International Journal of Operations Research and Artificial Intelligence

www.ijorai.reapress.com

J. Civ. Asp. Struct. Eng. Vol. 1, No. 1 (2025) 1–10.

Paper Type: Original Article

A Multiobjective Linear Programming Approach to DEA Ranking with Common Weights

Mostafa Albouyeh*

Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran; alebouyeh@gmail.com.

Citation:

Received: 8 January 2025 *Revised: 13 November 2024* *Accepted: 10 September 2024*	Alboyeh, M. (2025). A Multiobjective linear programming approach to DEA ranking with common weights. *International journal of operations research and artificial intelligence,* 1 (1), 1-10.

Abstract

In this paper, a novel approach is proposed for determining a Common Set of Weights (CSW) through multi-objective programming within the framework of Data Envelopment Analysis (DEA). In DEA, each Decision Making Unit (DMU) is evaluated under the most favorable conditions by selecting weights that maximize its own efficiency. To ensure a fair and unified assessment across all DMUs, a model is developed to identify a CSW. The proposed model involves fractional objective functions, which are subsequently transformed into an equivalent Multi-Objective Linear Programming (MOLP) problem. To solve the MOLP, we employ either the Multi-criterion Simplex Method (MSM) or the Weighted Sum Method (WSM). Finally, the derived CSW is used to assess and rank the efficient DMUs in a consistent manner.

Keywords: Multiple objective programming, Data envelopment analysis, Efficiency, Ranking, Common set of weights.

1 | Introduction

Data Envelopment Analysis (DEA) is a non-parametric technique introduced by Charnes et al. [1] for evaluating the relative efficiency of Decision-Making Units (DMUs). This approach, based on linear programming, is designed to assess the performance of units that consume multiple inputs to produce multiple outputs. In DEA, efficiency is defined as the ratio of the weighted sum of outputs to the weighted sum of inputs, with the maximum efficiency score normalized to 1. A DMU is considered efficient if it lies on the efficiency frontier; otherwise, it is classified as inefficient [2–4].

DEA models evaluate each DMU individually under the most favorable conditions by assigning it the optimal set of weights. While this allows each DMU to achieve its highest possible efficiency score, it leads to difficulty

Corresponding Author: alebouyeh@gmail.com

https://doi.org/10.48314/ijorai.v1i1.51

in making fair comparisons among all units. Therefore, identifying a Common Set of Weights (CSW) is crucial, as it provides a unified benchmark to evaluate and rank all DMUs under consistent conditions [5].

To address this, researchers such as Jahanshahloo et al. [6], [7] have proposed methods for finding common weights in DEA. In the present study, we transform a Multi-Objective Fractional Programming (MOFP) problem into a Multi-Objective Linear Programming (MOLP) problem. The transformed model is then solved using two distinct approaches: The Multi-criterion Simplex Method (MSM) and the Weighted Sum Method (WSM). These techniques facilitate the determination of a CSW in an efficient and structured manner.

Evaluating the efficiency of similar operational units is critical for maintaining competitiveness and ensuring continuous improvement. Institutions such as bank branches, schools, hospitals, power plants, and factories can benefit from DEA to identify their current performance levels and develop strategies for improvement. Awareness of current efficiency serves as the first step toward enhanced productivity and more informed managerial decisions [8], [9].

The structure of the paper is as follows:

In Section 2, we introduce the concepts of MOLP and Fractional Multiobjective Linear Programming (FMOLP), and discuss the Multicriterion Simplex Method (MSM) in detail. Section 3 presents both the fractional and multiplier models of DEA. In Section 4, a new model is proposed for determining a CSW. To solve this model, we employ two approaches: The MSM and the WSM. Section 5 is devoted to ranking the efficient Decision Making Units (DMUs) based on the obtained CSW. A numerical example is provided in Section 6 to illustrate the methodology. In Section 7, a real-world application involving 20 bank branches is examined. Finally, Section 8 concludes the paper with a summary of the main results.

2 | Multi-Objective Linear Programming

We consider the following MOLP problem:

$$\begin{aligned} &(\text{MOLP}) \max\ \big(f_1(x), f_2(x), \dots, f_k(x)\big),\\ &\text{s.t.}\quad Ax = b,\\ &\quad x \geq 0, \end{aligned} \tag{1}$$

where $f_i(x) = c^i x$ for $i = 1, \dots, k$ are the objective functions, $A \in R^{m \times n}$ is the constraint matrix, $b \in R^m$ is the right hand side vector and $x \in R^n$ is a vector of variables. We shall denote the feasible set of the MOLP by X. In the following we assume, without loss of generality, that X is non empty. The objective function can be written as $C^T x$, where $C \in R^{n \times k}$ has columns c^i. A solution $x^* \in X$ of MOLP is (Weakly) Pareto optimal if there is no $x \in X$ such that $C^T x \geq C^T x^*$ and $C^T x \neq C^T x^*$. If x^* is (Weakly) Pareto optimal, $C^T x^*$. is called (Weakly) efficient.

A fundamental result of multiobjective linear programming states that (Weakly) Pareto optimal solutions of MOLP can be characterized as optimal solutions of single objective linear programs with a convex combination of objectives c^i $i = 1, \dots, k$. Let $\Lambda = \{\lambda \in R^k\colon \lambda \geq 0,\ \sum_{i=1}^{k} \lambda_j = 1\}$ and $\Lambda^0 = \{\lambda \in R^k\colon \lambda > 0,\ \sum_{i=1}^{k} \lambda_j = 1\}$. We will refer to Λ respectively Λ^0 as parameter space or weight space.

Theorem 1. A feasible point $x \in X$ is (Weakly) Pareto optimal for MOLP if and only if there exists $\lambda \in \Lambda^0$ $(\lambda \in \Lambda)$ such that x is an optimal solution of

$$\begin{aligned} &(P(\lambda))\ \max\ c(\lambda)^T x, = \sum_{i=1}^{k} \lambda_i (c^i)^T (x),\\ &\text{s.t,}\quad Ax = b,\\ &x \geq 0. \end{aligned} \tag{2}$$

A proof of this result can be found in [10].

There exist other method for solving the MOLP, one of which is discussed in below.

2.1 | Multicriterion Simplex Method

The MSM is an extension of the classical simplex method used to solve Multiple Objective Linear Programming (MOLP) problems. Instead of optimizing a single objective function, MSM handles k objective functions simultaneously. The method constructs a simplex tableau that incorporates not only the constraint equations but also additional rows corresponding to each objective. These rows include the reduced costs for each non-basic variable with respect to each criterion. The aim is to identify solutions that are efficient (Pareto optimal), meaning no other feasible solution improves one objective without worsening at least one other.

In the MSM tableau, the top section contains the basic and non-basic variables, and the body includes the constraint coefficients and the current values of the basic variables. Below the constraint rows, there are k criterion rows, each representing one of the objectives. These rows show how changing a non-basic variable would affect each objective function. By analyzing the reduced costs in each criterion row, the decision-maker can determine potential entering variables and navigate toward a set of trade-off (Efficient) solutions.

The general form of MSM table to solve model MOLP in [1] is introduced as follow:

Table 1. MSM tableau structure for solving multicriteria optimization problems.

Basic Vars	x_1	…	x_m	x_{m+1}	…	x_j	…	x_n	RHS (x_i^0)
x_1	1	…	0	y_{1m+1}	…	y_{1j}	…	y_{1n}	x_1^0
…									
x_m	0	…	1	y_{mm+1}	…	y_{mj}	…	y_{mn}	x_m^0
z_1 (Obj 1)	0	…	0	z_{1m+1}	…	z_{1j}	…	z_{1n}	$c^1 \cdot x^0$
…									
z_k (Obj k)	0	…	0	z_k$_{m+1}$	…	z_k$_j$	…	z_k$_n$	$c^k \cdot x^0$

where:

I. y_{ij}: Coefficients in constraint rows.

II. z_{kj}: Reduced costs for objective k.

III. $c^k \cdot x^0$: Current value of the k-th objective function.

All variable, including the slacks, are divided into two groups: m basic variables $x_1 \dots x_m$ that are currently forming a solution, and (n-m) nonbasic variables $x_{m+1} \dots x_n$ whose value are, by definition, equal to zero. Since there are constraints and n variables, only m variables can be positive, while the remaining (n-m) variables are zero. We do not count the nonnegative condition as constraints, because they are automatically satisfied through the appropriate simplex method manipulations. A solution consisting of m basic and (n-m) nonbasic, zero-valued variables is referred to as a basic solution. The set basic variables $x_1 \dots x_m$ is often referred to as a basis. Basic solutions will generally be identified by superscripts, that is, x^1, x^2, and so on, usually indicting the order in which the were generated. Observe that each nonbasic variable in *Table 1* has the following associated column vector in the criteria row portion of the tableau: $z_j = (z_{1j}, \dots, z_{kj})^t$.

Also, with each basic solution (And its tableau) say, x^o there is associated a corresponding column vector of current values of the objective functions: $Cx^o = (c^1x^o, \dots, c^kx^o)^t$. Note that this vector Cx^o is also maintained and updated automatically by the performance of correct row operations on the individual tableaus. For any nonbasic variable x_j, such that $y_{rj} > 0$ for at least one $r = 1, \dots, m$, we define $\alpha_j = \text{Min}_r \frac{x_r^o}{y_{rj}}$. Thus α_j is the minimum of the ratios formed for the jth column; the corresponding minimum row r then determines which of the basic variables x_r is going to leave the basis. Then,

I. If there is a z_j consisting of only nonpositive components, not all of which are zero, then x^o must be dominated.

II. If there is a z_j consisting of only nonnegative components, not all of which are zero, then introducing x_j in to the basis would result in a dominated solution.

III. Consider two nonbasic variable x_j, x_k. If all the components of $\theta_j z_j$ are smaller or equal to the components of $\alpha_k z_k$ with at least one $\alpha_j z_{ij} < \alpha_k z_{ik}$, $r = 1, \dots, k$, then the solution resulting from introducing x_k would be dominated by the solution which would result from introducing x_j.

2.2 | Fractional Objective Functions

In many problem of a practical nature, especially in financial planning, one is often concerned about objective function which are defined as ratios, quotas, or fractions. This bring an aspect of nonlinearity into the analysis. For example, a ratio of two functions say, $h(x)$ profits to $g(x)$ sales, $\frac{h(x)}{g(x)}$ is a nonlinear function even though both $h(x)$ and $g(x)$ could be linear. Hosseinzadeh Lotfi [6] developed such a test through linearization of fractional functions by taking their derivatives.

If we could replace fractional function $f_i(x)$ sby some linear function $t_i(x)$ so that the nondominance-dominance relationships among the points of X would be preserved, then we could use MSM for testing the nondominance in fractional linear programming as well. Suppose that in *Model (2)*, $f_i(x) = \frac{h_i(x)}{g_i(x)}$ where $h(x)$ and $g(x)$ are linear functions. Therefore, we have a multiobjective fractional programming problem. The derivative of $f_i(x)$ around point $\bar{x}$ is as follows:

$$t_i(x) = \frac{\nabla h_i(\bar{x}) g_i(\bar{x}) - \nabla g_i(\bar{x}) h_i(\bar{x})}{[g_i(\bar{x})]^2}(x - \bar{x}).$$

herefore we can convert models which have fractional function to linear function by derivation around a point.

3 | Data Envelopment Analysis

Consider n DMUs with m inputs and s outputs. The input and output vectors of DMU_j $(j = 1, \dots, n)$ $X_j = (x_{1j}, \dots, x_{mj})^t$, $Y_j = (y_{1j}, \dots, y_{sj})^t$ where $X_j \geq 0$, $X_j \neq 0$, and $Y_j \geq 0$, $Y_j \neq 0$.

Suppose input and output weights are $v_i (i = 1, \dots, m)$, $u_r (r = 1, \dots, s)$, respectively. We also assume that the input and output components are nonegative. The efficiency of DMU_p is obtained by solving the following ratio programming problem:

$$\begin{aligned} \max \quad & \frac{\sum_{r=1}^{s} u_r y_{rp}}{\sum_{i=1}^{m} v_i x_{ip}}, \\ \text{s.t.} \quad & \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1, \quad j = 1, \cdots, n, \\ & u_r \geq \varepsilon, \quad v_i \geq \varepsilon, \quad r = 1, \dots, s, \quad i = 1, \dots, m. \end{aligned} \qquad (3)$$

where ϵ is positive non-Archimedean infinitesimal. The first set of constraints are for limiting the evaluating scale. The above mentioned problem is solved separatly for each DMU, and by solving this problem different set of weight are obtained. Instead of solving *Problem (5)* the following linear programming problem is solved:

$$
\begin{aligned}
&\max \quad \sum_{r=1}^{s} u_r y_{rp} 0.15\mathrm{cm}, \\
&\quad \text{s.t.} \quad \sum_{i=1}^{m} v_i x_{ip} = 1, \\
&\quad \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0, \quad j = 1, \cdots, n, \\
&\quad u_r \geq \varepsilon, \quad v_i \geq \varepsilon, \quad r = 1, \ldots, s, \quad i = 1, \ldots, m.
\end{aligned} \tag{4}
$$

This model is called CCR input orientation multiplier side problem.

4 | Common Set of Weights

In DEA calculating the efficiency of different DMUs, different set of weights are obtained, which seems to be unacceptable in reality. So the following model is used to find CSW which has same advantages that will be discussed later on. Consider the following problem:

$$
\begin{aligned}
&\max \quad \left\{ \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}}, \ j = 1, \ldots, n \right\}, \\
&\quad \text{s.t.} \quad \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1, \quad j = 1, \cdots, n, \\
&\quad u_r \geq \varepsilon, \quad v_i \geq \varepsilon, \quad r = 1, \ldots, s, \ \ i = 1, \ldots, m.
\end{aligned} \tag{5}
$$

The above problem is a fractional programming problem with multiple objectives, in which each objective function has goal equal to 1. As no priority is considered for the objective functions.

By manipulating *Model (5)* and adding the normalizing constraint $\sum_{r=1}^{s} u_r + \sum_{i=1}^{m} v_i = 1$, we have:

$$
\begin{aligned}
&\max \quad \left\{ \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}}, \quad j = 1, \ldots, n \right\}, \\
&\quad \text{s.t.} \quad \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0, \quad j = 1, \cdots, n, \\
&\quad \sum_{r=1}^{s} u_r + \sum_{i=1}^{m} v_i = 1, \\
&\quad u_r \geq \varepsilon, \quad v_i \geq \varepsilon, \quad r = 1, \ldots, s, \quad i = 1, \ldots, m.
\end{aligned} \tag{6}
$$

By using the fractional function approximation in $(u_1, \ldots, u_s, v_1, \ldots, v_m) = (\bar{u}_1, \ldots, \bar{u}_s, \bar{v}_1, \ldots, \bar{v}_m)$, *Model (6)* is converted into an MOLP as follows:

$$
\begin{aligned}
&\max \quad \{ \left(\sum_{r=1}^{s} u_r y_{rj} \right) \left(\sum_{i=1}^{m} \bar{v}_i x_{ij} \right) - \left(\sum_{i=1}^{m} v_i x_{ij} \right) \left(\sum_{r=1}^{s} \bar{u}_r y_{rj} \right), \ \ j = 1, \ldots, n \}, \\
&\quad \text{s.t.} \quad \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0, \quad j = 1, \cdots, n, \\
&\quad \sum_{r=1}^{s} u_r + \sum_{i=1}^{m} v_i = 1, \\
&\quad u_r \geq \varepsilon, \quad v_i \geq \varepsilon, \quad r = 1, \ldots, s, \quad i = 1, \ldots, m.
\end{aligned} \tag{7}
$$

5 | Complete Ranking of Decision-Making Units Using Common Set of Weights

In this section, we introduce a method for ranking of DMU using CSW. consider the *Problem (18)*, suppose that $E = \{j\text{: } DMU_j$ is efficient in the *Model (18)*$\}$.

To rank efficient DMUs, we omit the corresponding constraints of all efficient DMUs, and then evaluate efficient DMUs by following model:

$$\begin{aligned} &\max\ \{\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}},\quad j \in E\}0.15\text{cm}, \\ &\text{s.t.}\quad \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \le 1,\quad j = 1, \cdots, n, j \ne p, \\ &u_r \ge \varepsilon,\quad v_i \ge \varepsilon,\quad r = 1, \dots, s,\quad i = 1, \dots, m. \end{aligned} \tag{8}$$

By using the fractional function approximation in $(u_1, \dots, u_s, v_1, \dots, v_m) = (\bar{u}_1, \dots, \bar{u}_s, \bar{v}_1, \dots, \bar{v}_m)$, *Model (9)* is converted into an MOLP as follows:

$$\begin{aligned} &\max\quad \{\sum_{j\in A} [(\sum_{r=1}^{s} u_r y_{rj})(\sum_{i=1}^{m} \bar{v}_i x_{ij}) - (\sum_{i=1}^{m} v_i x_{ij})(\sum_{r=1}^{s} \bar{u}_r y_{rj})]\}0.15\text{cm}, \\ &\text{s.t.}\quad \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \le 0,\quad j = 1, \cdots, n, j \ne p, \\ &\sum_{r=1}^{s} u_r + \sum_{i=1}^{m} v_i = 1, \\ &u_r \ge \varepsilon,\quad v_i \ge \varepsilon,\quad r = 1, \dots, s,\quad i = 1, \dots, m. \end{aligned} \tag{9}$$

Therefore we can use objective function values for ranking efficient units.

6 | Numerical Example

We consider four DMUs with one input and one output in *Table 2* as follows:

Table 2. The data of the four DMUs.

	DMU_1	DMU_2	DMU_3	DMU_4
Inputs	1	2	4	3
Outputs	1	5	6	2

In *Table 2*, we examine a simplified DEA scenario involving four DMUs, each characterized by a single input and a single output. This structure enables a straightforward assessment of efficiency, where the performance of each DMU can be visualized and compared directly using input-output ratios. The input and output data are as follows: DMU_1 consumes 1 unit of input to produce 1 unit of output, DMU_2 uses 2 inputs to produce 5 outputs, DMU_3 utilizes 4 inputs for 6 outputs, and DMU_4 employs 3 inputs to yield 2 outputs.

To evaluate efficiency in this context, we calculate the output-to-input ratio (i.e., output divided by input) for each DMU. These ratios are: $DMU_1 = 1$, $DMU_2 = 2.5$, $DMU_3 = 1.5$, and $DMU_4 \approx 0.67$. Based on these results, DMU_2 demonstrates the highest efficiency, as it produces the greatest amount of output per unit of input. DMU_1 also shows relatively good performance with a ratio of 1.0, while DMU_4 is the least efficient among the four units. These findings suggest that DMU_2 lies on the efficiency frontier, potentially serving as a benchmark for other units.

From a DEA perspective, DMUs with higher efficiency scores are considered more productive, as they utilize fewer resources to generate more outputs. The presence of DMUs with varying efficiency levels highlights the need for comparative evaluation methods such as DEA. This example also illustrates the importance of identifying efficient frontiers and setting realistic performance targets. In further analysis, DEA models can be employed to assign appropriate weights to inputs and outputs, assess relative efficiency under multiple conditions, and guide managerial improvements across less efficient DMUs.

The proposed *Model (6)* for the data in *Table 2* is summarized as follows:

$$\begin{aligned} &\max\ \{\frac{u}{v},\ \frac{5u}{2v},\ \frac{6u}{4v},\ \frac{2u}{3v}\},\\ &\text{s.t.}\quad u-v\le 0,\quad 5u-2v\le 0,\\ &\quad 6u-4v\le 0,\quad 2u-3v\le 0,\\ &\quad u+v=1,\\ &\quad u\ge 0, v\ge 0. \end{aligned} \tag{10}$$

Model (8) corresponding to the data in *Table 2* is as follows:

$$\begin{aligned} &\max\{\frac{5}{7}u-\frac{2}{7}v,\frac{50}{7}u-\frac{20}{7}v,\frac{120}{7}u-\frac{48}{7}v,\frac{30}{7}u-\frac{12}{7}v\},\\ &\text{s.t.}\quad u-v\le 0,\quad 5u-2v\le 0,\\ &\quad 6u-4v\le 0,\quad 2u-3v\le 0,\\ &\quad u+v=1,\\ &\quad u\ge 0, v\ge 0. \end{aligned} \tag{11}$$

The optimal tableau of the MSM is as follows:

Table 3. The optimal table corresponding to the MSM.

	u	v	s_1	s_2	s_3	s_4	RHS
s_1	0	0	1	$-\frac{2}{7}$	0	0	$\frac{3}{7}$
u	1	0	0	$\frac{1}{7}$	0	0	$\frac{2}{7}$
s_3	0	0	0	$-\frac{10}{7}$	1	4	$\frac{8}{7}$
s_4	0	0	0	$-\frac{5}{7}$	0	1	$\frac{11}{7}$
v	0	1	0	$-\frac{1}{7}$	0	0	$\frac{5}{7}$
z^1-c^1	0	0	0	$\frac{1}{7}$	0	0	0
z^2-c^2	0	0	0	$\frac{10}{7}$	0	0	0
z^3-c^3	0	0	0	$\frac{24}{7}$	0	0	0
z^4-c^4	0	0	0	$\frac{6}{7}$	0	0	0

Pareto optimal solution is of MSM which $(u,v)=(\frac{2}{7},\frac{5}{7})$ are CSW for evaluating DMUs.

7 | An Application and Discussion

Evaluating the performance of bank branches is a complex task due to the multifaceted nature of banking services and the diversity of resources consumed and outputs delivered. DEA is a powerful tool for addressing this challenge, as it can accommodate multiple inputs and outputs simultaneously without requiring prior assumptions about their relationships. *Table 4* presents the input and output structure used to assess the efficiency of 20 bank branches, providing a rich dataset for comparative analysis. The selected inputs and outputs reflect both operational effort and service outcomes, enabling a balanced efficiency evaluation.

We consider the data of 20 bank branches with three inputs and eight outputs, in the following table:

Table 4. Inputs and outputs of bank branches.

Input	Output
1. Interest paid	1. Interest-free saving account
2. Personnel	2. Current account
3. Demand	3. Short-term
	4. long-term
	5. Other
	6. Loans
	7. Interest received
	8. Banking fees

The input and output data of this study are given in *Tables 5* and *6*.

Table 5. Inputs of bank branches.

DMUs	I_1	I_2	I_3
1	12963.87	37.75	134529
2	1483.69	22.74	10244
3	11756.02	25.5	42668
4	866.86	20.94	44128
5	4545.92	14.43	13043
6	9139.03	18.86	87981
7	308.67	25.66	97763
8	3185.7	26.27	7629
9	832.02	21.91	430513
10	11589.09	16.75	7859
11	2886.74	21.2	1192
12	3880.46	23.67	53209
13	6269.71	21.85	27506
14	2616.9	21.56	17988
15	2600.23	37.5	50229
16	4257	24.5	32618
17	6179.39	25.72	41817
18	528.92	16.28	22262
19	3679.91	28.76	16795
20	916.42	30.4	30326

Table 6. Outputs of bank branches.

DMUs	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
1	709874	2610439	172126	342833	228993	2761590	59161.51	9589.48
2	73560	317617	2326	26234	85324	252120	3504.68	180.38
3	435248	489666	6506	79194	824241	2043899	5817.55	760.9
4	29165	971640	1559	46160	354722	452931	622.48	244.75
5	216148	158130	845	90278	12256	552673	49570.82	20.7
6	514881	508735	103257	69604	396716	1751591	34115.58	1234.52
7	15676	172947	822	7373	287183	537567	1119.32	143.02
8	155799	269617	5451	83400	71509	1534505	10163.75	1868.82
9	13707	163397	5759	86775	135464	4829312	13520.68	1024.48
10	666679	242971	2319	5148	16992	2949072	21773.84	22.1
11	106682	408506	2439	173299	20643	1808353	2434.86	1193.46
12	190455	174657	9402	110027	69802	510656	2594.17	531.59

Table 6. Continued.

DMUs	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8
13	337298	293580	12340	148121	24997	768622	1946.46	94.83
14	133062	131262	15723	68920	150625	422274	422.05	461.59
15	152552	189936	11778	41074	420668	729915	2269.94	376.75
16	245412	106089	3651	12176	118734	448984	1200.53	102.69
17	392104	646367	6241	30690	375109	2144785	10428.37	295.05
18	27880	196292	2034	6021	6122	28007	365.28	144.75
19	204028	502730	3511	38927	1100502	1977217	8529.12	731.69
20	52156	71458	12850	14869	15344	277254	2761.24	162.163

We consider *Model (8)* for the data of *Tables 5* and *6*, which $(\bar{U}, \bar{V})$ is the optimal solution obtained form evaluating DMU_1 by *Model (4)*. Therefore, We put the optimal value of *Models (8)* and *(10)* in the *Table 7*.

Table 7. The efficiency of CSW model and ranking.

Efficient DMUs	Ranking Value	Rank	Common Set of Weights
DMU_1	1.16	4	$u_1 = 0.000100, u_2 = 0.000100, u_3 = 0.058890$
DMU_5	1.125	5	$u_4 = 0.000100, u_5 = 0.001235, u_6 = 0.001685$
DMU_9	2.257	2	$u_7 = 0.000100, u_8 = 0.000100$
DMU_{11}	1.306	3	$v_1 = 0.924065, v_2 = 0.000100, v_3 = 0.013525$
DMU_{19}	3.378	1	

The table presents five efficient DMUs evaluated using a CSW and ranked based on their efficiency scores. Among them, DMU_{19} stands out with the highest efficiency value of 3.378, securing rank 1, while DMU_5, with a score of 1.125, holds rank 5, the lowest among the efficient group. This variation reveals that even within the set of efficient DMUs, there are significant differences in how effectively each unit utilizes its resources to generate outputs.

The CSW values provide insight into which outputs or inputs are most influential in determining the efficiency of each DMU under a unified evaluation framework. For example, DMU_1 places negligible weight on u_1 and u_2 (0.000100), while u_3 receives a substantially higher weight (0.058890), indicating its dominant contribution to efficiency. In contrast, DMU_9 focuses entirely on outputs u_7 and u_8, reflecting a narrow but impactful performance profile. These patterns emphasize how different DMUs may rely on distinct strengths or services to achieve efficiency.

From a managerial perspective, these findings are highly informative. For instance, DMU_{11} ranks third, with a very high weight on v_1 (0.924065), suggesting that the first input is crucial to its operational success. Identifying such key factors helps managers understand what drives performance and where to direct improvement efforts. Furthermore, the application of CSW ensures consistency and fairness in comparison across units, enabling better benchmarking, resource allocation, and strategic planning within the organization.

8 | Conclusion

Finding a gradient hyperplane that forms the efficient frontier TvT_vTv is of significant importance in DEA. By solving the multiplier model, one can obtain such hyperplanes onto which DMUs are projected. However, identifying a common hyperplane with a shared gradient for all DMUs provides a more consistent and insightful evaluation framework. In conventional DEA, evaluating nnn DMUs requires solving nnn separate linear programming problems. In contrast, this paper proposes a unified approach in which all DMUs are evaluated simultaneously by solving a MOLP model. This can be done using either the MSM or the weighted sum approach. Evaluating all units under the same conditions using the proposed MOLP enhances comparability, and ranking efficient units based on the CSW yields more realistic and fair assessments.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European journal of operational research*, *2*(6), 429–444. https://doi.org/10.1016/0377-2217(78)90138-8

[2] Qi, X. G., & Guo, B. (2014). Determining common weights in data envelopment analysis with shannon's entropy. *Entropy*. *16*(12), 6394-6414. https://doi.org/10.3390/e16126394

[3] Roy, D., Cho, S., & Avdan, G. (2023). *Ergonomic risk and performance assessment using data envelopment analysis (DEA)*. https://doi.org/10.21872/2023IISE_1715

[4] Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA)–thirty years on. *European journal of operational research*, *192*(1), 1–17. https://doi.org/10.1016/j.ejor.2008.01.032

[5] G. H. Tzeng, H. F. Wang, U. P. Wen, P. L. Y. (1994). Multiple criteria decision making. *Proceedings of the 10th international conference: Expand and enrich the domains of thinking and application*. Springer. https://doi.org/10.1007/978-1-4612-2666-6

[6] Jahanshahloo, G. R., Memariani, A., Lotfi, F. H., & Rezai, H. Z. (2005). A note on some of DEA models and finding efficiency and complete ranking using common set of weights. *Applied mathematics and computation*, *166*(2), 265–281. https://doi.org/10.1016/j.amc.2004.04.088

[7] Hosseinzadeh Lotfi, F., Jahanshahloo, G. R., & Memariani, A. (2000). A method for finding common set of weights by multiple objective programming in data envelopment analysis. *Southwest journal of pure and applied mathematics*, *2000*(1), 44–54. https://eudml.org/doc/222875

[8] Cooper, W., Seiford, L., & Tone, K. (2007). *Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software*. http://dx.doi.org/10.1007/978-0-387-45283-8

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

[10] Steuer, R. E. (1986). Multiple criteria optimization. *Theory, computation, and application*. https://cir.nii.ac.jp/crid/1573668923835678848