130	0.1258	87.96	83.51	85.67	82.33
Algorithm 4.2.9 LL_1	130	0.1248	87.96	83.51	85.67	82.33
Algorithm 4.2.9 LL_2	130	0.1601	87.96	83.51	85.67	82.33
Test box 3	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	132	0.1381	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_1 [114]	132	0.1432	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_2 [114]	132	0.1856	NaN	80.00	NaN	79.53
Algorithm 4.2.9 L	132	0.1459	89.41	87.92	88.66	86.05
Algorithm 4.2.9 LL_1	132	0.1392	89.41	87.92	88.66	86.05
Algorithm 4.2.9 LL_2	132	0.1875	89.41	87.92	88.66	86.05
Test box 4	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	183	0.2216	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_1 [114]	183	0.1788	NaN	80.00	NaN	79.53
Algorithm of Suantai LL_2 [114]	183	0.2417	NaN	80.00	NaN	79.53
Algorithm 4.2.9 L	183	0.1860	89.78	87.92	88.84	86.05
Algorithm 4.2.9 LL_1	183	0.1857	89.78	87.92	88.84	86.05
Algorithm 4.2.9 LL_2	183	0.2468	89.78	87.92	88.84	86.05

From Table 43 we can see that Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2 in test box 1 are high of accuracy, precision and F1-score.

Table 44: **Data 3 (176 observations)** - Efficiency comparative of existing algorithms and our algorithms

Test box 1	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	115	0.1358	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_1 [114]	115	0.1407	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_2 [114]	115	0.1711	NaN	83.33	NaN	82.58
Algorithm 4.2.9 L	115	0.1322	92.67	89.31	90.96	87.88
Algorithm 4.2.9 LL_1	115	0.1380	92.67	89.31	90.96	87.88
Algorithm 4.2.9 LL_2	115	0.1718	92.67	89.31	90.96	87.88
Test box 2	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	140	0.1498	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_1 [114]	140	0.1586	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_2 [114]	140	0.2051	NaN	83.33	NaN	82.58
Algorithm 4.2.9 L	140	0.1600	91.41	88.28	89.82	87.12
Algorithm 4.2.9 LL_1	140	0.1778	91.41	88.28	89.82	87.12
Algorithm 4.2.9 LL_2	140	0.2052	91.41	88.28	89.82	87.12
Test box 3	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	203	0.2110	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_1 [114]	203	0.2551	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_2 [114]	203	0.2804	NaN	83.33	NaN	82.58
Algorithm 4.2.9 L	203	0.2128	91.00	89.76	90.38	87.88
Algorithm 4.2.9 LL_1	203	0.2209	91.00	89.76	90.38	87.88
Algorithm 4.2.9 LL_2	203	0.2802	91.00	89.76	90.38	87.88
Test box 4	**Iter**	**Time**	**Pre** (%)	**Rec** (%)	**F1** (%)	**Acc** (%)
Algorithm of Suantai L [114]	184	0.1898	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_1 [114]	184	0.1935	NaN	83.33	NaN	82.58
Algorithm of Suantai LL_2 [114]	184	0.2588	NaN	83.33	NaN	82.58
Algorithm 4.2.9 L	184	0.1929	91.31	89.61	90.45	87.88
Algorithm 4.2.9 LL_1	184	0.2030	91.31	89.61	90.45	87.88
Algorithm 4.2.9 LL_2	184	0.2611	91.31	89.61	90.45	87.88

From Table 44 we can see that Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2 in text box 1 are high of accuracy, precision and F1-score.

Tables 42-44 show that Algorithm 4.2.9 L, Algorithm 4.2.9 LL_1 and Algorithm 4.2.9 LL_2 have highly performance of accuracy, F1-score, recall and precision. And all of matrics of Algorithm 4.2.9 $(L) - (LL_2)$ are better than Algorithm of Suantai [114] with $(L) - (LL_2)$ models. It is the highest possibility of exactly classifying comparison to algorithms investigations. Both of training-validation loss and accuracy plots show that our algorithm has good fitting model for Mammographic dataset.

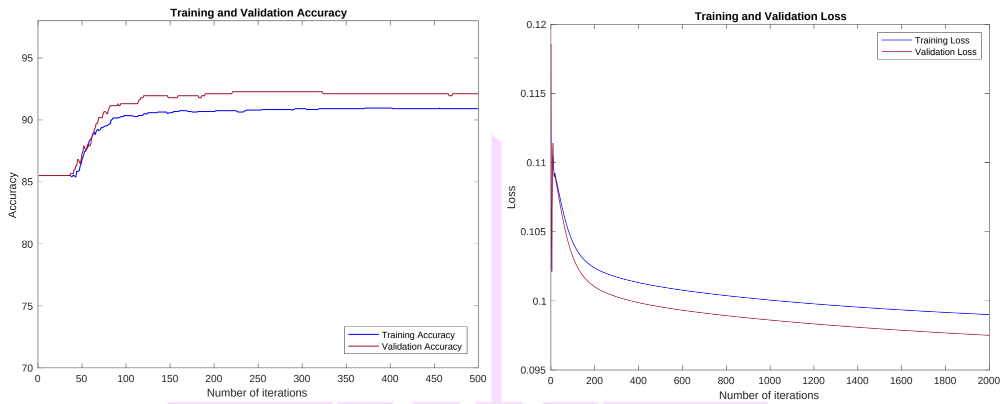

Figure 15: On the left is the accuracy plots of training and validation and on the right is the loss plots of training and validation of Algorithm 4.2.9 L, that use test box 2 in Data 1

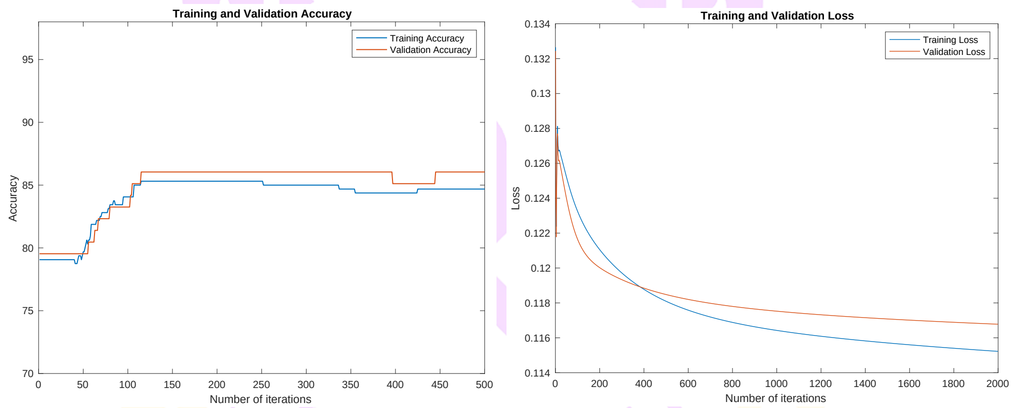

Figure 16: On the left is the accuracy plots of training and validation and on the right is the loss plots of training and validation of Algorithm 4.2.9 L, that use test box 1 in Data 2

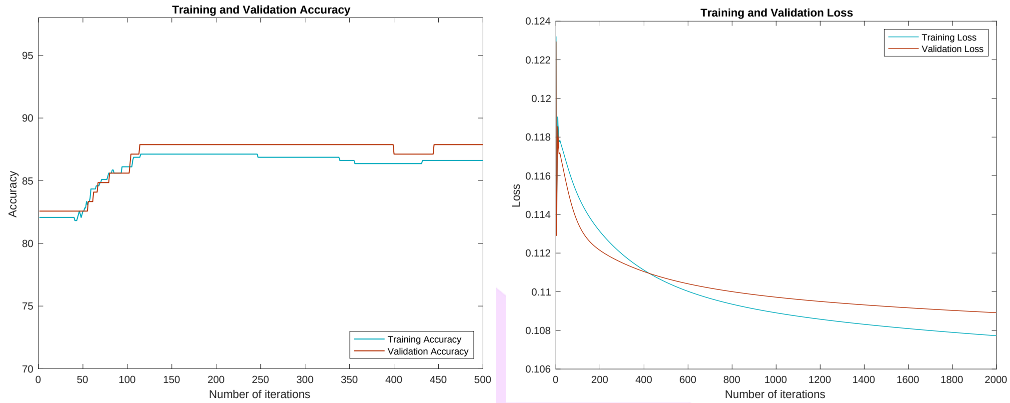

Figure 17: On the left is the accuracy plots of training and validation and on the right is the loss plots of training and validation of Algorithm 4.2.9 L , that use test box 1 in Data 3

From Figures 15-17, we saw that ours Algorithm 4.2.9 L has good fit model this means that our Algorithm 4.2.9 L appropriate learns the training dataset and generalizes well to classification the Mammographic dataset.

CHAPTER 6

CONCLUSIONS

Algorithm 5.5.1 ***(Modified viscosity type inertial extragradient algorithm for EP)***

Initialization: *Select* $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$, $\tau \in (0, \frac{1}{2}], \{\alpha_k\} \subset (0,1)$, *and* $\{\theta_k\} \subset [0, \frac{1}{3})$. ***Iterative step:*** *Let* $x_0, x_1 \in H$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = \alpha_k V(x_k) + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \underset{y \in C}{\operatorname{argmin}}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. *Calculate:*

$$z_k = \underset{y \in C}{\operatorname{argmin}}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. *Calculate the next iteration via:*

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

Replace k *by* $k+1$ *and return to* ***Step1.***

Lemma 5.5.2 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 5.5.1. Then there exists* $N > 0$ *such that*

$$\|x_{k+1} - u\|^2 \leqslant \|w_k - u\|^2 - \|x_{k+1} - w_k\|^2, \ \forall u \in EP(f, C), \ k \geqslant N.$$

Lemma 5.5.3 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 5.5.1. Then, for all* $u \in EP(f, C)$,

$$\begin{aligned} -2\alpha_k \langle x_k - u, x_k - V(x_k)\rangle \\ &\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\ &\quad + \alpha_{k+1}\|V(x_k) - x_{k+1}\|^2 - \alpha_k\|x_k - V(x_k)\|^2 - \theta_k\|x_k - u\|^2 \\ &\quad + \theta_{k-1}\|x_{k-1} - u\|^2 + (1 - 3\theta_{k+1} - \alpha_k)\|x_k - x_{k+1}\|^2. \end{aligned}$$

Lemma 5.5.4 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then* $\{x_k\}$ *generated by Algorithm 5.5.1 is bounded.*

Lemma 5.5.5 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 5.5.1. For each* $k \geqslant 1$, *define*

$$u_k = \|x_k - u\|^2 - \theta_{k-1}\|x_{k-1} - u\|^2 + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2.$$

Then $u_k \geqslant 0$.

Lemma 5.5.6 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be generated by Algorithm 5.5.1. Suppose*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0,$$

and

$$\lim_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k\|x_k - u\|^2) = 0.$$

Then $\{x_k\}$ *converges strongly to* $u \in EP(f, C)$.

Theorem 5.5.7 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then* $\{x_k\}$ *generated by Algorithm 5.5.1 strongly converges to the solution* $u = P_{EP(f,C)}V(u)$.

Algorithm 5.5.8 ***Initialization:*** *Select* $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$, $\tau \in (0, \frac{1}{2}], \{\alpha_k\} \subset (0,1)$, *and* $\{\theta_k\} \subset [0, \frac{1}{3})$. ***Iterative step:*** *Let* $x_0, x_1 \in H$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = \alpha_k x_0 + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \operatorname*{argmin}_{y \in C}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. *Calculate:*

$$z_k = \operatorname*{argmin}_{y \in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. *Calculate the next iteration via:*

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

Replace k *by* $k+1$ *and return to* ***Step1.***

Algorithm 5.5.9 ***Initialization:*** *Select* $\lambda_k \in (0, \frac{1}{2\max\{c_1,c_2\}}), \tau \in (0, \frac{1}{2}], \mu \in (0,1), \{\alpha_k\} \subset (0,1)$, *and* $\{\theta_k\} \subset [0, \frac{1}{3})$. ***Iterative step:*** *Let* $x_0, x_1 \in H$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = \alpha_k V(x_k) + (1-\alpha_k)x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = \operatorname*{argmin}_{y \in C}\{\lambda_k f(w_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 2. *Calculate:*

$$z_k = \operatorname*{argmin}_{y\in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - w_k\|^2\}.$$

Step 3. *Calculate the next iteration via:*

$$x_{k+1} = (1-\tau)w_k + \tau z_k.$$

and

$$\lambda_{k+1} = \begin{cases} \min\{\frac{\mu}{2}\frac{\|w_k-y_k\|^2+\|z_k-y_k\|^2}{f(w_k,z_k)-f(w_k,y_k)-f(y_k,z_k)}, \lambda_k\}, & \text{if } f(w_k, z_k) - f(w_k, y_k) - f(y_k, z_k) > 0, \\ \lambda_k, & \text{otherwise.} \end{cases}$$

Replace k by $k+1$ and return to ***Step1.***

Algorithm 5.5.10 ***(Modified viscosity type inertial subgradient extragradient algorithm - MVISE)***

Initialization: *Select* $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}$, $\tau \in (0, \frac{1}{2}]$, $\{\theta_k\} \subset [0, \frac{1}{3})$, $\{\alpha_k\} \subset (0, 1)$ ***Iterative step:*** *Let $x_0, x_1 \in H$ arbitrarily and calculate x_{k+1} as follows:*

Step1. *Compute:*

$$w_k = x_k + \theta_k(x_k - x_{k-1}),$$
$$y_k = \alpha_k V(x_k) + (1-\alpha_k)w_k,$$

and

$$z_k = \operatorname*{argmin}_{y\in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 2. *Choose $o_k \in \partial_2 f(y_k, \cdot)(z_k)$ such thata there exists $s_k \in N_C(z_k)$ satisfying*

$$s_k = y_k - \lambda_k o_k - z_k,$$

and construct a half-space

$$\Gamma_k \;=\; \{e \in H : \langle y_k - \lambda_k o_k - z_k, e - z_k\rangle \leqslant 0\}.$$

Compute

$$e_k \;=\; \operatorname*{argmin}_{y \in \Gamma_k}\{\lambda_k f(z_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 3. *Calculate:*

$$x_{k+1} \;=\; (1-\tau)y_k + \tau e_k.$$

*Replace k by $k+1$ and return to **Step1.***

Lemma 5.5.11 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let $\{x_k\}$ be a sequence generated by Algorithm 5.5.10. Then, for all $u \in EP(g, C)$, there exists $N > 0$ such that*

$$\|x_{k+1} - u\|^2 \leqslant \|y_k - u\|^2 - \|x_{k+1} - y_k\|^2, \;\; k \geqslant N.$$

Lemma 5.5.12 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Let $\{x_k\}$ be a sequence generated by Algorithm 5.5.10. Then, for all $u \in EP(g, C)$,*

$$\begin{aligned}
&-2\alpha_k\langle x_k - u, x_k - V(x_k)\rangle \\
&\geqslant \|x_{k+1} - u\|^2 - \|x_k - u\|^2 + 2\theta_{k+1}\|x_{k+1} - x_k\|^2 - 2\theta_k\|x_k - x_{k-1}\|^2 \\
&\quad + (1-\alpha_k)\theta_{k-1}\|x_{k-1} - u\|^2 - \theta_k\|x_k - u\|^2 + \alpha_{k+1}\|V(x_k) - x_{k+1}\|^2 \\
&\quad - \alpha_k\|x_k - V(x_k)\|^2 + (1 - 3\theta_{k+1} - \alpha_k)\|x_k - x_{k+1}\|^2.
\end{aligned}$$

Lemma 5.5.13 *Assume that Condition 4.1.1 and Condition 4.1.3 hold. Then, $\{x_k\}$ generated by Algorithm 5.5.10 is bounded.*

Lemma 5.5.14 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be a sequence generated by Algorithm 5.5.10. For each* $k \geqslant 1$*, define*

$$u_k = \|x_k - u\|^2 - \theta_{k-1}\|x_{k-1} - u\|^2 + 2\theta_k\|x_k - x_{k-1}\|^2 + \alpha_k\|x_k - V(x_k)\|^2.$$

Then $u^k \geqslant 0$*.*

Lemma 5.5.15 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Let* $\{x_k\}$ *be a sequence generated by Algorithm 5.5.10. Suppose*

$$\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0,$$

and

$$\lim_{k\to\infty} (\|x_{k+1} - u\|^2 - \theta_k\|x_k - u\|^2) = 0.$$

Then $\{x_k\}$ *converges strongly to* $u \in EP(f, C)$*.*

Theorem 5.5.16 *Suppose that Condition 4.1.1 and Condition 4.1.3 hold. Then,* $\{x_k\}$ *generated by Algorithm 5.5.10 strongly converges to the solution* $u = P_{EP(f,C)}V(u)$*.*

Algorithm 5.5.17 ***Initialization:*** *Select* $0 < \lambda_k \leqslant \lambda < \frac{1}{2\max\{c_1,c_2\}}, \tau \in (0, \frac{1}{2}], \{\theta_k\} \subset [0, \frac{1}{3}), \{\alpha_k\} \subset (0,1)$ ***Iterative step:*** *Let* $x_0, x_1 \in H$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$\begin{aligned} w_k &= x_k + \theta_k(x_k - x_{k-1}), \\ y_k &= \alpha_k x_0 + (1 - \alpha_k)w_k, \end{aligned}$$

and

$$z_k = \operatorname*{argmin}_{y\in C}\{\lambda_k f(y_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 2. *Choose* $o_k \in \partial_2 f(y_k, \cdot)(z_k)$ *such thata there exists* $s_k \in N_C(z_k)$ *satisfying*

$$s_k = y_k - \lambda_k o_k - z_k,$$

and construct a half-space

$$\Gamma_k = \{e \in H : \langle y_k - \lambda_k o_k - z_k, e - z_k\rangle \leqslant 0\}.$$

Compute

$$e_k = \operatorname*{argmin}_{y\in\Gamma_k}\{\lambda_k f(z_k, y) + \frac{1}{2}\|y - y_k\|^2\},$$

Step 3. *Calculate:*

$$x_{k+1} = (1-\tau)y_k + \tau e_k.$$

Replace k *by* $k+1$ *and return to* ***Step1.***

Algorithm 5.5.18 ***(An inertial projective Mann algorithm)***
Initialization: *Select* $\{\alpha_k\} \subset (0,1), \{\theta_k\} \subset [0, \frac{1}{3}), \{r_k\} \subset (0,\infty), \{\beta_k\} \subset (0, \frac{1}{L})$ *and* $\{\alpha_k\} \subset (0,1)$. ***Iterative step:*** *Let* $x_0, x_1 \in C$ *arbitrarily and start* $k = 0$. *Calculate* x_{k+1} *as follows:*
Step1. *Compute:*

$$w_k = x_k + \theta_k(x_k - x_{k-1}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k,$$

Step 2. *Calculate:*

$$x_{k+1} = P_E(\alpha_k w_k + (1-\alpha_k)y_k).$$

Replace k by $k+1$ and return to ***Step1.***

Theorem 5.5.19 *Suppose that Condition 4.2.1 hold. Then the sequence $\{x^k\}$ generated by Algorithm 5.5.18 converges weakly to $x^* \in \omega \cap E$.*

Algorithm 5.5.20 ***(Double relaxed inertial viscosity-type algorithm)***
Initialization: *Select $\{\theta_k\}, \{\delta_k\} \subset (-\infty, \infty), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty), \{\alpha_k\} \subset (0,1)$.* ***Iterative step:*** *Let $x_{-2}, x_{-1}, x_0 \in C$ arbitrarily and calculate x_{k+1} as follows:*

Step1. *Compute:*

$$w_k = x_k + \theta_k(x_k - x_{k-1}) + \delta_k(x_{k-1} - x_{k-2}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. *Calculate:*

$$x_{k+1} = \alpha_k V(x_k) + (1-\alpha_k)y_k.$$

Replace k by $k+1$ and return to ***Step1.***

Theorem 5.5.21 *Suppose that Condition 4.2.5 hold. Let $\{x_k\}$ be a sequence defined by Algorithm 5.5.20. Then the sequence $\{x_k\}$ converges strongly to $x^* = P_\omega V(x^*)$.*

Algorithm 5.5.22 ***Initialization:*** *Select* $\{\delta_k\} \subset (-\infty, \infty), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty), \{\alpha_k\} \subset (0, 1)$. ***Iterative step:*** *Let* $x_{-2}, x_{-1}, x_0 \in C$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = x_k + \delta_k(x_{k-1} - x_{k-2}),$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. *Calculate:*

$$x_{k+1} = \alpha_k V(x_k) + (1 - \alpha_k)y_k.$$

Replace k *by* $k+1$ *and return to* ***Step1.***

Algorithm 5.5.23 ***Initialization:*** *Select* $\{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0, \infty)$, *and* $\{\alpha_k\} \subset (0, 1)$. ***Iterative step:*** *Let* $x_0 \in C$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = x_k,$$

and

$$y_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)w_k.$$

Step 2. *Calculate:*

$$x_{k+1} = \alpha_k V(x_k) + (1 - \alpha_k)y_k.$$

*Replace k by $k+1$ and return to **Step1.***

Algorithm 5.5.24 ***(Double inertial Mann algorithm)***

Initialization: *Select* $\{\alpha_k\} \subset (0,1), \{\beta_k\} \subset (0, \frac{1}{L}), \{r_k\} \subset (0,\infty), \{\theta_k\}, \{\delta_k\} \subset (-\infty,\infty)$. ***Iterative step:*** *Let* $x_0, y_{-1}, y_0 \in H_1$ *arbitrarily and calculate* x_{k+1} *as follows:*

Step1. *Compute:*

$$w_k = T^f_{r_k}(I - \beta_k A^T(I - T^g_{r_k})A)x_k.$$

Step 2. *Calculate:*

$$y_{k+1} = (1-\alpha_k)x_k + \alpha_k w_k.$$

Step 3. *Calculate:*

$$x_{k+1} = y_{k+1} + \theta_k(y_{k+1} - y_k) + \delta_k(y_k - y_{k-1}).$$

*Replace k by $k+1$ and return to **Step1.***

Theorem 5.5.25 *Suppose that Condition 4.2.10 hold. Let* $\{x_k\}$ *be a sequence defined by Algorithm 5.5.24. Then the sequence* $\{x_k\}$ *converges weakly to* $x^* \in \omega$.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Agarwal, R. P., O'Regan, D., and Sahu, D. R. (2009). Fixed point theory for Lipschitzian-type mappings with applications. **New York: Springer**.

[2] Alvarez, F., and Attouch, H. (2001). An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. **Set-Valued Analysis**, 9, 3-11.

[3] Anh, P.N. (2013). A hybrid extragradient method extended to fixed point problems and equilibrium problems. **Optimization**, 62, 271-283.

[4] Antipin, A.S. (1995). The convergence of proximal methods to fixed points of extremal mappings and estimates of their rates of convergence. **Computational Mathematics and Mathematical Physics**, 35, 539-551.

[5] Antipin, A.S. (1997). Equilibrium programming: proximal methods. **Computational Mathematics and Mathematical Physics**, 37, 1285-1296.

[6] Antipin, A.S. (2000). Inverse iterative methods for saddle point and variational inequality problems. **Computational Mathematics and Mathematical Physics**, 40, 194-205.

[7] Arrow, K.J., and Debreu, G. (1954). Existence of an equilibrium for a competitive economy. **Econometrica**, 22, 265-290.

[8] Attouch, H., and Cabot, A. (2018). Convergence rates of inertial forward-backward algorithms. **SIAM Journal on Optimization**, 28(1), 849-874.

[9] Aubin, J. P. (1998). Optima and Equilibria: An Introduction to Nonlinear Analysis. **Springer-Verlag**, Berlin.

[10] Balestrino, R., abd Schapira, A. H. V. (2020). Parkinson disease. **European Journal of Neurology**, 27(1), 27-42.

[11] Bartle, R. G., and Sherbert, D. R. (2000). Introduction to real analysis (Vol. 2). **New York: Wiley**.

[12] Bauschke, H. H., and Borwein, J. M. (1996). On projection algorithms for solving convex feasibility problems. **SIAM review**, 38(3), 367-426.

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

[14] Benba, A., Jilbab, A., and Hammouch, A. (2016). Analysis of multiple types of voice recordings in cepstral domain using MFCC for discriminating between patients with Parkinson's disease and healthy people. **International Journal of Speech Technology**, 19, 449-456.

[15] Bhave, A., and Gaikwad, S. (2022). Prediction of cardiovascular disease using machine learning algorithms and ensemble techniques. **International Journal of Computer Science and Communication**, 13, 14-23.

[16] Biegler, L.T. (2010). **Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes**. MOS-SIAM Series on Optimization, Philadelphia.

[17] Bloem, B. R., Okun, M. S., and Klein, C. (2021). Parkinson's disease. **Lancet**, 397(10291), 2284-2303.

[18] Blum, E., and Oettli, W. (1994). From optimization and variational inequality to equilibrium problems. **The Mathematics Student**, 63, 127-

149.

[19] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In **Proceedings of the fifth annual workshop on Computational learning theory**, 144-152.

[20] Bot, R. I., Csetnek, E. R., and Lszl, S. C. (2016). An inertial forwardbackward algorithm for the minimization of the sum of two nonconvex functions. **EURO Journal on Computational Optimization**, 4, 3-25.

[21] Bozkurt, M. R., Yurtay, N., Yilmaz, Z., and Setkaya, C. (2014). Comparison of different methodologies for determining diabetes. **Turkish Journal of Electrical Engineering and Computer Sciences**, 22, 1044-1055.

[22] Brahim-Belhouari, S., and Bermak, A. (2014). Gaussian process for nonstationary time series prediction. **Computer Statistical Data Analysis**, 47, 705-712.

[23] Breiman, L. (2001). Random forests. **Machine learning**, 45, 5-32.

[24] Cegielski, A. (2012). Iterative methods for fixed point problems in Hilbert spaces. **Springer**, 2057.

[25] Celik, E., and Omurca, S. I. (2019). Improving Parkinson's disease diagnosis with machine learning methods. In **2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)**, 1-4.

[26] Censor, Y., and Elfving, T. (1994). A multiprojection algorithm using Bregman projection in a product space. **Numerical Algorithms**, 8, 221-239.

[27] Censor, Y., Gibali, A., and Reich, S. (2011) The subgradient extragradient method for solving variational inequalities in Hilbert spaces. **Journal of Optimization Theory and Applications**, 148, 318-335.

[28] Censor, Y., Gibali, A., and Reich, S. (2012). Extensions of Korpelevich's extragradient method for the variational inequality problem in Euclidean space. **Optimization**, 61(9), 1119-1132.

[29] Chatrati, S. P., Hossain, G., Goyal, A., Bhan, A., Bhattacharya, S., Gaurav, D., and Tiwari, S. M. (2020). Smart home health monitoring system for predicting type 2 diabetes and hypertension. **Journal of King Saud University: Computer and Information Sciences**, 34, 862-870.

[30] Cholamjiak, W. (2024). Fixed Point Optimization Algorithms and Their Applications. **Morgan Kaufmann**.

[31] Choubey, D. K., Kumar, M., Shukla, V., Tripathi, S., and Dhandhania, V. K. (2020). Comparative analysis of classification methods with PCA and LDA for diabetes. **Current Diabetes Reviews**, 16, 833-850.

[32] Combettes, P. L., and Hirstoaga, S. A. (2005). Equilibrium programming in Hilbert spaces. **J. Nonlinear Convex Anal.**, 6, 117-136.

[33] Cramer, J. S. (2002). The origins of logistic regression.

[34] Dafermos, S. (1980). Traffic equilibrium and variational inequalities. **Transportation Science**, 14, 42-54.

[35] Dang, V.H. (2018). Two hybrid algorithms for solving split equilibrium problems. **International Journal of Computer Mathematics**, 95(3), 561-583.

[36] Dashtipour, K., Tafreshi, A., Lee, J., and Crawley, B. (2018). Speech disorders in Parkinson?s disease: pathophysiology, medical management, and surgical approaches. **Neurodegenerative Disease Management**, 8(5), 337-348.

[37] Deng, D., and Kasabov, N. (2003). On-line pattern analysis by evolving self-organizing maps. **Neurocomputing**, 51, 87-103.

[38] Dromain, C., Boyer, B., Ferr, R., Canale, S., Delaloge, S., and Balleyguier, C. (2013). Computed-aided diagnosis (CAD) in the detection of breast cancer. **Eur J Radiol**, 82(3), 417-423.

[39] Durhan, G., Azizova, A., nder, ., Ksemehmeto?lu, K., Karakaya, J., Akp?nar, M. G., and et al. (2019). Imaging findings and clinicopathological correlation of breast cancer in women under 40 years old. **Eur J Breast Health**, 15(3), 147-152.

[40] Elter, M., Schulz-Wendtland, R., and Wittenberg, T. (2007). The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process. **Med Phys**, 34(11), 4164-4172.

[41] Eskidere, O., Karatutlu, A., and Unal, C. (2015). Detection of Parkinson's disease from vocal features using random subspace classifier ensemble. In **Proceedings of the Twelfth International Conference on Electronics, Computing, and Communication (ICECCO)**, 1-4. Almaty.

[42] Facchinei, F., and Pang, J.S. (2003). Finite-dimensional variational inequalities and complementarity problems. **Springger-Verlag**.

[43] Fan, K. (1972). A minimax inequality and applications. In O. Shisha (Ed.), **Inequalities**, Vol. III, Academic Press, New York, 103-113.

[44] Ferris, M.C., and Pang, J.-S. (1997). Engineering and economic applications of complementarity problems. **SIAM Review**, 39, 669-713.

[45] Flam, S.D., and Antipin, A.S. (1997). Equilibrium programming and proximal-like algorithms. **Mathematical Programming**, 78, 29-41.

[46] Freeman, K., Geppert, J., Stinton, C., Todkill, D., Johnson, S., Clarke, A., and et al. (2021). Use of artificial intelligence for image analysis in breast cancer screening programmes: systematic review of test accuracy. **BMJ**, 374, 1872.

[47] Frerking, J., and Westphal, U. (1989). On a property of metric projections onto closed subsets of Hilbert spaces. **Proceedings of the American Mathematical Society**, 105(3), 644-651.

[48] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological cybernetics**, 36(4), 193-202.

[49] Fukushima, M. (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. **Mathematical Programming**, 53, 99-110.

[50] Gaenslen, A., Swid, I., Liepelt-Scarfone, I., Godau, J., and Berg, D. (2011). The patients' perception of prodromal symptoms before the initial diagnosis of Parkinson?s disease. **Movement Disorders**, 26(4), 653-658.

[51] Giannessi, F., and Maugeri, A. (1995). **Variational Inequalities and Network Equilibrium Problems**. Plenum Press, New York, NY.

[52] Giannessi, F., Maugeri, A., and Pardalos, P.M. (2001). **Equilibrium Problems: Nonsmooth Optimization and Variational Inequality**

Models. Kluwer Academics Publishers, Dordrecht, Holland.

[53] Gibali, A. (2015). A new non-Lipschitzian projection method for solving variational inequalites in Euclidean spaces. **Journal of Nonlinear Analysis and Optimization**, 6(1), 41-51.

[54] Glowinski, R., Lions, J.L., and Tremolieres, R. (1981). **Numerical Analysis of Variational Inequalities**. North-Holland, Amsterdam, Holland.

[55] Grimm, L. J., Avery, C. S., Hendrick, E., and Baker, J. A. (2022). Benefits and risks of mammography screening in women ages 40 to 49 years. **J Prim Care Community Health**, 13, 21501327211058322.

[56] Guo, Z., Xie, J., Wan, Y., Zhang, M., Qiao, L., Yu, J., and et al. (2022). A review of the current state of the computer-aided diagnosis (CAD) systems for breast cancer diagnosis. **Open Life Sci**, 17(1), 1600-1611.

[57] Gwinner, J. (1983). On the penalty method for constrained variational inequalities. In J.-B. Hiriart-Urruty, W. Oettli, and J. Stoer (Eds.), **Optimization: Theory and Algorithms**, Vol. 86, Lecture Notes in Pure and Applied Mathematics, 197-211. Dekker, New York.

[58] Halpern, B. (1967). Fixed points of nonexpanding maps. **Bulletin of the American Mathematical Society**, 73, 957-961.

[59] He, Z. (2012). The split equilibrium problem and its convergence algorithms. **Journal of Inequalities and Applications**, 2012(1), 1-15.

[60] Hieu, D.V., Muu, L., and Anh, P.K. (2016). Parallel hybrid extragradient methods for pseudomonotone equilibrium problems and nonexpansive mappings. **Numerical Algorithms**, 73, 197-217.

[61] Hieu, D.V. (2017). An extension of hybrid method without extrapolation step to equilibrium problems. **Journal of Industrial Management and**

Optimization, 13, 1723-1741.

[62] Hieu, D.V. (2017). New subgradient extragradient methods for common solutions to equilibrium problems. **Computational Optimization and Applications**, 67, 571-594.

[63] Hieu, D.V. (2017). Halpern subgradient extragradient method extended to equilibrium problems. **Revista R. Acad. Cienc. Exactas Fs. Nat. Ser. A Math. RACSAM**, 111, 823-840.

[64] Hieu, D.V. (2018). An inertial-like proximal algorithm for equilibrium problems. **Math. Meth. Oper. Res.**, 88, 399-415.

[65] Huang, G. B., Zhu, Q. Y., and Siew, C. K. (2006). Extreme learning machine: theory and applications. **Neurocomputing**, 70, 489-501.

[66] Huberty, C. J. (1975). Discriminant analysis. Review of Educational Research, 45(4), 543-598.

[67] Jiang, Y., Nishikawa, R. M., Schmidt, R. A., Metz, C. E., Giger, M. L., and Doi, K. (1999). Improving breast cancer diagnosis with computer-aided diagnosis. **Academic Radiology**, 6(1), 22-33.

[68] Jiang, Y. (2002). Computer-aided diagnosis of breast cancer in mammography: Evidence and potential. **Technology in Cancer Research and Treatment**, 1(3), 211-216.

[69] Kanaungnit, P., Pongthavornkamol, N. W., and Khuhaprema, T. (2019). Breast cancer prevention and screening system in Thailand in health practitioners' perspectives. **Thai Cancer Journal**, 39(3).

[70] Katzen, J., and Dodelzon, K. (2018). A review of computer-aided detection in mammography. **Clinical Imaging**, 52, 305-309.

[71] Konnov, I.V. (2003). Application of the proximal point method to nonmonotone equilibrium problems. **Journal of Optimization Theory and Applications**, 119, 317-333.

[72] Korpelevich, G. (1976). The extragradient method for finding saddle points and other problems. **Matecon**, 12, 747-756.

[73] Kraikaew, R., and Saejung, S. (2014). Strong convergence of the Halpern subgradient extragradient method for solving variational inequalities in Hilbert spaces. **Journal of Optimization Theory and Applications**, 163(2), 399-412.

[74] Kreyszig, E. (1991). Introductory functional analysis with applications (Vol. 17). **John Wiley and Sons**.

[75] Kumari, V. A., and Chitra, R. (2013). Classification of diabetes disease using support vector machine. **International Journal of Engineering Research and Applications**, 3, 1797-1801.

[76] Larsen, M., Aglen, C. F., Hoff, S. R., Lund-Hanssen, H., and Hofvind, S. (2022). Possible strategies for use of artificial intelligence in screen-reading of mammograms, based on retrospective data from 122,969 screening examinations. **Eur Radiol**, 32(12), 8238-8246.

[77] Li, L. (2014). Diagnosis of diabetes using a weight-adjusted voting approach. In **Proceedings of the IEEE International Conference on Bioinformatics and Bioengineering**, 320-324.

[78] Li, Y., Zhang, C., Jia, Y., Wang, P., Zhang, X., and Xie, T. (2017) Simultaneous learning of speech feature and segment for classification of Parkinson disease. **Applications and Services (Healthcom)**, 1-6.

[79] Loizidou, K., Elia, R., and Pitris, C. (2023). Computer-aided breast cancer detection and classification in mammography: A comprehensive review. **Computers in Biology and Medicine**, 153, 106554.

[80] Lorenz, D. A., and Pock, T. (2015). An inertial forward-backward algorithm for monotone inclusions. **Journal of Mathematical Imaging and Vision**, 51, 311-325.

[81] Lukasiewicz, S., Czeczelewski, M., Forma, A., Baj, J., Sitarz, R., and Stanis?awek, A. (2021). Breast cancer epidemiology, risk factors, classification, prognostic markers, and current treatment strategies an updated review. **Cancers (Basel)**, 13(17).

[82] Mann, R. M., Hooley, R., Barr, R. G., and Moy, L. (2020). Novel approaches to screening for breast cancer. **Radiology**, 297(2), 266-285.

[83] Massari, H. E., Gherabi, N., Mhammedi, S., Ghandi, H., Bahaj, M., and Naqvi, M. R. (2022). The impact of ontology on the prediction of cardiovascular disease compared to machine learning algorithms. **International Journal of Online Engineering (iJOE)**, 18, 143-157.

[84] Mayrovitz, E. A., and HN, M. (Eds.) (2022). **Breast Cancer**. Brisbane: Exon Publications.

[85] McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, 5, 115-133.

[86] Moudafi, A. (1999). Proximal point algorithm extended to equilibrium problem. **Journal of Nonlinear Geometry**, 15, 91-100.

[87] Moudafi, A. (2003). Second-order differential proximal methods for equilibrium problems. **J. Inequal. Pure Appl. Math.**, 4, 18.

[88] Moudafi, A. (2011). Split monotone variational inclusions. **Journal of Optimization Theory and Applications**, 150, 275-283.

[89] Muangchoo, K. (2022). A new strongly convergent algorithm to solve pseudomonotone equilibrium problems in a real Hilbert space. **Journal of Mathematics and Computer Science**, 24, 308-322.

[90] Muu, L.D., and Oettli, W. (1992). Convergence of an adaptive penalty scheme for finding constrained equilibria. **Nonlinear Analysis: Theory, Methods and Applications**, 18(12), 1159-1166.

[91] Nagurney, A. (1993). **Network Economics: A Variational Inequality Approach**. Kluwer Academic Publishers, Dordrecht.

[92] Nash, J.F. (1950). Equilibrium points in n-person games. **Proceedings of the National Academy of Sciences of USA**, 36, 48-49.

[93] Nash, J.F. (1951). Non-cooperative games. **Annals of Mathematics**, 54, 286-295.

[94] Nikaido, H., and Isoda, K. (1955). Note on noncooperative convex games. **Pacific Journal of Mathematics**, 5, 807-815.

[95] Osilike, M. O., Aniagbosor, S. C., and Akuchu, B. G. (2002). Fixed points of asymptotically demicontractive mappings in arbitrary Banach spaces. **Panamerican Math. J.**, 12, 77-88.

[96] Patriksson, M. (1994). **The Traffic Assignment Problem: Models and Methods**. VSP International Science Publishers, Utrecht.

[97] Peeyada, P., Suparatulatorn, R., and Cholamjiak, W. (2022). An inertial Mann forward-backward splitting algorithm of variational inclusion problems and its applications. **Chaos, Solitons & Fractals**, 158, 112048.

[98] Polyak, B.T. (1964). Some methods of speeding up the convergence of iteration methods. **USSR Computational Mathematics and Mathematical Physics**, 4(5), 1-17.

[99] Quoc, T.D., Muu, L.D., and Nguyen, V.H. (2008). Extragradient algorithms extended to equilibrium problems. **Optimization**, 57, 749-776.

[100] Quinlan, J. R. (1986). Induction of decision trees. **Machine learning**, 1, 81-106.

[101] Quinlan, J. R. (2014). C4.5: Programs for Machine Learning. **Elsevier**.

[102] Sahan, S., Polat, K., Kodaz, H., and Gunes, S. (2005). The medical applications of attribute weighted artificial immune system (AWAIS): diagnosis of heart and diabetes disease. In **International Conference on Artificial Immune Systems, Springer**, 3627, 456-468.

[103] Sakar, B. E., Isenkul, M. E., Sakar, C. O., Sertbas, A., Gurgen, F., Delil, S., and Kursun, O. (2013). Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings. **IEEE Journal of Biomedical and Health Informatics**, 17(4), 828-834.

[104] Santos, P., and Scheimberg, S. (2011). An inexact subgradient algorithm for equilibrium problems. **Computational and Applied Mathematics**, 30, 91-107.

[105] Saxena, R., Sharma, S. K., Gupta, M., and Sampada, G. C. (2022). A novel approach for feature selection and classification of diabetes mellitus: Machine learning methods. **Computational Intelligence and Neuroscience**.

[106] Shehu, Y., and Iyiola, O. S. (2022). Weak convergence for variational inequalities with inertial-type method. **Applicable Analysis**, 101,

192-216.

[107] Shehu, Y., Izuchukwu, C., Yao, J. C., and Qin, X. (2023). Strongly convergent inertial extragradient type methods for equilibrium problems. **Applicable Analysis**, 102(8), 2160-2188.

[108] Shiraishi, J., Li, Q., Appelbaum, D., and Doi, K. (2011). Computer-aided diagnosis and artificial intelligence in clinical imaging. **Semin Nucl Med**, 41(6), 449-462.

[109] Signorini, A. (1959). Questioni di elasticita nonlinearizzata e semilinearizzata. **Rendiconti di Matematica e delle sue Applicazioni**, 18, 95-139.

[110] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). Using the Adap learning algorithm to forecast the onset of diabetes mellitus. In **Proceedings of the Annual Symposium on Computer Applications in Medical Care, American Medical Informatics Association**, 261-265.

[111] Stewart, C., et al. (1995). Speech dysfunction in early Parkinson's disease. **Movement Disorders**, 10(5), 562-565.

[112] Stewart, J. (2012). Calculus: early transcendentals. **Cengage Learning**.

[113] Suantai, S. (2005). Weak and strong convergence criteria of Noor iterations for asymptotically nonexpansive mappings. **J. Math. Anal. Appl.**, 311, 506-517.

[114] Suantai, S., Cholamjiak, P., Cho, Y.J., and Cholamjiak, W. (2016). On solving split equilibrium problems and fixed point problems of nonspreading multi-valued mappings in Hilbert spaces. **Fixed Point Theory Appl.**, 1-16.

[115] Takahashi, W. (2009). Introduction to nonlinear and convex analysis. **Yokohama Publishers**.

[116] Tan, K. K., and Xu, H. K. (1993). Approximating fixed points of non-expansive mappings by the Ishikawa iteration process. **Journal of Mathematical Analysis and Applications**, 178, 301-301.

[117] Thomas, T., Pradhan, N., and Dhaka, V. S. (2020). Comparative analysis to predict breast cancer using machine learning algorithms: A survey. In **2020 International Conference on Inventive Computation Technologies (ICICT), IEEE**, 192-196.

[118] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society**, 58(1), 267-288.

[119] Tran, D.Q., Dung, L.M., and Nguyen, V.H. (2008). Extragradient algorithms extended to equilibrium problems. **Optimization**, 57(6), 749-776.

[120] Wardrop, J.G. (1952). Some theoretical aspects of road traffic research. **Proceedings of the Institute of Civil Engineers, Part II**, 1, 325-378.

[121] World Health Organization. (2013). **Global action plan for the prevention and control of NCDs 2013-2020**. World Health Organization.

[122] Xu, H. K. (2002). Iterative algorithms for nonlinear operators. **J. London Math. Soc.**, 66, 240-256.

[123] Xu, H. K. (2003). An iterative approach to quadratic optimization. **J. Optim. Theory Appl.**, 116, 659-678.

[124] Yao, Y., Adamu, A., Shehu, Y., and Yao, J. C. (2024). Simple proximal-type algorithms for equilibrium problems. **Journal of Global Optimization**, 89, 1069-1098.

[125] Zhang, S. S., Lee, J. H., and Chan, C. K. (2008). Algorithms of common solutions to quasi variational inclusion and fixed point problems. **Applied Mathematics and Mechanics**, 29(5), 571-581.

[126] Zhao, Y., Liu, X., and Sun, R. (2021). Iterative algorithms of common solutions for a hierarchical fixed point problem, a system of variational inequalities, and a split equilibrium problem in Hilbert spaces. **Journal of Inequalities and Applications**, 1, 1-22.

BIOGRAPHY

BIOGRAPHY

Name Surname	Watcharaporn Yajai
Date of Birth	November 30, 1998
Place of Birth	Nan Province, Thailand
Address	1 Ropkamphangmeuangdantai Road, Nai Wiang Sub-District, Mueang Nan District, Nan Province, Thailand 55000

Education Background

2020	B.Sc. (Mathematics), Srinakarinwirot University, Bangkok, Thailand

Publications

Articles

1. **Yajai, W.**, Liawrungrueang, W., and Cholamjiak, W. (2025). A new double relaxed inertial viscosity-type algorithm for split equilibrium problems and its application to detecting osteoporosis health problems. Journal of Computational and Applied Mathematics, 116602.

2. **Yajai, W.**, Das, S., Yajai, S., and Cholamjiak, W. (2024). A modified viscosity type inertial subgradient extragradient algorithm for nonmonotone equilibrium problems and application to cardiovascular disease detection. Discrete and Continuous Dynamical Systems-S, 0-0.

3. **Yajai, W.**, Nabheerong, P., and Cholamjiak, W. (2024). A double inertial Mann algorithm for split equilibrium problems application to breast cancer screening. Journal of Nonlinear and Convex Analysis, 25(7), 1697-1716.

4. Suantai, S., **Yajai, W.**, Peeyada, P., Cholamjiak, W., and Chachvarat, P. (2022). A modified inertial viscosity extragradient type method for equilibrium problems application to classification of diabetes mellitus: Machine learning methods. AIMS Math, 8, 1102-1126.

5. Suantai, S., Kankam, K., Cholamjiak, W., and **Yajai, W.** (2022). Parallel hybrid algorithms for a finite family of G-nonexpansive mappings and its application in a novel signal recovery. Mathematics, 10(12), 2140.

Conference presentations

1. **Yajai, W.**, Cholamjiak, W. (August 29 - September 1, 2022). A Modified Inertial Viscosity Extragradient Type Methods For Equilibrium Problems Application To Classification of Diabetes Mellitus: Machine Learning Methods. In the 11^{th} International Eurasian Conference on Mathematical Sciences and Applications (IECMSA 2022), Yildiz Technical University, Istanbul, Turkey.

2. **Yajai, W.**, Cholamjiak, W. (September 8 - 9, 2022). Solving equilibrium problems using a modified inertial viscosity extragradient type method application to diabetes mellitus detection. In The international conference on Digital Image Processing and Machine Learning (ICDIPML 2022), University of Phayao, Phayao, Thailand.

3. **Yajai, W.**, Cholamjiak, W. (October 12 - 15, 2022). Diabetes mellitus detection using a modified inertial viscosity extragradient type method for equilibrium problems. In International

Workshop on Nonlinear Analysis and its Applications (IWNAA 2022), at Sakarya University of Applied Sciences and Sakarya University, Turkey.

4. **Yajai, W.**, Cholamjiak, W. (July 14, 2023). Cardiovascular disease detection using a modified viscosity type inertial subgradient extragradient algorithm for nonmonotone equilibrium problems. 14th International Conference on Fixed Point Theory and its Applications (ICFPTA 2023), Brasov, Romania.

5. **Yajai, W.**, Cholamjiak, W. (August 2 - 5, 2023). Solving nonmonotone equilibrium problems using a viscosity-type and Halpern-type inertial subgradient extragradient algorithm apply to classification for Cardiovascular disease. The 11th Asian Conference on Fixed Point Theory and Optimization (ACFPTO 2023), A-ONE The Royal Cruise Hotel, Chonburi, Thailand.

6. **Yajai, W.**, Cholamjiak, W. (August 6 - 8, 2024). A new double inertial Mann algorithm for solving split equilibrium problems application to breast cancer detection. The Fourth International Conference and Workshop on Applied Nonlinear Analysis (ICWANA 2024), at Bangsaen Heritage Hotel, Chonburi, Thailand.

7. **Yajai, W.**, Cholamjiak, W. (November 13 - 15, 2024). A double inertial embedded modified s-iteration algorithm for nonexpansive mappings in lung cancer classification. International Workshop on Nonlinear Analysis and its Applications (IWNAA 2024), School of Electrical Engineering University of Belgrade, Serbia.

8. **Yajai, W.**, Cholamjiak, W. (January 15 - 18, 2025). Efficient algorithms for fixed point and split equilibrium problems: applications in medical data classification. The 12th Asian Conference on Fixed Point Theory and Optimization 2025 (ACFPTO 2025), Duangtawan Hotel Chiang Mai, Chiang Mai, Thailand