Journal of Mathematical Extension
Vol. 19, No. 4 (2025) (4) 1-21
ISSN: 1735-8299
URL: http://doi.org/10.30495/JME.2025.3130
Original Research Paper

Determination of the Efficient Frontier of Weak and Strong Production Possibility Set using Genetic Algorithms

Sh. Malakouti
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University

R. Kargar *
Qom Branch, Islamic Azad University

Z. Taeb
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University

H. Bagherzade
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University

L. Karamali
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University

Abstract. This article uses genetic algorithms and geometric properties of hyperplanes and a constructive way to determine strong and weak hyperplanes in the collection entitled Efficient Frontier. To this end, a mapping with domain $\mathbb{R}^{m+s\geq 0}$ to the space of real numbers is defined. This function is introduced in such a way that it's optimal points, which are also Multiple, correspond to the normal vectors of the strong and weak supporting hyperplanes of the Production Possibility

Received: August, 2024; Accepted: Accepted: April 2025
* Corresponding Author

Set (PPS). The optimal points of this function are determined through an optimization algorithm, specifically a genetic algorithm. There exists a one-to-one correspondence between the optimal points of the introduced function and the set of supporting hyperplanes. Using the proposed method, the production function under PPS conditions is obtained.
It is evident that accurately determining the boundaries of the Production Possibility Set provides useful and valuable information, including returns to scale, benchmarks, and also the stability region.
Keywords: Genetic Algorithm (GA), Data Envelopment Analysis (DEA), Affine Independence, Efficiency Frontier, Defining Hyperplane, Farkas' Lemma.

1 Introduction

Artificial neural networks (ANN) and genetic algorithms (GA) are popular machine learning technologies. They were both developed in analogy to the structures and processes that occur in nature. Due to their desired properties, they are often used to solve various modeling and optimization problems [16, 18, 19]. Although these techniques are usually used separately, together they can extend the range of their possible applications. They can be applied to problems where it is difficult to find a clear analytical solution [9, 13].

The genetic algorithm (GA) was first introduced by John Holland in the 1960s as a method for solving optimization problems using an evolutionary search process. In his 1975 book Adaptation in Natural and Artificial Systems [6], Holland laid the foundational principles for GAs, emphasizing the use of natural selection and genetic principles to solve complex problems. Later, in the 1970s, David Goldberg, a student of Holland, significantly advanced the field by popularizing GAs and applying them to various practical optimization problems. In his 1989 book [7], Genetic Algorithms in Search, Optimization, and Machine Learning, Goldberg introduced more advanced concepts and demonstrated the power of GAs in real-world applications. Furthermore, K. A. De Jong's 1975 doctoral dissertation[8], Analysis of the Behavior of a Class of Genetic Adaptive Systems, provided important insights into the performance of genetic algorithms and their potential for adaptive problem-solving. These seminal works by Holland, Goldberg, and De Jong laid the foundation for the widespread adoption and continued development of genetic algorithms in various fields of optimization and machine learning. Since then, genetic algorithms have become one of the most powerful tools in computer science, engineering, and even biology. They quickly found applications in areas such as optimal design, machine learning, and intelligent systems, and achieving notable successes in many complex problems. In addition, various forms of evolutionary programming, as subsets of genetic algorithms, have been proposed and developed, each with specific applications.

Senior executives of an organization and government institutes tend to evaluate their Decision Making Units (DMUs) to promote the productivity of the system under their management through accurate and error-free scientific results. In the early 20th century, Data Envelopment Analysis (DEA) was discussed to evaluate the constituent units of a system based on mathematical principles, and some efficient models were presented in this modern science. Since DMUs' efficiency determines a system's efficiency, their fulfillment as accurately as possible attracted the attention of researchers in this field within a very short time. In 1957, Farell [10] proposed a non-parametric model for single-input and single-output units. In 1978, Charnes et al. [5] founded a so-called "CCR model" to evaluate homogeneous multiinput and multioutput units, which play a crucial role in evaluating the efficiency of DMUs today. The model had constant returns to scale; therefore, Banker et al [3] proposed the BCC model to assess the efficiency of data with variable returns to scale.

In DEA, the DMU located on the frontier is considered efficient, while the DMU within the production possibility set is classified as inefficient. The obtained frontier is called the "production function" in microeconomics. The production function is a function that maximizes outputs by combining inputs.

Estimation of the production function frontier allows the efficiency of DMUs to be calculated and the performance of units to be studied scientifically. Defining hyperplanes of the production frontier provides access to returns to scale and other indices that more accurately attract the attention of managers with minimal computational errors.

The determination of efficient hyperplanes is of utmost importance for the determination of reference units for inefficient DMUs. In radial or non-radial models, efficient DMUs are introduced as a reference for inefficient units. Several papers have been published on finding efficient frontier. In 2005, Jahanshahloo et al [15]. presented a technique for determining defining hyperplanes based on the number of units on a hyperplane by solving a binary planning problem. The method had two fundamental weaknesses, first, all generated hyperplanes were not "defining", and second, the algorithm was not able to specify all the defining hyperplanes. In 2007, Jahanshahloo et al [14] developed a method based on the fact that each defining hyperplane passes through at least $m+s$ or $m+s-1$ units. In this study, the affine independency of the units was not considered and the calculations of the presented algorithm were very complex. Aghayi et al [1], Amirteimoori et al [2], Ghazi et al [11] and Hosseinzadeh Lotfi et al [17] also discussed about finding defining hyperplanes. In this research, an attempt was made to determine the equation of hyperplanes using GA. In practice, the modern method gives the equation of hyperplanes more accurately in a minimum period.

By using the GA and DEA models, this paper attempts to propose a method to determine the defining hyperplanes of the Production Possibility Set (PPS). The paper is organized as follows: Major DEA discussions and a brief explanation of the GA are presented in the following section. Then the proposed method is fully explained in section 3 and a practical example is explained in section 4. The final section analyzes and studies the results.

2 Background

This section discusses the main concepts of DEA and GA. DEA provides senior managers with the estimation and analysis of the DMUs of a system without personal preferences. Literature provided some methods for estimating units, which are explained briefly in this chapter. Major definitions are described first.

Definition 2.1 ([12])**.** A set of vectors a $_1$, a $_2$, . . . , a $_n$ is called linear, in an independent n-dimensional space, if $\sum_{j=1}^{n} c_j a_j = 0$ implies that

$c_j = 0$ for $j = 1, \ldots, n$.

Definition 2.2 ([12]). A set of vectors such as $A = \{a_1, \ldots, a_n\}$ is called affine independent if $\{a_j - a_1 : j = 2, \ldots, n\}$ is linearly independent.

Definition 2.3 ([12]). If set $A = \{a_1, \ldots, a_n\}$ is affine-independent, then its affine grade is shown by afrank(A) and it is calculated as afrank(A)=n-1.

Definition 2.4 ([4]). vector $d \in P$ is called a direction or ray if $\forall \lambda > 0;\ \lambda d \in P$. Where P is an infinite arbitrary set.

Definition 2.5 ([4]). A direction that cannot be written as a positive linear combination of other directions is called extreme direction.

Assuming that the input vector X, generates the output vector Y and each vector includes at least a positive component, the dual non-negative (X, Y) is called an activity. A Production Possibility set (PPS), which is generated concerning production technology, is a set consisting of all of the mentioned activities. Assuming that n is the number of all Decision Making Units (DMUs) in a system. DMU_j is the j th activity including m inputs of $X_j = (x_{1j}, \ldots, x_{mj})$ and s output of $Y_j = (y_{1j}, \ldots, y_{sj})$.

That inputs and outputs vectors are non-negative, the PPS of T_c, which is obtained by the principles of Observation, Constant Returns to Scale, Feasibility, and Convexity, is as follows:

$$T_c = \left\{ (X, Y) |\ X \geq \sum_{j=1}^{n} \lambda_j X_j,\ Y \leq \sum_{j=1}^{n} \lambda_j Y_j, \quad \lambda_j \geq 0,\ j = 1, \ldots, n \right\}$$

This set is the production possibility set of the well-known CCR model [5]. The Frontier of the set, which is a linear piecewise surface, is considered as the efficiency frontier. If $DMU_0 = (X_0, Y_0)$ is non-dominated that is ; $\forall j$; $j = 1, \ldots, n$ $(-X_j, Y_j) \underset{\neq}{\leq} (-X_0, Y_0)$, then DMU_0 is called relatively efficient. Efficient DMUs lies on the efficiency frontier; otherwise, it is detected as an inefficient unit.

Considering the principles of Observation, Feasibility, and Convexity, the production possibility set is as follows:

$$T_v = \left\{ (X, Y) | X \geq \sum_{j=1}^{n} \lambda_j X_j \leq \sum_{j=1}^{n} \lambda_j Y_j,\ \sum_{j=1}^{n} \lambda_j = 1, \lambda_j \geq 0,\ j = 1, \ldots, n \right\}$$

Banker, Charnes, Cooper presented the following model, known as the BCC model.[3]

$$
\begin{aligned}
\min\quad & \theta - \varepsilon\Big(\sum_{i=1}^{m} s_i^- + \sum_{r=1}^{s} s_r^+\Big) \\
\text{s.t.}\quad & \sum_{j=1}^{n} \lambda_j X_{ij} + s_i^- = \theta\, X_{i0}; \quad i = 1, \ldots, m \\
& \sum_{j=1}^{n} \lambda_j Y_{rj} - s_r^+ = Y_{r0}; \quad r = 1, \ldots, s \\
& \sum_{j=1}^{n} \lambda_j = 1 \\
& \lambda_j \geq 0\ , s_i^- \geq 0, s_r^+ \geq 0;\ j = 1, \ldots, n,\ i = 1, \ldots, m,\ r = 1, \ldots, s.
\end{aligned}
$$

For each optimal solution of model (1), $\theta^* = 1$ and all the slacks equals zero that is $\sum_{i=1}^{m} s_i^{-*} + \sum_{r=1}^{s} s_r^{+*} = 0$, and the evaluated DMU_0 is called "strongly efficient". If $\theta^* = 1$ and $\sum_{i=1}^{m} s_i^{-*} + \sum_{r=1}^{s} s_r^{+*} \neq 0$, it is called weakly efficient. Otherwise the evaluated DMU_0 is not efficient.

The following model is the dual form of model (1), which is called the multiple form of the BCC model:

$$
\begin{aligned}
\max\quad & U^T Y_0 + u_0 \\
\text{s.t.}\quad & V^T X_0 = 1 \\
& U^T Y_j - V^T X_j + u_0 \leq 0 \qquad j = 1, \ldots, n \\
& U \geq 0, \qquad V \geq 0
\end{aligned}
\tag{1}
$$

The set of hyperplanes that construct the frontier of the T_v in the BCC model, is as follows:

$$
\begin{aligned}
S = \{&(U, V, u_o) | \forall j (j \in J; UY_j - VX_j + u_0 \leq 0), \\
&\exists\, j (j \in J; UY_j - VX_j + u_0 = 0)\}
\end{aligned}
$$

Where J is the set of strong efficient DMUs.

Some advanced models were then discussed in DEA to make system analysis more accurate and scientific. Effective algorithms have been presented with the advancement of different sciences, especially computer, in recent years. The Genetic Algorithm (GA) was one of the advanced algorithms utilized in the proposed method of the paper.

This paper applies Genetic Algorithms (GAs) among various metaheuristic approaches due to their many advantages, such as their ability to efficiently explore large search spaces, handle complex optimization problems, and avoid local optima. Compared to other popular metaheuristic algorithms like Simulated Annealing (SA) and Particle Swarm Optimization (PSO), GAs offer a more flexible and robust framework for optimization tasks.

While SA can sometimes be prone to getting stuck in local optima, and PSO can be sensitive to parameter settings, GAs tend to provide a better balance between exploration and exploitation, making them particularly well-suited for the problem addressed in this paper.

The GA is used for solving some optimization problems. The method acts are based on the principle of survival in nature and gene functions. A set of feasible solutions, called chromosomes, is generated and then measured under the solutions' fitness function (optimizer). In the next step, some are selected based on the ascending or descending order of solutions and using some random methods. A new generation is produced using gene, crossover, and mutation operators. The trend continues as long as achieving convergence conditions. The final solution obtained from the trend is not necessarily optimal, but it is acceptable. Different finalization conditions, such as a certain number of reproductions, limit the algorithm. There are various closing conditions for the algorithm to be finished; such as time, a certain number of reproductions, the optimum difference between two generations, and the tolerance of the optimum solutions. A GA flowchart is shown as follows (Figure 1):

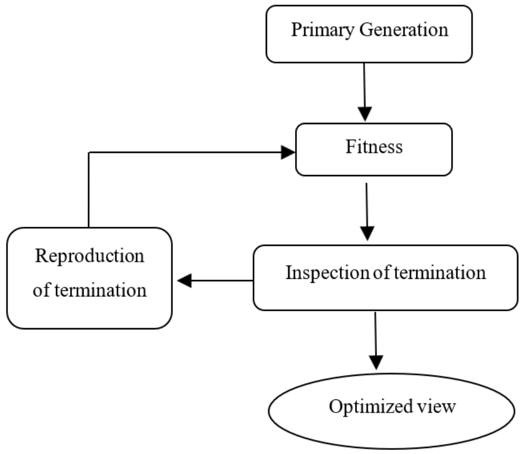

Figure 1: A *GA* flowchart

Sometimes, the current generation will be improved to determine a new improved generation or so-called genetic mutation. Below is the flowchart (Figure 2):

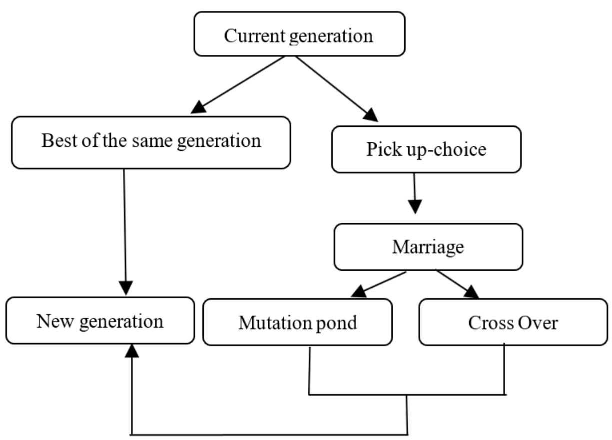

Figure 2: Genetic new generation

3 The Proposed Method

The determination of the efficiency frontier is crucial in DEA for accurately estimating units, conducting sensitivity analysis, and assessing returns to scale. This paper discuss a method to determine the defining hyperplane equations based on the PPS geometry structure. Some theorems are stated to understand the structure of the method accurately and they are proved. Then the proposed algorithm is described.

Theorem 3.1. *If F_S is defined as the following set,*

$$F_S = \{(X, Y)|(X, Y) \geq 0, \forall(U, V, u_0) \in S;\ UY - VX + u_0 \leq 0\},$$

then $T_v = F_S$.

Proof. Let $(X, Y) \in T_v$, hence (X, Y) is a point in the PPS of the BCC model and suppose $(U, V, u_0) \in S$, then

$$\exists(\lambda_1, \ldots, \lambda_n) \geq 0, \sum_{l=1}^{n} \lambda_l = 1,\ X \geq \sum_{l=1}^{n} \lambda_l X_l\, Y \leq \sum_{l=1}^{n} \lambda_l Y_l.$$

Hence, $-VX \leq -\sum_{l=1}^{n} \lambda_l V X_l$, $UY \leq \sum_{l=1}^{n} \lambda_l U Y_l$. Consequently,

$$UY - VX + u_0 \leq \sum_{l=1}^{n} \lambda_l U Y_l - \sum_{l=1}^{n} \lambda_l V X_l + u_0 = \sum_{l=1}^{n} \lambda_l (UY_l - VX_l + u_0) \leq 0.$$

It shows that $(X, Y) \in F_S$. □

The opposite is proven using proof by contradiction. It is assumed that $(X, Y) \in F_S$; however, $(X, Y) \notin T_v$. Therefore, the relative efficiency score of Model (2-3) is $\theta^* > 1$. Meantime there exist (U, V, u_0) in its dual model (2-4) in which $VX = 1$, $UY + u_0 > 1$, and $(U, V) \geq 0$, $UY_j - VX_j + u_0 \leq 0 \quad j = 1, \ldots, n$. As a result, $UY - VX + u_0 > 0$. On the other hand, $(U, V, u_0) \in S$, which this leads to a contradiction, and thus the theorem is proven.

Theorem 3.2. *If the set of extreme directions of set S is denoted by S_R and also $F_{S_R} = \{\,(X, Y)|\ \forall(U, V, u_0) \in S_R;\ UY - VX + u_0 \leq 0\}$ is assumed, then, $F_{S_R} = F_S$.*

Proof. It is obvious that $F_S \subseteq F_{S_R}$. Now, assume that $(X, Y) \in F_{S_R}$ and, $(U, V, u_0) \in S$, then $\exists(\lambda_1, \ldots, \lambda_K) \geq 0, (U_i, V_i, u_{0_i}) \in S_R$; $(U, V, u_0) = \sum_{i=1}^{K} \lambda_i (U_i, V_i, u_{0_i})$, where K is assumed the cardinal of S_R. Therefore, $(X, Y) \in F_S$. □

Lemma 3.3. *If inefficient and weakly efficient units are deleted from the set of observations and the PPS generated by principles of Observation, Feasibility, and Convexity, is called T_{v_0}, then $T_v = T_{v_0}$.*

Proof. Let (X_0, Y_0) is an inefficient or weakly efficient unit, and if $(X_1, Y_1) = \theta^*(X_0, Y_0) + (-S^-, S^+)$ is assumed to be a Pareto efficient unit, then $(X_1, -Y_1) \leq (X_0, -Y_0)$. Therefore, according to the Feasibility principle, (X_0, Y_0) is produced by (X_1, Y_1). On the other hand, (X_1, Y_1) is the convex combination of some of the observed strong efficient units. Therefore, $T_v = T_{v_0}$, hence $S_J = S_R$, where J is the set of strong efficient units. □

Theorem 3.4. $(U, V, u_0) \in S_R$ *and* $(U, V) > 0$ *if and only if*

1. *For any observed* $DMU_j = (X_j, Y_j)\,; j = 1, .., n$ *, it can be assumed that* $UY_j - VX_j + u_0 \leq 0$.
2. *A set of* T_v *with afrank* $= m + s - 1$, *is binding on the hyperplane defined by* (U, V, u_0).

Proof. Proof of the forward direction: If condition (A) is met, it will be clear that $(U, V, u_0) \in S$. Condition (B) shows that the set

$$A_1 = \{(X^t, Y^t) : (X^t, Y^t) \in T_v,\ t = 0, 1, 2, 3, \ldots, m + s\}$$

with afrank $(A_1) = m + s - 1$ is binding on the hyperplane defined by (U, V). Therefore, (U, V) is perpendicular to $A = A_1 - \{(X^0, Y^0)\}$ and $rank(A) = m+s-1$. Consequently, only one of its collinear vectors may have this property, because the kernel of the matrix whose rows are made up of the elements of A has only one dimension. As a result, (U, V, u_0) is not a non-negative combination of S members; i.e. $(U, V, u_0) \in S_R$.

Proof of the reverse direction: It is assumed that $(U, V, u_0) \in S_R$. Therefore, Condition (A) is met. Farkas' lemma is used for proving

Condition (B). If the affine rank of the desired set such as A of T_v, in which $UY - VX + u_0 = 0$ hyperplane is binding, is lower than $m+s-1$, as per definition S, $A \neq \phi$. Therefore, the above equation is a facet with $k < m + s - 1$ dimension. Consequently, there is at least a k -member subset of S_R in which

$$\begin{array}{ll} \forall\,(X,Y) \quad (X,Y) \in A & : U_t Y - V_t X + {u_0}^t = 0 \quad\; t = 1,\ldots,k \\ \forall t\ t = 1,\ldots,k\ \exists\left(X^t, Y^t\right) \in T_v\ - A & :\ U_t Y^t - V_t X^t + {u_0}^t = 0 \end{array}$$

If $(X_0, Y_0) \in A$, the equation $U_t Y_0 - V_t X_0 + {u_0}^t < 0$ will be obtained for $k+1 \le t \le p$. It is claimed that

$$\sum_{t=1}^{k} c_t \left(U_t, -V_t\right) = (U, -V) \quad c_t \ge 0\ , t = 1,\ldots,k.$$

If the claim is not valid, as per Farkas' lemma, there is a (X_1, Y_1) in which

$$\begin{array}{l} U_t Y_1 - V_t X_1 \le 0 \quad\; t = 1,\ldots,k \\ UY_1 - VX_1 > 0 \end{array}$$

Now, $\mu > 0$ is selected in a way that

$$\begin{array}{l} X_0 + \mu X_1 \ge 0 \\ Y_0\ + \mu Y_1 \ge 0 \\ U_t Y_0 - V_t X_0 + {u_0}^t + \mu\left(U_t Y_1 - V_t X_1\right) \le 0 \quad\; t = k+1,\ldots,l \end{array}$$

Therefore, $(X_D, Y_D) = (X_0 + \mu X_1, Y_0\ + \mu Y_1) \in T_v$, but $UY_D - VX_D + u_0 > 0$. Thus, the claim is true. On the other hand, $u_0 = \sum_{t=1}^{k} c_t u_0^t$ if $\sum_{t=1}^{k} c_t \left(U_t Y_0 - V_t X_0 + {u_0}^t\right) = 0$ and $UY_0 - VX_0 + u_0 = 0$. Therefore, (U, V, u_0) is not an extreme direction, and this contradiction originated form $k \neq m + s - 1$ assumption. □

Theorem 3.5. *Let $(U, V, u_0) \in S$ and k includes the number of zero components of vector (U, V), then $(U, V, u_0) \in S_R$ if and only if the hyperplane defined by (U, V, u_0) is binding on a set of strong efficient units with afrank $= m + s - k - 1$.*

Proof. Proof of the forward direction: Let $(U, V, u_0) \in S_R$ is the normal vector of the defining hyperplane binding on the strong efficient unit (X_0, Y_0), and k is the number of zero components of (U, V), then $u_0 = VX_0 - UY_0$. Since DMU_o is an extreme unit, it should be binding on $m + s - k - 1$ of the following equation:

$$\begin{aligned}&U(Y_j - Y_0) - V(X_j - X_0) \leq 0 \; j \in J - \{0\}\\ &U, V \geq 0\end{aligned}$$

Therefore, affine rank of this set equals $m + s - k - 1$.

Proof of the reverse direction: It is assumed that $k = 0$. Therefore it is binding on a set with the affine rank of $m + s - 1$, as per Theorem (3), $(U, V, u_0) \in S_R$. Otherwise, if $k > 0$ let $P = P_U \bigcup P_V$, in which P_U and P_V are the zero components of U and V, respectively, without loss of generality, it is assumed that $P_{\boldsymbol{U}} = \{1, 2, \dots, k_U\}$. The hyperplane passes through the strong efficient units such as $A_0 = \{(X_t, Y_t) : t \in J\}$ with $m + s - k - 1$ affine rank, i.e. rank of $B_0 = A_0 - (X_0, Y_0)$ is equal to $m + s - k - 1$. Therefore, $\eta_1 > 0$ is available:

$$(X_{P_1}, Y_{P_1}) = (X_0, Y_0) - \eta_1 e_{m+1} \in T_v$$

Therefore, the hyperplane H_0, passes through (X_{P_1}, Y_{P_1}). On the other hand, at least one hyperplane such as H_1 with normal vector $(U, V) \neq 0$, $U_1 = 1$ passes through A_0. Therefore,

$$\begin{gathered}\{(X, Y) : UY - VX + u_0 = 0\} \cap \{(X, Y) : (X, Y) = (X_0, Y_0) + te_{m+1},\\ t \in \mathbb{R}\} = \{(X_0, Y_0)\}.\end{gathered}$$

Consequently, $(X_{P_1}, Y_{P_1}) \notin Affine\{A_0\}$ and $B_1 = B_0 \cup \{e_1\}$ dimension is equal $m + s - (k - 1) - 1$. As a result, the affine rank of $A_1 = A_0 \cup \{(X_{p_1}, Y_{p_1})\}$ equals B_1 dimension. This action continues k_U times, and H_0 passes through a set such as $A_{k_u} = A_0 \cup \{(X_{P_i}, Y_{P_i}) : i = 1, \dots, k_U\}$ that $afrank(A_{k_u}) = m + s - (k - k_{\boldsymbol{U}}) - 1$.

Without loss of generality, it is reassumed that $P_V = \{1, 2, \dots, k_V\}$. As $k_U + k_V = k$, then

$$(X_{Q_1}, Y_{Q_1}) = (X_0, Y_0) + e_1 \in T_v.$$

Therefore, the hyperplane H_0 passes through (X_{Q_1}, Y_{Q_1}). On the other hand, at least one hyperplane, such as H_{k_U+1} with the normal vector $(U, V) \neq 0$, $V_1 = 1$, passes through A_{k_U}. Therefore,

$$\{(X, Y) : UY - VX + u_0 = 0\} \cap \{(X, Y) : (X, Y) = (X_0, Y_0) + te_1, \\ t \in \mathbb{R}\} = \{(X_0, Y_0)\}.$$

Hence, $(X_{Q_1}, Y_{Q_1}) \notin Affine\{A_{k_U}\}$ and $B_{k_U+1} = B_{k_U} \bigcup \{e_1\}$ dimension equals $m + s - (k - k_U - 1) - 1$. Therefore, the affine rank $A_{k_U+1} = A_{k_U} \bigcup \{(X_{q_1}, Y_{q_1})\}$ equals B_{k_U+1} dimension.

This action continues k_U times and H_0 passes through a set such as $A_k = A_{k_u} \bigcup \{(X_{Q_i}, Y_{Q_i}) : i = 1, \ldots, k_V\}$ with $afrank(A_k) = m + s - (k - k_{\boldsymbol{U}} - k_V) - 1 = m + s - 1$. Therefore, H_0 pass through A_k set, which enjoys $m+s-1$ affine rank. As per Theorem 3.1, $(U, V, u_0) \in S_R$. $\square$

Theorem 3.1 states that the production possibility set of BCC model (T_v), is the intersection of all half-spaces corresponding to the supporting hyperplanes of T_v. Theorem 3.2 states that to compute F_{S_R}, it is not necessary to consider all supporting hyperplanes; it is sufficient to extract the extreme normal vectors from the normal vectors of the supporting hyperplanes. The entire PPS can then be constructed from the intersection of their half-spaces. Theorem 3.4 and theorem 3.5 state that a defining hyperpalne, is a hyperplane binding on strong efficient points with *afrank* equals to $m + s - k - 1$. Therefore, based on the four stated theorems, to determine the efficiency frontier of the production possibility set T_v, it is sufficient to obtain the extreme normal vectors of the defining hyperplanes of T_v.

According to what has been stated, the objective is to identify supporting hyperplanes. For this purpose, a function has been designed whose optimal points correspond to the normal vectors of the supporting hyperplanes of the production possibility set. Using the heuristic GA algorithm, the optimal points of the function are obtained. Since GA or any other optimization algorithm does not find multiple optimal points, the function is modified by applying sequential penalties to ensure that each new optimal point is unique. To achieve this, it is checked that the set of points lying on the new hyperplane does not belong to the family whose elements are the sets of points lying on the previously identified supporting hyperplanes. This process continues until no further optimal points exist. Since GA guarantees that it finds an optimal point for any function, no supporting hyperplane remains undiscovered.

The following algorithm is expressed aiming at finding S_R set.

Algorithm 1 The Proposed Algorithm

```
Begin
Let $i = 1$, $opt = 0$, $J = \phi$, $P = \phi$, $\Gamma = \phi$, $optset = \phi$, $\varepsilon = 10^{-5}$
Obtain the strongly efficient DMUs using Model (1) and assign their indices to set J.
Perform steps A, B, and C as long as $opt = 0$ .
Construct the function $f$ as follows:
    $$f(U,V) : [0,1]^{m+s} \to \mathbb{R} \qquad , \qquad f(U,V) = \beta_1 + \beta_2$$
To compute $\beta_1$ and $\beta_2$ , $u_0$ is first obtained in the form of $u_0 = \max\{UY_j - VX_j : j \in J\}$.
Then, the set $A = \{ (X_j, Y_j) : \frac{|UY_j - VX_j - u_0|}{\|(U,V)\|_2} \leq \varepsilon : j \in J\}$ is computed.
Finally, $\beta_1$ and $\beta_2$ is obtained from the following rule:
if $A \in \Gamma$, then
    $\beta_2 = M$ & $\beta_1 = M$
else
    if $afrank(A) = m + s - k - 1$, then
        $\beta_2 = 0$ & $\beta_1 = 0$
    else
        $\beta_2 = 0$ & $\beta_1 = M$.
    end if
end if
Where $k$ is the number of zero components of the vector $(U,V)$.
Let $populationsize = (m + s)200 + 100i$ & $[(U_i, V_i), opt] = GA(f)$.
If $opt = 0$, then place:
    $$u_{0_i} = \max\{U_iY_j - V_iX_j : j \in J\}$$
    $$P = P \cup \{(U_i, V_i, u_{0i})\}, \quad optset = \{j \in J : \frac{|U_iY_j - V_iX_j - u_{0i}|}{\|(U_i, V_i)\|_2} \leq \varepsilon\},$$
    $$\Gamma = \Gamma \bigcup \{optset\}, \qquad i = i + 1.$$
Print the set of vectors $P$.
End.
```

4 Numerical Example

The example mentioned in this section includes 30 DMUs, each with two inputs and one output (see Table 1). Executing step 3 of the algorithm, ten units are detected strongly efficient using model (1).

Table 1: Efficiency Score of observed DMUs.

DMU No.	I1	I2	O	Efficiency
1	25	2	8	1.000
2	10	11	17	0.535
3	14	9	38	1.000
4	10	9	13	0.490
5	9	18	38	1.000
6	10	10	12	0.442
7	5	8	19	1.000
8	9	10	16	0.563
9	9	11	17	0.570
10	1	25	8	1.000
11	8	11	16	0.580
12	7	6	21	1.000
13	8	12	19	0.644
14	7	9	14	0.612
15	6	2	8	1.000
16	8	9	19	0.728
17	2	5	8	1.000
18	4	3	8	1.000
19	1	7	8	1.000
20	8	8	15	0.639
21	9	8	13	0.548
22	6	11	15	0.640
23	5	11	11	0.545
24	10	10	14	0.488
25	7	10	17	0.685
26	9	8	14	0.575
27	11	13	17	0.473
28	7	8	10	0.528
29	9	7	12	0.555
30	8	9	14	0.575

Using eight strongly efficient units and applying the proposed algorithm, the equations of both weak and strong hyperplanes are derived. In this case, 16 defining hyperplanes are obtained, and the coefficients of each shown in Table 2.

Table 2: Coefficients of defining hyperplanes obtained by the proposed algorithm

hyperplane No.	V_1	V_2	U	u_0
1	0	0.677734	0	-1.35547
2	0.693633	0	0	-0.69363
3	0.65168	0.641688	0.30395	-2.02895
4	1.808718	0.629447	0.716524	-0.38058
5	0.460859	0.256476	0.195411	-0.64081
6	0.906329	0.925516	0	-6.40186
7	0.603664	0.276575	0.241313	-0.60919
8	0.769659	1.672368	0.567599	-3.4219
9	0.774714	0.382786	0.361728	-0.06304
10	0.641618	0.355105	0.328283	0.308317
11	0.488245	0.977283	0	-4.88403
12	0.31464	0.151618	0	-1.37597
13	0.00004	0	0.849259	32.2439
14	0.411073	1.072416	0.359943	-1.72894
15	0	1.045754	0.244703	-0.11307
16	0.845976	0	0.225641	0.960589

Figure 3 shows the hyperplanes obtained by the algorithm. For example $-0.677\,V_2 - 1.355 = 0$ is one of the weak defining hyperplane of PPS or is $0.303\,U - 0.651\,V_1 - 0.641\,V_2 - 2.028 = 0$ a strong defining hyperplane.

The above algorithm identifies the efficient hyperplanes of T_v set. Concerning the advantages of the GA, this method can be implemented for discrete and continuous variables. Moreover, the technique can execute efficiently with a large number of variables.

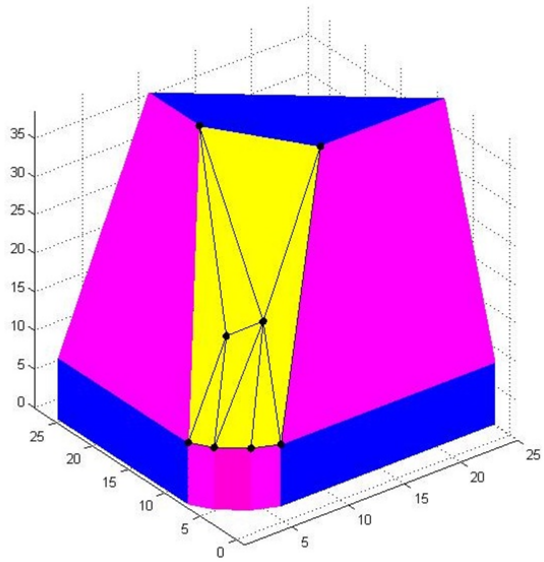

Figure 3: The hyperplanes obtained by the algorithm

5 Conclusion

The proposed method determines the defining frontier of the BCC model including weak and strong hyperplanes, without the necessity of solving linear programming models. Additionally, the optimization function is defined in such a way that it ensures the results are obtained with minimal computational error. As the proposed algorithm uses the GA, it has its specific advantages and it is capable of identifying all PPS hyperplanes for systems with a large number of DMUs, multiple inputs, and outputs, while maintaining low computational complexity. The equations of the defining hyperplanes in data envelopment analysis are crucial, as understanding the efficiency frontier allows for the easy evaluation of returns to scale, identification of reference points, and sensitivity analysis. Additionally, meta-heuristic algorithms, such as PSO, could be utilized to enhance the execution of the algorithm. Furthermore, alternative DEA models may be explored as potential replacements to improve the analysis.

Acknowledgment

"This article derived from PhD degree thesis in the Islamic Azad University, Shahr-e Rey branch."

References

[1] N. Aghayi and Z. G. Beigi, Identification of strong and weak hyperplanes in data envelopment analysis. *Procedia - Social and Behavioral Sciences*, 129 (2014), 365–371. https://doi.org/10.1016/j.sbspro.2014.03.686

[2] A. Amirteimoori and S. Kordrostami, Generating strong defining hyperplanes of the production possibility set in data envelopment analysis. *Applied Mathematics Letters*, 25(3) (2012), 605–609. https://doi.org/10.1016/j.aml.2011.09.007

[3] R. D. Banker, A. Charnes and W. W. Cooper, Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, 30(9) (1984), 1078–1092. https://doi.org/10.1287/mnsc.30.9.1078

[4] M. S. Bazaraa J. J. Jarvis and H. D. Sherali, *Linear Programming and Network Flows* (4th ed.). John Wiley & Sons (2011). https://www.wiley.com/en-us/Linear+Programming+and+Network+Flows%2C+4th+Edition-p-9780470462720

[5] A. Charnes, W. W. Cooper and E. Rhodes, Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6) (1978), 429–444. https://doi.org/10.1016/0377-2217(78)90138-8

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press (1975). https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/

[7] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison-Wesley (1989).

[8] K. A. De Jong, Analysis of the behavior of a class of genetic adaptive systems. *Doctoral dissertation*, University of Michigan (1975). https://hdl.handle.net/2027.42/4507

[9] Z. Dokur, Segmentation of MR and CT images using a hybrid neural network trained by genetic algorithms. *Neural Processing Letters*, 16(3) (2002), 211–225. https://doi.org/10.1023/A:1021769530941

[10] M. J. Farrell, The measurement of productive efficiency. *Journal of the Royal Statistical Society: Series A (General)*, 120(3) (1957), 253–281. https://doi.org/10.2307/2343100

[11] A. Ghazi, F. Hosseinzadeh Lotfi and M. Sanei, Finding the strong efficient frontier and strong defining hyperplanes of production possibility set using multiple objective linear programming. *Operational Research*, 22(1) (2022), 1–34. https://doi.org/10.1007/s12351-019-00542-9

[12] K. Hoffmann and R. A. Kunze, *Linear Algebra*. Prentice-Hall (1971). https://www.amazon.com/Linear-Algebra-2nd-Kenneth-Hoffman/dp/0135367972

[13] C. E. Henderson, W. D. Potter , R. W. McClendon and G. Hoogenboom, Predicting aflatoxin contamination in peanuts: a genetic algorithm/neural network approach. *Applied Intelligence*, 12(3) (2000), 183–192. https://doi.org/10.1023/A:1008310906900

[14] G. R. Jahanshahloo, F. H. Lotfi, H. Z. Rezai and F. R. Balf, Finding strong defining hyperplanes of production possibility set. *European Journal of Operational Research*, 177(1) (2007), 42–54. https://doi.org/10.1016/j.ejor.2006.04.038

[15] G. R. Jahanshahloo, F. H. Lotfi and M. Zohrehbandian, Finding the piecewise linear frontier production function in data envelopment analysis. *Applied Mathematics and Computation*, 163(1) (2005), 483–488. https://doi.org/10.1016/j.amc.2004.03.055

[16] C. Karul, S. Soyupak and C. Yurteri, Neural network models as a management tool in lakes. *Shallow Lakes' 98: Trophic Interactions in Shallow Freshwater and Brackish Waterbodies*, (1999), 139–144. https://link.springer.com/chapter/10.1007/978-94-017-2986-4_14

[17] F. H. Lotfi, G. R. Jahanshahloo, M. R. Mozaffari and J. Gerami, Finding DEA-efficient hyperplanes using MOLP efficient faces. *Journal of Computational and Applied Mathematics*, 235(5) (2011), 1227–1231. https://doi.org/10.1016/j.cam.2010.09.022

[18] K. A. Smith and J. N. Gupta, Neural networks in business: techniques and applications for the operations researcher. *Computers & Operations Research*, 27(11–12) (2000), 1023–1044. https://doi.org/10.1016/S0305-0548(99)00147-9

[19] R. C.Wu, Neural network models: foundations and applications to an audit decision problem. *Annals of Operations Research*, 75 (1997), 291–301. https://doi.org/10.1023/A:1018996301520

Sharif Malakouti
PhD Student of Mathematics
Department of Mathematics
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University
Tehran, Iran.
E-mail: Sharif.malakouti@iau.ir

Reza Kargar
Professor of Mathematics
Department of Mathematics
Science and Research Branch, Islamic Azad University
Tehran, Iran.
E-mail:Reza_kargar@srbiau.ir

Zohreh Taeb
Professor of Mathematics
Department of Mathematics
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University
Tehran, Iran.
E-mail: taeb_zt@iause.ac.ir

Hadi Bagerzadeh Valami
Professor of Mathematics
Department of Mathematics
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University
Tehran, Iran.
E-mail: hadi.bagherzade@iause.ac.ir

Leila Karamali
Professor of Mathematics
Department of Mathematics
Shahr-e Rey (Yadegar-e Emam) Branch, Islamic Azad University
Tehran, Iran.
E-mail: lm_karamali@iause.ac.ir