جامعة القاهرة
كلية الدراسات العليا للبحوث الإحصائية

المؤتمر السنوى الرابع والخمسون
للإحصاء وعلوم الحاسب
وبحوث العمليات

بحوث العمليات

9-11 ديسمبر 2019

Cairo University
Faculty of Graduate Studies for Statistical Research

The 54th Annual Conference on Statistics, Computer Sciences and Operations Research

Operations Research

9-11, Dec, 2019

Operations Research

A Unified Mathematical Model for Output Oriented Data Envelopment Analysis with Uncertainty

Basma E. El-Demerdash[1] Assem A. Tharwt[2] Ihab A. El-Khodary[3]

Abstract

An organization's objective is to make better decisions at all levels of the firm to improve performance. Typically, organizations are multi-faceted and complex systems that use uncertain information. Therefore, making quality decisions to improve organizational performance is a daunting task. Data Envelopment Analysis (DEA) evaluates and measures the relative efficiency of decision-making units (DMUs) that use multiple inputs and outputs to provide non-objective measures without making any specific assumptions about data. In many situations, inputs and outputs are volatile and complex so that it is difficult to measure them in an accurate way. Instead the data can be given as a stochastic and/or fuzzy variable. Therefore, the main aim of this paper is introducing a new DEA model to handle the expected uncertainty types. The develop model allows input and/or output variables to be defined as deterministic, randomness and/or vagueness depending on the nature of uncertainty in the variables. Implementation of the model was presented through the illustrative example to illustrate the model functionality. In addition, the results are compared with another different DEA model. Managers can rely on the developed model to assess relative efficiency in business complex systems associated with different uncertainty natures.

Keywords:

Data Envelopment Analysis; Efficiency Change; Uncertainty Variations; Output Oriented; Model Validation.

1. Introduction

One of the most important principles in any business is the principle of efficiency; where the best possible economic effects (outputs) are attained with as little economic sacrifices as possible (inputs). In order to assess the relative efficiency of a business unit, it is necessary to consider the conditions and operation results of other units of the same kind and to determine the real standing of the results of such a comparison. In fact, in a real evaluation problem input and output data of entities evaluated often fluctuate. These fluctuating data can be represented as linguistic variables or random variables. Over the past three decades Data Envelopment Analysis (DEA) has emerged as a useful tool for business entities and organizations to evaluate their activities. DEA evaluates the efficiencies of a set of

[1] Assistant Lecturer, Department of Operations Research and Management, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt. E-mail: basma.ezzat@cu.edu.eg.

[2] Professor, College of Business Administration, American University in the Emirates, UAE.
E-mail: assem.tharwat@aue.ae.

[3] Professor, Department of Operations Research and Decision Support, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt. E-mail: e.elkhodary@fci-cu.edu.eg.

homogenous decision-making units (DMUs) with multiple performance criteria. Data used in evaluating the efficiency usually consists of multiple inputs and multiple outputs which represent possibly conflicting criteria. DEA evaluates the relative efficiency of DMUs by using a ratio of the weighted sum of outputs to the weighted sum of inputs with the weights being variable.

The objective of DEA is to assess the relative efficiency of each DMU based on peer-group comparisons and identifies the source and the amount of inefficiency in each input relative to each output for a target DMU (El-Demerdash et al., 2013). There are some limitations of DEA that have to be considered. Because DEA is a methodology focused on frontiers, small changes in data can change efficient frontiers significantly. Therefore, to successfully apply DEA, we have to have accurate measurement of inputs and outputs. However, in some situations it is difficult to measure inputs and outputs in an accurate way to obtain precise data. Because of many important reality observations either stochastic or fuzzy in nature. As a result, DEA efficiency measurement may be sensitive to such variations Cooper et al. (2011). A DMU which is rated as efficient relative to other DMUs may turn inefficient if such uncertainty variations are considered, or vice versa. In another word, if the collected data for a variable are not represented in the correct form nature, then the resulting efficiencies will be erroneous and misleading because of a high sensitivity of the efficiency scores to the realized levels of inputs or outputs.

The rest of the paper is organized as follows. Section 2 includes the literature review. Section 3 includes the proposed unified output-oriented DEA model that handles deterministic variables and variables with different uncertainty variations. Section 4 includes an illustrative example and model validation. The paper is ending with section 5 that includes the conclusion.

2. Literature Review

Many efforts that have been made recently in DEA model to handle the randomness or vagueness in data. Therefore, in this section, we provide in the first section a brief review of the most widely cited literature relevant to chance-constrained programming approach that is used for solving stochastic DEA models. Moreover, in the second section, the α-cut approach that is used for solving fuzzy DEA models.

2.1. Data Envelopment Analysis under Stochastic Environment

For handling stochastic data, Land et al. (1993) extended the output-oriented DEA approach to the case of all outputs have a random nature using chance-constrained programming while all inputs are deterministic. Olesen and Petersen (1995) used chance-constrained DEA to evaluate efficiency when there exists an unknown amount of noise in all data. Cooper et al. (2004) extended the output-oriented DEA model by use chance-constrained method for treating congestion in DEA through dealing with all inputs and outputs to have random variation. Desai et al. (2005) proposed a chance-constrained DEA model that allows all output variables have random variations, but all input variables are deterministic. Talluri et al. (2006) proposed a model for vendor selection which a stochastic input-oriented DEA model that considered all output variables are random in nature and all input variables are deterministic. Razavyan and Tohidi (2008) proposed the output-oriented DEA method for ranking stochastic efficient of DMUs, by defining the efficient DMU via joint randomness comparisons of all variables with other comparable DMUs.

Khodabakhshi (2010) developed an output-oriented super-efficiency stochastic DEA model that considers all data variables have random in nature. El-Khodary, et al. (2010) developed a stochastic input-oriented DEA model assume that all input variables are random in nature and all outputs are deterministic. Wu et al. (2012) proposed a stochastic output-oriented DEA model to deals with the existence of random errors in the collected data. El-Demerdash et al. (2013) developed an algorithm to help any organization for evaluating their performance given that some inputs are stochastic. The developed algorithm is for a stochastic input-oriented DEA model based on the chance-constrained programming. El-Demerdash et al. (2014) used modified stochastic input-oriented DEA model that conducted by El-Demerdash et al. (2013) for the comparison of evaluating the relative efficiency scores of Faculties of Computers and Information in Egypt. Azadeh, A., et al. (2015) developed a stochastic output-oriented DEA model that have all observations are random in nature and applied for assessment of Iranian electricity distribution units from 2001 to 2011. El-Demerdash et al. (2016) developed stochastic input-oriented DEA that assumed that some of input/output variables are random in nature and remining variables are deterministic. Liu et al. (2017) developed two stochastic output-oriented DEA model to obtain the upper and lower bounds of the quantile efficiency under a constraint and then achieve an interval efficiency evaluation.

It is evident that the developed stochastic DEA models adopted either original or output-oriented DEA model, but still very few considered input-oriented DEA model. It is noticed that, available stochastic DEA models consider either all output variables or all input variables or all variables as random in nature, although some might have a deterministic in nature. In addition to, they were considered the stochastic input/output variables are identically normal distributed and pairwise independent.

2.2. Data Envelopment Analysis under Fuzzy Environment

There are many papers that have been made in the DEA models to handle the vagueness in all data. The α-cut DEA approach was proposed by Girod (1996) and it is perhaps the most popular fuzzy DEA model. Kao and Liu (2000) developed a method to find the membership functions of the fuzzy efficiency measures when the output observations are fuzzy numbers. The idea was based on the α-cuts principle to transform a fuzzy DEA model to a family of crisp DEA models. Entani et al. (2002) proposed the output-oriented fuzzy DEA model that considers all data have fuzzy in nature used α-cut approach with an interval efficiency consisting of the efficiencies obtained from the pessimistic and the optimistic viewpoints. Liu et al. (2007) developed a modified fuzzy output-oriented DEA model used α-level approach to handle fuzzy in all observations and incomplete information on weight indices in product design evaluation. Chiang and Che (2010) proposed a weight-restricted fuzzy DEA ranking methodology using α-level approach by considers all data have fuzzy in nature. Zerafat et al. (2010) proposed a fuzzy DEA model used α-level approach to retain fuzziness of the model by maximizing the membership functions of all fuzzy variables.

Khoshfetrat and Daneshvar (2011) developed a fuzzy CCR DEA model using the α-cut approach, to transform the given fuzzy data to interval numbers and then computed lower bounds of fuzzy inputs and outputs for each factor weight. Azadeh et al. (2011) proposed a flexible approach that composed of fuzzy DEA and artificial neural network to applied on location optimization of solar plants when considers all data are fuzzy. Zerafat et al. (2012) proposed a fuzzy DEA model that considers some of the input and output variables have fuzzy uncertainty information from the intervals within the α-cut approach. Hatami-Marbini et al. (2016) applied fuzzy DEA model used an α-cut approach developed by Saati et al. (2002) for identifying supplier performance. Hatami-Marbini et al. (2017) proposed a fuzzy output-oriented DEA model that considers the vagueness of information to identify supplier performance that can be a flexible cross-efficiency evaluation methodology. Tharwat et al. (2019) proposed input-oriented DEA model by assuming some of input/output variables are fuzzy in nature and remining variables are deterministic. From surveying the literature, we reached main conclusion. The available fuzzy DEA models consider all output and/or input variables as fuzzy in nature, although some might have a deterministic in nature.

3. A Developed Unified Output Oriented DEA Model with Uncertainty Variations

In real world problems, exact data may not always be available due to the existence of uncertainty and we needed to evaluate the performance of any comparable institutions to assure the quality given that some of the input and /or output variables might have as uncertainty in nature. So, it was necessary to develop a model to deal with this case. There are basically two main types of DEA models: a constant return to scale (CRS) model, there is a direct proportion between changing outputs and changing inputs and variables return to scale (VRS) model, changing inputs may not affect outputs in direct proportion. Therefore, in this section, we aim to develop a unified output-oriented VRS DEA model to handle different uncertainty types of variables. The proposed model allows some input and/output variables to be uncertainty (stochastic/fuzzy) in nature while keeping other variables deterministic. In this section we had concern about output-oriented DEA model which which defines the frontier by seeking the maximum proportional increase in output production, with input levels held fixed, for each DMU. the output oriented VRS DEA model for measuring the efficiency level of p^{th} *DMU* is as follow:

$$
\begin{aligned}
& Max \quad Z_p = \emptyset \\
\text{s.t.} \quad & \sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} \quad , \forall k = 1 \dots s \\
& \sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \quad , \forall j = 1 \dots m \qquad\qquad (\boldsymbol{M-1}) \\
& \sum_{i=1}^{n} \lambda_i = 1 \\
& \lambda_i \geq 0, (i = 1,2, \dots, n)
\end{aligned}
$$

where $\emptyset$ represents efficiency level for p^{th} DMU; k = 1 to 's' (no. of outputs); j = 1 to 'm' (no. of inputs); i = 1 to 'n' (no. of DMUs); y_{ik} = amount of output k produced by DMU i; x_{ij} = amount of input j utilized by DMU i; λ_i: weight given to DMU i.

Proposition 1: Assume that some of the input and/or output observations are random variables, the equivalent chance-constraint unified output-oriented DEA model for measuring the efficiency level of p^{th} DMU for the model *(M - 1)* is presented as below:

$$Max\ Z_p = \emptyset$$

$$s.t.$$

$$pr\left\{\sum_{i=1}^{n}\lambda_i y_{ik} \geq \emptyset y_{pk}\right\} \geq (1-\alpha_k) \qquad ,\forall k \in K_S$$

$$\sum_{i=1}^{n}\lambda_i y_{ik} \geq \emptyset y_{pk} \qquad ,\forall k \in K_D$$

$$pr\left\{\sum_{i=1}^{n}\lambda_i x_{ij} \leq x_{pj}\right\} \geq (1-\alpha_j) \qquad ,\forall j \in J_S \qquad (\boldsymbol{M-2})$$

$$\sum_{i=1}^{n}\lambda_i x_{ij} \leq x_{pj} \quad ,\forall j \in J_D$$

$$\sum_{i=1}^{n}\lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2,\ldots,n)$$

Where α_j: significance level for input j, α_k: significance level for output k, J_D is the set of deterministic inputs, J_S is the set of stochastic inputs, J is the set of all inputs, where $J_D \cup J_s = J$ and K_D is the set of deterministic outputs, K_s is the set of stochastic outputs, and K set of all outputs, where $K_D \cup K_s = K$. For the detailed proof see Land et al. (1993).

Proposition 2: Assume that the random output variable $(y_{ik} \in K_s)$ are identically normal distributed and pairwise dependent $(cov(y_{ik}, y_{pk}) \neq 0)$, then the equivalent deterministic nonlinear model for the unified output oriented VRS DEA model for the model presented in (*M-2*) as below:

$$Max\ Z_p = \emptyset$$

$$s.t.$$

$$\sum_{i=1}^{n}\lambda_i y_{ik} \geq \emptyset y_{pk} \qquad ,\forall k \in K_D$$

$$\sum_{i=1}^{n}\lambda_i \mu_{ik} - \emptyset\mu_{pk} \geq e_k\sqrt{\left(\lambda_p - \emptyset\right)^2\sigma_{pk}^2 + \sum_{\substack{i=1\\ i\neq p}}^{n}\lambda_i^2\sigma_{ik}^2 + 2\lambda_i\lambda_p cov(y_{ik}, y_{pk})}\ ,\forall k \in K_s$$

$$\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \quad , \forall j \in J_D$$

$$pr\left\{\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj}\right\} \geq (1 - \alpha_j) \qquad , \forall j \in J_S \qquad (\boldsymbol{M-3})$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2, \dots, n)$$

Where μ_{ik}: mean of output *k* for DMU *i*, σ_{ik}^2: variance of output *k* for DMU *i*.

Proof: assume that each output $y_{ik} \in K_s$ is identically normal distributed with a mean μ_{ik} and variance σ_{ik}^2. Further, assume that they are pairwise dependent ($cov(y_{ik}, y_{pk}) \neq 0$). Now, define the random variable for all output $k \in K_s$

$$u_k = \sum_{i=1}^{n} \lambda_i y_{ik} - \emptyset y_{pk} \qquad (1)$$

with mean:

$$E(u_k) = \sum_{i=1}^{n} \lambda_i \mu_{ik} - \emptyset \mu_{pk} \equiv \mu_{uk} \qquad (2)$$

and variance:

$$var(u_k) = \left(\lambda_p - \emptyset\right)^2 \sigma_{pk}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ik}^2 + 2\lambda_i \lambda_p cov\left(y_{ik}, y_{pk}\right) \equiv \sigma_{u_k}^2 \qquad (3)$$

Since, $y_{ik}'s$ are normally distributed, so is variable u_k,

$$z_k = \frac{u_k - \mu_{u_k}}{\sigma_{u_k}} \qquad (4)$$

Hence,

$$pr\left\{\sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk}\right\} = pr\{u_k \geq 0\} = pr\left\{z_k \geq \frac{-\mu_{u_k}}{\sigma_{u_k}}\right\} \qquad (5)$$

Given the symmetry property of normal distribution,

$$pr\left\{z_k \geq \frac{-\mu_{u_k}}{\sigma_{u_k}}\right\} = pr\left\{z_k \leq \frac{\mu_{u_k}}{\sigma_{u_k}}\right\} = \varphi\left(\frac{\mu_{u_k}}{\sigma_{u_k}}\right) \qquad (6)$$

where $\varphi(\,)$ is the cumulative standard distribution function. The random inequality restriction in the chance-constrained DEA problem can be replaced by the equivalent restriction (Land et al. (1993)).

$$\varphi\left(\frac{\mu_{u_k}}{\sigma_{u_k}}\right) \geq (1-\alpha_k) \tag{7}$$

$$\varphi\left(\frac{\mu_{u_k}}{\sigma_{u_k}}\right) \geq \varphi(e_k) \tag{8}$$

$\varphi(e)$ is obtained from the table of standard normal distribution. Hence the last equation can be written as,

$$\mu_{u_k} \geq e\sigma_{u_k} \tag{9}$$

Substitute equations (2) and (3) in equation (9)

$$\sum_{i=1}^{n} \lambda_i \mu_{ik} - \emptyset \mu_{pk} \geq e_k \sqrt{\left(\lambda_p - \emptyset\right)^2 \sigma_{pk}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ik}^2 + 2\lambda_i \lambda_p cov_{(y_{ik}, y_{pk})}} \tag{10}$$

Proposition 3: Assume that the random input variable ($x_{ij} \in J_s$) are identically normal distributed and pairwise dependant ($cov_{(x_{ij}, x_{pj})} \neq 0$), then the equivalent deterministic nonlinear model for the unified output oriented DEA model presented in the model (*M*-*3*) is as:

$$Max\ Z_p = \emptyset$$

$$s.t.$$

$$\sum_{i=1}^{n} \lambda_i \mu_{ik} - \emptyset \mu_{pk} \geq e_k \sqrt{\left(\lambda_p - \emptyset\right)^2 \sigma_{pk}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ik}^2 + 2\lambda_i \lambda_p cov_{(y_{ik}, y_{pk})}} \quad , \forall k \in K_s$$

$$\sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} \quad , \forall k \in K_D \qquad (\boldsymbol{M-4})$$

$$\sum_{i=1}^{n} \lambda_i \mu_{ij} - \mu_{pj} \leq e_j \sqrt{\lambda_p^2 \sigma_{pj}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ij}^2 + 2\lambda_i \lambda_p cov_{(x_{ij}, x_{pj})}} \quad , \forall j \in J_S$$

$$\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \quad , \forall j \in J_D$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2, \dots, n)$$

Where μ_{ij}: mean of input *j* for DMU *i*, σ_{ij}^2: variance of input *j* for DMU *i*.

Proof: assume that each input $x_{ij} \in J_s$ is identically normal distributed with a mean μ_{ij} and variance σ_{ij}^2. Further, assume that they are pairwise dependent ($cov_{(x_{ij}, x_{pj})} \neq 0$). Now, define the random variable for all input $j \in J_S$

$$u_j = \sum_{i=1}^{n} \lambda_i x_{ij} - x_{pj} \tag{11}$$

with mean:

$$E(u_j) = \sum_{i=1}^{n} \lambda_i \mu_{ij} - \mu_{pj} \equiv \mu_{u_j} \tag{12}$$

and with variance:

$$var(u_j) = \lambda_p^2 \sigma_{pj}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ij}^2 + 2\lambda_i \lambda_p cov(x_{ij}, x_{pj}) \equiv \sigma_{u_j}^2 \tag{13}$$

Since the $x_{ij}'s$ are normally distributed with a mean μ_{u_j} and variance $\sigma_{u_j}^2$, therefore the variable u can be transformed into its equivalent standardized normal value z_j, as follows:

$$z_j = \frac{u_j - \mu_{u_j}}{\sigma_{u_j}} \tag{14}$$

Hence,

$$pr\left\{\sum_{i=1}^{m} \lambda_i x_{ij} \leq \theta x_{pj}\right\} = pr\{u_j \leq 0\} = pr\left\{z_j \leq \frac{-\mu_{u_j}}{\sigma_{u_j}}\right\} \tag{15}$$

Given the symmetric property of the normal distribution, then:

$$pr\left\{z_j \leq \frac{-\mu_{u_j}}{\sigma_{u_j}}\right\} = pr\left\{z_j \geq \frac{\mu_{u_j}}{\sigma_{u_j}}\right\} = 1 - \varphi\left(\frac{\mu_{u_j}}{\sigma_{u_j}}\right) \tag{16}$$

where $\varphi(\,)$ is the cumulative standard distribution function.

The random inequality restriction in the chance-constrained DEA problem can be replaced by the equivalent restriction (Land et al. (1993)):

$$1 - \varphi\left(\frac{\mu_{u_j}}{\sigma_{u_j}}\right) \geq (1 - \alpha_j) \tag{17}$$

$$\varphi\left(\frac{\mu_{u_j}}{\sigma_{u_j}}\right) \leq \varphi(e_j) \tag{18}$$

$\varphi(e)$ is obtained from the table of standard normal distribution. Hence eq. (18) can be written as

$$\mu_{u_j} \leq e_j \sigma_{u_j} \tag{19}$$

Substitute eq. (12) and (13) in eq. (19). i.e.,

$$\sum_{i=1}^{n} \lambda_i \mu_{ij} - \mu_{pj} \leq e_j \sqrt{\lambda_p^2 \sigma_{pj}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ij}^2 + 2\lambda_i \lambda_p cov(x_{ij}, x_{pj})} \tag{20}$$

Proposition 4: Assume that some of the input and/or output observations are linguistic variables, the equivalent fuzzy output-oriented VRS DEA model for measuring the efficiency level of p^{th} *DMU* for the model *(M - 1)* is presented as below:

$$Max \quad \tilde{Z}_p = \emptyset$$

$s.t.$

$$\sum_{i=1}^{n} \lambda_i \tilde{y}_{ik} \geq \emptyset \tilde{y}_{pk} \qquad , \forall k \in K_F$$

$$\sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} \qquad , \forall k \in K_D$$

$$\sum_{i=1}^{n} \lambda_i \tilde{x}_{ij} \leq \tilde{x}_{pj} \qquad , \forall j \in J_F \qquad \qquad (\boldsymbol{M-5})$$

$$\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \qquad , \forall j \in J_F$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2, \dots, n)$$

Where $\tilde{x}_{ij}$: fuzzy number for input j utilized by DMU i, $\tilde{y}_{ik}$: fuzzy number for output k produced by DMU i, J_D is the set of deterministic inputs, J_F is the set of fuzzy inputs, J is the set of all inputs, where $J_D \cup J_F = J$ and K_D is the set of deterministic outputs, K_F is the set of fuzzy outputs, and K set of all outputs, where $K_D \cup K_F = K$. For the detailed proof see Girod (1996).

Proposition 5: Suppose that the linguistic output variables ($\tilde{y}_{ik} \in K_F$) are follow triangular membership function, then the equivalent crisp model for output-oriented VRS DEA model presented in the model (*M-5*) is as:

$$Max \quad \tilde{Z}_p = \emptyset$$

$s.t.$

$$\sum_{i=1}^{n} \lambda_i \tilde{y}_{ik} \geq \emptyset \tilde{y}_{pk} \qquad , \forall k \in K_F$$

$$\alpha y_{ik}^M + (1-\alpha) y_{ik}^L \leq \tilde{y}_{ik} \leq \alpha y_{ik}^M + (1-\alpha) y_{ik}^U \qquad , \forall k \in K_F, i = 1,2, \dots, n$$

$$\sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} \qquad , \forall k \in K_D$$

$$\sum_{i=1}^{n} \lambda_i \tilde{x}_{ij} \leq \tilde{x}_{pj} \qquad , \forall j \in J_F \qquad \qquad (\boldsymbol{M-6})$$

$$\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \qquad , \forall j \in J_F$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2, \dots, n)$$

Where α: α-cut level for linguistic output variables, y_{ik}^L: the lower value of linguistic output variable k produced by DMU i, y_{ik}^M: median value of linguistic output variable k produced by DMU i, y_{ik}^U: the upper value of the linguistic output variable k produced by DMU i

Proof: For model (*M-5*) assumed that a triangular membership function for the linguistic numbers that are used for expressing linguistic outputs as follows:

$$\mu_{\tilde{y}_{ik}} \begin{cases} 0 & , y_{ik} \leq y_{ik}^L \\ \frac{y_{ik}-y_{ik}^L}{y_{ik}^M-y_{ik}^L} & , y_{ik}^L \leq y_{ik} \leq y_{ik}^M \\ \frac{y_i^U-y_{ik}}{y_{ik}^U-y_{ik}^M} & , y_{ik}^M \leq y_{ik} \leq y_{ik}^U \\ 0 & , y_{ik} \geq y_{ik}^U \end{cases} \tag{21}$$

$$\tilde{y}_{ik} = \left(y_{ik}^L, y_{ik}^M, y_{ik}^U\right), \quad 0 \leq y_{ik}^L \leq y_{ik}^M \leq y_{ik}^U \rightarrow \tilde{y}_{ik} \in [y_{ik}^L, y_{ik}^U] \tag{22}$$

To find α-cuts of $\tilde{y}_{ik}$, arithmetical operations on triangular linguistic numbers are defined in eq. 21. With this operation, an interval that shows lower and upper bounds in different α-levels is found. Application of α-cut interval operations to linguistic outputs are as follows:

$$\mu_{\tilde{y}_{ik}} \geq \alpha \begin{cases} \frac{y_{ik}-y_{ik}^L}{y_{ik}^M-y_{ik}^L} \geq \alpha \\ \frac{y_{ik}^U-y_{ik}}{y_{ik}^U-y_{ik}^M} \geq \alpha \end{cases} \tag{23}$$

$$\tilde{y}_{ik} \in [\alpha y_{ik}^M + (1-\alpha) y_{ik}^L, \alpha y_{ik}^M + (1-\alpha) y_{ik}^U\ ] \tag{24}$$

Proposition 6: Suppose that the linguistic input variables ($\tilde{x}_{ij} \in J_F$) are follow triangular membership function, then the equivalent crisp model for the output-oriented VRS DEA model using α-cut approach presented in the model (*M-6*) is as:

$$\begin{aligned}
& Max\ \left(\tilde{Z}_p\right)_\alpha = \emptyset \\
& s.t. \\
& \sum_{i=1}^{n} \lambda_i \tilde{y}_{ik} \geq \emptyset \tilde{y}_{pk} && , \forall k \in K_F \\
& \alpha y_{ik}^M + (1-\alpha) y_{ik}^L \leq \tilde{y}_{ik} \leq \alpha y_{ik}^M + (1-\alpha) y_{ik}^U && , \forall k \in K_F, i = 1,2,\dots,n \\
& \sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} && , \forall k \in K_D && (\boldsymbol{M-7}) \\
& \sum_{i=1}^{n} \lambda_i \tilde{x}_{ij} \leq \tilde{x}_{pj} && , \forall j \in J_F \\
& \alpha x_{ij}^M + (1-\alpha) x_{ij}^L \leq \tilde{x}_{ij} \leq \alpha x_{ij}^M + (1-\alpha) x_{ij}^U && , \forall j \in J_F, i = 1,2,\dots,n \\
& \sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} && , \forall j \in J_D
\end{aligned}$$

$$\sum_{i=1}^{n} \lambda_i = 1$$
$$\lambda_i \geq 0, (i = 1,2, \dots, n)$$

Where α: α-cut level for linguistic input variables j, x_{ij}^{L}: the lower value of input linguistic variable j utilized by DMU i, x_{ij}^{M}: median value of input linguistic variable j utilized by DMU i, x_{ij}^{U}: the upper value of input linguistic variable j utilized by DMU i.

Proof: For model (*M-6*) assumed that a triangular membership function for the linguistic input numbers that are used for expressing linguistic inputs as follows:

$$\mu_{\tilde{x}_{ij}} \begin{cases} 0 & , x_{ij} \leq x_{ij}^{L} \\ \frac{x_{ij}-x_{ij}^{L}}{x_{ij}^{M}-x_{ij}^{L}}, & x_{ij}^{L} \leq x_{ij} \leq x_{ij}^{M} \\ \frac{x_{ij}^{U}-x_{ij}}{x_{ij}^{U}-x_{ij}^{M}}, & x_{ij}^{M} \leq x_{ij} \leq x_{ij}^{U} \\ 0 & , x_{ij} \geq x_{ij}^{U} \end{cases} \tag{25}$$

$$\tilde{x}_{ij} = \left(x_{ij}^{L}, x_{ij}^{M}, x_{ij}^{U}\right), \quad 0 \leq x_{ij}^{L} \leq x_{ij}^{M} \leq x_{ij}^{U} \rightarrow \tilde{x}_{ij} \in \left[x_{ij}^{L}, x_{ij}^{U}\right] \tag{26}$$

To find α-cuts of $\tilde{x}_{ij}$ arithmetical operations on triangular linguistic numbers are defined in eq. 25. With this operation, an interval that shows lower and upper bounds in different α-levels is found. Application of α-cut interval operations to linguistic inputs are as follows:

$$\mu_{\tilde{x}_{ij}} \geq \alpha \begin{cases} \frac{x_{ij}-x_{ij}^{L}}{x_{ij}^{M}-x_{ij}^{L}} \geq \alpha \\ \frac{x_{ij}^{U}-x_{ij}}{x_{ij}^{U}-x_{ij}^{M}} \geq \alpha \end{cases} \tag{27}$$

$$\tilde{x}_{ij} \in \left[\alpha x_{ij}^{M} + (1-\alpha)\, x_{ij}^{L}, \alpha x_{ij}^{M} + (1-\alpha)\, x_{ij}^{U}\right] \tag{28}$$

Proposition 7: Assume that the set of input/output variables may include deterministic, stochastic, and fuzzy variables, consider the output oriented VRS case, then the unified output oriented VRS DEA model be presented as follows:

$$Max\ \left(\tilde{Z}_p\right)_{\alpha} = \emptyset$$
$$s.t.$$
$$\sum_{i=1}^{n} \lambda_i y_{ik} \geq \emptyset y_{pk} \qquad , \forall k \in K_D$$
$$\sum_{i=1}^{n} \lambda_i \mu_{ik} - \emptyset \mu_{pk} \geq e_k \sqrt{\left(\lambda_p - \emptyset\right)^2 \sigma_{pk}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ik}^2 + 2\lambda_i \lambda_p cov(y_{ik}, y_{pk})} \qquad , \forall k \in K_S$$

$$\sum_{i=1}^{n} \lambda_i \tilde{y}_{ik} \geq \emptyset \tilde{y}_{pk} \qquad ,\forall k \in K_F \qquad (\boldsymbol{M-8})$$

$$\alpha y_{ik}^{M} + (1-\alpha) y_{ik}^{L} \leq \tilde{y}_{ik} \leq \alpha y_{ik}^{M} + (1-\alpha) y_{ik}^{U} \qquad ,\forall k \in K_F, i = 1,2,\dots,n$$

$$\sum_{i=1}^{n} \lambda_i x_{ij} \leq x_{pj} \quad ,\forall j \in J_D$$

$$\sum_{i=1}^{n} \lambda_i \mu_{ij} - \mu_{pj} \leq e_j \sqrt{\lambda_p^2 \sigma_{pj}^2 + \sum_{\substack{i=1 \\ i \neq p}}^{n} \lambda_i^2 \sigma_{ij}^2 + 2\lambda_i \lambda_p cov(x_{ij}, x_{pj})} \quad ,\forall j \in J_F$$

$$\sum_{i=1}^{n} \lambda_i \tilde{x}_{ij} \leq \tilde{x}_{pj} \quad ,\forall j \in J_F$$

$$\alpha x_{ij}^{M} + (1-\alpha) x_{ij}^{L} \leq \tilde{x}_{ij} \leq \alpha x_{ij}^{M} + (1-\alpha) x_{ij}^{U} \qquad ,\forall j \in J_F, i = 1,2,\dots,n$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

$$\lambda_i \geq 0, (i = 1,2,\dots,n).$$

Where $J_D \cup J_S \cup J_F = J$ and $K_D \cup K_S \cup K_F = K$.

Proof: For proposing a unified output oriented VRS DEA model that allow input/output variables to be defined as deterministic, randomness and/or vagueness depending on the nature of uncertainty in the variables. The first constraint in model *(M-1)* is responsible for deterministic output variables in nature, in the model (*M-4*) represent the equivalent nonlinear deterministic for the constraint after applied chance constraint approach to be responsible for stochastic output variables in nature. Moreover, in the model (*M-7*) represent the equivalent linear crisp for the constraint after applied α-cut approach to be responsible for fuzzy output variables in nature.

The second constraint in model (*M-1)* is responsible for deterministic input variables in nature, in the model (*M-4*) represent the equivalent nonlinear deterministic for the constraint after applied chance constraint approach to be responsible for stochastic input variables in nature. Moreover, in the model (*M-7*) represent the equivalent linear crisp for the constraint after applied α-cut approach to be responsible for fuzzy input variables in nature.

4. Model Implementation

In this section, implementation of the developed unified output-oriented VRS DEA model (see *M-8*) to illustrate the functionality of model.

4.1. Illustrative Example

The proposed problem at hand discuss the performance evaluation of seven DMUs with three input variables, the nature of the first two is deterministic while the third one is assumed to be fuzzy. In the other hand there are two output variables with deterministic and stochastic nature respectively. The values presentations for the proposed variables are presented in tables 1-3. Furthermore, the α-cut level for the fuzzy variable is considered 0.5, while the level of significance for stochastic variable is 5%.

Table 1 Data for the deterministic variables for the DMUs

DMU	Inputs		Output
	Input 1	Input 2	Output 1
A	6.11	4.36	0.21
B	3.66	2.54	0.12
C	1.44	0.48	0.14
D	1.21	0.23	0.10
E	2.75	1.40	0.10
F	4.18	2.74	0.06
G	6.39	3.36	0.18

Table 2 Data for the fuzzy variables for the DMUs

DMU	Input 3		
	Lower	Middle	Upper
A	1.76	7.27	12.27
B	3.85	4.65	5.53
C	1.33	1.88	3.38
D	0.78	1.48	2.06
E	3.22	3.63	4.61
F	4.30	6.13	8.03
G	4.40	8.00	10.68

Table 3. Data for the stochastic variable for the DMUs

DMU	Output 2				
	μ	σ^2	***Covariances***		
A	0.19	0.08	$cov(y_A, y_B)$ =0.006	$cov(y_B, y_D) = 0.006$	$cov(y_C, y_G)$ =0.004
B	0.11	0.12	$cov(y_A, y_C) = 0.003$	$cov(y_B, y_E) = 0.006$	$cov(y_D, y_E) = 0.004$
C	0.10	0.06	$cov(y_A, y_D) = 0.004$	$cov(y_B, y_F) = 0.006$	$cov(y_D, y_F) = 0.004$
D	0.07	0.08	$cov(y_A, y_E) = 0.004$	$cov(y_B, y_G) = 0.01$	$cov(y_D, y_G) = 0.006$
E	0.09	0.08	$cov(y_A, y_F) = 0.004$	$cov(y_C, y_D)$ =0.003	$cov(y_E, y_F) = 0.004$
F	0.07	0.08	$cov(y_A, y_G)$ =0.006	$cov(y_C, y_E)$ =0.003	$cov(y_E, y_G) = 0.006$
G	0.18	0.12	$cov(y_B, y_C) = 0.004$	$cov(y_C, y_F)$ =0.003	$cov(y_F, y_G) = 0.006$

To evaluate the performance through the relative efficiency for each DMU, the developed model has been implemented for solving the problem under investigation. GAMS (General Algebraic Modeling System) programming language software is used to solve the constructed formulation based on the given values. The relative efficiency levels for DMUs are shown in Table 4, the results reveal that **DMUs *C, D,*** and ***E*** are efficient, while **DMUs *A, B, F,*** and ***G*** are inefficient

Table 4 Relative efficiency level for each DMU for developed model

DMU	Unified output Oriented DEA Model Efficient Set	DMU	Unified output Oriented DEA Model Inefficient Set
C	1	A	0.28
D	1	B	0.83
E	1	F	0.50
		G	0.55

4.2. Result Validation

Since, DEA is a nonparametric technique, statistical hypothesis tests are difficult to conduct, and assessing the strength or fit of the resulting model is correspondingly difficult. However, the devolved model must be validated. The most approach that used by various researchers for validating their developed DEA models is the comparison with dissimilar DEA models (Azadeh et al. (2015) and Wang & Chin (2011)). From the literature review, it is evident that there is no other DEA model handled two different natures of input/output variables. Therefore, the current illustrative example will be compared with the relative efficiency that obtained of with a deterministic DEA model in which all the variables are deterministic. To implement the select model, a difuzzification for the third input variable and distcochasication for the second output variable are done. The models are constructed and solved and the obtained relative efficiency levels for each DMU are shown in Table 5.

Table 5 Relative efficiency level for each DMU for deterministic DEA Model

DMU	Deterministic DEA Model Efficient Set	DMU	Deterministic DEA Model Inefficient Set
A	1	B	0.70
C	1	E	0.64
D	1	F	0.39
G	1		

Investigating the results shown in Table 6, it is evident that, efficiency results have been changed significantly across models. The ***DMUs C and D*** are efficient in the two models, while the ***DMUs B and F*** are inefficient in the two model with sever inefficiency levels obtained by the developed model. In addition, the three remaining ***DMUs*** have different scenarios, the ***DMUs A and G*** transferred from the efficiency side to the inefficiency one, while the last ***DMU E*** considered an efficient unit based on the new model. In conclusion, the model assumption regarding the nature of non-deterministic variables significantly affects the resulting efficiency. This will lead to a completely different interpretation for the results which will affect the remedy planning for the inefficient units. Therefore, it is recommended to clearly determine the nature of each variable before implementing the model to reduce the misleading results which influenced by the obtained relative efficiency levels.

Table 6 Relative efficiency level for each DMU for the two DEA Model

DMU	Developed DEA Model Efficient levels	Deterministic DEA Model Efficient levels
A	0.28	1
B	0.83	0.70
C	1	1
D	1	1
E	1	0.64
F	0.50	0.39
G	0.55	1

5. Conclusion

In real-life problems, values for input and/or output variables include uncertainty, this uncertainty may be randomness or vagueness in nature. Therefore, we have proposed a unified DEA model to conduct performance assessments in uncertainty environments that able to deal with some of variables either input/output have vagueness or randomness in nature and the remaining variables are deterministic in nature. The model is tailored for fuzzy variables with triangular membership functions and solved using the α-level approach and for stochastic variables with chance constraints programming approach.

Through the illustrative example provided, wrong assumptions for variables nature significance affect model result. A DMU which is rated as efficient relative to other DMUs may turn inefficient if such uncertainty variations are considered, or vice versa. In another word, if the collected data for a variable are not represented in the correct form nature, then the resulting efficiencies will be erroneous and misleading because of a high sensitivity of the efficiency scores to the realized levels of inputs or outputs. Therefore, it is necessary to identify the nature of the variables to apply the appropriate DEA model and achieving reliable results.

References

Azadeh, Ali and Alem, Seyed Mostafa, (2010). “A flexible deterministic, stochastic and fuzzy Data Envelopment Analysis approach for supply chain risk and vendor selection problem: Simulation analysis”. Expert Systems with Applications. 37(12): 7438-7448.

Azadeh, A., Sheikhalishahi, M., and Asadzadeh, S.M., (2011). “A flexible neural network-fuzzy data envelopment analysis approach for location optimization of solar plants with uncertainty and complexity”. Renewable Energy. 12: 3394-3401.

Azadeh, A., S. Motevali Haghighi, M. Zarrin, and S. Khaefi., (2015). "Performance evaluation of Iranian electricity distribution units by using stochastic data envelopment analysis. International Journal of Electrical Power & Energy Systems. 73: 919-931.

Chiang, Tzu-An and Che, ZH, (2010). “A fuzzy robust evaluation model for selecting and ranking NPD projects using Bayesian belief network and weight-restricted DEA”. Expert Systems with Applications. 37(11): 7408-7418.

Cooper, William W., Honghui Deng, Zhimin Huang, and Susan X. Li., (2004). “Chance constrained programming approaches to congestion in stochastic data envelopment analysis”. European Journal of Operational Research. 155(2): 487-501.

Cooper, W. W., Seiford, L. M., Zhu, J., (2011). “Data envelopment analysis: history, models, and interpretations”. In Handbook on data envelopment analysis, Springer US.

Desai, A., Ratick, S. J., & Schinnar, A. P., (2005). “Data envelopment analysis with stochastic variations in data”. Socio-Economic Planning Sciences. 39(2), 147-164.

El-Demerdash, B. E., El-Khodary, I. A., and Tharwat, A. A., (2013). “Developing a Stochastic Input Oriented Data Envelopment Analysis (SIODEA) Model”. International Journal of Advanced Computer Science & Applications. 4(4): 40-44.

El-Demerdash, B. E., El-Khodary, I. A., and Tharwat, A. A., (2013). “Developing a Stochastic Input Oriented Data Envelopment Analysis (SIODEA) Model”. International Journal of Advanced Computer Science & Applications. 4(4): 40-44

El-Demerdash, Basma E and El-Khodary, Ihab A and Tharwat, Assem A., (2014). “Performance evaluation for Egyptian's Computers and Information Faculties: Using a Stochastic Input Oriented Data Envelopment Analysis Model”. International Journal of Computer Applications. 89(5).

El-Demerdash, Basma E., Ihab A. El-Khodary, and Assem A. Tharwat., (2016). "A Stochastic Data Envelopment Analysis Model Considering Variation in Input and Output Variables". International Journal of Data Envelopment Analysis and *Operations Research*. 2(1): 1-6.

El-Khodary, I., El-Demerdash, B., and Tharwat, A. (2010) “An Algorithm for Evaluating the Performance of Higher Education Organizations in Egypt Using a Stochastic DEA” Performance Management and Measurement with Data Envelopment Analysis. 75-82.

Egilmez, Gokhan and Gumus, Serkan and Kucukvar, Murat and Tatari, Omer, (2016). “A fuzzy data envelopment analysis framework for dealing with uncertainty impacts of input-output life cycle assessment models on eco-efficiency assessment”. Journal of cleaner production. 129: 622-636.

Entani, T., Maeda, Y., and Tanaka, H., (2002). “Dual models of interval DEA and its extension to interval data”. European Journal of Operational Research, 1: 32-45.

Girod, O.A., (1996) “Measuring technical efficiency in a fuzzy environment” Doctoral dissertation, Virginia Polytechnic Institute and State University.

Hatami-Marbini, Adel and Agrell, Per J and Tavana, Madjid and Khoshnevis, Pegah, (2017). “A flexible cross-efficiency fuzzy data envelopment analysis model for sustainable sourcing”. Journal of Cleaner Production. 142: 2761-2779.

Kao, C., and Liu, S.T., (2000). “Fuzzy efficiency measures in data envelopment analysis”. Fuzzy Sets and Systems, 3: 427-437.

Khodabakhshi, M., (2010). “An output oriented super-efficiency measure in stochastic data envelopment analysis: Considering Iranian electricity distribution companies”. Computers & Industrial Engineering. 58(4): 663-671.

Khoshfetrat, S., and Daneshvar, S. (2011) “Improving weak efficiency frontiers in the fuzzy data envelopment analysis models” Applied Mathematical Modelling, 1: 339-345.

Land, K.C., Lovell, C.A.K., Thore, S. (1993) “Chance-constrained data envelopment analysis”. Managerial and Decision Economics.14, 541–554.

Liu, Y.P., Gao, X.L., and Shen, Z.Y., (2007). “Product design schemes evaluation based on fuzzy DEA”. Computer Integrated Manufacturing Systems-Beijing. 11: 2099-2104.

Liu, Wenli and Wang, Ying-Ming and Lyu, Shulong., (2017). “The upper and lower bound evaluation based on the quantile efficiency in stochastic data envelopment analysis”. Expert Systems with Applications. 85: 14-24.

Manzoni, Alex and Islam, Sardar MN., (2007) “Measuring collaboration effectiveness in globalised supply networks: a data envelopment analysis application”. International Journal of Logistics Economics and Globalisation. 1(1):77-91.

Olesen, O. B., and Petersen, N.C. (1995) "Incorporating Quality into Data Envelopment Analysis." International Journal of Production Economics 39: 117-135.

Razavyan, S., Tohidi, G., (2008). “Ranking of efficient DMUs with stochastic data”. In International Mathematical Forum. 3(2): 79-83.

Talluri, S., Narasimhan, R., Nair, A., (2006). “Vendor performance with supply risk: A chance-constrained DEA approach”. International Journal of Production Economics. 100(2): 212-222.

Tharwat A., El-Demerdash B., El-Khodary I., (2019). “Developing an Input Oriented Data Envelopment Analysis Model with Fuzzy Uncertainty in Variables”. Creative Business and Social Innovations for a Sustainable Future. Advances in Science, Technology & Innovation. 117-122.

Wang, Ying-Ming and Chin, Kwai-Sang (2011). Fuzzy data envelopment analysis: A fuzzy expected value approach. Expert Systems with Applications. 38(9), 11678—11685.

Wu, Jie and Sun, Jiasen and Liang, Liang, (2012). “Cross efficiency evaluation method based on weight-balanced data envelopment analysis model”. Computers & Industrial Engineering. 63(2): 513-519.

Zerafat Angiz, L.M., Emrouznejad, A., and Mustafa, A., (2010). “Fuzzy assessment of performance of a decision making units using DEA: A non-radial approach”. Expert Systems with Applications, 7: 5153-5157.

Zerafat Angiz, L.M., Emrouznejad, A., and Mustafa, A., (2012). “Fuzzy data envelopment analysis: A discrete approach”. Expert Systems with Applications, 3: 2263-2269.

A Literature Review on U-Shaped Assembly Line Balancing Problem

Naglaa Ragaa[1], Mohammad Zakaraia[2]

Abstract

An assembly line balancing problem is a problem of finding an optimal assignment of the assembly tasks to stations to achieve some specific objectives. The U-shaped assembly line balancing problem (UALBP) is one of its classes that the operators can perform the assembly tasks in the entrance and exit sides of the line. This paper attempts to provide up to date a review of 44 papers on the U-shaped assembly line balancing problem. The main objective is to help the researchers to have a new vision for future points of research. Thus, the paper includes an analysis of the previous work presented on the UALBP, which ended by classifying the problem into a set of classes that leads to such vision.

Keywords: U-shaped assembly line balancing problem (UALBP), integer 0-1 programming, combinatorial optimization, meta-heuristics, branch and bound, dynamic programming.

1. Introduction

An assembly line balancing problem (ALBP) is the problem of optimizing the assignment of the tasks to the assembly stations to achieve some specific objectives. The main constraints of any assembly line balancing problem can be divided into three sets of constraints, which are as follows:

- The assignment constraints, which are the set of constraints that ensures that each task has to be assigned in only one station.
- The cycle time constraints, which are the set of constraints that ensures that the total time of the assigned tasks in each station doesn't exceed the cycle time, where the cycle time is the time between two successive finished products.
- The precedence constraints, which is the set of constraints that ensures that each task can't be assigned before its predecessors.

The U-shaped assembly line is one of the assembly lines where the operators have two sides to assemble the tasks, which are the entrance and the exit sides. Such type of assembly lines allows more flexibility than the straight assembly lines through controlling the number of operators to adapt the changes in demand. UALBP includes another set of constraints which is related to the successors, where any task can be assigned either after assigning its predecessors or its successors.

The main objective of this paper is to present a literature review on UALBP to guide on the future points of research. This paper is organized as follows: Section 2 presents the

[1] Associate Professor in Operations Research, The Faculty of graduate studies for statistical research, Cairo University

[2] PhD Candidate in Operations Research, The Faculty of graduate studies for statistical research, Cairo University

mathematical model of the problem. Section 3 shows the summary of literature review. Section 4 provides the analysis of the literature review. Finally, Section 5 considers the conclusion and future points of research.

2. The mathematical model

Miltenburg and Wijngaard (1994) introduced the first mathematical formulation for UALBP, where the objective is to minimize number of stations. The formulation of the problem is as follows:

Notations:

The set of tasks	$F = \{k \mid k = 1, 2, \ldots, n\}$
The set of precedence constraints	$P = \{(x, y) \mid x \text{ must be performed before } y\}$
The set of task times	$T = \{t(k) \mid k = 1, 2, \ldots, n\}$
The processing time of task k	$t(k)$
The cycle time	C
The assigned tasks at station i.	$S_i = \{k \mid \text{task } k \text{ is done at station } i\}$
Assigned station variable	$ST_i = \begin{cases} 1, & S_i \neq \emptyset, \\ 0, & Otherwise \end{cases}$

The mathematical model:

$$Minimize \sum_{i}^{n} ST_i \tag{1}$$

$$\bigcup_{i=1}^{n} S_i = F \tag{2}$$

$$S_i \bigcap_{i \neq j} S_j = \emptyset \tag{3}$$

$$\sum_{k \in S_i} t(k) \leq C, i = \{1,2, \ldots, n\} \tag{4}$$

For each task y:

$$\begin{aligned} & if\ (x, y) \in P, x \in S_i, y \in S_j\ then\ i \leq j, for\ all\ x; or \\ & if\ (y, z) \in P, z \in S_i, y \in S_j\ then\ i \leq j, for\ all\ z \end{aligned} \tag{5}$$

The objective function (1) is to minimize the number of stations. The constraints (2) and (3) are the assignment constraints. The set of constraints (4) represents the cycle time constraints, where the total processing time of any station must not exceed the cycle time. The set of constraints (5) shows that each task has to be assigned either after assigning its immediate predecessors or its immediate successors.

3. A Review on U-shaped assembly line balancing problem

This section shows some information about the previous work in the UALBP. The problem was first presented by Miltenburg and Wijngaard (1994), where they developed dynamic programming approach based on heuristic to sovle the problem. Urban (1998) presented an integer programming formulation for the problem. Ajenblit and Wainwright

(1998) developed a genetic algorithm to minimize the number of stations. Nakade and Ohno (1999) solved the problem by usnig heuristic approach. Their paper handled multi objectives, where they minimized the cycle time and the number of assigned workers. Erel et al. (2001) developed a simulated annealing approach to minimize the number of stations of the problem. Gokcen et al. (2005) solved the problem by using shortest route approach, where they seeked to minimize the number of stations. Gokcen and Agpak (2006) used a multi criteria dections making approach for achieving several conflicing goals.

Baykasoglu (2006) developed a simulated annealing approach to miximize the smoothness index and to minimize the number of stations. Keun et al. (2006) presented endosymbiotic evolutionary algorithm to maximize the workload smothness. Hwang et al. (2008) developed a genetic algorithm to optimize multi objectives, which are minimzing the number of stations and minimzing the variation of workload. Kara (2008) proposed a simulated annealing approach to minimze the absolute deviation of workload. Özcan and Toklu (2009) used Simulated annealing and genetic algorithm to maximize the line efficiency and to minimize the smoothness index. Hwang and Katayama (2009) developed evolutionary algorithm to minimize number of stations and the variation of workload. Kara et al. (2009) developed an integer programming approach to deal with the problem under uncertainty. Their objectives are to minimize the deviation of two goals, which are the number of stations and the cycle time. Baykasoğlu and Dereli (2009) used an ant colony optimization to minimize the number of stations. Sabuncuoglu et al. (2009) developed an ant colony optimization to minimze the cycle time. Simaria et al. (2009) presented an ant colony optimization to minimize the number of workers.

Yegul et al. (2010) proposed a multi pass assignment heuristic to minimize the number of stations. Bagher et al (2011) used a hybrid evolutionary algorithm to solve the problem under uncertainty. The objecives were to minimize the number of stations, total idle time, and non-completion probabilites of each station. Kara et al. (2011) produced an integer programming formulation the deals with a multi-objective model, where the model is to minimize the total cost assciated with work station untilization, assistant, assignment and equipment allocation. Kazemi et al. (2011) developed a genetic algorithm to minimize the total costs associated with the number of stations and task duplication costs. Hamzadayi and Yildiz (2012) presented a genetic algorithm to minimize number of stations and maximize the smothness index. Rabbani et al. (2012) developed a heuristic algorithm based on genetic algorithm to minimize number of stations and minimize the cycle time. Agpak and Yegul (2012) used GAMS/CPLEX-12 mathematical programming software package to solve a multi-objecive model that minimizes the number of stations and number of positions in the two sided ALBP. Rabbani et al. (2012) developed a genetic algorithm that solves a muli-objective model, where the objecive functions are to minimize the number of stations, maximize the line efficiency, and minimize the variation of workload. Avikal et al. (2013) used the critical path method to minimize number of stations and maximize labor productivity. Hamzadayi and Yildiz (2013) proposed a simulated annealing approach for minimizing the number of stations for mixed models u-shaped assembly line balancing problem. Nourmohammadi et al. (2013) presented a multi-objective model that maximizes the line efficiency and minimizes the variation of workload. They solved such multi-objective model by using an evalutionary algorithm.

Fattahia et al. (2014) presented a new formulation for the problem, where the objective of this formulation is to minimize the number of stations. Jayaswal and Agarwal (2014) developed a simulated annealing approach to minimize the total annual cost of work station utilization, equipment operation and assistant employment. Hazır and Dolgui (2015) proposed a bender decomposition algorithm for minimizing the cycle time. Ogan and Azizogl (2015) developed a branch and bound algorithm for minimizing the total equipment cost. Manavizadeh et al. (2015) solved a multi-objective model that minimizes the wastage in each station and minimizes the work overload. Their developed appoach was a heuristic algorithm. Alavidoost et al. (2016) used two-phase interactive fuzzy programming approach to minimize number of stations and cycle time under uncertainty. Nilakantan and Ponnambalam (2016) developed a particle swarm optimization approach for minimizing the cycle time and maximizing the production rate. Alavidoost et al. (2016) developed a two-phase interactive fuzzy programming approach for solving a multi-objective model that minimizes the number of stations and the cycle time. Fathi and Álvarez (2016) developed a heuristic based simulated annealing for solving a multi-objective model that minimizes the number of stations and the balance efficiency. Alavidoost et al. (2017) developed a modified genetic algorithm that deals with the problem under uncertainty. The objectives were to minimize number of stations, maximize the fuzzy balance efficiency, and minimize the fuzzy idle time precentage.

Li et al. (2017) developed heuristic approach based on multiple rules to minimize the cycle time. Oksuz et al. (2017) used artificial bee colony algorithm and genetic algorithm to maximize the line effiecincy. Li et al. (2018) proposed branch, bound and remember algorithm to maximize the number of stations. Zhang et al. (2018) developed migrating birds optimization algorithm for minimizing the cycle time. Kızılkaya et al. (2019) developed a particale swarm optimization algorithm to minimize the nubmer of stations under uncertainty. Babazadeh and Javadian (2019) developed a novel meta-heurisic approach for solving a multi-objective model that minimizes the number of stations and the cycle time

4. Analysis of the Literature Review:

Over the past decade most researches focused on ALBP with its simplified assumptions, while the UALBP has begun appearing on 1994 by Miltenburg and Wijngaard. This section shows most of the papers that were puplished on UALBP to help classifying the problem and to show the developed approaches for solving it. **Table 1** shows the review of UALBP by listing the objective of each paper, the type of model (deterministic or uncertainy), (single or multi-objectives) and the developed approachs for solving the problem:

Table 1: Summary of the Literature Review on the U-shaped assembly line balancing problems

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
1	Miltenburg & Wijngaard (1994)	Minimize No. of stations	Deterministic	Single	Dynamic programming and heuristics
2	Urban (1998)	Minimize No. of stations	Deterministic	Single	Integer programming formulation

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
3	Ajenblit & Wainwright (1998)	Minimize No. of stations	Deterministic	Single	Genetic algorithm
4	Nakade & Ohno (1999)	Minimize the cycle time & No. of assigned workers	Deterministic	Multi-objectives	Heuristic approach
5	Erel, et al. (2001)	Minimize No. of stations	Deterministic	Single	Simulated annealing
6	Gokcen, et al. (2005)	Minimize No. of stations	Deterministic	Single	Shortest route approach
7	Gokcen & Agpak (2006)	Serveral conflicting goals can be considered simultanous	Deterministic	Multi-objectives	Multi criteria decsion making approach
8	Baykasoglu (2006)	Maximize the smothness index and Minimize No. of stations	Deterministic	Multi-objectives	Simulated annealing
9	Keun, et al. (2006)	Maximize the workload smothness	Deterministic	Single	Endosymbiotic evolutionary algorithm
10	Hwang, et al. (2008)	Minimize No. of stations & the variation of workload	Deterministic	Multi-objectives	Genetic algorithm
11	Kara, (2008)	Minimize the absolute deviation of workloads	Deterministic	Single	Simulated annealing
12	Özcan & Toklu (2009)	Maximize the line efficiency and minimize the smoothness index	Deterministic	Multi-objectives	Simulated annealing and genetic algorithm
13	Hwang & Katayama (2009)	Minimize No. of stations and the variation	Deterministic	Multi-objectives	Evolutionary algorithm

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
		of workload			
14	Kara, et al. (2009)	Minimize the deviation of two goals, which are the No. of stations & cycle time	Uncertainty	Multi-objectives	Integer programming method
15	Baykasoğlu & Dereli (2009)	Minimize No. of stations	Deterministic	Single	Ant colony optimization
16	Sabuncuoglu, et al. (2009)	Minimize the cycle time	Deterministic	Single	Ant colony optimization
15	Simaria, et al. Vilarinho (2009)	Minimize the No. of workers	Deterministic	Single	Ant colony optimisation
18	Yegul, Agpak, & Yavuz (2010)	Minimize No. of stations	Deterministic	Single	Multi-pass assignment heuristic
19	Bagher, Zandieh, & Farsijani (2011)	Minimmize No. of stations, idle time, & non-completion probabilites of each station	Uncertainty	Multi-objectives	Hybrid evolutionary algorithm
20	Kara, et al. (2011)	Minimize the total cost associated with work station utilisation, assistant assignment & equipment allocation	Deterministic	Multi-objectives	Integer programming formulation
21	Kazemi, et al. (2011)	Minimize the total costs associated with the No. of stations & task duplication costs	Deterministic	Single	Genetic algorithm
22	Agpak & Yegul (2012)	Minimize No. of stations in UALBP &	Deterministic	Multi-objectives	GAMS/CPLEX-12 mathematical programming

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
		minimize the No. of positions in the two sided ALBP			software package
23	Hamzadayi & Yildiz (2012)	Minimize No. of stations & maximize the smothness index	Deterministic	Multi-objectives	Genetic algorithm
24	Rabbani, et al. (2012)	Minimize No. of stations and minimize the cycle time	Deterministic	Multi-objectives	Heuristic algorithm based on genetic algorithms
25	Rabbani, et al. (2012)	Minimize No. of stations, maximize the efficiency and minimize the variation of workload	Deterministic	Multi-objectives	Genetic algorithm
26	Avikal, et al. (2013)	Minimize number of stations & maximize labor productivity	Deterministic	Multi-objectives	Critical path method
27	Hamzadayi & Yildiz (2013)	Minimize No. of stations for mixed models UALBP	Deterministic	Single	Simulated annealing
28	Nourmohammadi, et al. (2013)	Maximize the line efficiency and minimize the variation of workload	Deterministic	Multi-objectives	Evolutionary algorithm
29	Fattahia, et al. (2014)	Minimize No. of stations	Deterministic	Single	New formulation for the problem
30	Jayaswal & Agarwal (2014)	Minimize the total annual cost of work station utilization, equipment operation and	Deterministic	Multi-objectives	Simulated annealing

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
		assistant employment			
31	Hazır & Dolgui (2015)	Minimize the cycle time	Uncertainty	Single	Bender decomposition algorithm
32	Manavizadeh, et al. (2015)	Minimize the wastage in each station & minimize the work overload	Deterministic	Multi-objectives	Heuristic algorithm
33	Ogan & Azizogl (2015)	Minimize the total equipment cost	Deterministic	Single	A branch and bound algorithm
34	Alavidoost, Babazadeh, & Sayyar (2016)	Minimize the No. of stations & cycle time	Uncertainty	Multi-objectives	Two-phase interactive fuzzy programming approach
35	Fathi & Álvarez (2016)	Minimize No. of stations & maximize the balance efficiency	Deterministic	Multi-objectives	Heuristic based on simulated annealing
36	Nilakantan & Ponnambalam (2016)	Minimize the cycle time and maximize the production rate	Deterministic	Multi-objectives	Particle swarm optimization
37	Alavidoost, et al. (2017)	Minimize No. of stations, maximize the fuzzy balance efficiency & minimize the fuzzy idle time precentage	Uncertainty	Multi-objectives	Modified genetic algorithm
38	Li, et al. (2017)	Minimize cycle time	Deterministic	Single	Heuristic approach based on multiple rules
39	Oksuz, et al. (2017)	Maximize the line efficiency	Deterministic	Single	Artificial Bee Colony Alg. & Genetic Algorithm
40	Li, et al. (2018)	Minimize No.	Deterministic	Single	Branch, bound

No.	Author(s)	Objective (s)	Type of Model	Single / Multi Objectives	The Approach Used
		of stations			and remember algorithm
41	Zhang, et al. (2018)	Minimize the cycle time	Deterministic	Single	Migrating birds optimization algorithm
42	Sresracoo, et al. (2018)	Minimize No. of stations	Deterministic	Single	Differential Evolution Algorithm
43	Babazadeh & Javadian (2019)	Minimize No. of stations and cycle time	Uncertainty	Multi-objectives	A novel meta-heuristic approach
44	Kızılkaya, et al. (2019)	Minimize No. of stations	Uncertainty	Single	particle swarm optimization

Table 1 represents the data matrix that can help researchers to have an approximatly full review about history of the UALBP. The next sub-section provides analysis of this review, which covers the kind of model (single or multi-objectives), besides (determinstic or uncertain) and finally, we consider the classification according to the main objective of UALBP as a model in each and different papers besides, showing the different approaches used to the solve the problem.

- **The model Type (Single or Multi-objectives / deterministic or uncertainty)**

Figure 1 shows a pie chart that divides UALBP into two categories according to its model types, which are the single objective and the multi-objectives. It depicts that about 55% of the researches have focused on the single objective category, while 45% have focused on the multi-objectives.

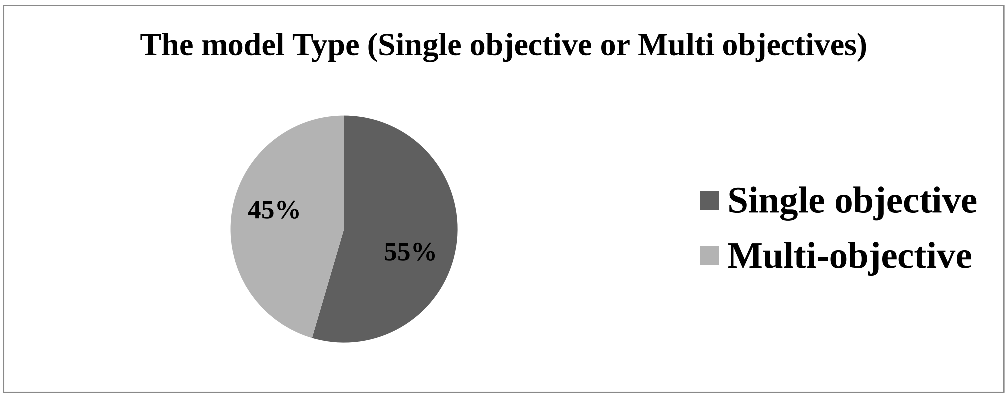


Figure 1: The model type: Single objective or Multi-objectives

Also, the researches about UALBP are classified into two main kinds of models, which are deterministic or uncertainty models. Thus, as in Figure 21 the pie chart shows the ratios where it found 55% of models single objective while 45% are multi-objectives models.

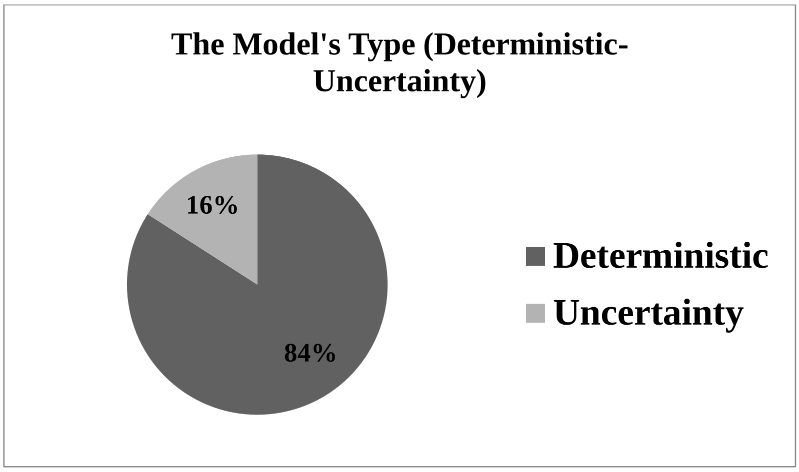


Figure 2: The model type (deterministic models or uncertainty)

It can be seen from Figure 2 that majority of research on this topic have been devoted to deterministic case of the problem with ratio 84% while the analysis are demonstrated for a small portion of models including uncertainty models. Therefore, the future points of research may have to deal more with the uncertainty case of this problem.

- ## The Objective of Model

There are several objectives required for the problem as shown in **Table 1**. Table 2 illustrates the frequency & the relative percentage for each objective as follow:

Table 2: The percentage of contributions for each objective

Objective	Frequency	Relative Percentage
Minimize number of stations	26	41%
Minimize the cycle time	10	16%
Maximize the balance efficiency	5	8%
Minimize variation of workload	4	6%
Minimize the total costs associated with the number of stations	3	5%
Maximize the workload smoothness	2	3%
Minimize the idle time	2	3%
Maximize the production rate	2	3%
Minimize number of assigned workers	1	2%
Maximize the smoothness index	1	2%
Minimize the smoothness index	1	2%
Minimize the number of workers	1	2%
Minimize non-completion probabilities of each station	1	2%
Minimize the variation of workload	1	2%
Minimize the wastage in each station	1	2%
Minimize the workload	1	2%

Objective	Frequency	Relative Percentage
Minimize the total equipment cost	1	2%
Maximize balance efficiency	1	2%

From Table 2, the main focus in the previous work concentrates on minimizing the number of stations and minimizing the cycle time. So, the future points of research may take into consideration the other objectives due to the little efforts on them.

- The used Approaches

Due to the combinatorial nature of the problem, which makes it an NP-hard problem, most of contributions that developed to solve it are meta-heuristics. Figure 3 is a bar chart that shows a comparison between the approaches that were developed to solve the problem according to three categories, which are classical approaches such as the branch and bound and the dynamic programming, heuristic approaches, and meta-heuristic approaches:

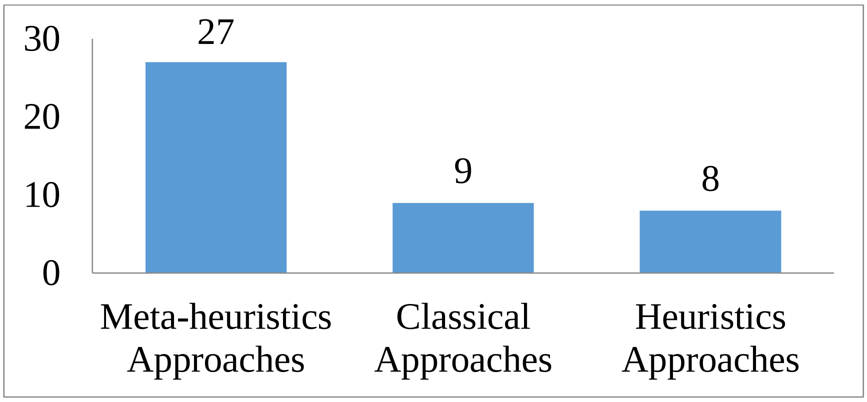


Figure 3: The comparison of the developed approaches to solve the problem

5. Conclusion

The review has covered most of the published papers in the U-shaped assembly line balancing problem. After analyzing the review, it found that most of researches have focused on solving a single objective specially minimizing the number of stations and cycle time. The decision type of most papers is under certainty, which is the deterministic case of processing time. Due to the combinatorial nature of the problem, which makes it NP-hard problem, most of developed approaches are meta-heuristics.

References

Agpak, K., & Yegul, M. F. (2012) "Two-Sided U-Type Assembly Line Balancing Problem". International Journal of Production Research, 50, 5035-5047.

Ajenblit, D. A., & Wainwright, R. L. (1998) "Applying genetic algorithms to the U-shaped assembly line balancing problem". The 1998 IEEE International Conference on, 96-101.

Alavidoost, M. H., Zarandi, M. H., Tarimoradi, M., & Nemati, Y. (2017) "Modified Genetic Algorithm for Simple Straight and U-Shaped Assembly Line Balancing with Fuzzy Processing Times". Journal of Intelligent Manufacturing, 28, 313-336.

Alavidoost, M., Babazadeh, H., & Sayyar, S. (2016) "An Interactive Fuzzy Programming Approach For Bi-Objective Straight And U-Shaped Assembly Line Balancing Problem". Applied Soft Computing, 40, 221-235.

Avikal, S., Jain, R., Mishra, P., & Yadav, H. (2013) "A Heuristic Approach for U-Shaped Assembly Line Balancing to Improve Labor Productivity". Computers & Industrial Engineering, 64, 895–901.

Babazadeh, H., & Javadian, N. (2019) "A novel meta-heuristic approach to solve fuzzy multi-objective straight and U-shaped assembly line balancing problems". Soft Computing, 23, 8217–8245.

Bagher, M., Zandieh, M., & Farsijani, H. (2011) "Balancing Of Stochastic U-Type Assembly Lines: An Imperialist Competitive Algorithm". The International Journal of Advanced Manufacturing Technology, 54, 271–285.

Baykasoglu, A. (2006) "Multi-rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems". Journal of Intelligent Manufacturing, 17, 217-232.

Baykasoğlu, A., & Dereli, T. (2009) "Simple and U-Type Assembly Line Balancing By Using an Ant Colony Based Algorithm". Mathematical and Computational Applications, 14, 1-12.

Erel, E., Sabuncuoglu, & Aksu, B. A. (2001) "Balancing of U-type assembly systems using simulated annealing". International Journal of Production Research, 39, 3003- 3015.

Fathi, M., & Álvarez, M. J. (2016) "A New Heuristic-Based Bi-Objective Simulated Annealing Method for U-Shaped Assembly Line Balancing". European J. Industrial Engineering, 10, 145-169.

Fattahia, A., Elaoudb, S., & Turkay, E. S. (2014) "A Novel Integer Programming Formulation with Logic Cuts for The U-Shaped Assembly Line Balancing Problem". International Journal of Production Research, 52, 1318–1333.

Gokcen, H., & Agpak, K. (2006) "A goal programming approach to simple U-line balancing problem". European Journal of Operational Research, 171, 557-585.

Gokcen, H., Agpak, K., Gencer, C., & Kizilkaya, E. (2005) "A shortest route formulation of simple U-type assembly line balancing problem". Applied Mathematical Modelling, 29, 373-380.

Hamzadayi, A., & Yildiz, G. (2012) "A Genetic Algorithm Based Approach for Simultaneously Balancing and Sequencing of Mixed-Model U-Lines with Parallel Workstations and Zoning Constraints". Computers & Industrial Engineering, 62, 206–215.

Hamzadayi, A., & Yildiz, G. (2013) "A Simulated Annealing Algorithm Based Approach for Balancing and Sequencing of Mixed-Model U-Lines". Computers & Industrial Engineering, 66, 1070-1084.

Hazır, Ö., & Dolgui, A. (2015) "A Decomposition Based Solution Algorithm For U-Type Assembly Line Balancing With Interval Data". Computers & Operations Research, 59, 126-131.

Hwang, R. K., Katayama, H., & Gen, M. (2008) "U-shaped assembly line balancing problem with genetic algorithm". International Journal of Production Research, 46, 4637–4649.

Hwang, R., & Katayama, H. (2009) "A multi-decision genetic approach for workload balancing of mixed-model U-shaped assembly line systems". International Journal of Production Research, 47, 3797–3822.

Jayaswal, S., & Agarwal, P. (2014) "BalancingU-Shaped Assembly Lines with Resource Dependent Tasktimes: A Simulated Annealing Approach". Journal of Manufacturing Systems, 33, 522-534.

Kara, Y. (2008) "Line balancing and model sequencing to reduce work overload in mixedmodel U-line production environments". Engineering Optimization, 40, 669–684.

Kara, Y., Özgüven, C., Yalçın, N., & Atasagun, Y. (2011) "Balancing Straight And U-Shaped Assembly Lines With Resource Dependent Task Times". International Journal of Production Research, 49, 6387–6405.

Kara, Y., Paksoy, T., & Chang, C.-T. (2009) "Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing". European Journal of Operational Research, 195, 335–347.

Kazemi, S. M., Ghodsi, R., Rabbani, M., & Tavakkoli-Moghaddam, R. (2011) "A Novel Two-Stage Genetic Algorithm for A Mixed-Model U-Line Balancing Problem with Duplicated Tasks". The International Journal of Advanced Manufacturing Technology, 55, 1111–1122.

Keun, K. Y., Yun, K. J., & Yeongho, K. (2006) "An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines". European Journal of Operational Research, 168, 838–852.

Kızılkaya, E., Yılmaz, A., Özcan, D., Gencer, C., & Bali, Ö. (2019) "Balancing stochastic U-lines using particle swarm optimization". Journal of Intelligent Manufacturing, 30, 97–111.

Li, M., Tang, Q., Zheng, Q., Xia, X., & C.A.Floudas. (2017) "A Rules–Based Heuristic Approach for The U-Shaped Assembly Line Balancing Problem". Applied Mathematical Modelling, 48, 423-439.

Li, Z., Kucukkoc, I., & Zhang, Z. (2018) "Branch, bound and remember algorithm for U-shaped assembly line balancing problem". Computers & Industrial Engineering, 124, 24-35.

Manavizadeh, N., Rabbani, M., & Radmehr, F. (2015) "A New Multi-Objective Approach in Order to Balancing and Sequencing U-Shaped Mixed Model Assembly Line Problem: A Proposed Heuristic Algorithm". The International Journal of Advanced Manufacturing Technology, 79, 415–425.

Miltenburg, G. J., & Wijngaard, J. (1994) "The U-line Line Balancing Problem". MANAGEMENT SCIENCE, 40, 1378-1388.

Nakade, K., & Ohno, K. (1999) "An optimal worker allocation problem for a U-shaped production line". International Journal of Production Economics, 60-61, 353-358.

Nilakantan, J. M., & Ponnambalam, S. (2016) "Robotic U-Shaped Assembly Line Balancing Using Particle Swarm Optimization". Engineering Optimization, 48, 1-22.

Nourmohammadi, A., Zandieh, M., & Tavakkoli-Moghaddam, R. (2013) "An Imperialist Competitive Algorithm for Multi-Objective U-Type Assembly Line Design". Journal of Computational Science, 4, 393-400.

Ogan, D., & Azizogl, M. (2015) "A Branch and Bound Method for The Line Balancing Problem in U-Shaped Assembly Lines with Equipment Requirements". Journal of Manufacturing Systems, 36, 46–54.

Oksuz, M. K., & Satoglu, S. I. (2017) "U-Shaped Assembly Line Worker Assignment and Balancing Problem: A Mathematical Model and Two Meta-Heuristics". Computers & Industrial Engineering, 112, 246-263.

Özcan, U., & Toklu, B. (2009) "A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems". Journal of Intelligent Manufacturing, 20, 29-123.

Rabbani, M., Kazemi, S. M., & Manavizadeh, N. (2012) "Mixed Model U-Line Balancing Type-1 Problem: A New Approach". Journal of Manufacturing Systems, 31, 131–138.

Rabbani, M., Moghaddam, M., & Manavizadeh, N. (2012) "Balancing of Mixed-Model Two-Sided Assembly Lines with Multiple U-Shaped Layout". The International Journal of Advanced Manufacturing Technology, 59, 1191–1210.

Sabuncuoglu, I., Erdalerel, & Ardaalp. (2009) "Ant Colony Optimization for The Single Model U-Type Assembly Line Balancing Problem". International Journal of Production Economics, 120, 287-300.

Simaria, A., Sa´, M. Z., & Vilarinho, P. (2009) "Meeting Demand Variation Using Flexible U-Shaped Assembly Lines". International Journal of Production Research, 47, 3937–3955.

Sresracoo, P., Kriengkorakot, N., Kriengkorakot, P., & Chantarasamai, K. (2018) "U-Shaped Assembly Line Balancing by Using Differential Evolution Algorithm". Mathematical and Computational Applications, 23, 1-21.

Urban, T. L. (1998) "Note. Optimal Balancing of U-Shaped Assembly Lines". Management Science, 44(5), 738-741.

Yegul, M. F., Agpak, K., & Yavuz, M. (2010) "A New Algorithm For U-Shaped Two-Sided Assembly Line Balancing". Transactions of the Canadian Society for Mechanical Engineering, 34, 225-241.

Zhang, Z., Tang, Q., Han, D., & Li, Z. (2018) "Enhanced migrating birds optimization algorithm for U-shaped assembly line balancing problems with workers assignment". Neural Computing and Applications, 1-15.

Crow Search Algorithm for Solving Resources Allocation of Timetable Problem

Naglaa Ragaa[1], Abdalla Eldhshan[2]

Abstract

The resources allocation problem attempts to find an optimal assignment of a set of activities to limited resources in order to minimize the violated cost incurred by the assignment process. Timetabling problem, as a one of resources allocation problems concerns with the assignment of a set of activities (teachers, courses, and subjects) to specific number of resources (rooms, days, and timeslots). Typically, for building a feasible solution, a set of hard constraints must be satisfied, and to improve the quality of the feasible solution, another set of soft constraints should be satisfied. Accordingly, one of recent meta-heuristic optimization, Crow Search Algorithm (CSA) is a nature-inspired of intelligent behavior of crows, which it uses for solving optimization problems. The proposed CSA introduced to solve the resources allocation of timetable problem. The results of the experiment casts a new light on using the CSA for solving the timetable problem, this yields increasingly good results comparing with other techniques.

Keywords: Resources Allocation; Timetable Problem; Metaheuristic Algorithms; Crow Search Algorithm

1. Introduction

In past decades, Resources allocation plays an important role in many fields. It is frequently facing in manufacturing, educations, project managements, transportations, etc. Resources allocation of timetable problem is one of the famous education problems. Timetable has classified as NP-hard problem Burke and Petrovic (2002). Since, the problem consists of multi-dimensional elements and multi objectives. Generating a feasible timetable is very exhausting work. Traditional methods need hard effort and time to satisfy all problem constraints. The constraints of timetable problem can be classified into two types. Hard constraints, which never violated, and soft constraints, which should be satisfied but it might violate. Generally, Resources allocation of timetable is a process of assignment a set of activities (regular meeting between students and teachers for a specific subject) into a limited number resources (timeslots of working days and rooms) Eludire and Akanbi (2017). The objective of timetabling problem is to maximize the quality of a feasible solution. In this work, CSA has presented as a new intelligent method inspired by crows behavior of saving excess food in hiding places and retrieve it when the food are needed. It belongs to metaheuristic population-based approaches, as well as it explores the search space to find a solution with high quality Askarzadeh (2016). The proposed CSA investigated to solve the resources allocation of timetable problem. Hdtt (**h**ar**d** **t**ime**t**able) benchmark is used to examine the efficiency of the proposed CSA comparing with other literature techniques. .

---

[1] Associate Professor in Operations Research, The Faculty of Graduate Studies for Statistical Research, Cairo University

[2] PhD Candidate in Operations Research, The Faculty of Graduate Studies for Statistical Research, Cairo University

This paper consists of seven sections. Section 1 demonstrates a general overview of the research. In section 2, literature review is described. In Section 3, the resources allocation of timetable problem definition is introduced. In Section 4, the crow search algorithm is proposed to solve resources allocation of timetable problem. In Section 5, the proposed CSA for recourses allocation of timetable is described. Section 6 analyzes, testing and the experimental results of the proposed algorithm. Finally, Section7 includes the conclusion.

2. Literature Review

Optimization process is the way of finding the best solution of a set of feasible solutions with consideration of the problem constrains. Optimization problems have two types of variables discrete or continuous. Discrete optimization problems named also combinatorial optimization problems Arora et al. (1994). Resources allocation of timetable problem is a combinatorial optimization problem, Artificial intelligence approaches and other optimization techniques are introduced for solving this problem. Simulated annealing Zhang et al. (2010) tabu search Lü and Hao (2010) and great deluge Shaker and Abdullah (2009) techniques of single-based metaheuristic algorithms are introduced for solving this problem type. As well as, Genetic algorithms Raghavjee and Pillay (2012); El-Sherbiny et al. (2015), artificial bee colony Bolaji et al. (2014) are proposed as population based metaheuristic algorithms. Not only metaheuristic techniques used to solve the problem but also heuristic methods Pimmer and Raidl (2013) and Integer programming Phillips et al. (2015) as a traditional approach are used. Although, there are many methods and approaches are adapted deals with complex optimization problems, no one could efficiently solve all the classes of optimization problems Wolpert and Macready (1997).

Recently, one of the newest metaheuristic optimizer, crow search algorithm (CSA) has been introduced by Askarzadeh (2016). The CSA emulates the behavior of crows as intelligent birds. He demonstrated that the CSA has better and competitive results, which were discussed the effectiveness of the technique that can be implemented easily. Furthermore, in another research, Askarzadeh (2016) used CSA for optimization of the PV/wind/tidal/battery system. Abdelaziz and Fathy el al. (2017) modified the CSA for optimal design of the radial distribution network by optimal selection of the suitable conductor to minimize the capital and energy loss costs. Liu el al. (2017) proposed an evaluation model of groundwater quality by improved Extreme Learning Machine (ELM) to resolve fuzziness of the water quality evaluation and incompatibility of water parameters. Crow Search Algorithm (CSA) was used to optimize the input weights and thresholds of hidden-layer neurons of the ELM. Mohamadi and Abdi (2018) modified the CSA to solve the non-convex economic load dispatch problem. They proposed an elitist approach for selecting the target crow. In addition, they adjusted the flight length of crows based on crows distance. Rizk-Allah el al. (2019) presented the chaotic crow search algorithm (CCSA) for solving fractional optimization problems (FOPs). They tried to refine the global convergence speed and enhance the exploration/exploitation tendencies. The proposed CCSA has integrated the chaos theory (CT) into the CSA. Javidi el al. (2019) presented an enhanced crow search algorithm for optimum design of structure. However, there has been less previous evidence for employing the CSA for combinatorial optimization problems and needs more deep investigations.

3. Resources Allocation of Timetable Problem Definition

Resources allocation have identical entities that are collected and interacted together to achieve a certain goal. Lazarev and Nekrasov (2017) listed the common entities of resource scheduling problem.

- **Activities:** Let $A = \{a_1.a_2 \dots, a_N\}$ is the set of given activities (events, jobs, tasks, etc.) which need a set of resources to process or execute.
- **Resources**: the physical entities that are used in the scheduling process (e.g. machines, rooms, workers, raw material, etc.). Each resource can process only one task, and the set of all resources designated as R. In case there are M types of resources the set R includes M subsets of typified resources:

$$\boldsymbol{R} = \bigcup_{\boldsymbol{m}=\boldsymbol{1}}^{\boldsymbol{M}} \boldsymbol{R_m} \qquad (\boldsymbol{1})$$

Timetable is known as multi-dimensional assignment problem, which combining the shard objects (teacher. class. subject. room. day. and period). The teacher assigned to teach a subject for a class in room at a specific time of the day. This assignment has to consider a set of hard and soft constraints.

Let *T* is a set of teachers.

C is a set of classes.

S is a set of subjects.

R is a set of rooms.

P is a set of periods.

D is a set of days.

$A_{t.s}$ is a set of periods when teacher t of subject s is available, and the decision variable is:

$$x_{tcsrpd} = \begin{cases} 1 \quad if\ teacher\ t\ teaches\ subject\ s\ for\ class\ c\ \ in\ room\ r \\ \qquad\qquad at\ \ period\ p\ in\ day\ d \\ 0\ otherwise \end{cases} \qquad (\boldsymbol{2})$$

3.1 Hard constraints

The set of hard constraints are the main element of solving timetable problem. The solution must satisfy all the hard constraints to be a feasible solution. The set of main timetable hard constraints as follows. Zhang et al. (2010)

HC1: A teacher cannot be assigned to more than one class during any time slot.

$$\sum_{c \in C} x_{tcsrpd} \ \leq 1 \quad \forall\ \ t \in T_s\,.s \in S.r \in R.p \in P.d \in D. \qquad (\boldsymbol{3})$$

HC2: A class cannot be assigned more than one teacher for any time slot.

$$\sum_{s \in S} \sum_{t \in T_s} x_{tcsrpd} \leq 1 \quad \forall\, c \in C.r \in R.p \in P.d \in D. \qquad (\boldsymbol{4})$$

HC3: A room cannot be allocated more than once for any time slot.

$$\sum_{t\in T_s}\sum_{c\in C}\sum_{s\in S} x_{tcsrpd} \leq 1 \quad \forall\ r \in R.p \in P.d \in D. \tag{5}$$

HC4: A teacher t. teaching subject s. is to be assigned $l_{t.s}$ lectures (periods or time slots) per week.

$$\sum_{c\in C}\sum_{r\in R}\sum_{d\in D}\sum_{p\in P} x_{tcsrpd} = l_{t.s} \quad \forall\ s \in S.t \in T_s\,. \tag{6}$$

HC5: A class c must attend l_c lectures per week.

$$\sum_{s\in S}\sum_{t\in T_s}\sum_{r\in R}\sum_{d\in D}\sum_{p\in P} x_{tcsrpd} = l_c \quad \forall\ c \in C. \tag{7}$$

3.2 Soft Constraints:

The set of soft constraints used to measure the solution quality. Every problem has a set of soft constraints upon the enterprise rolls such as teacher preference time, Consecutive or isolated of lectures, and room utilization.

3.3 Fitness Function:

Fitness function is used to evaluate the solution fitness. That is differentiating between the solutions. Doulaty et al. (2013)

$$eval(f) = \frac{1}{1 + \mathrm{cost(f)}} \tag{8}$$

Since, cost(f) represents the number of violating constraints and it can be calculated as:

$$\boldsymbol{cost(f) = \sum_{i=1}^{ct} n_i(f) \times w_i} \tag{9}$$

Where, ct is the count of constraints. $n_i(f)$ is the penalty of violating constraint i and w_i is the weight of the constraint *i*.

The objective is to minimize the count of violated soft constraints for the feasible solution which already generated by obtaining the hard constrains.

4. Crow Search Algorithm

Crows are extremely intelligent birds. They have a large brain relative to their body size and very close to human brain based on brain-to-body ratio. They have many incredible skills such as problem solving and communication. Self-awareness is a one of the amazing skills which have tool-making ability. The researches show that the crow can remember faces and never forget. Furthermore, it has the ability of using helping tools for food searching journey.

Besides that, they can remember their hiding places of food for long period of time Askarzadeh (2016).

Crows follow other birds to see where its food will be stored, that to steal the food once it goes. Just it had the food, the crow makes set of prevarications to deceive other watching birds, they are using their own experience to protect themselves to be a victim later, and to determine the safest actions to avoid their caches from being pilfered.

Based on the above-mentioned, Askarzadeh (2016) listed the principles of CSA:

- Crows live in the form of flock.
- Crows memorize the position of their hiding places.
- Crows follow each other to do thievery.
- Crows protect their caches from being pilfered by a probability.

Let (d), The number of environment dimensions,

N: The number of crows (flock size or population size).

g_{max} : The maximum number generations.

$x^{i.g}$: The position of crow i at generation g (iteration),

Where, $i = 1.2. \dots . N;\ g = 1.2. \dots . g_{max}$.

$x^{i.g} = \left[x_1^{i.g}. x_2^{i.g}. \dots . x_d^{i.g}\right]$ is the vector of search space.

p^{ig} : Represents the best hiding food place obtained by crow i at generation$g$.

Actually, the crow keeps its best position experience and moves in the environment to find out better food sources (hiding places).

Assume that at generation g, crow j wants to visit its hiding place, p^{jg}. At this generation, crow i decided to follow crow j to discover the hiding place of crow j. In this situation, two cases may be happen:

Case I: Crow j does not know that crow i is following it. Then, crow i will steal the hiding place of crow j. In this case, for next generation the position of crow i is obtained as follows:

$$x^{i.g+1} = x^{i.g} + r_i \times fl^{i.g} \times \left(p^{ig} - x^{j.g}\right) \qquad (\mathbf{10})$$

Where r_i is a random number with uniform distribution between 0 and 1 and $fl^{i.g}$ is the flight length of crow i at generation g.

Case II: Crow j knows that crow i is following it. Then, in order to protect its cache from pilfered, crow j will fool crow i by going to another position of the search space.

Generally, the two cases can be summaries as follows:

$$x^{i.g+1} = \begin{cases} x^{i.g} + r_i \times fl^{i.g} \times \left(p^{ig} - x^{j.g}\right) & r_j \geq AP^{j.g} \\ a\ random\ possion & otherwise \end{cases} \qquad (\mathbf{11})$$

Where r_j is a random number with uniform distribution between 0 and 1 and $fl^{i.g}$ is the flight length of crow i at generation g, while $AP^{j.g}$ is the awareness probability of crow j at generation g.

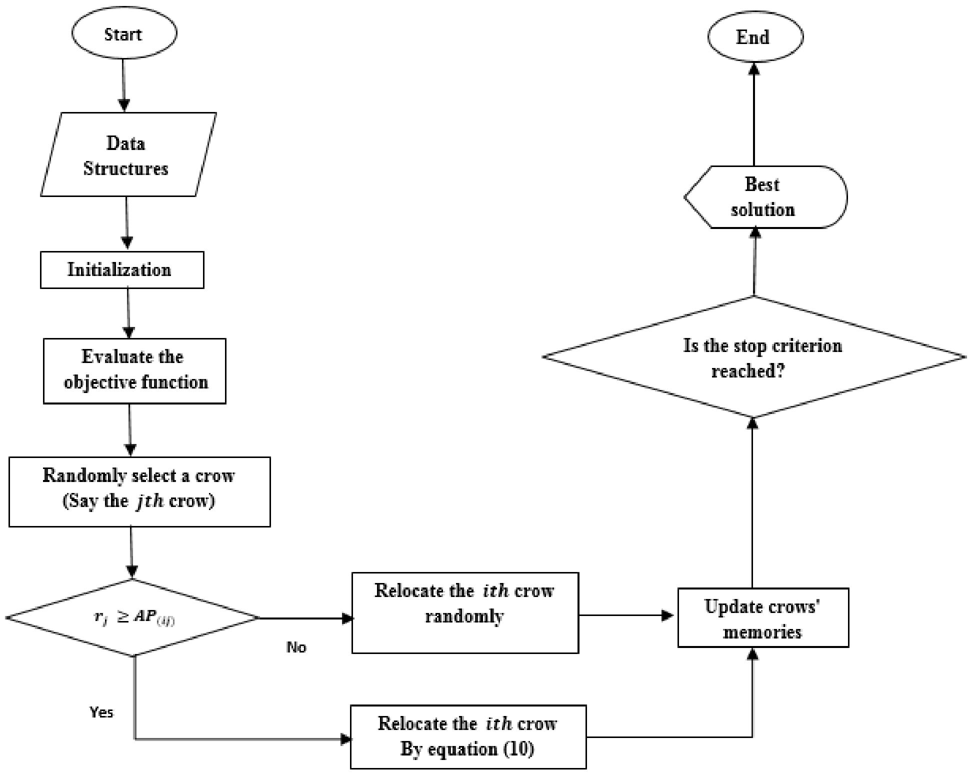


Figure 1 Flowchart of Crow Search Algorithm (Zolghadr-Asli el al. (2018))

5. CSA for Recourses Allocation of Timetable

5.1. Solution Structure

Constructing a reasonable solution structure notably is absent to create a direct relationship between the problem entities (dimensions). Hence, Working with multidimensional problem is very complicated to solve. Therefore, it is useful to associate the related entities into two-dimensional.

Timetable as a resources allocation problem can be abstracted into two main entities. The first main entity is the available resources $S = \{s_1. s_2. \dots. s_j\}$. which contains the combination of periods $P = \{1.2. \dots. p\}$. rooms $R = \{1.2. \dots. r\}$. and days $\mathrm{D} = \{1.2. \dots. \mathrm{d}\}$. The combination can represent as a vector with length $(\mathrm{p} \times \mathrm{r} \times \mathrm{d})$. The second dimension is activities, that can be represented as $A = \{a_1. a_2. \dots. a_i\}$. this dimension concatenates subjects $\mathrm{S} = \{1.2. \dots. \mathrm{s}\}$. teachers $\mathrm{T} = \{1.2. \dots. \mathrm{t}\}$. and classes (group of students) $\mathrm{C} = \{1.2. \dots. \mathrm{c}\}$.

Figure 2 gives an illustration of solution structure, the activities are ordered as the dimension of the allocation array, and the resource integer encoding s_j will assign to the activity a_i. The proposed solution structure is not only used for representing the problem dimensions, but it avoids violating the first three hard constrains of the problem which described in equations (3, 4, and 5) El-Sherbiny et al. (2015).

Activity	a_1	a_2	a_3			a_{i-1}	a_i
Resource	3	4	19	s_j	...		77

Figure 2 Solution Structure

Example:

Assume that there are 2 rooms, 3 days, and 2 periods per day, And there are 10 activities need available resource to allocate. Then, the number of the available resources will be:

$$= 2\ (\text{rooms}) \times 3(\text{days}) \times 2(\text{periods})$$

$$= \ 12\ \ (\text{avilible resources}).$$

- The activity list:

A 1	**A 2**	**A 3**	**A 4**	**A 5**	**A 6**	**A 7**	**A 8**	**A9**	**A 10**

- The available resources list:

S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**

The resource will select randomly from list of the available resources to allocate. Starting from the first activity A1, select random resources let it S4 and remove S4 from available resource. Then, S4 allocate to activity A1. The second activity, select random resource let it S7 and remove S7 from available resource, then Resource S7 allocate to activity A2 and the same procedure applies to all activity to get the final solution structure.

- Solution Structure

A 1	**A 2**	**A 3**	**A 4**	**A 5**	**A 6**	**A 7**	**A 8**	**A9**	**A 10**
S4	**S7**	**S9**	**S12**	**S3**	**S6**	**S1**	**S10**	**S8**	**S5**

- The reminder available resources list:

~~**S1**~~	**S2**	~~**S3**~~	~~**S4**~~	~~**S5**~~	~~**S6**~~	~~**S7**~~	~~**S8**~~	~~**S9**~~	~~**S10**~~	**S11**	~~**S12**~~

5.2. The Initial Population

As mentioned before in problem definition, there are group of activities (Teacher, Class and Subject) required a suitable resource (Room, Day and Period) to create the crows (solutions). For each new crow, the proposed algorithm is ordered as list of activities then, it is started with the first event selecting randomly resource from the available resources, and then, the selected resource allocates to the current activity. Figure 1 illustrates the basic framework of CSA, and the steps of the CSA are stated as follows (Javidi el al. (2019)):

Crow Search Algorithm
Step 1: Initialize algorithm parameters: Assign number of population(n), maximum number of generation g_{max}, flight length (fl) and awareness probability (AP). **Step 2:** Generate initial solutions randomly, check feasibility and fill memory $Population = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix}$ $Memory \quad = \begin{bmatrix} m_1^1 & m_2^1 & \dots & m_d^1 \\ m_1^2 & m_2^2 & \dots & m_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_1^n & m_2^n & \dots & m_d^n \end{bmatrix}$ Where d is number of decision variable (problem dimension). **Step 3:** Calculate objective function (fiteness function) by equation (9). **Step 4:** Generate new solution by equation (11). **Step 6:** Calculate objective function of new solutions like in step3. **Step 7:** Update the memory: If the new solution is better than its best, the memory will be updated. Otherwise, the memory should not be updated: **Step 8:** Repeat steps 4 to 7 until the termination criterion is satisfied and retune the best global solution for all population.

5.3. Reproduction

The proposed approach depends only on the mutation operator for generating the new population (crows) from its main crow source. The previous work El-Sherbiny et al. (2015) has experimented 16 type of mutation on the hdtt instances. It used the Relative Percentage Deviation (RPD) to select the fittest mutation type. This mutation is based on generating a random number for NS ϵ $[1.\mathrm{a}]$. where a is the length of the available events. Then, the basic two point swap is performed NS times. The NS is calculated as following:

$$NS = Int\big(Rand(1.a)\big) \qquad (\mathbf{12})$$

6. Experimental Results

The datasets bank of OR Library provides timetable datasets called HDTT. All Instances and its description are available and share the official URL website[3]. HDTT denotes to the instances are **h**ar**d** **t**ime**t**abling problems. All periods must be used with very little or no options for each allocation. Each week has a five working days and each day has a six periods with 30 timeslots are available per room.

CSA implemented using Visual Studio 2010 in C# language. Output results of all runtimes have measured using the Intel Core 7 and 8GB RAM.

[3] http://people.brunel.ac.uk/mastjjb/jeb/orlib/tableinfo.html.

The parameter values that are used for solving Course Timetable Problem CTTP are (10) population size. (10000) generation number. (0.01) awareness probability (AP). The proposed CSA are compared with other literature methods used to solve the timetable problems for the same benchmark instances, which described as follows:

- **SA1** Simulated Annealing Algorithm Abramson and Dang (1993).
- **SA2** Simulated Annealing Heuristic Randall and Abramson (2001).
- **TS** stand to Tabu Search. **GS** a Greedy Search. **NN-T2** and **NN-T2** are Hopfield Neural Networks Smith et al. (2003).
- **DWTAN** and **CPMF** are neural network approaches Carrasco and Pato (2004).
- **SA3** is a Simulated Annealing variant Gendreau and Potvin (2005).
- **TFH** Timeslot-filling heuristic and **EAH** event-assignment heuristic. The both of them are a heuristic methods described in Pimmer and Raid (2013).
- **GAHCO** denotes to Genetic Algorithm with Hill Clamping Optimization El-Sherbiny et al. (2015).
- **CSA** denotes to the proposed crow search algorithm.

Table1 compares the results of CSA with other methods. The first column indicates the count of violated constraints for the best result achieved, and the second column for the average of violated constraints of 20 runs. The results show that the CSA is the best of all methods, which yields zero cost as the best cost and the average cost values for all instances. In addition, no cost all datasets. The performance of the proposed algorithm is competitive when compared to the other methods, and performs the best finding timetables for all 20 runs conducted for each data set.

Table1. Comparative result of the CSA with other methods

Method	HDTT4		HDTT5		HDTT6		HDTT7		HDTT8	
	Best Cost	Avg Cost	Best Cost	Avg Cost	Best Cost	Avg Cost	Best Cost	Avg Cost	Best Cost	Avg Cost
SA1	-	-	**0.0**	0.7	**0.0**	2.5	2.0	2.5	2.0	2.5
SA2	**0.0**	**0.0**	**0.0**	0.3	**0.0**	0.8	**0.0**	1.2	**0.0**	1.9
TS	**0.0**	0.2	**0.0**	2.2	3.0	5.6	4.0	10.9	13.0	17.2
GS	5.0	8.5	11.0	16.2	19.0	22.2	26.0	30.9	29.0	35.4
NN-TT2	**0.0**	0.1	**0.0**	0.5	**0.0**	0.8	**0.0**	1.1	**0.0**	1.4
NN-TT3	**0.0**	0.5	**0.0**	0.5	**0.0**	0.7	**0.0**	1.0	**0.0**	1.2
CPMF	5.0	10.7	8.0	13.2	11.0	18.7	18.0	25.6	15.0	28.6
DWTAN	**0.0**	**0.0**	**0.0**	0.4	**0.0**	1.65	**0.0**	2.1	0.0	3.25
SA3	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.4
EAH	**0.0**	**0.0**	2.0	5.4	6.0	7.9	9.0	12	13.0	15
TFH	**0.0**	**0.0**	**0.0**	0.6	**0.0**	2.1	**0.0**	2.5	**0.0**	3.1
GAHCO	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	0.6
CSA	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**

7. Conclusion

This study proposes the crow search algorithm CSA for solving the resources allocation of timetable problem. The results indicate that the crow search algorithm provide a good potential mechanism compared with other mechanisms. CSA gives optimal solutions with one hundred percentages of fitness values within a few seconds for all datasets of the benchmark. So, the proposed approach can add as a new method for solving resources allocation problems. The future work needs more investigations for tuning the proposed CSA parameters. Looking forward, further applies for CSA to similar combinatorial optimization problems could prove quite beneficial.

References

- Abdelaziz, A. Y., & Fathy, A. (2017). A novel approach based on crow search algorithm for optimal selection of conductor size in radial distribution networks. Engineering Science and Technology, an International Journal, 20(2), 391-402.
- Abramson, D., & Dang, H. (1993). School timetables: A case study in simulated annealing. In Applied simulated annealing (pp. 103-124). Springer, Berlin, Heidelberg.
- Arora, J. S., Huang, M. W., & Hsieh, C. C. (1994). Methods for optimization of nonlinear problems with discrete variables: a review. Structural optimization, 8(2-3), 69-85.
- Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Computers & Structures, 169, 1-12.
- Askarzadeh, A. (2016). Electrical power generation by an optimised autonomous PV/wind/tidal/battery system. IET Renewable Power Generation, 11(1), 152-164.
- Bolaji, A. L. A., Khader, A. T., Al-Betar, M. A., & Awadallah, M. A. (2014). University course timetabling using hybridized artificial bee colony with hill climbing optimizer. Journal of Computational Science, 5(5), 809-818.
- Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. European Journal of Operational Research, 140(2), 266-280.
- Carrasco, M. P., & Pato, M. V. (2004). A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. European Journal of Operational Research, 153(1), 65-79.
- Doulaty, M., Derakhshi, M. F., & Abdi, M. (2013). Timetabling: A state-of-the-art evolutionary approach. International Journal of Machine Learning and Computing, 3(3), 255.
- El-Sherbiny, M. M., Zeineldin, R. A., & El-Dhshan, A. M. (2015). Genetic algorithm for solving course timetable problems. International Journal of Computer Applications, 124(10).
- Eludire, A. A., & Akanbi, C. O. (2017). A Conceptual Approach to Resources Allocation Scheduling. Journal of Advances in Mathematics and Computer Science, 1-12.
- Gendreau, M., & Potvin, J. Y. (2005) "Metaheuristics in combinatorial optimization", Annals of Operations Research, 140(1), 189-213.
- Javidi, A., Salajegheh, E. & Salajegheh, J. (2019) "Enhanced crow search algorithm for optimum design of structures", Applied Soft Computing, 77, 274-289.
- Lazarev, A., & Nekrasov, I. (2017) "Mathematical models for enterprise resource scheduling: Complexity of key approaches to problem formulation. In 2017 Tenth

International Conference Management of Large-Scale System Development (MLSD) (pp. 1-5). IEEE.

- Liu, D., Liu, C., Fu, Q., Li, T., Imran, K. M., Cui, S., & Abrar, F. M. (2017). ELM evaluation model of regional groundwater quality based on the crow search algorithm. Ecological indicators, 81, 302-314.
- Lü, Z., & Hao, J. K. (2010). Adaptive tabu search for course timetabling. European Journal of Operational Research, 200(1), 235-244.
- Mohammadi, F., & Abdi, H. (2018). A modified crow search algorithm (MCSA) for solving economic load dispatch problem. Applied Soft Computing, 71, 51-65.
- Phillips, A. E., Waterer, H., Ehrgott, M., & Ryan, D. M. (2015). Integer programming methods for large-scale practical classroom assignment problems. Computers & Operations Research, 53, 42-53.
- Pimmer, M., & Raidl, G. R. (2013). A timeslot-filling heuristic approach to construct high-school timetables. In Advances in Metaheuristics (pp. 143-157). Springer, New York, NY.
- Raghavjee, R., & Pillay, N. (2012, November). A comparison of genetic algorithms and genetic programming in solving the school timetabling problem. In 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC) (pp. 98-103). IEEE.
- Randall, M., & Abramson, D. (2001). A general meta-heuristic based solver for combinatorial optimisation problems. Computational optimization and applications, 20(2), 185-210.
- Rizk-Allah, R. M., Hassanien, A. E., & Bhattacharyya, S. (2018). Chaotic crow search algorithm for fractional optimization problems. Applied Soft Computing, 71, 1161-1175.
- Shaker, K., & Abdullah, S. (2009, October). Incorporating great deluge approach with kempe chain neighbourhood structure for curriculum-based course timetabling problems. In 2009 2nd Conference on Data Mining and Optimization (pp. 149-153). IEEE.
- Smith, K. A., Abramson, D., & Duke, D. (2003). Hopfield neural networks for timetabling: formulations, methods, and comparative results. Computers & industrial engineering, 44(2), 283-305.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), 67-82.
- Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. C. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. European Journal of Operational Research, 203(3), 550-558.
- Zolghadr-Asli, B., Bozorg-Haddad, O., & Chu, X. (2018). Crow search algorithm (CSA). In Advanced Optimization by Nature-Inspired Algorithms (pp. 143-149). Springer, Singapore.

A Novel Differential Evolution Optimizer for Solving Constrained Non-Linear Integer and Mixed-Integer Global Optimization Problems

[a]Ali Khater Mohamed, [b,c]Ali Wagdy Mohamed

Abstract

This paper proposes an enhanced directed Differential Evolution algorithm (MI-EDDE) to solve global constrained optimization problems that consist of mixed / non-linear integer variables. The MI-EDDE (Mixed Integer – Enhanced Directed Differential Evolution) algorithm, which is based on the constraints violation, introduces a new mutation rule that sort all individuals ascendingly due to their constraint violations (cv) value and then the new proposed mutation scheme is applied. This new mutation scheme shown to enhance the global and local search capabilities and increases the convergence speed. Eighteen test problems with different features are tested to evaluate the performance of MI-EDDE, and a comparison is made with four state-of-the-art evolutionary algorithms. The results show superiority of MI-EDDE to the four algorithms in terms of the quality, efficiency and robustness of the final solutions. Moreover, MI-EDDE shows a superior performance in solving two high dimensional problems and finding better solutions than the known optimal solution.

Keywords: Evolutionary computation**,** Differential evolution, Global optimization**,** Novel mutation, handling constraints.

1 Introduction

Optimization is the selection of the best result (regarding some criterion) from some set of available alternatives for a given problem. Real life optimization problems arise in different areas such as engineering and science, in which the optimization problems involve different types of objective functions and constraints with different types of variables (Mohamed and Sabry, 2012). Most of these problems involve continues variables and integer variables, so it can be formulated as mixed integer non-linear programming problems (MINLP), and classified as a class of NP complete problems, and can be considered as difficult problems due to their combinatorial nature (Costa and Oliveira, 2001; Lin, Hwang and Wang, 2004). This class of optimization problems arises very often in real life problems and applications (Sandgren, 1990; Maldonado et al., 2014). Due to the objective function and constraints nature, there exist two types of MINLP: convex MINLP which involves the minimization of convex objective function over a convex feasible region, and non-convex MINLP that considers the minimization of objective function and/or the continuous relaxation of a non-convex feasible region (Hsieh, Lee and You, 2015). According to the difficult characteristics mentioned above and such diverse area, many attempts have been made during past decades to solve MINLP using many types of deterministic or exact approaches and stochastic or meta-heuristic algorithms (Ng et al., 2005).

----

a. Department of Computer Science, Faculty of Computer Science, October University for Modern Science and Arts (MSA), 6th October City, Giza 12451, Egypt. akhater@msa.eun.eg

b. Operations Research Department, Faculty of Graduate studies for Statistical Research, Cairo University, Giza 12613, Egypt [c]Wireless Intelligent Networks Center (WINC), Nile University, Egypt. aliwagdy@gmail.com

Various deterministic methods for solving convex MINLP (Abhishek, Leyffer and Linderoth, 2010; Belotti *et al.*, 2013) and several exact methods for solving non-convex MINLP (Trespalacios and Grossmann, 2014) are addressed, but there are two main drawbacks for those methods. First: they require that the objective function is differentiable, second: they assume the convexity and continuity of the objective function and constraints. So, they can be applied only for well-structured problems. Additionally, most of those methods can deal with small scale problems in a reasonable time. But, for large problems, there is no efficient algorithm to obtain a feasible solution in a reasonable time, because MINLP are considered NP-hard problems. Thus, many limitations are considered in these methods. So, there was important need to develop stochastic approaches to solve high dimensional, high combinatorial and high nonlinear MINLP.As an alternative, metaheuristics have attracted a wide range of researchers, with a main advantage of tackling the drawbacks of deterministic/exact methods mentioned above, as metaheuristics do not require the objective function and constraints to be differentiable, convex, and continuous. Despite these algorithms have no guarantee for the global solution, but they can obtain the near optimal solutions in a reasonable time. However, the parameter tuning is considered as the main drawback of these algorithms. Motivated by metaheuristic's success, more research is needed to be able to find solutions for MINLP. Differential evolution (Storn and Price, 1997), considered as the most recent EAs, is a stochastic population-based search method for solving optimization problems. DE only differs from EAs in the sense that in DE, distance and direction information is used to guide the search process. A comprehensive review for DE and its applications is made by (Das and Suganthan, 2011). Consequently, this analysis and discussion motivated us to present an enhanced mutation scheme that is able to balance the exploration capability and exploitation tendency and to enhance the convergence speed of the algorithm without using extra parameters. MI-EDDE is tested and analyzed by solving a set of 18 test problems with different characteristics. The proposed algorithm shows superior performance compared to four state-of-art evolutionary algorithms. The paper is organized as follows. The formulation of the MINLPs is introduced in section 2. Section 3 introduces the DE algorithm. The proposed MI-EDDE is presented in detail in section 4. The computational results of these benchmark problems are reported and discussed in section 5. Finally, the paper is concluded in section 6.

2. Problem Statement and constrained handling

Generally, MINLP can be expressed as follows: (Gao, et al. (2011) & Lin et al. (2004))

Min $f(X,Y)$ **,** $X=(x_1,x_2,....,x_C)\in R^C$**,** $Y=(y_1,y_2...,y_I)\in R^I$ **(1)**

subject to:

$$g_j(X,Y)\le 0,\ j=1,...,n \quad \textbf{(2)}$$

$$h_k(X,Y)=0, k=n+1,...,m \quad (3)$$

Where $(X,Y)\in\Omega\subset S^{C+I}$, *C*: number of continuous variables, *I*: number of integer variables, Ω: the feasible region, and S is (*C*+*I*) dimensional rectangular space in $R^C\cup R^I$ with parametric constraints $l_i\le x_i\le u_i, 1\le i\le C$ where l_i and u_i are lower and upper bounds for a continuous decision variable x_i, $l_i\le y_i\le u_i, 1\le i\le I$ where l_i and u_i are lower and upper bounds for an integer decision variable y_i, respectively. For an inequality constraint that satisfies $g_j(X,Y) \le 0,\ j = 1,...,n$ at any points $(X,Y) \in \Omega$, is active at that point. All equality constraints $h_k(X,Y) = 0, k = n+1,...,m$ are considered active at all points of Ω. Dealing only with inequality constraints is the main feature for most constraint-handling approaches used with EAs. Therefore, transformation of equality constraints into equality constraints of the form $\left|h_j(\vec{x},\vec{y})\right| - \varepsilon \le 0$ is required, where ε is the tolerance level. Deb's (Deb, 2000) constraint handling procedure is used in order to handle the constraints. Obviously, the constraint violation vector for any solution is considered as m dimensions' vector as there are m constraints in the formulation of MINLP presented. Using a tolerance (ε), the constraint violation of a decision vector on the jth constraint is given by

$$\mathrm{cv}_j(\mathrm{X},\mathrm{Y}) = \begin{cases} \max(0, \mathrm{g}_j(\mathrm{X},\mathrm{Y}), \mathrm{j}=1,...,\mathrm{n} \\ \max(0, \left|\mathrm{h}_j(\mathrm{X},\mathrm{Y})\right| - \varepsilon), \mathrm{j}=\mathrm{n}+1,...,\mathrm{m} \end{cases} \tag{4}$$

If a decision vector or an individual (X,Y) satisfies the jth constraint, $cv(X,Y)$ is equal to zero. As discussed (Venkatraman and Yen, 2005), a normalization for each constraint violation is made. so, the maximum violation of each constraint is given by

$$cv_{\max}^{j} = \max_{X \in S} cv_j(X,Y) \tag{5}$$

Normalized constraint violations, obtained by using the maximum constraint violation values, are added together to get a scalar constraint violation $cv(X,Y)$ for that individual, which takes a value between 0,1.

$$cv(X,Y) = \frac{1}{m} \sum_{j=1}^{m} \frac{cv_j(X,Y)}{cv_{\max}^{j}} \tag{6}$$

3. Differential Evolution

In simple DE, DE/rand/1/bin (Fan and Lampinen, 2003), an initial population of NP vectors $\vec{X}_i, i = 1, 2, ..., NP$, is uniformly generated within lower and upper boundaries (x_j^L, x_j^U). Mutation and crossover are used to generate a trial vector by evolving the individuals of the population. The parent and its trial vector compete in order to select the vector to survive to the next generation (Mohamed and Suganthan, 2018). The steps of DE are:

3.1. Initialization of a population

Initial population in DE, as the starting point for the process of optimization, is created by assigning a random chosen value for each decision variable in every vector, as indicated in equation (7)

$$x_{ij}^{0} = l_j + rand_j(u_j - l_j) \tag{7}$$

Where l_j, u_j: the lower and upper boundaries for x_j, $rand_j$: a random number uniform [0, 1].

3.2. Mutation

A mutant vector v_i^{G+1} is generated for each target vector X_i^G at generation G according to equation (8)

$$v_i^{G+1} = X_{r_1}^G + F * (X_{r_2}^G - X_{r_3}^G), r_1 \neq r_2 \neq r_3 \neq i \tag{8}$$

Where $r_1, r_2, r_3 \in \{1,.., NP\}$ are randomly. The mutation factor $F \in [0,2]$ according to (Mohamed and Suganthan, 2018). A new value for the component of mutant vector is generated using (7) if it violates the boundary constraints.

3.3. Recombination (crossover)

Crossover has two popular types: binomial and exponential, in the binomial crossover, a mix between the target vector and the mutated vector using (9), to yield the trial vector u_i^{G+1}.

$$\mathrm{u}_{\mathrm{ij}}^{\mathrm{G+1}} = \begin{cases} \mathrm{v}_{\mathrm{ij}}^{\mathrm{G+1}}, \mathrm{rand(j)} \leq \mathrm{CR} \ \text{ or } \ \mathrm{j=rand(i)}, \\ \mathrm{x}_{\mathrm{ij}}^{\mathrm{G}}, \ \mathrm{rand(j)} > \mathrm{CR} \ \text{ and } \ \mathrm{j \neq rand(i)}, \end{cases} \tag{9}$$

Where $j = 1, 2, ..., D$, $\mathrm{rand(j)} \in [0,1]$ is the *jth* evaluation of a uniform random generator number. $CR \in [0,1]$ is the crossover rate, $\mathrm{rand(i)} \in \{1, 2, ..., \mathrm{D}\}$ is a randomly chosen index which ensures that u_i^{G+1} gets at least one element from v_i^{G+1}; otherwise, the population remains without change. In the exponential crossover, a starting index l and a number of components w are chosen randomly from the ranges {1, D} and {1, D-l} respectively. The values of variables in locations l to l+w from v_i^{G+1} and the remaining locations from the x_i^G are used to produce the trial vector u_i^{G+1}.

3.4. Selection

Greedy scheme for fast convergence of DE, the child u_i^{G+1} is compared with its parent x_i^G to select the better for the next generation according to the selection scheme in equation (10).

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, \ f(u_i^{G+1}) \leq f(x_i^G) \\ x_i^G, \quad \text{otherwise} \end{cases} \tag{10}$$

Standard DE algorithm is given in figure 1

Begin
G = 0
generate an initial population randomly $\vec{x}_i^G \; i = 1, \ldots, NP$
Calculate $f(\vec{x}_i^G) \; i = 1, \ldots, NP$
For G=1 to GEN **Do**
For i=1 to NP **Do**
Randomly select $r1 \neq r2 \neq r3 \neq i \in [1, NP]$
j_{rand}= randint(1,D)
For j=1 to D **Do**
If ($rand_j[0,1] < CR$ **or** $j = j_{rand}$) **Then**
$u_{i,j}^{G+1} = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$
Else
$u_{i,j}^{G+1} = x_{i,j}^G$
End If
End For
Verify Boundary constraints
If($f(\vec{u}_i^{G+1}) \leq f(\vec{x}_i^G)$)**Then**
$\vec{x}_i^{G+1} = \vec{u}_i^{G+1}$
Else
$\vec{x}_i^{G+1} = \vec{x}_i^G$
End If
End For
G=G+1
End For
End

Fig 1 Pseudo code of the standard DE

4. The proposed algorithm (MI-EDDE):

In this section, enhanced directed DE algorithm (MI-EDDE) is discussed in detail, the constraint-handling mechanism and integer variables handling used in this paper are described & the pseudo-code of the algorithm is explained.

4.1 Novel mutation scheme

The fundamental strategy DE/rand/1 developed by Storn and Price (Storn and Price, 1997) is considered as the most successful scheme in the literature (Wagdy Mohamed, Sabry and Farhat, 2011; Mohamed, 2018). In this scheme, a difference vector for two randomly selected vectors is added to a third randomly chosen vector called the base vector. The strategy can maintain the population diversity and global search ability without any bias for any search direction, but with slower convergence speed. DE/rand/2 is a scheme with an extra difference vector that may lead to better perturbation which increases the exploration ability of the search space. However, the two mentioned schemes use the best solution found so far to increase the local search tendency to fasten the convergence speed of the algorithm, but the diversity and the exploration capability of the algorithm may get slower or completely lost in a very small number of generations, that causes stagnation and/or premature convergence. Most of the recent algorithms tried to overcome the weakness of DE/rand/1 and DE/rand/2, by combining different control parameter settings with different strategies of trial vector generation, so it can deal with various problems with different characteristics(Mallipeddi *et al.*, 2011). To improve the performance of the optimization process, an adaptive control parameter algorithm, named JADE, with a new mutation strategy "DE/current-to-*p*best" is introduced by (Zhang and Sanderson, 2009). Consequently, challenge is still opened for proposing new mutation strategies that is able to increase the possible achievement of obtaining promising results in complex and large-scale optimization problems and also improving the search ability. Therefore, this research proposes a new mutation scheme based on the constraint violation as a metric for evaluation only with a view of balancing the global exploration ability and the local exploitation tendency and enhancing the convergence rate of the algorithm (A. K. Mohamed and Mohamed, 2019; A. W. Mohamed and Mohamed, 2019). Firstly, all individuals are sorted ascendingly according to their constraint violations *(cv)* values. Then the population is divided into three clusters (best, better and worst) with sizes 100*p*%, (NP-2) * 100*p*% and 100*p*% respectively. Where *p* is the proportion of the partition with respect to the total number of individuals in the population (NP). MI-EDDE selects three random individuals, one of each partition to implement the mutation process. The proposed mutation vector is generated as follows:

$$v_i^{G+1} = x_r^G + F \cdot (x_{p\_best}^G - x_{p\_worst}^G) \qquad (11)$$

Where x_r^G is a random chosen vector from the middle (*NP-2*(100*p*%)) individuals, $x_{p\_best}^G$ and $x_{p\_worst}^G$ are randomly chosen as one of the top and bottom 100*p*% individuals in the current population, respectively, with $p \in (0\%, 50\%)$, F is the mutation factor that is generated due to uniform distribution in the interval (0.1,1). Learning from the best individual and avoiding the position of the worst individual is the main idea of the proposed novel mutation. It is obvious from Eq. (11), that there are two benefits from the proposed mutation. First: the current individual is attracted to the best solution, not the best-so-far, thus avoiding the premature convergence. Second: the position of the worst individual is avoided, which enhances the exploitation by guiding the search process to some new promising regions. i.e. it concentrates the exploitation of some sub-regions of the search space. It's obvious that following the same directions of the best vectors and the opposite directions for the worst vectors can lead to the global solution easily. So, sharing and utilizing the information of the best and worst vectors in the proposed directed mutation scheme can balance the global exploration and local exploitation.

4.2 Constraint handling

From the literature, Deb's feasibility rules and multi-objective optimization technique are the most famous rules that used to handle the constraints during last decade. However, to balance both the population diversity and convergence speed of the proposed algorithm, the constraint violation (i.e. cv>0) and/or constraint satisfaction (i.e. cv<=0) as a metric for direct evaluation of all individuals during the optimization process is only considered to form the novel mutation strategy. Thus, the comparison of the individuals and the selection process are only based on their constraint violation or constraint satisfaction. In fact, it can be deduced from the real-world problems and the benchmark problems that will be tested in section 5 that there are two benefits of using (cv) only as a metric for evaluation unlike Deb's feasibility rule and multi-objective optimization technique. Firstly, most of the real-world problems as well as the benchmark problems contain equality constraints. Consequently, the ratio between the feasible region and the search space is very small or approximately zero which means that there is no need to use objective function as a metric for evaluation as all individuals are infeasible. Alternatively, it must use constraint violation metric for evaluation to guide the search process to the very tiny feasible region. Secondly, on the other hand, in case of high ratio between the feasible region and search space, i.e. most of individuals are feasible, using the objective function only as a metric for evaluation will significantly increase the possibility of plunging into local minimum. As aforementioned in section 2, a transformation for the equality constraints to inequality constraints by a tolerance value ε is made. In the experiments, the tolerance value ε is adjusted like (Mezura-Montes and Coello Coello, 2005). The parameter ε decreases from generation to generation using the following equation:

$$\varepsilon(g+1) = \begin{cases} \dfrac{\varepsilon(g)}{1.0168}, if\, \varepsilon > 10^{-4} \\ 10^{-4}, otherwise \end{cases} \tag{12}$$

Where g: the generation number and the initial $e(0) = 5$ with all problems. Regarding boundary-handling method, the re-initialization method is used (see Eq. 7). The pseudo-code of MI-EDDE is in Fig. 2.

4.3 Integer Variables Handling

DE in its form is able only to handle COP with continuous decision variables. However, a very few attempts, in the literature, for handling the integer variables by transforming the canonical form of DE are recorded (Li and Zhang, 2014; Mohamed, 2017). Two simple modifications are proposed by MI-EDDE to enhance the convergence speed, by searching directly for the integer variables in the integer space and to avoid the unnecessary searches in real number field. First: rounding the initial population and the boundary constraint verification to the nearest integer by using the round operator. Second: rounding the trial vector after the proposed mutation and cross over is applied. Therefore, the initialization and mutation are as follows:

Initialization: $x_{ij}^{0} = round\,(l_j + rand_j * (u_j - l_j))$.

boundary constraint verification: $x_{ij}^{G+1} = round\,(l_j + rand_j * (u_j - l_j))$.

New mutation: $v_i^{G+1} = round(x_r^G + F * (x_{p\_best}^G - x_{p\_worst}^G))$

```
Begin
  G=0
  Generate an initial population randomly x_i^G ∀i, i = 1,..., NP, use cr= 0.5
  Evaluate cv(x_i^G), ∀i, i = 1,..., NP
  For G=1 to GEN Do
    Calculate (ε) using equation (13).
    For i=1 to NP Do
      F =rand[0.1,1]
      j_rand= randint(1,D)
```

Choose $x_{p\_best}^{G}$ randomly from (top individuals partition)
Choose $x_{p\_worst}^{G}$ randomly from (bottom individuals partition)
Choose x_{r}^{G} randomly from (middle individuals partition)
For j=1 to D **Do**
If ($rand_i[0,1]$< CR **or** j= j_{rand}) **Then**
$v_{i}^{G+1} = x_{r}^{G} + F \cdot (x_{p\_best}^{G} - x_{p\_worst}^{G})$
Else
$u_{ij}^{G+1} = x_{ij}^{G}$
End If
End For
Verify Boundary constraints
If ($\vec{u}_{i}^{G+1}$ is better than $\vec{x}_{i}^{G}$ (based on Deb's selection criteria)) **Then**
$\vec{x}_{i}^{G+1} = \vec{u}_{i}^{G+1}$
Else
$\vec{x}_{i}^{G+1} = \vec{x}_{i}^{G}$
End If
Evaluate $cv(\vec{x}_{i}^{G+1}), \forall i, i = 1,..., NP$.
End For
G=G+1
End For
End

Fig 2 Description of MI_EDDE algorithm

5. Experiments and Discussion:

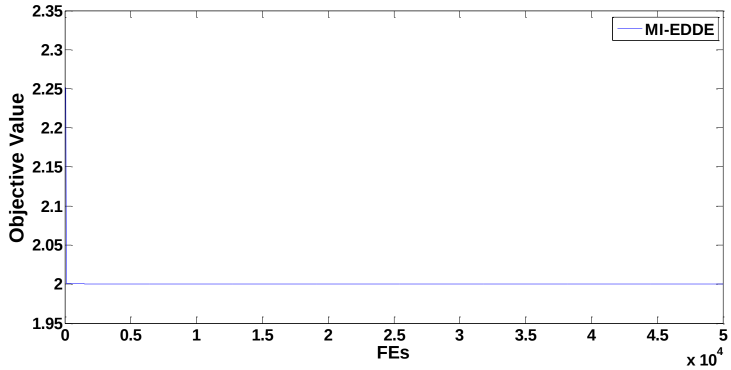
A set of 18 integer and mixed integer COP in (Deep *et al.*, 2009) are tested to confirm the feasibility and efficiency of MI-EDDE. All problems are nonlinear except problem 16, where the number of decision variables varies from 2 to 100. Although, based on a self-check, we found few differences in problems 6,16 and 17 in (Deep *et al.*, 2009) and the original references. So, table 1 represents the corrections of these problems. The three control parameters of MI-EDDE, population size *NP*, cross over rate *Cr*, and the scale factor *F* are set to 50, 0.5, and uniform (0.1,1) respectively. The maximum number of generations is set to 1000, so the maximum number of function evaluations FEs is 50,000 for all test problems. The experiments were executed on an Intel Core i5-CPU ,1.80GHz and 4GB-RAM. The code of MI-EDDE is coded using MATLAB. 50 independent runs are performed for each problem and statistical results including the best, median, mean, worst and the standard deviation are provided in table 2. Moreover, the success of each run is considered only if [abs(f(x)-f(x*)) <0.01], where f(x) is the achieved value of the objective function, x* is the best solution or optimal value found, and f(x*) is the known best or optimal value function. The convergence figures for some problems are presented in Fig 3.

Table 1: Better solutions by MI-EDDE than MI-LXPM [40]

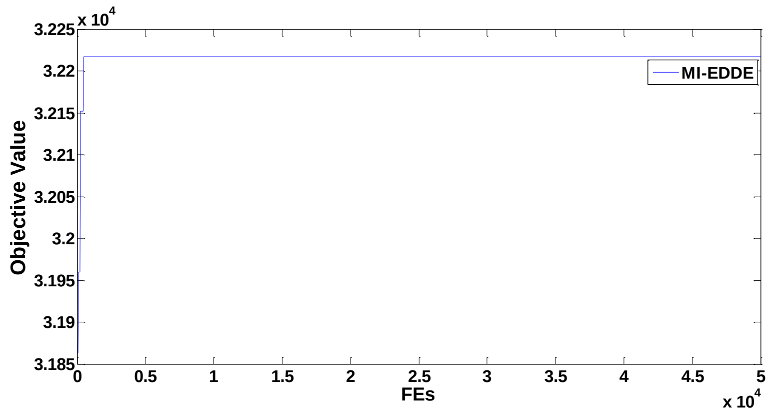
Problem	Solution	O. F
10	**1.0 ,1.0 ,0.0 ,1.0 ,0.0 ,1.0 ,1.0 ,0.0**	0.953197
16	**99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99**	1352439
17	**97, 99, 98, 99, 98, 5, 2, 18, 7, 5, 99, 98, 99, 98, 99, 34, 75, 75, 96, 95, 94, 99, 98, 99, 97, 99, 99, 99, 98, 97, 99, 5, 99, 99, 99, 99, 1, 2, 98, 99, 98, 99, 1, 47, 82, 23, 9, 5, 3, 9, 98, 98, 98, 95, 96, 99, 77, 94, 77, 84, 95, 95, 97, 79, 62, 99, 98, 41, 90, 84, 96, 97, 72, 96, 85, 60, 93, 97, 95, 95, 97, 93, 91, 92, 98, 64, 60, 89, 53, 38, 95, 63, 96, 3, 34, 90, 67, 65, 24, 36**	304151776

Table 2: Results of MI-EDDE in 50 runs

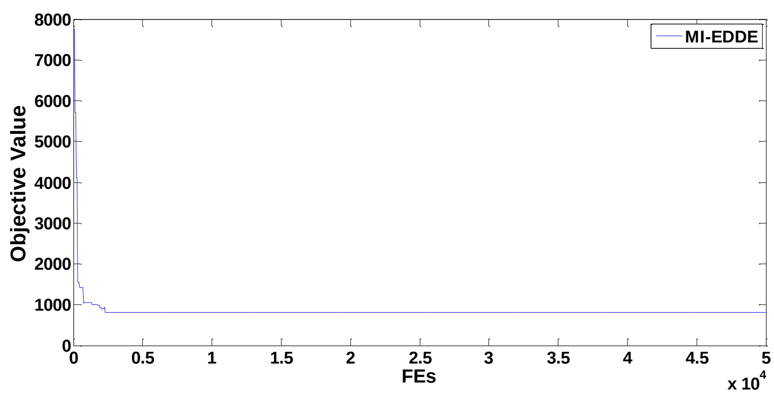
Problem	known	Best	Worst	Mean	Median	S. D
1	2.000E+00	2.000E+00	2.000E+00	2.000E+00	2.000E+00	0.000E+00
2	2.124E+00	2.124E+00	2.124E+00	2.124E+00	2.124E+00	0.000E+00
3	1.077E+00	1.077E+00	1.077E+00	1.077E+00	1.077E+00	0.000E+00
4	-6.962E+03	-6.962E+03	-6.962E+03	-6.962E+03	-6.962E+03	0.000E+00
5	-6.800E+01	-6.800E+01	-6.800E+01	-6.800E+01	-6.800E+01	0.000E+00
6	-6.000E+00	-6.000E+00	-6.000E+00	-6.000E+00	-6.000E+00	0.000E+00
7	9.925E+01	9.924E+01	9.924E+01	9.924E+01	9.924E+01	0.000E+00
8	3.557E+00	3.557E+00	3.557E+00	3.557E+00	3.557E+00	0.000E+00
9	3.222E+04	3.222E+04	3.222E+04	3.222E+04	3.222E+04	0.000E+00
10	9.435E-01	9.532E-01	9.532E-01	9.532E-01	9.532E-01	0.000E+00
11	8.000E+00	8.000E+00	8.000E+00	8.000E+00	8.000E+00	0.000E+00
12	1.400E+01	1.400E+01	1.500E+01	1.426E+01	1.400E+01	4.430E-01
13	-4.263E+01	-4.263E+01	-4.263E+01	-4.263E+01	-4.263E+01	0.000E+00
14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
15	8.070E+02	8.070E+02	8.070E+02	8.070E+02	8.070E+02	0.000E+00
16	1.030E+06	1.352E+06	1.351E+06	1.352E+06	1.352E+06	4.770E+02
17	3.031E+08	3.042E+08	3.041E+08	3.042E+08	3.042E+08	6.145E+02
18	1.000E+00	1.000E+00	9.999E-01	1.000E+00	1.000E+00	3.490E-06



F1



F9



F15

Fig 3 Convergence graph (median curves) of MI-EDDE on some of the test functions.

The success percentage (Ps), the ratio between the number of successful runs and total number of runs, and the number of function evaluations FEs in the successful runs for each test problem including the best, worst, mean, median and standard deviation are provided in Table 3.

Table 3: Number of FEs, Successful rate (Ps) provided by MI-

Problem	Best	Worst	Mean	Median	Std.	Ps
1	5.00E+01	1.40E+03	5.15E+02	5.00E+02	3.55E+02	100
2	5.00E+01	8.50E+02	2.00E+02	1.50E+02	1.54E+02	100
3	9.50E+02	2.70E+03	1.64E+03	1.55E+03	3.83E+02	96
4	1.23E+04	3.43E+04	1.70E+04	1.65E+04	2.93E+03	100
5	1.00E+02	8.00E+02	3.74E+02	3.50E+02	2.01E+02	100
6	5.00E+01	5.00E+01	5.00E+01	5.00E+01	0.00E+00	100
7	1.25E+03	6.10E+03	2.89E+03	2.65E+03	1.27E+03	100
8	2.20E+03	9.90E+03	4.89E+03	4.65E+03	1.36E+03	98
9	3.00E+02	1.55E+03	8.96E+02	9.00E+02	2.59E+02	100
10	5.00E+01	1.50E+02	5.70E+01	5.00E+01	2.26E+01	100
11	5.00E+01	7.00E+02	2.12E+02	1.25E+02	1.90E+02	100
12	1.50E+02	5.00E+03	1.69E+03	1.43E+03	9.82E+02	98
13	5.00E+01	5.00E+01	5.00E+01	5.00E+01	0.00E+00	100
14	2.50E+02	1.50E+03	8.31E+02	8.25E+02	2.13E+02	100
15	2.45E+03	4.95E+03	3.58E+03	3.60E+03	5.89E+02	100
16	2.00E+02	8.00E+02	5.06E+02	5.50E+02	1.38E+02	100
17	1.00E+02	1.30E+03	7.33E+02	7.25E+02	2.23E+02	100
18	5.00E+01	6.00E+02	2.20E+02	2.50E+02	1.24E+02	100

An experiment is conducted to evaluate the benefits of MI-EDDE and the results are compared against four state-of-the-art evolutionary algorithms (MI-LXPM (Deep et al., 2009), RST2ANU (Mohan and Nguyen, 1999), AXNUM (Li and Gen, 1996), MIPDE (Gao, Ren and Gao, 2011)).

Table 4 represents the experimental results for the five algorithms, where Ps: the success rate, percentage of the successful runs to total runs and mean FEs: the average number of objective function evaluations used by the algorithm in successful runs. The results for (MI-LXPM, RST2ANU and AXNUM) were taken from (Deep et al., 2009) and the results for MIPDE was taken from (Gao, Ren and Gao, 2011). Only 14 benchmark problems were solved by MIPDE, with exception to problems 14, 16,17 and 18. It is obvious from results in Table 4, that MI-EDDE has obtained the optimal solution with a success rate of 100% for 15 test functions, and success rates 96%, 98% and 98% for problems 3,8 and 12 respectively, which indicates the stability and robustness of MI-EDDE, with no need for the fine tuning of the parameters. Moreover, to measure the efficiency of MI-EDDE, the sum of the mean FEs for the five algorithms are compared, and the one produced by MI-EDDE is found to be the smallest. Therefore, MI-EDDE is considered the most efficient with the smallest total mean (FEs). Fig.4 show the performance of the five algorithms in terms of total Mean FEs for all test functions. Generally, MI-EDDE proves the superiority to all the compared EAs algorithms in terms of success rate and efficiency.

Table 4: Results obtained by using MI-EDDE, MI-LXPM, RST2ANU, AXNUM and MIPDE algorithms											
		MI-EDDE		MI-LXPM		RST2ANU		AXNUM		MIPDE	
problem	Dim	Mean FEs	PS	Mean FEs	PS	Mean FEs	PS	Mean FEs	PS	Mean FEs	PS
1	2	515	100	172	84	173	47	1728	86	9493	100
2	2	200	100	64	85	657	57	82	67	1683	100
3	3	1641.67	96	18608	43	221129	4	65303	35	9828	100
4	2	16950	100	10933	95	1489713	2	45228	82	10644	97
5	3	374	100	671	100	2673	75	13820	95	565	100
6	4	50	100	84	100	108	100	432	100	36	100
7	3	2892	100	7447	59	-	-	16077	45	1265	100
8	7	4891.837	98	3571	41	180859	15	1950	3	13310	95
9	5	896	100	100	100	189	100	4946	100	451	100
10	8	57	100	258	93	545	100	700	33	1375	100
11	5	212	100	171	100	2500	100	863	97	204	100
12	7	1690.217	98	299979	71	6445	29	380115	19	6293	84
13	2	50	100	77	99	35	100	456	91	27	100
14	3	831	100	78	100	214	100	1444	100	-	-
15	5	3578	100	2437	92	3337	19	267177	9	4195	97
16	40	506	100	1075	100	1114	100	2950	100	-	-
17	100	733	100	600	100	2804	100	600	100	-	-
18	8	220	100	250	100	697	100	256	102	-	-
Total mean FEs		36287.724		346575		1913761		804127		59369	

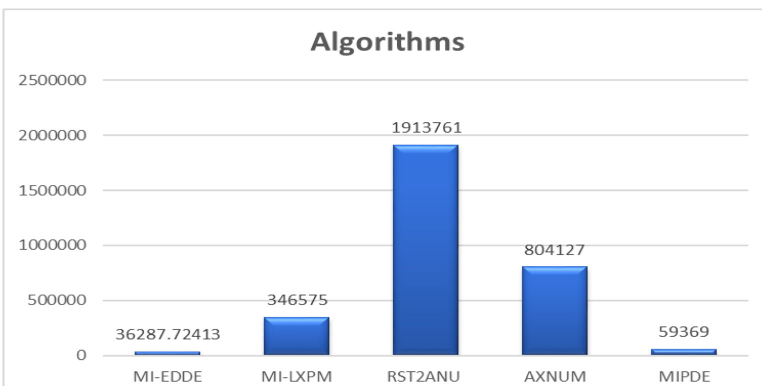


Fig 4 The performance of MI-EDDE, RST2ANU, MIPDE, MI-LXPM & AXNUM in terms of total Mean of FEs.

6. Conclusion

The paper represented an enhanced directed Differential Evolution algorithm for solving constrained, integer and mixed integer optimization problems MI-EDDE. The MI-EDDE algorithm, which is based on the constraints violation (*cv*), a new mutation rule is introduced, which is based on dividing the population into three clusters (best, better and worst) with sizes

100p%, (NP-2) * 100p% and 100p% respectively, Where p is the proportion of the partition with respect to the total number of individuals in the population (NP) and then selecting three random individuals, one of each partition to implement the mutation process. This new mutation scheme is shown to enhance the global and local search capabilities and to increase the convergence speed of the proposed algorithm. A set of 18 test problems with different characteristics are tested to evaluate the performance of MI-EDDE, and the results are compared with four state-of-the-art evolutionary algorithms (MI-LXPM, RST2ANU, AXNUM and MIPDE). The results showed that the proposed algorithm is superior to the four algorithms in terms of the quality, efficiency and robustness of the final solutions. Moreover, MI-EDDE shows a good performance in solving two high dimensional problems and finding better solutions than the optimal known. So, MI-EDDE is superior to the compared algorithms. Additionally, MI-EDDE performance will be investigated by solving constrained and unconstrained multi objective optimization problems.

References

- Abhishek, K., Leyffer, S. and Linderoth, J. (2010) ‘FilMINT: An outer approximation-based solver for convex mixed-integer nonlinear programs’, INFORMS Journal on Computing, 22(4), pp. 555–567. doi: 10.1287/ijoc.1090.0373.

- Belotti, P. (2013) ‘Mixed-integer nonlinear optimization’, Acta Numerica. Cambridge University Press, pp. 1–131. doi: 10.1017/S0962492913000032.

- Costa, L. and Oliveira, P. (2001) ‘Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems’, Computers and Chemical Engineering. Pergamon, 25(2–3), pp. 257–266. doi: 10.1016/S0098-1354(00)00653-0.

- Das, S. and Suganthan, P. N. (2011) ‘Differential evolution: A survey of the state-of-the-art’, IEEE Transactions on Evolutionary Computation, 15(1), pp. 4–31. doi: 10.1109/TEVC.2010.2059031.

- Deb, K. (2000) ‘An efficient constraint handling method for genetic algorithms’, Computer Methods in Applied Mechanics and Engineering. Elsevier, 186(2–4), pp. 311–338. doi: 10.1016/S0045-7825(99)00389-8.

- Deep, K. (2009) ‘A real coded genetic algorithm for solving integer and mixed integer optimization problems’, Applied Mathematics and Computation. Elsevier, 212(2), pp. 505–518. doi: 10.1016/j.amc.2009.02.044.

- Fan, H. Y. and Lampinen, J. (2003) ‘A trigonometric mutation operation to differential evolution’, Journal of Global Optimization. Springer, 27(1), pp. 105–129. doi: 10.1023/A:1024653025686.

- Gao, Yuelin, Ren, Z. and Gao, Yang (2011) ‘Modified differential evolution algorithm of constrained nonlinear mixed integer programming problems’, Information Technology Journal, 10(11), pp. 2068–2075. doi: 10.3923/itj.2011.2068.2075.

- Hsieh, Y. C., Lee, Y. C. and You, P. S. (2015) ‘Solving nonlinear constrained optimization problems: An immune evolutionary based two-phase approach’, Applied Mathematical Modelling. Elsevier Inc., 39(19), pp. 5759–5768. doi: 10.1016/j.apm.2014.12.019.

- Li, H. and Zhang, L. (2014) 'A discrete hybrid differential evolution algorithm for solving integer programming problems', Engineering Optimization. Taylor and Francis Ltd., 46(9), pp. 1238–1268. doi: 10.1080/0305215X.2013.836637.

- Li, Y. X. and Gen, M. (1996) 'Nonlinear mixed integer programming problems using genetic algorithm and penalty function', in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 2677–2682. doi: 10.1109/icsmc.1996.561362.

- Lin, Y. C., Hwang, K. S. and Wang, F. S. (2004) 'A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems', Computers and Mathematics with Applications. Elsevier Ltd, 47(8–9), pp. 1295–1307. doi: 10.1016/S0898-1221(04)90123-X.

- Maldonado, S. (2014) 'Feature selection for Support Vector Machines via Mixed Integer Linear Programming', Information Sciences. Elsevier Inc., 279, pp. 163–175. doi: 10.1016/j.ins.2014.03.110.

- Mallipeddi, R. et al. (2011) 'Differential evolution algorithm with ensemble of parameters and mutation strategies', in Applied Soft Computing Journal. Elsevier, pp. 1679–1696. doi: 10.1016/j.asoc.2010.04.024.

- Mezura-Montes, E. and Coello Coello, C. A. (2005) 'A simple multimembered evolution strategy to solve constrained optimization problems', IEEE Transactions on Evolutionary Computation, 9(1), pp. 1–17. doi: 10.1109/TEVC.2004.836819.

- Mohamed, A. K. and Mohamed, A. W. (2019) 'Real-parameter unconstrained optimization based on enhanced AGDE algorithm', in Studies in Computational Intelligence. Springer Verlag, pp. 431–450. doi: 10.1007/978-3-030-02357-7_21.

- Mohamed, A. W. (2017) 'An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems', International Journal of Machine Learning and Cybernetics. Springer Verlag, 8(3), pp. 989–1007. doi: 10.1007/s13042-015-0479-6.

- Mohamed, A. W. (2018) 'A novel differential evolution algorithm for solving constrained engineering optimization problems', Journal of Intelligent Manufacturing. Springer New York LLC, 29(3), pp. 659–692. doi: 10.1007/s10845-017-1294-6.

- Mohamed, A. W. and Mohamed, A. K. (2019) 'Adaptive guided differential evolution algorithm with novel mutation for numerical optimization', International Journal of Machine Learning and Cybernetics. Springer Verlag, 10(2), pp. 253–277. doi: 10.1007/s13042-017-0711-7.

- Mohamed, A. W. and Sabry, H. Z. (2012) 'Constrained optimization based on modified differential evolution algorithm', Information Sciences. Elsevier, 194, pp. 171–208. doi: 10.1016/j.ins.2012.01.008.

- Mohamed, A. W. and Suganthan, P. N. (2018) 'Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation', Soft

Computing. Springer Verlag, 22(10), pp. 3215–3235. doi: 10.1007/s00500-017-2777-2.

- Mohan, C. and Nguyen, H. T. (1999) ‘Controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems’, Computational Optimization and Applications. Kluwer Academic Publishers, 14(1), pp. 103–132. doi: 10.1023/A:1008761113491.

- Ng, C. K. et al. (2005) ‘Discrete filled function method for discrete global optimization’, Computational Optimization and Applications. Springer, 31(1), pp. 87–115. doi: 10.1007/s10589-005-0985-7.

- Sandgren, E. (1990) ‘Nonlinear integer and discrete programming in mechanical design optimization’, Journal of Mechanical Design, Transactions of the ASME. American Society of Mechanical Engineers Digital Collection, 112(2), pp. 223–229. doi: 10.1115/1.2912596.

- Storn, R. and Price, K. (1997) ‘Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces’, Journal of Global Optimization. Springer Netherlands, 11(4), pp. 341–359. doi: 10.1023/A:1008202821328.

- Trespalacios, F. and Grossmann, I. E. (2014) ‘Review of mixed-integer nonlinear and generalized disjunctive programming methods’, Chemie-Ingenieur-Technik. Wiley-VCH Verlag, pp. 991–1012. doi: 10.1002/cite.201400037.

- Venkatraman, S. and Yen, G. G. (2005) ‘A generic framework for constrained optimization using genetic algorithms’, IEEE Transactions on Evolutionary Computation, 9(4), pp. 424–435. doi: 10.1109/TEVC.2005.846817.

- Wagdy Mohamed, A., Sabry, H. Z. and Farhat, A. (2011) ‘Advanced differential evolution algorithm for global numerical optimizatiom’, in ICCAIE 2011 - 2011 IEEE Conference on Computer Applications and Industrial Electronics, pp. 156–161. doi: 10.1109/ICCAIE.2011.6162123.

- Zhang, J. and Sanderson, A. C. (2009) ‘JADE: Adaptive differential evolution with optional external archive’, IEEE Transactions on Evolutionary Computation, 13(5), pp. 945–958. doi: 10.1109/TEVC.2009.2014613.