Scholars Journal of Engineering and Technology
OPEN ACCESS
Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

Enhanced Predictive Data Modeling for Specialized Sciences using Least Squares Convex Optimization

Tran Dang Hung[1*]

[1]Ho Chi Minh City University of Industry and Trade

DOI: https://doi.org/10.36347/sjet.2025.v13i05.002 | **Received:** 18.03.2025 | **Accepted:** 24.04.2025 | **Published:** 17.05.2025

***Corresponding author:** Tran Dang Hung
Ho Chi Minh City University of Industry and Trade

Abstract **Review Article**

This paper introduces an improved approach to predictive data modeling by leveraging least squares optimization and global convex analysis. We begin with the construction of a linear predictive model and apply the least squares method to minimize residual error. Subsequently, we incorporate global convex optimization techniques to refine the model using quadratic forms. This approach offers enhanced prediction accuracy and robustness for specialized scientific datasets. The methodology is further translated into algorithmic pseudocode suitable for large-scale data programming. Real-world examples and visual illustrations validate the efficacy of the proposed model.

Keywords: Least Squares, Global Convex Optimization, Predictive Modeling, Quadratic Form, Data Fitting.

1. INTRODUCTION

With the rise of artificial intelligence and data-driven science, predictive modeling has become essential in specialized domains. Traditional modeling techniques often rely on linear assumptions and are prone to overfitting or inefficiencies in high-dimensional spaces. This paper introduces a hybrid approach - Least Squares Convex Optimization (LO) - that combines classical least squares methods with modern convex optimization frameworks to enhance predictive performance.

2. THEORETICAL BACKGROUND

In this section, we introduce the foundations of our background theories, namely linear least squares problem and global convex optimization.

2.1 Linear Least Squares Problem

The linear least squares problem is a computational problem of primary importance, which originally arose from the need to fit a linear mathematical model to given observations. To reduce the influence of errors in observations one would then like to use a greater number of measurements than the number of unknown parameters in the model. A simple example involving the relationship between spring length and suspended objects is described below.

Example: Consider Linear Spring Model

Given measurements of spring length vs. mass:

- Data: $x = [1, 2, 3, 4, 5], y = [5.2, 5.6, 5.9, 6.3, 6.7]$
- How do you predict the length of spring?

What is the best solution?

Solution If we plot these databases, our points lie exactly on a line and we going to have the solution, but some of them do not. Since the points all lie close to a line, the problem "almost" has a solution by assuming that they approximate lie on a line. It means that $y = a + bx$.

To find the coefficients a and b, we could try plugging y and x for each data point into $y = a + bx$, yielding the linear system

$\begin{cases} a+b=5.2 \\ a+2b=5.6 \\ a+3b=5.9 \\ a+4b=6.3 \\ a+5b=6.9 \end{cases}$	(1)

Unfortunately, this system cannot have any solutions.

In this section we develop an approximation method that gives us a way to change a linear system that has no solutions into a new system that has a solution. Our method is such that we change the system as little as possible, so that the solution to the new system can serve as an approximate solution for the original system. To find a solution to this example, we need some of the development tools below. ■

The resulting problem is to "solve" an overdetermined linear system of equations. In matrix terms, given a vector $y \in R^m$ and a matrix A $(m \times n$ - matrix), $m > n$, we want to find a vector $x \in R^n$such that Ax is the "best" approximation to y. We have a linear system

$Ax = y$	(2)

With the changing the vector y in (2) into a new vector $\hat{y} \in R^m$ such that has a solution

$Ax = \hat{y}$	(3)

There are many possible ways of defining the "best" solution. We choice one of which that can lead to a simple computational problem is to let x be a solution to the minimization problem

$r_0^2 = \min_x \|y - \hat{y}\|^2$	(4)

where $r = y - \hat{y}$ as the residual vector and $\|\cdot\|$ denotes the Euclidean vector norm. Problem (4) is called a **linear least squares problem** and x that satifies (4) is a linear least squares solution of the system (3). A least squares solution minimizes $\|r\|^2 = r_1^2 + r_2^2 + \ldots + r_m^2 = \sum_{i=1}^{m} r_i^2$, i.e. the sum of the squared residuals [1].

Definition 1: Least Squares Solution

Given a linear system $Ax = y$ (A is an $m \times n$ - matrix, $x \in R^n, y \in R^m$) that has no solutions, we find an approximate solution by solving $Ax = \hat{y}$, where $\hat{y} \in R^m$ such that $\min_x \|y - \hat{y}\|^2$. This approach is called Least Squares Regression (or Linear Regression), and a solution $\hat{x}$ to $Ax = \hat{y}$is called a **least**

Squares Solution.

An alternative definition:

If matrix A, an $m \times n$ - matrix, and vertor $y \in R^m$, then a least squares solution to $Ax = y$ is a vector $\hat{x}$ in R^nsuch that

$$\|A\hat{x} - y\| \le \|Ax - y\|, \forall x \in R^n$$

Definition 2: Orthogonal Projector

Let S be a nonzero subspace with orthogonal basis $\{v_1, v_2, \ldots v_n\}$.Then the projection of u onto S (denoted $p_S u$) is given by

$p_s u = \dfrac{v_1.u}{\|v_1\|} v_1 + \dfrac{v_2.u}{\|v_2\|} v_2 + \ldots + \dfrac{v_n.u}{\|v_n\|} v_n$	(5)

Theorem 3: Let y be a vector and S = col(A), S a subspace. Then the vector closest to y in S that satisfies (4) is given by $\hat{y} = p_S y$.

Proof of theorem 3

Let a vector $s \in S$, and $\hat{y} = p_S y$.

we have $y - \hat{y} \in S^{\perp}$,

and $s, \hat{y} \in S \Rightarrow \hat{y} - s \in S$ (because S is a subspace)

Therefore $y - \hat{y}$ and $\hat{y} - s$are orthogonal, so we have

$$\|y-s\|^2 = \|(y-\hat{y})+(\hat{y}-s)\|^2$$
$$= ((y-\hat{y})+(\hat{y}-s)).((y-\hat{y})+(\hat{y}-s))$$
$$= (y-\hat{y}).(y-\hat{y})+(y-\hat{y}).(\hat{y}-s)+(\hat{y}-s).(y-\hat{y})+(\hat{y}-s).(\hat{y}-s)$$
$$= (y-\hat{y}).(y-\hat{y})+(\hat{y}-s).(\hat{y}-s)+2((\hat{y}-s).(y-\hat{y}))$$
$$= \|y-\hat{y}\|^2+\|\hat{y}-s\|^2$$

So we can deduce

$$\|y-s\|^2 \geq \|y-\hat{y}\|^2, \forall s \in S$$

Therefore, no vector in *S* is closer to y than $\hat{y}$, so $\hat{y}$is the vector in *S* that is closest to *y*. Furthermore, there is equality only when $\hat{y}$ = *s*, so $\hat{y}$ is the unique closest point. ■

This approach requires an orthogonal basis for *S* = col(*A*), so the problem in this example is still have no solution. If the columns of *A* are linearly independent, i.e., rank(*A*) = *n*, then another convenient approach is to use the following theories. We call it global convex optimization (GO). For details, see [1-3].

2.2 Global Convex Optimization (GCO)

Definition 4: Convex Set and Convex Functions

i. **Convex Set**. A set Ω is convex if for any $x_1, x_2 \in \Omega$ and any $\theta$with $0 \leq \theta \leq 1$, we have $\theta x_1 + (1-\theta)x_2 \in \Omega$

ii. **Convex Functions.** A function $f: R^n \rightarrow R$is convex if *dom f* is a convex set and if for all $x, y \in domf$, and $\theta$with $0 \leq \theta \leq 1$, we have $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$

Theorem 5: Let a function $f: R^n \rightarrow R$. We now assume that *f* is twice differentiable, that is, its second deriva tive $\nabla^2 f$ (Hessian) exists at each point in *dom f*, an convex set. The following statements are equivalent.

i. *f* is convex.

ii. $f(y) \geq f(x) + \nabla f(x)^T(y-x), \forall x, y \in domf$.

iii. $\nabla^2 f(x) \geq 0, \forall x \in domf$.

Proof of Theorem 5

Firstly, we prove that *(i)* and *(ii)* are equivalent

Let's consider the case n = 1: We prove that a differentiable function $f: R \rightarrow R$is convex if and only if $f(y) \geq f(x) + f'(x)(y-x), \forall x, y \in domf$

Since the convexity of $domf$, $\forall x, y \in domf, \forall \in R, 0 \leq \theta \leq 1$ we have

$$x + \theta(y-x) = \theta y + (1-\theta)x \in domf.$$

and *f* is convex,$\forall x, y \in domf, \forall \in R, 0 < \theta \leq 1$ we have

$$f(x+\theta(y-x)) = f(\theta y + (1-\theta)x) \leq \theta f(y) + (1-\theta)f(x)(6)$$

divide both sides (6) by θ, we obtain

$$f(y) - f(x) \geq \frac{f(x+\theta(y-x)) - f(x)}{\theta} \rightarrow f'(x)(y-x), when\theta \rightarrow 0(7)$$

Therefore

$$f(y) \geq f(x) + f'(x)(y-x)(8)$$

Otherwise,$\forall x, y \in domf, x \neq y$, and $0 \leq \theta \leq 1$, and let $z = \theta x + (1-\theta)y$. Applying (8) we have

$$f(x) \geq f(z) + f'(z)(x-z)(9)$$

and

$$f(y) \geq f(z) + f'(z)(y-z)(10)$$

Multiplying (9) by θ, (10) by $1-\theta$, then adding them together, we have

$$\theta f(x) + (1-\theta)f(y) \geq f(z) = f(\theta x + (1-\theta)y)$$

which proves that *f* is convex.

In general case, we have $f: R^n \rightarrow R$.

Let $x, y \in R^n$, consider funtion g defined by

$$g(t) = f(x+th), 0 \leq t \leq 1, h = y - x \in R^n(11)$$

and it is clear that

$$g'(t) = \nabla f(x+th)^T h(12)$$

Assume first that *f* is convex, and we can imply that *g* is convex, and applying (8) we have

$$g(t) \geq g(x) + g'(x)(t - x)(13)$$

by take $t = 1, x = 0$, we have

$$g(1) \geq g(0) + g'(0)$$

which means

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)(14)$$

We have (14) holds for all *x*, *y* and we need to prove that *f* is convex

If $t = ty + (1- t)x$ Î $dom\ f$ and $s = \tilde{t}y + (1- \tilde{t})x$ Î $dom\ f$, we have

$$g(t) = f(ty + (1 - t)x) \geq f(\tilde{t}y + (1 - \tilde{t})x) + \nabla f(\tilde{t}y + (1 - \tilde{t})x)^T (y - x)(t - \tilde{t})$$

or

$$g(t) \geq g(\tilde{t}) + g'(\tilde{t})(t - \tilde{t})(15)$$

We have (15) which implies that *g* is convex.
Next, we prove that *(ii)* and *(iii)* are equivalent

By the funtion g which consider above and $\forall t, \tilde{t}$ and $\tilde{t} < t$we have
$g(t) \geq g(\tilde{t}) + g'(\tilde{t})(t - \tilde{t})(16)$ and

$$g(\tilde{t}) \geq g(t) + g'(t)(\tilde{t} - t)(17)$$

Therefore

$$g'(\tilde{t})(t - \tilde{t}) \leq g(t) - g(\tilde{t}) \leq g'(t)(t - \tilde{t})(18)$$

Dividing both sides by $(t - \tilde{t})^2$ we get

$$\frac{g'(t) - g'(\tilde{t})}{t - \tilde{t}} \geq 0(19)$$

Let $t \to \tilde{t}$, we have

$$g''(t) = h^T \nabla^2 f(x + th)^T h \geq 0(20)$$

or

$$\nabla^2 f(x) \geq 0(21)$$

Conversely, we have (21) and prove (14) is true

If $t = ty + (1- t)x$ Î $dom\ f$ and $s = \tilde{t}y + (1- \tilde{t})x$ Î $dom\ f$, $t < s$. There exists r such that $t < r < s$, by applying the expansion of Taylor's formula, we have

$$g(t) = g(s) + g'(s)(t - s) + g''(r)\frac{(t - s)^2}{2!}(22)$$

Because $g''(t) \geq 0$, we have

$$g(t) \geq g(s) + g'(s)(t - s)(23)$$

Let $t = 1$ and $s = 0$, we have

$$g(1) \geq g(0) + g'(0)$$

or

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \blacksquare$$

Definition 6: A function $f: R^n \to R$ is a quadratic form if *f* that has the form

$$f(x) = x^T Ax(24)$$

where *A* is an $n \times n$ symmetric matrix called the *matrix of the quadratic form*.
With the quadratic form satisfying formula (24) we have the gradient vector

$$\nabla f_A(x) = 2Ax(25)$$

and the Hesse matrix

$$\nabla^2 f_A(x) = 2A(26)$$

The inequality *(ii)* shows that if $\nabla f(x) = 0$, then for all $y \in dom f, f(y) \geq f(x)$, i.e., x is a global minimizer of the function *f*. We have corollary.

Corollary 7: Let a function $f: R^n \to R$. We now assume that *f* is differentiable, *dom f* is a convex set x_0 Î $dom\ f$, such that $\nabla f(x_0) = 0$. We have

If f is convex funtion then x_0 is a global minimizer of the function f and denote

$$f(x_0) = \underset{x \in dom\, f}{\text{Gmin}}\, f(x)$$

Theorem 8: Given A is an $m \times n$ matrix, $n \leq m$, such that $rank(A) = n$ and vector $b \in R^m$. We have

$$\underset{x \in R^n}{\text{Gmin}} \|Ax - b\|^2 = (A^T A)^{-1} A^T b \qquad (27)$$

where Gmin is global minimum

Proof

Let funtion $f: R^n \to R$ be defined by

$$f(x) = \|Ax - b\|^2$$

where A is an $m \times n$ matrix, $n \leq m$, such that $rank(A) = n$ and vector $b \in R^m$. We have

$$\begin{aligned} f(x) &= \|Ax - b\|^2 = \langle Ax - b, Ax - b \rangle = (Ax - b)^T (Ax - b) \\ &= (x^T A^T - b^T)(Ax - b) \\ &= x^T A^T Ax - x^T A^T b - b^T Ax + b^T b \\ &= x^T (A^T A) x - (2b^T Ax) + b^T b (because (x^T A^T b)^T = b^T (A^T)^T (x^T)^T = b^T Ax) \end{aligned}$$

the gradient vector of funtion f

$$\nabla f(x) = 2(A^T A)x - 2b^T A = 2(A^T A)x - 2A^T b$$

and the Hesse matrix of funtion f

$$\nabla^2 f(x) = 2(A^T A)$$

with $h \in R^m$, we have

$$h^T \nabla^2 f(x) h = 2[h^T (A^T A) h] = 2[(h^T A^T)(Ah)] = 2\|Ah\|^2 \geq 0$$

It means

$$\nabla^2 f(x) \geq 0$$

Furthermore, $rank(A^T A) = rank(A) = n$, so $(A^T A)$ is an invertible matrix.

Therefore, applying theorem 5 to solve the equation $\nabla f(x) = 0$, $x = (A^T A)^{-1} A^T b$, is the global minimum of funtion f. ■

3. PREDICTIVE DATA MODEL

3.1 Data Model

We are given m data sets $(x_{i1}, x_{i2}, \ldots, x_{in}), i = 1, \ldots, m$ and m relevant information $y_i, i = 1, \ldots, m$, which is denoted by D, an $m \times n$ matrix, $m > n$, and by I, an $m \times 1$ matrix

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, I = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

The problem is how to predict the value of y. The solution is to find a function f that matches this data as closely as possible. The function f is called the model (predictor function). In this paper, we consider the linear model below.

We consider a family of functions $f_i: R^n \to R, i = 1, \ldots, p$, with common domain $\text{dom}\, f_i = D$. With each $\theta = (\theta_1, \theta_2, \ldots, \theta_p) \in R^p$ we associate the function $f: R^n \to R$ given by

$$f(x) = \theta_1 f_1(x) + \theta_2 f_2(x) + \ldots + \theta_p f_p(x) = \sum_{i=1}^{p} \theta_i f_i(x) \quad (28)$$

where $f_i: R^n \to R, i = 1, \ldots, p$ is called the set of basic functions, the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_p) \in R^p$, model parameters, is our optimization variable (or coefficient vector).

In applications, the basic functions are specially chosen, using prior knowledge or experience, to reasonably model functions of interest with the finite-dimensional subspace of functions. In many fields, families of functions are often used as: Polynomials, Piecewise-linear functions, Piecewise polynomials and splines, etc. [7].

With the basic functions given, problem (28) becomes the problem of finding $\theta = (\theta_1, \theta_2, \ldots, \theta_p) \in R^p$ such that f best fits (consistent) y. We compute the residual of f and $y = (y_1, y_2, \ldots, y_m)$ to know the goodness of fit of the model, denoted by r, and given by.

$$r = y - f (29)$$

or

$$r_i = y_i - (\theta_1 f_1(x^i) + \theta_2 f_2(x^i) + \ldots + \theta_p f_p(x^i))$$

with $x^i = (x_{i1}, x_{i2}, \ldots, x_{in}), i = 1, \ldots, m$

Therefore

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} - \begin{bmatrix} f_1(x^1) & f_2(x^1) & \ldots & f_p(x^1) \\ f_1(x^2) & f_2(x^2) & \ldots & f_p(x^2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x^m) & f_2(x^m) & \ldots & f_p(x^m) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}$$

and by denoted

$$A = \begin{bmatrix} f_1(x^1) & f_2(x^1) & \ldots & f_p(x^1) \\ f_1(x^2) & f_2(x^2) & \ldots & f_p(x^2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x^m) & f_2(x^m) & \ldots & f_p(x^m) \end{bmatrix}, \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

then we rewrite expression (29)

$$r = y - A\theta$$

Then the problem (28) becomes

$$\min_{q} \left\| Aq - y \right\|^2 \qquad (30)$$

Applying theorem 8 for A, an $m \times n$ matrix, $n \leq m$, such that $rank(A) = n$ and vector $b = y \in R^m$we have the value of θthat makes *f* is the best consist of *y* below

$$q_0 = \operatorname{Gmin}_{q \hat{\mathrm{I}} R^p} \left\| Aq - y \right\|^2 = (A^T A)^{-1} A^T y$$

3.2 Modified Model

In data prediction models, model modification is often called solution update. We have several cases where modification is needed. We can add rows of matrix *A* (in model (30)), delete rows, or both to the model; otherwise, we can add, delete columns of matrix *A*, or both. This arises because the data is coming in sequence. The modification must be done with as few operations and as little storage as possible, it must satisfy the following two conditions

i. Reasonable computational cost with low algorithmic complexity. For example, recomputing QR decomposition is too expensive because it requires $O(mn^2)$ operations.
ii. Real-time feedback, the solution must be accurate within the limits of the data and the conditions of the problem; a stable method must be used.

There are many methods for model updating, such as: *QR* decomposition method, recursive least squares algorithm, using aggregation functions in fuzzy mathematics, etc. These methods are still commonly used in some cases, although they do not satisfy the requirement. Therefore, we introduce a positive diagonal weight matrix,

$$W = diag(w_1, w_2, \ldots, w_m), 0 < w_i \leq 1$$

Problem (30) is equivalent to a weighted linear least squares problem

$$\min_{q} \left\| \tilde{A}q - \tilde{y} \right\|^2 \qquad (31)$$

where $\tilde{A} = W^{\frac{1}{2}}A, \tilde{y} = W^{\frac{1}{2}}y$

However, the inclusion of the diagonal weight matrix in problem (30) changes the size greatly in applications, making this approach unstable. When A is an $m \times n$matrix with m much larger than n, up to thousands of times, solving problems (30) is computationally very difficult when the input data is large. It is a stiff problem for solving [1-8].

To solve this stiff problem, we consider the important special case where only the first p equations are weighted (p is also the number of basic functions f_i). That is, $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$, $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$and matrix $D = diag(w_1, w_2, \ldots, w_p)$ is obtained from matrix $W = diag(w_1, w_2, \ldots, w_m)$, where A_1 is an $p \times n$ matrix, $A_2$is an $(m - p) \times n$matrix, vector $y_1 \in R^p$, vector $y_2 \in R^{m-p}$. And then we have

$$\min_{q} \left\| \begin{pmatrix} D^{\frac{1}{2}} A_1 \\ A_2 \end{pmatrix} q - \begin{pmatrix} D^{\frac{1}{2}} y_1 \\ y_2 \end{pmatrix} \right\|^2 \qquad (32)$$

3.3 Model Update Strategy

To accommodate streaming data or system changes, modify matrix A by adding/removing rows or columns. Updates must be computationally efficient to maintain real-time applicability.

4. Implementation and Examples

Example 1: Linear Spring Model

Given measurements of spring length vs. mass:

- Data: $x = [1, 2, 3, 4, 5]$, $y = [5.2, 5.6, 5.9, 6.3, 6.7]$
- Model: $y = a + bx$

Solution yields: $y = 4.83 + 0.37x$

Example 2: ICPI (Increase in Consumer Price Index) Prediction Model

Quarterly ICPI data is modeled as a cubic polynomial: $f(x) = a + bx + cx^2 + dx^3$

Solution yields: $y = 1.20 + 1.25x - 0.28x^2 + 0.02x^3$

5. Algorithmic Framework

We propose the following modular pipeline for predictive modeling:

1. data = collect_data()
2. data_cleaned = preprocess_data(data)
3. features = extract_features(data_cleaned)
4. train_data, test_data = split_data(features, test_size=0.2)
5. model = select_model(model_type='LinearRegression')
6. trained_model = train_model(model, train_data)
7. model_performance = evaluate_model(trained_model, test_data)
8. predictions = model.predict(new_data)
9. optimal_params = optimize_model(predictions, trained_model)
10. update_system(optimal_params)

This structure ensures adaptability for large-scale data systems and scientific applications.

6. CONCLUSION AND FURTHER STUDY

The paper recommends a linear model for predictive data which combines the advantages of both calculus and algebra and called LSCO.

- Convex least squares optimization allows modeling of predictive data in a linear form. LSCO models help to accurately predict the output value when we have any input data.
- Positive diagonal weight matrix is used to fit the data or update the model with as few operations and as little storage as possible.

In the future, two studies will be:

- Multi-objective optimization extensions such as model:

$$\min_{q} \left\| A_1 q - y_1, A_2 q - y_2, \ldots, A_k q - y_k \right\|^2$$

where A_i is an $m_i \times n$ matrix, vector $b_i \in R^{m_i}$, $i = 1, \ldots, k$

- Application to medical prediction systems (e.g., human hearing, neural implants)
- Integration with deep learning architectures for hybrid modeling

REFERENCES

1. Stephen Boyd; Lieven Vandenberghe. Convex Optimization. Cambridge University Press (2004).
2. Jasbir S. Arora. Introduction to Optimum Design, 3rd ed. Elsevier (2012).
3. Åke Björck. Numerical Methods for Least Squares Problems. SIAM - Society for Industrial and Applied Mathematics (1996).
4. Åke Björck. Numerical Methods in Matrix Computations. Springer (2015).
5. Nocedal, J.; Wright, S. J. Numerical Optimization. Springer 2006.
6. Jacob C. Illustrating Evolutionary Computation with Mathematica. Morgan Kaufmann Publishers, San Francisco, California (2001).
7. Laguna M.; Gonz´ alez-Velarde J.-L. Computing Tools for Modeling, Optimization and Simulation. Kluwer Academic Publishers, Dordrecht/Boston/ London, 2000.
8. Stephen Boyd; Lieven Vandenberghe. Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Cambridge University Press, 2018.
9. Panos M. Pardalos; H. Edwin Romeijn. Handbook Of Global Optimization. Kluwer Academic Publishers, Dordrecht/Boston/London (2001).
10. Keith Nicholson W. Linear Algebra with Applications. McGraw Hill (2019)
11. Howard Anton; Chris Rorres. Elementary Linear Algebra: Applications Version, 11th Edition. Wiley (2013).