ELSEVIER European Journal of Operational Research 126 (2000) 51–68

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

Using DEA to obtain efficient solutions for multi-objective 0–1 linear programs

Fuh-Hwa Franklin Liu [a,*], Chueng-Chiu Huang [b], Yu-Lee Yen [a]

[a] *Department of Industrial Engineering and Management, National Chiao Tung University, Box 17, 1001 Ta Shueh Road, Hsin Chu, Taiwan 300, ROC*

[b] *Department of Industrial Management, National Yunlin University of Science and Technology, Touliu, Taiwan 640, ROC*

Received 16 November 1998; accepted 23 March 1999

Abstract

This paper concerns the problem of a service-oriented public sector entity to allocate limited resources to different activities while keeping conflicting objectives in mind. The Multi-objective Resource Allocation Problem (MRAP) is to select activities to be performed. The authors formulate the problem as a multi-objective 0–1 linear problem. The authors implement Data Envelopment Analysis (DEA) with the Banker, Charnes and Cooper's (BCC) model to measure the Decision Making Unit's (DMU) efficiency. In this study, the production function is a mathematical statement relating the technological relationship between the objectives and resources of MRAP. Each DMU presents a technological relationship, i.e. DMU presents a relationship between resources and objectives. This relationship gives information about the use of resources and satisfactoriness of objectives. The inputs and outputs, respectively, outline resources and objectives. The production possibility set represents feasible solutions for MRAP. Moreover, due to the multiple objectives of problems, the method derives a solution set instead of an optimal solution in single objective ones. This solution set, a well-known *efficient solutions set*, forms the decision set of problems. Each DMU results from an *alternative*, a combination of *activities*. The production possibility set presents all the candidates of DMU. The set of alternatives resulting in efficient DMUs is efficient solutions of MRAP. The authors developed a two-stage algorithm to generate and evaluate DMUs. The first stage generates a DMU with the maximum of the *distance function*. The second stage is then used to evaluate the *efficiency* of the generated DMU.

Keywords: Data envelopment analysis; Multi-objective mathematical programming; 0–1 linear programming

1. Introduction and literature review

This paper concerns the problem of a service-oriented public sector entity to allocate m resources to Q activities while keeping s conflicting objectives in mind. The allocating process selects the activities to be

* Corresponding author. Tel.: +886-3-573-1739; fax: +886-3-572-9100.
E-mail address: fliu@cc.nctu.edu.tw (F.-H.F. Liu).

PII: S 0 3 7 7 - 2 2 1 7 (9 9) 0 0 2 7 9 - 9

performed. The authors formulate the Multi-objective Resource Allocation Problem (MRAP) as a multi-objective 0–1 linear problem. In this paper *alternative* stands for a combination of activities. If there are Q activities, the total possible number of alternatives would be 2^Q.

Matrix $\boldsymbol{A}$, with element a_{iq} denotes the usage of the ith resource to perform the qth activity. Matrix $\boldsymbol{C}$, with element c_{rq} denotes the profit of the rth objective to perform the qth activity. Vector $\boldsymbol{b}$, with element b_i denotes resource limitation of the ith resource. $\boldsymbol{W} = [w_1, w_2, \ldots, w_Q]^{\mathrm{t}}$ denotes the decision variable vector. If the qth activity is performed, set $w_q = 1$, otherwise, $w_q = 0$. The set of feasible solutions is $\Omega = \{\boldsymbol{W} : \boldsymbol{AW} \leqslant \boldsymbol{b},\ \boldsymbol{W} \in \boldsymbol{B}^Q\}$ where $\boldsymbol{B}$ is binary. This MRAP can be formulated as follows:

$$(\mathbf{P}_0): \quad \text{maximize}\{\boldsymbol{CW} : \boldsymbol{W} \in \Omega\}. \tag{1.1}$$

Each solution of $(\mathbf{P}_0)$ is a combination of activities. The product vectors $\boldsymbol{AW}$ and $\boldsymbol{CW}$ represent the usage of resources and the satisfaction of objectives, respectively. $\overline{\boldsymbol{W}} \in \Omega$ is called *an efficient solution of* $(\mathbf{P}_0)$ if there is no $\boldsymbol{W} \in \Omega$ such that $\boldsymbol{CW} \geqslant \boldsymbol{C}\overline{\boldsymbol{W}}$ and $\boldsymbol{CW} \neq \boldsymbol{C}\overline{\boldsymbol{W}}$. The set of solutions is called an *efficient set*.

The *Decision-Making Unit* (DMU) is the decision unit during the decision process. Charnes et al. [5] considered DMUs in the form of not-for-profit entities rather than the more customary "firms" or "industries". The DMU results from an alternative. That is, given an alternative, one can obtain both the usage of resources and satisfaction of objectives. Based on the method of Data Envelopment Analysis (DEA), the authors developed a two-stage algorithm to generate and evaluate DMUs. The first stage generates a DMU with the maximum of the *distance function*. The second stage is then used to evaluate the *efficiency* of the generated DMU.

Techniques for solving multi-objective mathematical programs (MMP) can be classified in terms of time for eliciting decision maker's (DM) preferences [7]. The eliciting time is classified as *prior to*, *during* and *after* optimization. As the DM articulates preference *prior to* optimization, solution techniques generally derive optimal solutions in terms of utility function. By introducing a utility function, the MMP can be formulated as a single objective mathematical programming problem. However, determining the explicit form of the utility function may require too much time and effort. *During* optimization, DM interacts with the computer approach. These algorithms require the DM to provide a set of weights for the competing objectives. Hence, the DM's weighting would subjectively influence the optimal solution. These algorithms are not an objective method. After optimization, the DM only determines an alternative from the efficient set. This method alleviates drawbacks from algorithms used *during* optimization. The methods of generating efficient solutions are mathematically well defined and completely objective. These methods have been criticized for both their long computation times in generating the entire efficient set and their cognitive burden on the DM in selecting a solution out of infinite number of alternatives.

Numerous algorithms have been designed to solve multiple objective linear programs (MOLP). However, integer multiple objective linear programs have not received the algorithmic attention that continuous problems have. The literature available on this topic is limited. Moreover, since algorithmic development has been limited, the reported computational experience is almost non-existent. Herein we briefly review some literatures.

Pasternak and Passy [12] conducted an earlier study on designing solving methods for integer MOLPs. They used the concept of implicit enumeration to resolve zero–one bi-criterion linear programs. Three examples were presented and solved; however, extensive computational experience was not discussed. Bitran [4] used relaxation techniques to generate efficient solutions. He defined a relaxation problem and proved the efficient solutions of the relaxation problem that are feasible to original problem would also be efficient in the original problem. Bitran [4] also reported some computational results. More recently, Deckro and Winkfsky [6] reported computational results in terms of implicit enumeration compared to Bitran's works. They claimed that their studies compared favorably with Bitran's results.

Resource allocation problems have received algorithmic attention in [9,13–15,17]. Gilbert et al. [9] formulated the land allocation problem as a multi-objective integer programming model and applied interactive multi-objective optimization to generate efficient solutions. Ramanathan and Ganesh [12] used an Analytic Hierarchy Process (AHP) to solve resource allocation problems. They concentrated on two approaches, *expected priority* and *benefit–cost* ratio. However, their works were limited to the criteria (in AHP model) that are sought to be maximized. Schniederjans and Santhanam [14] applied zero–one goal programming to select information system projects. The works of Teng and Tzeng [18] were concerned with selecting transportation investment alternatives. In their study, relationships among alternatives do not necessary independent. They designed a SENTRA method to attain the near-optimal solution. Sinuany-Stern [17] suggested a multiple objectives network optimization model for multi-layer budget allocation.

2. Related theories and principal results

2.1. Production efficiency

Prior to introducing the measure of *productive efficiency*, we first describe the production function. The economic theory of the production function is a mathematical statement relating quantitatively to the purely technological relationship between the output of the process and inputs of the factors of production [14]. The distinct kinds of goods and services usable in production technology are referred to as the factors of production of that technology and, for any set of these factors, the production function defines the maximal output realizable therefrom. The *production possibility set* (*curve*) presents all feasible technological relationships (production technology) of the production function. This presentation can be studied in terms of benefit–cost analysis, utility function or statistical methods.

One purpose of the production function is to study the efficiency of production technology. Below is a brief review of the methods used to measure productive efficiency with single and multiple outputs.

2.1.1. Single output

Farrell [8] was the first to introduce a non-parametric approach to measure productive efficiency [15]. Instead of estimating conventional production functions, he started from the observed input–output coefficients of a set of 'firms', as a standard for measuring the efficiency of the firms. Then, he fitted a frontier function to the points as a piecewise linear function. This frontier function is called the *efficient production function*; it is used as a reference for comparing the efficiency of various firms relative to the frontier surface. However, his studies were limited to single input and output factors of production.

2.1.2. Multiple outputs

Charnes, Cooper and Rhodes [5] introduced a ratio definition of efficiency, formulated as a fractional form (CCR ratio definition). This form generalized the single-output to single-input classical engineering-science ratio definition to multiple outputs and inputs without requiring pre-assigned weights. Measuring the efficiency of any DMU is done by maximizing a ratio of weighted outputs to weighted inputs as long as the similar ratios of every DMU is less than or equal to unity. The CCR ratio definition can be formulated as a problem of fractional programming denoted as ($\mathbf{P}_1$). In ($\mathbf{P}_1$), the objective function is a ratio form presenting the DMU to be evaluated and constraints are similar ratios for every DMU. Then the weights of inputs and outputs can be determined by solving this problem. Charnes et al. [5] proved that the fractional programming problem can be transformed into a linear programming problem. However, their studies were limited to constant *returns to scale*. The meaning of returns to scale will be described in the next section. Banker, Charnes and Cooper (BCC) [1] generalized Charnes et al.'s work. They used a decision variable to qualify return to scale and transformed a fractional programming problem into a linear programming problem, denoted as $(\mathbf{P}_2)$, in terms of Shephard's [16] distance function. They also obtained the same efficient conditions as Charnes et al.'s model [5].

$$(\mathbf{P}_1) \qquad \max \quad \frac{\sum_{r=1}^{s} u_r y_{r0}}{\sum_{i=1}^{m} v_i x_{i0}}, \tag{2.1}$$

$$\text{s.t.} \quad \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leqslant 1, \quad j = 1, \ldots, n, \;\; u_r, v_i \geqslant 0, \;\; r = 1, \ldots, s, \;\; i = 1, \ldots, m. \tag{2.2}$$

y_{rj}, x_{ij} ($\geqslant 0$) are the known outputs and inputs of DMU-j and the u_r and v_i are the variable weights to be determined by the solution of problem $(\mathbf{P}_1)$. The efficiency of one member of this reference set $j = 1, \ldots, n$ DMUs is to be rated relative to the others.

The DMU weight is distinguished by assigning it the subscript 0 in the function for optimization (but preserving its original subscript in the constraints). The indicated maximization then accords this DMU the most favorable weighting that the constraints allow.

2.2. Measuring DMU efficiency

The authors implement DEA with the BCC model to measure the DMU's efficiency. In this study, the production function is a mathematical statement relating the technological relationship between the objectives and resources of MRAP. Each DMU presents a technological relationship, i.e. DMU presents a relationship between resources and objectives. This relationship gives information about the use of resources and satisfactoriness of objectives. The inputs and outputs respectively outline resources and objectives. The production possibility set represents feasible solutions for MRAP. Moreover, due to the multiple objectives of problems, the method derives a solution set instead of an optimal solution in single objective ones. This solution set, a well-known efficient solutions set, forms the decision set of problems. Each DMU results from an *alternative*, a combination of *activities*. The production possibility set presents all candidates of DMU. The set of alternatives resulting in efficient DMUs is efficient solutions of $(\mathbf{P}_0)$ (see Property 1 in the following section).

Below is an approach to finding efficient solutions to an MRAP. Note that the DMUs forming the efficient production possibility frontier are the efficient solutions of problem $(\mathbf{P}_0)$.

The $(\mathbf{P}_2)$ model is used to evaluate the efficiency of DMU-d, and d could be any one of the DMUs. Before introducing the $(\mathbf{P}_2)$ model, however, some additional notation is required.

u_{rd} represents the weight of rth objective corresponding to DMU-d.
v_{id} represents the weight of ith resource corresponding to DMU-d.
u_{0d} represents the returns-to-scale factor corresponding to DMU-d.
$\boldsymbol{A}_i$ is the ith row vector of $\boldsymbol{A}$ matrix in the problem $(\mathbf{P}_0)$.
$\boldsymbol{C}_r$ is the rth row vector of $\boldsymbol{C}$ matrix in the problem $(\mathbf{P}_0)$.
$\boldsymbol{W}_d = [w_{1d}, w_{2d}, \ldots, w_{Qd}]^{\mathrm{t}}$ denotes the transpose of the vector. $w_{qd} = 1$ if perform qth activity of DMU-d. Otherwise, $w_{qd} = 0$.

For the ith resource input (usage) request by alternative-d with vector $\boldsymbol{W}_d$,

$$0 \leqslant x_{id} = \boldsymbol{A}_i \cdot \boldsymbol{W}_d = \sum_{q=1}^{Q} a_{iq} w_{qd} \leqslant b_i. \tag{2.3}$$

For the rth satisfactory objective output obtained from alternative-d with vector $\boldsymbol{W}_d$,

$$0 \leqslant y_{rd} = \boldsymbol{C}_r \cdot \boldsymbol{W}_d = \sum_{q=1}^{Q} c_{rq} w_{qd} \leqslant g_r. \tag{2.4}$$

And $\boldsymbol{D}_d = (\boldsymbol{x}_d, \boldsymbol{y}_d)$ is the DMU resulted from alternative-d, where

$$\begin{aligned}\boldsymbol{x}_d &= [x_{1d}, x_{2d}, \ldots, x_{md}], \quad \boldsymbol{y}_d = [y_{ld}, y_{2d}, \ldots, y_{sd}],\\ \boldsymbol{X} &= [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^t, \quad \boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]^{\mathrm{t}},\\ \boldsymbol{b} &= [b_1, \ldots, b_i, \ldots, b_m], \quad \boldsymbol{g} = [g_1, \ldots, g_r, \ldots, g_s].\end{aligned}$$

Assume that each positive output vector can be produced from a positive input vector.

The $(\mathbf{P}_2)$ model derived by Banker et al. is below.

$$(\mathbf{P}_2) \qquad \max \quad h_d = \sum_{r=1}^{s} u_{rd} y_{rd} - u_{0d}, \tag{2.5}$$

s.t.

$$\sum_{r=1}^{s} u_{rd} y_{rj} - \sum_{i=1}^{m} v_{id} x_{ij} - u_{0d} \leqslant 0, \quad j = 1, \ldots, n, \tag{2.6}$$

$$\sum_{i=1}^{m} v_{id} x_{id} = 1, \tag{2.7}$$

$u_{rd}, v_{id} \geqslant \varepsilon \forall i, r\, u_{0d}$ are unrestricted in sign. $\varepsilon > 0$ is a small real number.

The inequality (2.6) is derived from

$$0 \leqslant \frac{\sum_{r=1}^{s} u_{rd}\, y_{rj}}{\sum_{i=1}^{m} v_{id}\, x_{ij} + u_{0d}} \leqslant 1. \tag{2.8}$$

u_{0d} denotes an offset of a fraction of weight between objective satisfaction and resource usage. There are three types of u_{0d}. If $u_{0d} < 0$, the percentage increase in the resource's usage exceeds the percentage increase in the objective's satisfaction. This is the so-called case of Increasing Returns to Scale (IRS). If IRS is present and DM decides to double current usage of resources, then its satisfaction of objectives would be more than double.

If $u_{0d} > 0$, the percentage increase in the resource's usage is less than the percentage increase in the objective's satisfactory. This is the so-called case of Decreasing Returns to Scale (DRS). If DRS is present and DM decides to double current usage of resource, then its satisfaction of objectives would be less than double.

For $j = 1, \ldots, n$, the dual problem of $(\mathbf{P}_2)$ is denoted as follows:

$$(\mathbf{P}_2') \qquad \min \quad h_d' = \omega_d - \varepsilon \left[\sum_{i=1}^{m} s_i - \sum_{r=1}^{s} s_k' \right] \tag{2.9}$$

s.t.

$$\omega_d x_{id} - \sum_{j=1}^{n} x_{ij} \lambda_j - s_i = 0, \quad i = 1, \ldots, m. \tag{2.10}$$

$$\sum_{j=1}^{n} y_{rj}\lambda_j - s'_r = y_{rd} \quad r = 1,\ldots,s \tag{2.11}$$

$$\sum_{j=1}^{n} \lambda_j = 1, \quad \lambda_j, s_i, s'_r \geqslant 0 \ \forall j,i,r, \tag{2.12}$$

where s_i and s'_r are slack variables. λ_j and ω_d, respectively, denote the dual variables corresponding to (2.6) and (2.7). Note that the $(\mathbf{P}_2)$ model is equivalent to a fractional programming problem, expressed as follows:

$$(\mathbf{P}''_2) \qquad \max \quad z_d = \frac{\sum_{r=1}^{s} u_{r\,d} y_{rd} - u_{0d}}{\sum_{i=1}^{m} v_{id} x_{id}}, \tag{2.13}$$

$$\text{s.t.} \quad \frac{\sum_{r=1}^{s} u_{r\,d} y_{rj} \quad - u_{0d}}{\sum_{i=1}^{m} v_{id} x_{ij}} \leqslant 1, \quad j = 1,\ldots,n, \tag{2.14}$$

$u_{rd}, v_{id} \geqslant 0 \forall i, r, u_{0d}$ unrestricted in sign.

In fact, a transformation of variables is needed to obtain $(\mathbf{P}''_2)$, see [5, p. 432]. Now delineate the efficient condition for a DMU. Let z'_d denote the objective value of the reciprocal (inefficiency) measure version of $(\mathbf{P}''_2)$. Since a change in variables does not alter the value of function. Thus, $z'^*_d = h'^*_d = h^*_d$, and $z^*_d = 1/h^*_d$. Also the desired relative weights have been obtained at this time. Thus, nothing more is required than the solution of $(\mathbf{P}''_2)$ or $(\mathbf{P}'_2)$ in order to determine whether $z^*_d < 1$ or, correspondingly, $z'^*_d > 1$ with efficiency prevailing if and only if $z^*_d = z'^*_d = 1$.

Now consider the slack variables in $(\mathbf{P}'_2)$. Let s^*_i $(i = 1,\ldots,m)$ and s'^*_r $(r = 1,\ldots,s)$ denote the value of slack variable in the equivalent equations (2.10) and (2.11) of the optimal solution. If s'^*_r has any positive components, it is possible to increase the associated objectives (outputs) in the amount of these slack variables without altering any of the λ^* values or without violating any constraints. Similarly, if s^*_i has any positive component, one could reduce the resource (input) in the same way. Thus, in either case, the DMU being evaluated has not achieved (relative) efficiency even with $h'^*_d = 1$. For easy reference, one can summarize what is involved as:

1. $h'^*_d = 1$, and
2. the slack variables are all zero in $(\mathbf{P}'_2)$.

According to the complementary slackness properties of primal–dual problems, if the prime variables u^*_{rd} and v^*_{id} are positive and $h^*_d = 1$, then the DMU-d is an DEA-efficient solution of problem $(\mathbf{P}_2)$. Formally, $\boldsymbol{D}_d = (\boldsymbol{x}_d, \boldsymbol{y}_d)$ is the vector of DMU-d resulting from alternative-d with vector $\boldsymbol{W}_d = [w_{1d}, w_{2d}, \ldots, w_{Qd}]^{\mathrm{t}}$. If $h^*_d = 1$, then $\boldsymbol{D}_d$ is called DEA-efficient.

Now one can prove that an alternative of the multi-objective model $\mathbf{P}_0$ which results in a DEA-efficient DMU is an efficient solution of $\mathbf{P}_0$.

Property 1. *Let $\boldsymbol{D}_d = (\boldsymbol{x}_d, \boldsymbol{y}_d)$ be the vector representing DMU-d resulting from alternative-d. If DMU-d is DEA-efficient, then alternative-d is an efficient solution.*

Proof. Let $\boldsymbol{W}_\beta$ is the vector of alternative β. If alternative-d is not an efficient solution, then there is an alternative-β such that at least one objective r results in $\boldsymbol{C}_r\boldsymbol{W}_\beta > \boldsymbol{C}_r\boldsymbol{W}_d$, where $\boldsymbol{C}_r$ is the rth row vector of matrix $\boldsymbol{C}$. Then consider the DMU-β. Since $y_{rd} = \boldsymbol{C}_r\boldsymbol{W}_d$, $y_{r\beta} = \boldsymbol{C}_r\boldsymbol{W}_\beta$. Then in $(\mathbf{P}'_2)$, the rth objective constraint in evaluating DMU-β is

$$\sum_{j=1}^{n} y_{rj}\lambda_j \geqslant y_{r\beta} > y_{rd}.$$

Contracting all slack variables of DEA-efficient DMU, $\boldsymbol{D}_d$, are zero. □

Banker et al. [1] indicated that the optimal solution of ($\mathbf{P}_2$) is respect to a DMU. In this paper, let $\boldsymbol{u}_d = [u_{1d}^*, u_{2d}^*, \ldots, u_{sd}^*], \boldsymbol{v}_d = [v_{1d}^*, v_{2d}^*, \ldots, v_{md}^*]$ and u_{0d}^*, denote optimal solution of ($\mathbf{P}_2$) respects DMU-*d*. If DMU-*d* is efficient, $\boldsymbol{u}_d\boldsymbol{y}_d - \boldsymbol{v}_d\boldsymbol{x}_d = u_{0d}^*$ is the supporting hyperplane for the production possibility set constructed by efficient DMUs.

2.3. Distance function

In a discrete solution space, $\boldsymbol{G}'$ is a convex region in that space. Let $\boldsymbol{G}^E$ denote the set of feasible solution points outside region $\boldsymbol{G}'$. Define the distance function of a point, $\boldsymbol{D}$, in set $\boldsymbol{G}^E$ to $\boldsymbol{G}'$ as

$$\delta_1(\boldsymbol{D}) = \min\left\{\|\boldsymbol{D} - \boldsymbol{D}_0\| : \boldsymbol{D}_0 \in \boldsymbol{G}'\right\}. \tag{2.15}$$

The distance function, δ_1 ($\boldsymbol{D}$) is a quadratic integer programming problem. In this case, define a support function of $\boldsymbol{G}'$ as

$$\delta_2(\boldsymbol{D}) = \sup\{\boldsymbol{D}\boldsymbol{D}_0 : \boldsymbol{D}_0 \in \boldsymbol{G}'\}. \tag{2.16}$$

The difficulty of solving $\delta_1(\boldsymbol{D})$ motivates the following property.

Property 2. *For a given* $\boldsymbol{D}$, if $\boldsymbol{D}_a$ *is the optimal solution of* $\delta_2(\boldsymbol{D})$, *i.e.* $\delta_2(\boldsymbol{D}) = \boldsymbol{D}\boldsymbol{D}_a$, *then* $\boldsymbol{D}_a$ *is also the optimal solution of* $\delta_1(\boldsymbol{D})$, *i.e.* $\delta_1(\boldsymbol{D}) = \|\boldsymbol{D} - \boldsymbol{D}_a\|$.

Proof. If $\boldsymbol{D}_a$ is not the optimal solution of $\delta_1(\boldsymbol{D})$, there is a $\boldsymbol{D}_b \in \boldsymbol{G}'$ such that $\|\boldsymbol{D} - \boldsymbol{D}_a\| > \|\boldsymbol{D} - \boldsymbol{D}_b\|$ which implies $\|\boldsymbol{D}_a\|^2 - \|\boldsymbol{D}_b\|^2 - 2\boldsymbol{D}\boldsymbol{D}_a + 2\boldsymbol{D}\boldsymbol{D}_b > 0$. Since $\boldsymbol{D}, \boldsymbol{D}_a$ and $\boldsymbol{D}_b$ are nonnegative bounded integer, $\|\boldsymbol{D} - \boldsymbol{D}_a\| > \|\boldsymbol{D} - \boldsymbol{D}_b\|$ implies $\|\boldsymbol{D}_a\|^2 - \|\boldsymbol{D}_b\|^2 > 0$. Hence, $\boldsymbol{D}\boldsymbol{D}_a < \boldsymbol{D}\boldsymbol{D}_b$. Contracting $\boldsymbol{D}_a$ is the optimal solution of $\delta_2(\boldsymbol{D})$. □

Assume the supporting hyperplanes of set $\boldsymbol{G}'$ form a convex hull, denoted as $H(\boldsymbol{G}')$. Hence, the surface of $H(\boldsymbol{G}')$ is a continuous bounded set.

Nemhauser and Wolsey [11, p. 107] indicated that the optimal bounded solution of $\{\sup \boldsymbol{D}\boldsymbol{D}_a : \boldsymbol{D}_a \in H(\boldsymbol{G}')\}$ is also an optimal solution of $\delta_2(\boldsymbol{D})$. Therefore, $\delta_2(\boldsymbol{D})$ could be redefined as

$$\delta_2(\boldsymbol{D}) = \sup\{\boldsymbol{D}\boldsymbol{D}_a : \boldsymbol{D}_a \in H(\boldsymbol{G}')\}. \tag{2.17}$$

Then $\delta_2(\boldsymbol{D})$ can be solved as a linear programming problem, thus saving much computation time.

Property 3. $\delta_2(\boldsymbol{D})$ *is a convex function.*

Proof. For any two points $\boldsymbol{D}_a$, $\boldsymbol{D}_b \in \boldsymbol{G}^E$ and for a point $\boldsymbol{D}_c \in H(\boldsymbol{G}')$. By definition of $\delta_2(\boldsymbol{D})$ function, one gets

$$\mu\boldsymbol{D}_a\boldsymbol{D}_c \leqslant \mu \sup\{\boldsymbol{D}_a\boldsymbol{D}_c\} \text{ and } (1-\mu)\boldsymbol{D}_b\boldsymbol{D}_c \leqslant (1-\mu)\sup\{\boldsymbol{D}_b\boldsymbol{D}_c\}, \quad 0 \leqslant \mu \leqslant 1.$$

Combining these two inequalities, one obtains

$$\mu \boldsymbol{D}_a\boldsymbol{D}_c + (1-\mu)\boldsymbol{D}_b\boldsymbol{D}_c \leqslant \mu \sup\{\boldsymbol{D}_a\boldsymbol{D}_c\} + (1-\mu)\sup\{\boldsymbol{D}_b\boldsymbol{D}_c\}, \quad 0 \leqslant \mu \leqslant 1.$$

Since for each $\boldsymbol{D}_c \in H(\boldsymbol{G}')$ the above inequality would hold. One obtains

$$\sup\{\mu \boldsymbol{D}_a\boldsymbol{D}_c + (1-\mu)\boldsymbol{D}_b\boldsymbol{D}_c\} \leqslant \mu \sup\{\boldsymbol{D}_a\boldsymbol{D}_c\} + (1-\mu)\sup\{\boldsymbol{D}_b\boldsymbol{D}_c\}, \quad 0 \leqslant \mu \leqslant 1.$$

Thus, $\delta_2(\boldsymbol{D})$ is a convex function. □

Property 4. *Let $\boldsymbol{D}'$ is a given point of $\boldsymbol{G}'$. If $\delta_2(\boldsymbol{D}') = \boldsymbol{D}'\boldsymbol{D}^*$, then $\boldsymbol{D}^*$ is a subgradient of $\delta_2(\boldsymbol{D})$ at $\boldsymbol{D}'$.*

Proof. Since $\delta_2(\boldsymbol{D})$ is a convex function, if $\delta_2(\boldsymbol{D}) \geqslant \delta_2(\boldsymbol{D}') + \boldsymbol{D}^*(\boldsymbol{D} - \boldsymbol{D}')\ \forall \boldsymbol{D} \in \boldsymbol{G}^E$, then $\boldsymbol{D}^*$ is a subgradient of δ_2 $(\boldsymbol{D})$ at $\boldsymbol{D}'$. Since $\delta_2(\boldsymbol{D}) = \sup\{\boldsymbol{D}\boldsymbol{D}_0\} = \boldsymbol{D}\boldsymbol{D}^* \geqslant \boldsymbol{D}\boldsymbol{D}_0\ \forall \boldsymbol{D}_0 \in H(\boldsymbol{G}')$, then $\delta_2(\boldsymbol{D}) = \boldsymbol{D}\boldsymbol{D}^* \geqslant \boldsymbol{D}\boldsymbol{D}_0 \forall\ \boldsymbol{D} \in \boldsymbol{G}^E$, where $\boldsymbol{D}^* \in H(\boldsymbol{G}')$. Thereby, $\delta_2(\boldsymbol{D}) \geqslant \boldsymbol{D}\boldsymbol{D}^* + \boldsymbol{D}'\boldsymbol{D}^* - \boldsymbol{D}'\boldsymbol{D}^* = \boldsymbol{D}'\boldsymbol{D}^* + (\boldsymbol{D} - \boldsymbol{D}')\boldsymbol{D}^* = \delta_2(\boldsymbol{D}') + (\boldsymbol{D} - \boldsymbol{D}')\boldsymbol{D}^*$, $\forall \boldsymbol{D} \in \boldsymbol{G}^E$. □

3. Two-stage algorithm

The authors developed a two-stage algorithm to generate and evaluate DMUs. The first stage generates a DMU with the maximum of the *distance function* as described in Section 2.3. The second stage is then used to evaluate the efficiency of the generated and evaluated DMU. Each DMU results from an *alternative*. The alternative is a combination of *activities*.

3.1. Framework of the algorithm

The algorithm has two: Stage-I, and Stage-II. Stage-0 initializes the algorithm. Stage-I generates a further DMU. Stage-II evaluates the efficiency of the DMU. Notations used in the algorithm are presented below. An iteration index, k, was added to notations to indicate which DMU would be generated at iteration.

$\underline{\boldsymbol{G}}$: the set of all the possible DMUs (production possibility set) of the MRAP.
$\boldsymbol{G}_k$: a subset of $\underline{\boldsymbol{G}}$, denotes the set of evaluated DMUs at the beginning of the k_{th} iteration.
$\boldsymbol{G}'_k$: a subset of $\boldsymbol{G}_k$, denotes the efficient DMUs.
$H(\boldsymbol{G}'_k)$: the convex hull constructed by $\boldsymbol{G}'_k$.
$\boldsymbol{G}^H_k$: contains all the non-evaluated and evaluated DMUs inside the convex hull $H(\boldsymbol{G}'_k)$ at the beginning of iteration k.
$\boldsymbol{G}^E_k$: a subset of $\underline{\boldsymbol{G}}$, denotes DMUs outside $H(\boldsymbol{G}'_k)$.

3.2. Stage-0: Initialization

A set of alternatives triggers the two-stage algorithm. The authors arbitrarily chose a set of alternatives satisfying resource constraints. Stage-0 is summarized below.

Step 0.1: $k = 0$. Arbitrarily choose a set of feasible alternatives. Obtain the DMU according to each alternative. Calculate the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$. Let $\boldsymbol{G}_0$ be the set of these DMUs. Go to Step 0.2.

Step 0.2: Evaluate each DMU's efficiency in set $\boldsymbol{G}_0$ by solving ($\mathbf{P}_2$). If the prime variables u^*_{kd} and v^*_{id} are positive and $h^*_d = 1$, the evaluated DMU is DEA-efficient and is also a member of $\boldsymbol{G}'_0$. Follow step 2.2 to construct $H(\boldsymbol{G}'_0)$. Go to Stage-I. If no DMU is DEA-efficient, go to step 0.1.

3.3. Stage-I (generation)

Stage-I aims to augment the set $H(\boldsymbol{G}'_k)$. At the beginning of Stage-I, the updated (from Stage-II) $\boldsymbol{G}_k^E, \boldsymbol{G}_k^H, \boldsymbol{G}'_k$, and $H(\boldsymbol{G}'_k)$ are given. Based on this information, seek a further DMU contained in $\boldsymbol{G}_k^E$ from $H(\boldsymbol{G}'_k)$. If one exists, proceed with the algorithm. Otherwise, stop the algorithm and the alternative results in a member of efficient DMU, $\boldsymbol{G}'_{k\cdot}$, being the efficient solution of $(\mathbf{P}_0)$. By incorporating the distance function, the purpose of stage-I is equivalent to solving the problem, $\max\{\delta_1(\boldsymbol{D})\colon \boldsymbol{D} \in \boldsymbol{G}'_k\}$. Based on Property 2, however, one can use $\delta_2(\boldsymbol{D})$ instead of $\delta_1(\boldsymbol{D})$. That is, using $\max\{\delta_2(\boldsymbol{D}) : \boldsymbol{D} \in \boldsymbol{G}_k^E\}$ solves the problem, $\max\{\delta_1(\boldsymbol{D}) : \boldsymbol{D} \in \boldsymbol{G}_k^E\}$. Now the problem is maximizing a convex function, $\delta_2(\boldsymbol{D})$. Bazaraa et al. [3] developed a necessary condition for such a problem. They indicated that several local maximal satisfying the condition exist, but there is no local information at such solutions that could lead to better points as the minimization case does. Based on the above discussion, the purpose of stage-I is the same as finding the optimal solution of $(\mathbf{P}_3)$. Due to Property 4, use a subgradient optimization method to solve $(\mathbf{P}_3)$. Based on the basic step of the subgradient optimization method, three steps are used in Stage-I.

$$(\mathbf{P}_3) \quad \max\{\delta_2(\boldsymbol{D}) : \boldsymbol{D} \in \boldsymbol{G}_k^E\}. \tag{3.1}$$

Step 1.1: Initial DMU

Solve problem $(\mathbf{P}_4)$ to find initial alternative d. If there exists, go to Step 1.2. Otherwise, stop. Current alternatives resulting in efficient DMUs, $\boldsymbol{G}'_k$, are efficient solutions of $(\mathbf{P}_0)$.

$$(\mathbf{P}_4) \qquad \max \quad \sum_{q=1}^{Q} w_{qd}, \tag{3.2}$$

s.t.

$$\sum_{q=1}^{Q}\left[\sum_{r=1}^{s} u_{rj}^* c_{rq} - \sum_{i=1}^{m} v_{ij}^* a_{iq}\right] w_{qd} > u_{0j}^* \quad \forall j \in \boldsymbol{G}'_k, \tag{3.3}$$

$$0 \leqslant \sum_{q=1}^{Q} c_{rq} w_{qd} \leqslant g_r, \quad r = 1, \ldots, s, \tag{3.4}$$

$$0 \leqslant \sum_{q=1}^{Q} a_{pq} w_{qd} \leqslant b_i, \quad i = 1, \ldots, m,$$

$$w_{qd} \in \{0, 1\} \tag{3.5}$$

g_r and b_i are bounded constraints of objectives and resources, respectively.

The convex hull of all the efficient DMUs in $\boldsymbol{G}'_k$ is constructed by the associated supporting hyperplanes:

$$\boldsymbol{u}_j \cdot \boldsymbol{y}_j - \boldsymbol{v}_j \cdot \boldsymbol{x}_j \leqslant u_{0j}^* \quad \forall j \in \boldsymbol{G}'_k. \tag{3.6}$$

For any DMU which is not contained in $H(\boldsymbol{G}'_k)$ and results from an alternative, say, d, its $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$ must satisfy the set of constraints

$$\boldsymbol{u}_j \cdot \boldsymbol{y}_d - \boldsymbol{v}_j \cdot \boldsymbol{x}_d > u_{0j}^* \quad \forall j \in \boldsymbol{G}'_k. \tag{3.7}$$

Inequality (3.7) can be rewritten as

$$\begin{aligned} \boldsymbol{u}_j \cdot \boldsymbol{y}_d - \boldsymbol{v}_j \cdot \boldsymbol{x}_d &= \sum_{r=1}^{s} y_{rd} u_{rj}^* - \sum_{i=1}^{m} x_{id} v_{ij}^* \\ &= \sum_{r=1}^{s} u_{rj}^* \sum_{q=1}^{Q} c_{rq} w_{qd} - \sum_{i=1}^{m} v_{ij}^* \sum_{q=1}^{Q} a_{iq} w_{qd} \\ &= \sum_{q=1}^{Q} \left[\sum_{r=1}^{s} u_{rj}^* c_{rq} - \sum_{i=1}^{m} v_{ij}^* a_{iq} \right] w_{qd} > u_{0j}^* \quad \forall j \in \boldsymbol{G}_k'. \end{aligned} \tag{3.8}$$

Any DMU can also satisfy the boundary conditions as (2.3) and (2.4). Define the objective of ($\mathbf{P}_4$) because the authors subjectively hoped more activities could be performed. Other objective functions should not be restricted.

Step 1.2: Obtain subgradient

Let $\boldsymbol{D}'$ be DMU resulting from alternative-*d*. Solve $\delta_2(\boldsymbol{D})' = \sup\{\boldsymbol{D}'\boldsymbol{D} : \boldsymbol{D} \in H(\boldsymbol{G}_k')\}$ for optimal solution $\delta_2(\boldsymbol{D}') = \boldsymbol{D}'\boldsymbol{D}^*$. Go to Step 1.3.

At the beginning of this step, $\boldsymbol{D}'$, a DMU resulting from the initial alternative, $H(\boldsymbol{G}_k')$ and $\boldsymbol{G}_k^E$ must be determined. Then, starting from $\boldsymbol{D}'$, one needs to know what direction would obtain a further DMU. One of the candidates (directions) is a subgradient. This gives a subgradient of $\delta_2(\boldsymbol{D})$ at $\boldsymbol{D}'$. Due to Property 4, a subgradient of $\delta_2(\boldsymbol{D})$ at $\boldsymbol{D}'$ can be easily obtained. However, $\delta_2(\boldsymbol{D})$ uses $H(\boldsymbol{G}_k')$ to describe an efficient frontier formed by $\boldsymbol{G}_k'$. Hence, $H(\boldsymbol{G}_k')$ must be constructed. (Constructing $H(\boldsymbol{G}_k')$ is delineated in stage-II)

Step 1.3: Find a more distant DMU

Solve the following problem ($\mathbf{P}_2$) in terms of the linear search method, $\boldsymbol{D} = \boldsymbol{D}' + p\boldsymbol{D}^*$ to obtain a DMU, $\boldsymbol{D}$, more distant from $H(\boldsymbol{G}_k')$. If one exists, replace $\boldsymbol{D}'$ by $\boldsymbol{D}$ and repeat Step 1.2. Otherwise, let $\boldsymbol{G}_{k+1} = \boldsymbol{G}_k' \cup \{\boldsymbol{D}'\}$ and go to Stage-II.

At the beginning of this step, $\boldsymbol{D}'$ and a subgradient $\boldsymbol{D}^*$ of $\delta_2(\boldsymbol{D})$ at $\boldsymbol{D}'$ must be determined. Recall that Stage-I is equivalent to solve ($\mathbf{P}_3$). Since ($\mathbf{P}_3$) is a discrete optimization problem. Based on Property 4, a subgradient optimization method would be involved. The subgradient optimization method improves the objective function in terms of step size and subgradient. By introducing step size and subgradient, a closer optimal solution would be obtained. Bazaraa and Sherali [2] have introduced convergence and computational efficient procedures for selecting step size. However, a generally subgradient optimization method occurred in closed convex set. The ($\mathbf{P}_3$) problem is discrete. Hence, restrict the integrity of step size(p). Now develop a linear search, $\boldsymbol{D} = \boldsymbol{D}' + p\boldsymbol{D}^*$, algorithm to resolve ($\mathbf{P}_3$). Because of the integrity of $\boldsymbol{D}$, p must be one of the three types, −1, 0, or 1. If more improvement is not attempted, stop and update $\boldsymbol{G}_{k+1}, \boldsymbol{G}_{k+1} = \boldsymbol{G}_k' \cup \{\boldsymbol{D}'\}$. Otherwise, replace $\boldsymbol{D}'$ by $\boldsymbol{D}$ and redo Step 1.2.

3.4. Stage-II: Evaluation

Stage-II aims to evaluate the efficiency of DMU contained in $\boldsymbol{G}_k$. The method used is DEA with Banker et al.'s [1] model. In the approach outlined here, the authors separated Stage-II into two steps, although these two steps are finished simultaneously. Each step is explained below.

Step 2.1: Evaluate efficiency: Calculate the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices by $\boldsymbol{G}_{k+1}$ and solve problem ($\mathbf{P}_2$) to evaluate efficiency. After that, obtain $\boldsymbol{G}_{k+1}'$ and optimal solutions $\boldsymbol{u}_d$, $\boldsymbol{v}_d$, and u_{0d}^* corresponding to efficient DMU-*d*. Go to Step 2.2.

At the beginning of Stage-II, an updated $\boldsymbol{G}_{k+1}$ is given. This step is then used to evaluate the efficiency of members contained in $\boldsymbol{G}_{k+1}$. Assume the total number of DMUs contained in $\boldsymbol{G}_{k+1}$ is n. The method used is

DEA with the ($\mathbf{P}_2$) model. Then n linear programs of the ($\mathbf{P}_2$) model should be solved to obtain efficiency of each DMU. In other words, assigning $d = 1, 2, \ldots, n$ to the ($\mathbf{P}_2$) model. Each linear program differs only in objective function (2.5) and constraint (2.7). If the prime variables u^*_{kd} and v^*_{id} are positive and $h^*_d = 1$, the evaluated DMU is DEA-efficient and is also a member of $\boldsymbol{G}'_{k+1}$.

After the DEA is evaluated, new efficient DMUs, formed set $\boldsymbol{G}'_{k+1}$, are identified with DEA-efficient conditions. Based on these efficient DMUs, $\boldsymbol{G}'_{k+1}$, the convex hull, $H(\boldsymbol{G}'_{k+1})$, constructed by $\boldsymbol{G}'_{k+1}$, are determined.

Step 2.2: Construct $H(\boldsymbol{G}'_k)$:

Let $H(\boldsymbol{G}'_{k+1}) = \{(\boldsymbol{x}_d, \boldsymbol{y}_d)\colon \boldsymbol{u}_d\boldsymbol{y}_d - \boldsymbol{v}_d\boldsymbol{x}_d \leqslant u^*_{0d},\ d \in \boldsymbol{G}'_{k+1}, (\boldsymbol{x}_d, \boldsymbol{y}_d) \in \boldsymbol{G}^H_{k+1}\}$. Go to Stage-I, update $k = k + 1$.

Recall that since $\boldsymbol{u}_d\,\boldsymbol{y}_d - \boldsymbol{v}_d\,\boldsymbol{x}_d = u^*_{0d}$ is a supporting hyperplane of $\boldsymbol{G}^H_{k+1}$ and $\boldsymbol{x}_d$, $\boldsymbol{y}_d$ are bounded, $\boldsymbol{u}_d$, $\boldsymbol{v}_d$ and u'_{0d} are the optimal solution of ($\mathbf{P}_2$) corresponding to efficient DMU-d in $\boldsymbol{G}'_{k+1}$. Moreover, $\boldsymbol{u}_d\,\boldsymbol{y}_d - \boldsymbol{v}_d\,\boldsymbol{x}_d \leqslant u'_{0d}$, $d \in \boldsymbol{G}'_{k+1}$, for each $(\boldsymbol{x}_d, \boldsymbol{y}_d) \in \boldsymbol{G}'_{k+1}$. One can construct $H(\boldsymbol{G}'_{k+1})$ in terms of the set of constraints, $\boldsymbol{u}_d\,\boldsymbol{y}_d - \boldsymbol{v}_d\,\boldsymbol{x}_d \leqslant u'_{0d}$, and bounded value of $\boldsymbol{x}_d$, $\boldsymbol{y}_d$. Hence, $H(\boldsymbol{G}'_{k+1}) = \{(\boldsymbol{x}_d, \boldsymbol{y}_d)\colon \boldsymbol{u}_d\,\boldsymbol{y}_d - \boldsymbol{v}_d\,\boldsymbol{x}_d \leqslant u^*_{0d}, (\boldsymbol{x}_d, \boldsymbol{y}_d) \in \boldsymbol{G}^H_{k+1}\}$.

The efficient solutions of MRAP can be obtained from the final $\boldsymbol{G}'_k$ of this recursive two-stage algorithm. Each *alternative* resulting in a member of $\boldsymbol{G}'_k$ is an efficient solution of MRAP (following Property 1). Fig. 1 portrays this three-stage algorithm.

3.5. Illustrative example

Consider the following multi-objective 0–1 linear program:

$$
\begin{aligned}
\max\quad & 3w_1 + 6w_2 + 5w_3 - 2w_4 + 3w_5,\\
\max\quad & 6w_1 + 7w_2 + 4w_3 + 3w_4 - 8w_5,\\
\max\quad & 5w_1 - 3w_2 + 8w_3 - 4w_4 + 3w_5,\\
\text{s.t.}\quad & -2w_1 + 3w_2 + 8w_3 - w_4 + 5w_5 \leqslant 13,\\
& 6w_1 + 2w_2 + 4w_3 + 4w_4 - 3w_5 \leqslant 15,\\
& 4w_1 - 2w_2 + 6w_3 - 2w_4 + w_5 \leqslant 11,\\
& w_1, w_2, w_3, w_4, w_5 \in \{0\ 1\}.
\end{aligned}
$$

The production possibility set, $\underline{\boldsymbol{G}}$, is

$$
\underline{\boldsymbol{G}} = \left\{ \begin{array}{l}
\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} : \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \leqslant \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \leqslant \begin{bmatrix} 13 \\ 15 \\ 11 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \leqslant \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \leqslant \begin{bmatrix} 17 \\ 20 \\ 16 \end{bmatrix}, \\
\boldsymbol{y} \geqslant 0 \text{ can be produced from } \boldsymbol{x} \geqslant 0
\end{array} \right\}.
$$

Stage-0: *Initialization.*

Step 0.1. Since there are five decision variables, the total number of possible alternatives are $2^5 = 32$. The authors arbitrarily chose four initial feasible alternatives, $\boldsymbol{W}_1 = [1\ 1\ 1\ 0\ 0]$, $\boldsymbol{W}_2 = [1\ 0\ 1\ 0\ 1\ ]$, $\boldsymbol{W}_3 = [1\ 1\ 0\ 0\ 0]$ and $\boldsymbol{W}_4 = [0\ 0\ 1\ 1\ 0]$. Using Eqs. (2.3) and (2.4), these alternatives resulted in DMUs, $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$ and $\boldsymbol{D}_4$,

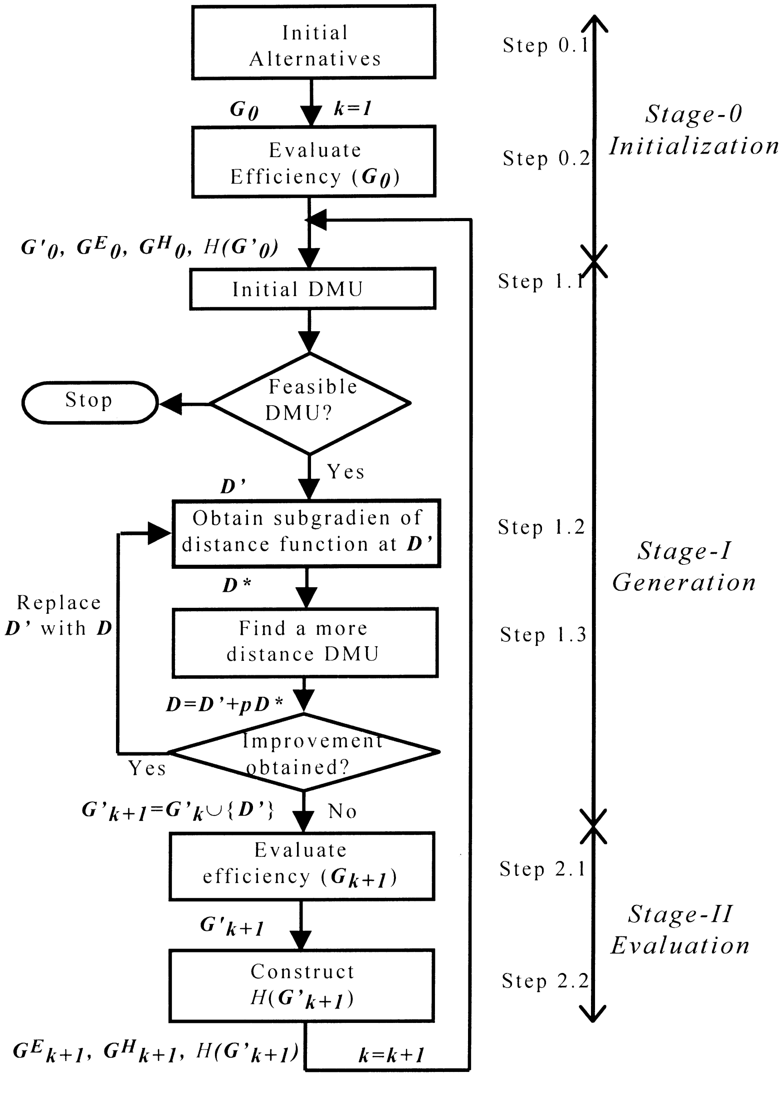

Fig. 1. Framework of the algorithm.

respectively. $\boldsymbol{D}_1 = [9, 12, 8, 14, 17, 10]$, $\boldsymbol{D}_2 = [11, 7, 11, 11, 2, 16]$, $\boldsymbol{D}_3 = [1, 8, 2, 9, 13, 2]$ and $\boldsymbol{D}_4 = [7, 8, 4, 3, 7, 4]$. $\boldsymbol{X}$, $\boldsymbol{Y}$ matrices could also be obtained. These four DMUs are to be evaluated, therefore $\boldsymbol{G}_0 = \{1, 2, 3, 4\}$, where number 1, 2, 3 and 4 represent the index of DMU-d. Go to step 0.2.

$$\boldsymbol{X} = \begin{bmatrix} 9 & 12 & 8 \\ 11 & 7 & 11 \\ 1 & 8 & 2 \\ 7 & 8 & 4 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} 14 & 17 & 10 \\ 11 & 2 & 16 \\ 9 & 13 & 2 \\ 3 & 7 & 4 \end{bmatrix}.$$

Step 0.2: Let $d = 1$, evaluate DMU-*d*, $\boldsymbol{D}_1$, the ($\mathbf{P}_2$) model is formulated as follows.

$$\begin{aligned}
\max \quad & h_1 = 14u_{11} + 17u_{21} + 10u_{31} - u_{01},\\
\text{s.t.} \quad & 14u_{11} + 17u_{21} + 10u_{31} - 9v_{11} - 12v_{21} - 8v_{31} - u_{01} \leqslant 0,\\
& 11u_{11} + 2u_{21} + 16u_{31} - 11v_{11} - 7v_{21} - 11v_{31} - u_{01} \leqslant 0,\\
& 9u_{11} + 13u_{21} + 2u_{31} - v_{11} - 8v_{21} - 2v_{31} - u_{01} \leqslant 0,\\
& 3u_{11} + 7u_{21} + 4u_{31} - 7v_{11} - 8v_{21} - 4v_{31} - u_{01} \leqslant 0,\\
& 9v_{11} + 12v_{21} + 8v_{31} = 1,
\end{aligned}$$

$u_{11}, u_{21}, u_{31}, v_{11}, v_{21}$ and $v_{31} \geqslant \varepsilon$, u_{01} is unrestricted in sign.

By a similar method evaluate $\boldsymbol{D}_2$, $\boldsymbol{D}_3$ and $\boldsymbol{D}_4$. The solutions are listed in Table 1. Since $h'_4 \neq 1$, $\boldsymbol{D}_4$ is not an efficient DMU. Hence $\boldsymbol{G}'_0 = \{1, 2, 3\}$ were obtained.

Following Eq. (3.6), $H(\boldsymbol{G}'_0)$ is constructed as:

$$\begin{aligned}
& 0.0001y_{11} + 0.0187y_{21} + 0.0680y_{31} - 0.0001x_{11} - 0.0261x_{21} - 0.0867x_{31} \leqslant 0,\\
& 0.0001y_{12} + 0.0166y_{22} + 0.0604y_{32} - 0.0001x_{12} - 0.0232x_{22} - 0.0760x_{32} \leqslant 0,\\
& 0.0001y_{13} + 0.0492y_{23} + 0.1795y_{33} - 0.0001x_{13} - 0.0684x_{23} - 0.2265x_{33} \leqslant 0.
\end{aligned}$$

Go to **Stage-I**.

Iteration $k = 1$.

Stage-I: *Generation.*

Step 1.1: Solving the following 0–1 integer problem by implicit enumeration.

$$\begin{aligned}
(\mathbf{P}_4) \qquad \max \quad & w_{1d} + w_{2d} + w_{3d} + w_{4d} + w_{5d},\\
\text{s.t.} \quad & -0.0507w_{1d} + 0.0484w_{2d} - 0.0061w_{3d} - 0.1470w_{4d} + 0.0458w_{5d} > 0,\\
& -0.0411w_{1d} + 0.0409w_{2d} + 0.0005w_{3d} - 0.1327w_{4d} + 0.0418w_{5d} > 0,\\
& -0.1234w_{1d} + 0.1224w_{2d} - 0.3911w_{4d} + 0.1234w_{5d} \leqslant 0,\\
& 0 \leqslant -2w_{1d} + 3w_{2d} + 8w_{3d} - w_{4d} + 5w_{5d} \leqslant 13,\\
& 0 \leqslant 6w_{1d} + 2w_{2d} + 4w_{3d} + 4w_{4d} - 3w_{5d} \leqslant 15,\\
& 0 \leqslant 4w_{1d} - 2w_{2d} + 6w_{3d} - 2w_{4d} + w_{5d} \leqslant 11,\\
& 0 \leqslant 3w_{1d} + 6w_{2d} + 5w_{3d} - 2w_{4d} + 3w_{5d} \leqslant 20,\\
& 0 \leqslant 6w_{1d} + 7w_{2d} + 4w_{3d} + 3w_{4d} - 8w_{5d} \leqslant 21,\\
& 0 \leqslant 5w_{1d} - 3w_{2d} + 8w_{3d} - 4w_{4d} + 3w_{5d} \leqslant 19,\\
& w_{1d}, w_{2d}, w_{3d}, w_{4d} \text{ and } w_{5d} \in 0, 1.
\end{aligned}$$

Table 1
Data obtained at stage 0 of the example

Primal variable	DMU-*d*			
	1	2	3	4
u^*_{1d}	0.0001	0.0001	0.0001	0.0001
u^*_{2d}	0.0187	0.0166	0.0492	0.0122
u^*_{3d}	0.0680	0.0604	0.1795	0.1701
v^*_{1d}	0.0001	0.0001	0.0001	0.0001
v^*_{2d}	0.0261	0.0232	0.0684	0.0001
v^*_{3d}	0.0867	0.0760	0.2265	0.2496
u^*_{0d}	0	0	0	0
h^*_d	1	1	1	0.7764

The optimal solution as $\boldsymbol{W}_d = [1\ 1\ 0\ 0\ 1]$ resulting in a new DMU $\boldsymbol{D}' = [6\ 3\ 3\ 12\ 5\ 5]$. Go to step 1.2.

Step 1.2: Solve $\delta_2(\boldsymbol{D}')$. $\boldsymbol{D}^* = [13\ 15\ 11\ 20\ 21\ 13]$ by rounding $y_{3d} = 13.8195$ down to 13. Go to step 1.3.

Step 1.3: Improving $\boldsymbol{D}$ in terms of linear search strategy, $\boldsymbol{D} = \boldsymbol{D}' + p\boldsymbol{D}^*$. $\boldsymbol{D}$ could not be improved. Hence the new DMU is $\boldsymbol{D}'$. Let $\boldsymbol{D}' = [6\ 3\ 3\ 12\ 5\ 5]$ to be DMU-*4*. $\boldsymbol{G}_l = \boldsymbol{G}' \cup \boldsymbol{D}' = \{1,2,3,4\}$. Go to **Stage-II**.

Stage-II: *Evaluation.*

Step 2.1: The new $\boldsymbol{X}$, $\boldsymbol{Y}$ matrices determined by the DMUs in $\boldsymbol{G}_1$ are as follows:

$$\boldsymbol{X} = \begin{bmatrix} 9 & 12 & 8 \\ 11 & 7 & 11 \\ 1 & 8 & 2 \\ 6 & 3 & 3 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} 14 & 17 & 10 \\ 11 & 2 & 16 \\ 9 & 13 & 2 \\ 12 & 5 & 5 \end{bmatrix}.$$

Solve problem ($\mathbf{P}_2$) to evaluate DMU-1, 2, 3, 4 of $\boldsymbol{G}_1$. The solutions are listed in Table 2. Since the prime variables are positive and the objective function values equal 1, $\boldsymbol{G}_j' = 1,2,3,4$.

Step 2.2: After the DEA is evaluated, $\boldsymbol{G}_l = \{1,2,3,4\}$ and $H(\boldsymbol{G}_1')$ were obtained.

$$0.0001y_{11} + 0.0333y_{21} + 0.0454y_{31} - 0.0327x_{11} - 0.0588x_{21} - 0.0001x_{31} \leqslant 0.0223,$$

$$0.0001y_{12} + 0.0432y_{22} + 0.0588y_{32} - 0.0423x_{12} - 0.0762x_{22} - 0.0001x_{32} \leqslant 0.0286,$$

$$0.640y_{13} + 0.0663y_{23} + 0.0902y_{33} - 0.0649x_{13} - 0.1169x_{23} - 0.0001x_{33} \leqslant 0.0435,$$

$$0.0163y_{14} + 0.0896y_{24} + 0.1218y_{34} - 0.0876x_{14} - 0.1580x_{24} - 0.0001x_{34} \leqslant 0.0585.$$

Go to **Stage-I**.

Iteration $k = 2$.

Stage-I: *Generation.*

Step 1.1: Solving the following 0–1 integer problem by implicit enumeration:

$$\begin{aligned} (\mathbf{P}_4) \quad & \max \quad w_{1d} + w_{2d} + w_{3d} + w_{4d} + w_{5d}, \\ & \text{s.t.} \quad 0.1393w_{1d} - 0.1180w_{2d} - 0.0005w_{3d} - 0.2842w_{4d} - 0.1171w_{5d} > 0.0223, \\ & \quad 0.1823w_{1d} - 0.1552w_{2d} - 0.0073w_{3d} - 0.3672w_{4d} - 0.1564w_{5d} > 0.0286, \\ & \quad 0.4688w_{1d} + 0.1492w_{2d} + 0.3194w_{3d} - 0.6924w_{4d} - 0.0417w_{5d} > 0.0435, \\ & \quad 0.4223w_{1d} - 0.2190w_{2d} + 0.0809w_{3d} - 0.7952w_{4d} - 0.2666w_{5d} > 0.0585, \\ & \quad 0 \leqslant -2w_{1d} + 3w_{2d} + 8w_{3d} - w_{4d} + 5w_{5d} \leqslant 13, \\ & \quad 0 \leqslant 6w_{1d} + 2w_{2d} + 4w_{3d} + 4w_{4d} - 3w_{5d} \leqslant 15, \\ & \quad 0 \leqslant 4w_{1d} - 2w_{2d} + 6w_{3d} - 2w_{4d} + w_{5d} \leqslant 11, \end{aligned}$$

Table 2
Data obtained in the first iteration of the example

Primal variable	DMU-d			
	1	2	3	4
u_{1d}^*	0.0001	0.0001	0.0640	0.0163
u_{2d}^*	0.0333	0.0432	0.0663	0.0896
u_{3d}^*	0.0454	0.0588	0.0902	0.1218
v_{1d}^*	0.0327	0.0423	0.0649	0.0876
v_{2d}^*	0.0588	0.0762	0.1169	0.1580
v_{3d}^*	0.0001	0.0001	0.0001	0.0001
u_{0d}^*	0.0223	0.0286	0.0435	0.0585
h_d^*	1	1	1	1

$$0 \leqslant 3w_{1d} + 6w_{2d} + 5w_{3d} - 2w_{4d} + 3w_{5d} \leqslant 20,$$
$$0 \leqslant 6w_{1d} + 7w_{2d} + 4w_{3d} + 3w_{4d} - 8w_{5d} \leqslant 21,$$
$$0 \leqslant 5w_{1d} - 3w_{2d} + 8w_{3d} - 4w_{4d} + 3w_{5d} \leqslant 19,$$
$$w_{1d}, w_{2d}, w_{3d}, w_{4d} \text{ and } w_{5d} \in \{0, 1\}.$$

The optimal solution is $\boldsymbol{W}_d = [1\ 0\ 1\ 0\ 0]$ resulting in a new DMU $\boldsymbol{D}' = [6\ 10\ 10\ 8\ 10\ 13]$. Go to step 1.2.

Step 1.2: Solve $\delta_2 = (\boldsymbol{D}')$. $\boldsymbol{D}^* = [13\ 15\ 11\ 0\ 21\ 13]$ by rounding $y_{ld} = 0.0063$ and $y_{3d} = 13.8481$ down to 0 and 13, respectively. Go to step 1.3.

Step 1.3: Improving $\boldsymbol{D}$ in terms of linear search strategy, $\boldsymbol{D} = \boldsymbol{D}' + p\boldsymbol{D}^*$. $\boldsymbol{D}$ could not be improved. Hence the new DMU is $\boldsymbol{D}'$. Let $\boldsymbol{D}'$ be DMU-5. $\boldsymbol{G}_2 = \boldsymbol{G}_1' \bigcup \{\boldsymbol{D}'\} = \{1, 2, 3, 4, 5\}$. Go to **Stage-II**.

Stage-II: *Evaluation*.

Step 2.1: The new $\boldsymbol{X}$, $\boldsymbol{Y}$ matrices were determined by the DMUs in $\boldsymbol{G}_2$ as follows:

$$\boldsymbol{X} = \begin{bmatrix} 9 & 12 & 8 \\ 11 & 7 & 11 \\ 1 & 8 & 2 \\ 6 & 3 & 3 \\ 6 & 10 & 10 \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} 14 & 17 & 10 \\ 11 & 2 & 16 \\ 9 & 13 & 2 \\ 12 & 5 & 5 \\ 8 & 10 & 13 \end{bmatrix}.$$

The evaluated solutions are listed in Table 3. Only the first two DMUs satisfy the conditions of efficient DMU. Hence, $\boldsymbol{G}_2' = \{1, 2\}$.

Step 2.2: $H(\boldsymbol{G}_k')$ were found as:

$$0.0001y_{11} + 0.0120y_{21} + 0.0896y_{31} - 0.0125x_{11} - 0.0001x_{21} - 0.1108x_{31} \leqslant 0.1017,$$
$$0.0001y_{12} + 0.0030y_{22} + 0.0621y_{32} - 0.0179x_{12} - 0.0001x_{22} - 0.0729x_{32} \leqslant 0.0000.$$

Go to **Stage-I**.

Iteration $k = 3$.

Stage-I: *Generation*.

Step 1.1: Solve the following 0–1 integer problem.

$$\begin{aligned} (\mathbf{P}_4) \qquad \max \quad & w_{1d} + w_{2d} + w_{3d} + w_{4d} + w_{5d}, \\ \text{s.t.} \quad & 0.1015w_{1d} - 0.0003w_{2d} + 0.0001w_{3d} - 0.0889w_{4d} + 0w_{5d} > 0.1017, \\ & 0.0724w_{1d} - 0.0728w_{2d} - 0.0717w_{3d} - 0.0763w_{4d} + 0.0005w_{5d} > 0, \\ & 0 \leqslant -2w_{1d} + 3w_{2d} + 8w_{3d} - w_{4d} + 5w_{5d} \leqslant 13, \end{aligned}$$

Table 3
Data obtained in the second iteration of the example

Primal variable	DMU-d				
	1	2	3	4	5
u_{1d}^*	0.0001	0.0001	0.0829	0.0509	0.0359
u_{2d}^*	0.0102	0.0030	0.0001	0.0001	0.0001
u_{3d}^*	0.0896	0.0621	0.1264	0.0777	0.0548
v_{1d}^*	0.0125	0.0179	0.2223	0.1368	0.0964
v_{2d}^*	0.0001	0.0001	0.0972	0.0600	0.0422
v_{3d}^*	0.1108	0.0729	0.0000	0.0000	0.0000
u_{0d}^*	0.1017	0	0	0	0
h_d^*	1	1	1	1	1

$$
\begin{aligned}
&0 \leqslant 6w_{1d} + 2w_{2d} + 4w_{3d} + 4w_{4d} - 3w_{5d} \leqslant 15,\\
&0 \leqslant 4w_{1d} - 2w_{2d} + 6w_{3d} - 2w_{4d} + w_{5d} \leqslant 11,\\
&0 \leqslant 3w_{1d} + 6w_{2d} + 5w_{3d} - 2w_{4d} + 3w_{5d} \leqslant 20,\\
&0 \leqslant 6w_{1d} + 7w_{2d} + 4w_{3d} + 3w_{4d} - 8w_{5d} \leqslant 21,\\
&0 \leqslant 5w_{1d} - 3w_{2d} + 8w_{3d} - 4w_{4d} + 3w_{5d} \leqslant 19,\\
&w_{1d}, w_{2d}, w_{3d}, w_{4d} \text{ and } w_{5d} \in \{0, 1\}.
\end{aligned}
$$

There is not a feasible solution. Stop.

Current alternatives give efficient solutions. That is, alternatives [1 1 1 0 0] and [1 0 1 0 1] result in efficient solutions.

All the alternatives and their resources' usage and objective satisfactoriness are listed in Table 4. There are 17 infeasible alternatives, 12 non-efficient alternatives, and 3 efficient alternatives. The efficient alternative [1 1 1 1 1] is not found in the procedure.

4. Conclusions

Resource allocation problems are close to our daily activities, whether it is personal financial management or national capital planning. Such activities involve allocating limited resources to activities while keeping conflicting objectives in mind. Each activity is either performed or not. One can formulate this problem as a multi-objective 0–1 linear program. Solution methods for multi-objective 0–1 linear programs have been developed [7]. Some computational results have been reported in the literature [4,6]. However, algorithms for large problems have not been presented previously.

This paper presents a new method for solving multi-objective 0–1 linear programs. This new method focuses on generating partially efficient solutions. The method is divided into two stages. Stage-I generates a DMU that maximizes the distance function. Stage-II then evaluates efficiency of generated and evaluated DMUs. In developing the process, we assume that all resources and objectives are nonnegative and bounded. The assumption for some objectives and resources may be not practical, however. Because of limited resources, it is hoped to utilize them efficiently. Moreover, based on the essence of DEA, this study is only concerned with public sectors. That is, the resource allocation problem solved is done in public sectors.

Development of the computer program for operational using this developed method is underway. The authors hope to compare their results further with the methods developed by Bitran [4] and Deckro and Winkofsky [6]. The authors also hope to test their algorithm with larger problems.

Finally, some future research issues addressed are summarized below. These issues include disadvantages and incomplete portions of the two-stage algorithm developed here.

1. The linear search, Step 1.3, for improving the initial alternative, may be a burdensome strategy. In the illustrative example, since the problem is too small, one cannot recognize the drawback of this strategy. The authors suspect however this procedure would significantly influence efficiency for solving larger scale problems.
2. This algorithm focuses on solving multi-objective 0–1 linear programs. It is also capable of solving multi-objective integer linear programming problems.
3. Practically speaking, the assumption of nonnegative and bounded objectives and resources may need to be relaxed. This relaxation would violate Property 2, however.
4. One may need to add *points at infinity* constraints to $H(\boldsymbol{G}'_k)$ so optimal solutions of δ_2 $(\boldsymbol{D})$ are all integers. 'Points at infinity' is one of Farrel's more awkward concepts. However, Charnes et al. [5] do not incorporate this concept in their results.

Table 4
Whole alternatives for the example

No. d	Alternative $(w_1, w_2, w_3, w_4, w_5)$	Resource usage (x_{1d}, x_{2d}, x_{3d})	Objectives satisfied (y_{1d}, y_{2d}, y_{3d})
1	(0,0,0,0,1)[a]	(5,−3,1)	(3,−8,3)
2	(0,0,0,1,0)[a]	(−1,4,−2)	(−2,3,−4)
3	(0,0,1,0,0)	(8,4,6)	(5,4,8)
4	(0,1,0,0,0)[a]	(3,2,−2)	(6,7,−3)
5	(1,0,0,0,0)[a]	(−2,6,4)	(3,6,5)
6	(0,0,0,1,1)[a]	(1,1,−1)	(1,−5,−1)
7	(0,0,1,0,1)[a]	(13,1,7)	(8,−4,11)
8	(0,1,0,0,1)[a]	(8,−1,−1)	(9,−1,0)
9	(1,0,0,0,1)[a]	(3,3,5)	(6,−2,8)
10	(0,0,1,1,0)	(7,8,4)	(3,7,4)
11	(0,1,0,1,0)[a]	(2,6,−4)	(4,10,−7)
12	(1,0,0,1,0)[a]	(−3,10,2)	(1,9,1)
13	(0,1,1,0,0)	(11,6,4)	(11,11,5)
14	(1,0,1,0,0)	(6,10,10)	(8,10,13)
15	(1,1,0,0,0)	(1,8,2)	(9,13,2)
16	(0,0,1,1,1)[a]	(12,5,5)	(8,−4,11)
17	(0,1,0,1,1)[a]	(7,3,−3)	(7,2,−4)
18	(1,0,0,1,1)	(2,7,3)	(4,1,4)
19	(0,1,1,0,1)[a]	(16,3,5)	(14,3,8)
20	(1,0,1,0,1)[b]	(11,7,11)	(11,2,16)
21	(1,1,0,0,1)	(6,3,3)	(12,5,5)
22	(0,1,1,1,0)	(10,10,2)	(9,14,1)
23	(1,0,1,1,0)	(5,14,8)	(6,13,9)
24	(1,1,0,1,0)[a]	(0,12,0)	(7,16,−2)
25	(1,1,1,0,0)[b]	(9,12,8)	(14,17,10)
26	(1,1,1,1,0)[a]	(8,16,6)	(12,20,6)
27	(1,1,1,0,1)[a]	(14,9,9)	(17,9,13)
28	(1,1,0,1,1)	(5,9,1)	(10,8,1)
29	(1,0,1,1,1)	(10,11,9)	(9,5,12)
30	(0,1,1,1,1)[a]	(15,7,3)	(12,6,4)
31	(1,1,1,1,1)[b]	(13,13,7)	(11,12,9)
32	(0,0,0,0,0)	(0,0,0)	(0,0,0)

[a] The alternative resulting in an infeasible DMU.
[b] Efficient alternative (efficient solution).

5. One may need to include a convergence analysis in this algorithm.
6. Reducing the total number of efficient solutions to a manageable size is critical for the DM. This is because a large number of efficient solutions complicate the task of selection. Some studies have examined this topic, however [10].
7. The algorithm developed here is not guaranteed to generate all efficient solutions. Future work could be focused on generating all efficient solutions.

References

[1] R.D. Banker, A. Charnes, W.W. Cooper, Some models for estimating technical and scale inefficiencies in data envelopment analysis, Management Science 30 (9) (1984) 1078–1092.
[2] M.S. Bazaraa, H.D. Sherali, On the choice of step size in subgradient optimization, European Journal of Operational Research 7 (1981) 380–388.
[3] M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear Programming, Wiley, New York, 1993.

[4] G.R. Bitran, Theory of algorithms for linear multiple objective programs with zero–one variables, Mathematical Programming 17 (1979) 362–390.
[5] A. Charnes, W.W. Cooper, E. Rhodes, Measuring the efficiency of decision making units, European Journal of Operational Research 2 (1978) 429–444.
[6] R.F. Deckro, E.P. Winkofsky, Solving zero–one multiple objective programs through implicit enumeration, European Journal of Operational Research 12 (1983) 362–374.
[7] G.W. Evans, An overview of techniques for solving multi-objective mathematical programs, Management Science 30 (11) (1984) 1268–1282.
[8] M.J. Farrel, The measurement of productive efficiency, Journal of Royal Statistical Society, Series A, 120 (1957) 253–290.
[9] K.C. Gilbert, D.D. Holmes, R.E. Rosenthal, A multi-objective discrete optimization model for land allocation, Management Science 31 (12) (1985) 1509–1522.
[10] S.B. Graves, J.L. Ringuest, J.F. Brad, Recent developments in screening methods for non-dominated solutions in multi-objective optimization, Computer and Operation Research 19 (7) (1992) 683–694.
[11] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiely, New York, 1988.
[12] H. Pasternak, V. Passy, Bicriterion mathematical programs with Boolean variables, in: J.L. Cochrane, M. Zeleny (Eds.), Multiple Criterion Decision Making, University of South Carolina Press, Columbia, SC, 1973.
[13] R. Ramanathan, L.S. Ganesh, Using AHP for resource allocation problems, European Journal of Operational Research 80 (1995) 410–417.
[14] M.J. Schniederjans, R.A. Santhanam, Multi-objective constrainted resource information system project selection method, European Journal of Operational Research 70 (2) (1993) 244–253.
[15] L.K. Sengupta, Efficiency analysis by production frontiers, in: The Nonparametric Approach, Kluwer Academic Publishers, Boston, 1989.
[16] R.W. Shephard, Theory of Cost and Production Functions, Princeton University Press, Princeton, NJ, 1970.
[17] Z. Sinuany-Stern, A network optimization model for budget planning in multi-objective hierarchical systems, Journal of the Operational Research Society 44 (3) (1993) 297–308.
[18] J.Y. Teng, G.H. Tzeng, A multiobjective programming approach for selecting non-independent transportation investment alternatives, Transportation Research B 30 (4) (1996) 291–307.