Available online at sjuoz.uoz.edu.krd

journals.uoz.edu.krd

Science Journal of University of Zakho

Vol. 12, No. 1, pp. 116 – 137 January-March, 2024

p-ISSN: 2663-628X
e-ISSN: 2663-6298

SOLVING STOCHASTIC TRANSPORTATION ELECTRICITY PROBLEM WITH FUZZY INFORMATION ON PROBABILITY DISTRIBUTION USING MATLAB PROGRAM

Halgurd Namiq Azeez [a,] , Abdulqader Othman Hamad Ameen [a]

[a] Mathematics Department, Faculty of Science and Health, Koya University, Koya, Kurdistan Region, Iraq.

(halgurd.namiq@koyauniversity.org)

Received: 6 Oct., 2023 / Accepted: 21 Feb., 2024 / Published: 30 Mar., 2024. https://doi.org/10.25271/sjuoz.2024.12.1.1212

ABSTRACT:

This study focuses on MATLAB code programs of the entire stages of solving Stochastic Transportation Linear Programming Problems with Fuzzy Uncertainty Information on Probability Distribution Space (STLPPFI) with its algorithm outlines. A MATLAB code program of STLPPFI problem solver with algorithm outlines are proposed to solve STLPPFI model problems, and it utilizes many concepts as Alpha-Cut technique, Truth Degrees technique, Linear Fuzzy Membership Function (LFMF), Trapezoidal Fuzzy Number $(TpFN)$, Triangular Fuzzy Number $(TrFN)$, Linear Fuzzy Ranking Function (LFRF), Expectation Weighted Summation technique (EWS) and analyzing cases via second condition test of alpha-cut technique. The STLPPFI problem solver is utlized to convert STLPPFI into its corresponding equivalent Deterministic Transportation Linear Programming Problem (DTLPP) via defuzzifying from fuzziness on probability distribution space and derandomization randomness of problem formulation respectively. In addition, Dual-Simplex algorithm method with Vogel Approximation Algorithm Method (VAM) are used to obtain optimal solution from DTLPP. All MATLAB code programs with their proposed algorithm outlines are new except Dual-Simplex and VAM. The MATLAB code program of STLPPFI problem solver are more efficient along with a numerical example on electricity field illustrating practicability of this proposed MATLAB code program with its algorithm. Finally, the solution procedure illustrates the MATLAB code program of the proposed method is practical and applicable in the fields of energy and industry as it facilitates the method of transforming the energy at the lowest cost, least time running and is commercially applicable. Comparative comments are provided between Dual-Simplex and VAM in solution process.

KEYWORDS: Alpha-Cut Technique Algorithm; Stochastic Transportation Problem; Fuzzy Information Probability Distribution; Truth Degrees Technique MATLAB Program; Expectation Weighted Summation Algorithm.

1. INTRODUCTION TO STLPPFI

Linear Programming Problems LPPs have been important subjects in solving optimization life problems, and the simplex algorithm method has improved to solve them. This improvement has led to extending Deterministic Transportation Linear Programming Problems DTLPP from LPP as a special case of it, and it is one of the most useful mathematical models. Additionally, DTLPP has three popular algorithms: the North-West Corner Method, Matrix Minima Cost Method (MMCM) and Vogel Approximation Method (VAM) for solving these kinds of problems and finding Basic Feasible Solution (BFS) for DTLPP. However, these methods are not sufficient to stop, so both algorithm methods, the Stepping Stone Method (SSM) and Modify Distribution Method (MODI) are used to find optimal solution for DTLPP. Sometimes the three methods mentioned above that are used to find BFS do not yield the optimal solution. The mean goals of DTLPP are to minimize the total transport costs, increasing the amount of transporting goods/objects as possible, increasing availability/production sources and decreasing non-useful demand/requirement endpoints as possible by removing unnecessary points and reducing waste points which does not appear if and only if availability/production sources and demand/requirement endpoints are balanced since the problem cannot be solved without a balance condition (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017).

Now, DTLPP steps to complexity and challenges will be added to it via discovering real-life problems as fuzziness and randomness of problem which are motivating to search for more efficient algorithms and program outlines to solve Stochastic Transportation Linear Programming Problems with Fuzzy Uncertainty Information on Probability Distribution Space STLPPFI. Where problem formulation has randomness in objective cost coefficients and fuzziness in linear inequalities polyhedral sets of information probability distribution space, we focus on transporting electricity power sector problems which has non-deterministic values of transporting cost coefficients and non-deterministic values of creation probability distribution parameters intervals with weight rank in importance of probability distribution. The (LPP) mathematical formulation contain a (max/min) linear objective function subject to set of linear constraint satisfies equations/inequalities, with non-negative unrestricted variable set. A DTLPP usually contains minimizing linear objective function or minimizing total transporting cost of shipping objects, subject to both availability and requirement linear constraints set where total availabilities satisfy total requirements, parameters set are non-negative (Reeb, James Edmund;Leavengood, Scott A, 2002; Sengamalaselvi, 2017; Sharma, 1974; Winston, Wayne L;Goldberg, Jeffrey B, 2004).

An Stochastic Transportation Linear Programming Problems (STLPP) is a DTLPP when parameters are random and represented by probability distributions (Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2010; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Ameen, 2015; Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling, 2015; Hamadameen, Abdulqader Othman;Hassan, Nasruddin, 2018). The probability distribution space $(\Omega, 2^{\Omega}, P)$ of an STLPP in many cases is unknown, undetermined, and un-specified since it has fuzzy information, unknown distribution, then should be

* Corresponding author

determined/specified as first step in solution procedures. Further, an STLPP's under fuzzy information on probability and described by fuzzy linear inequalities polyhedral set are called Stochastic Transportation Linear Programming Problems with Fuzzy Uncertainty Unknown Information on Probability Distribution Space STLPPFI (A. Edward Samuel;M. Venkatachalapathy, 2011; Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Abdelaziz, Fouad Ben;Masri, Hatem, 2010; Ameen, 2015; Appati, Justice Kwame;Gogovi, Gideon Kwadwo;Fosu, Gabriel Obed, 2015; Dharani, K;Selvi, D, 2018; Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling, 2015; Mahdavi-Amiri, N;Nasseri, SH, 2006; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007) and (Sengamalaselvi, 2017; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993; Sakawa, Fundamentals of fuzzy set theory, 1993).

2. PRELIMINARIES OF FUZZY CONCEPTS AND POLYHEDRAL SET TYPES

This section reviews some necessary definitions of fuzzy concepts along with stating two kinds of polyhedral sets that are related to certainty information on probability distribution as follows:

2.1 Basic Definitions

2.1.1 Definition of Fuzzy Set: Let X be a universal set, $\widetilde{A} \subseteq X$. $\widetilde{A}$ is called a fuzzy/non-exact set that contains ordered pairs, $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)), \forall x \in X\}$ where $\mu_{\widetilde{A}}(x)$ is membership function of $x \in \widetilde{A}$ (i.e., a characteristic/indicator function for $\widetilde{A}$ that shows to what degree $x \in \widetilde{A}$), if the height of fuzzy set is one, then fuzzy set is normal, where the height of a fuzzy set is the largest membership value attained by any point in the set (Ameen, 2015; Dharani, K;Selvi, D, 2018; Mahdavi-Amiri, N;Nasseri, SH, 2006; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007; Sakawa, Fundamentals of fuzzy set theory, 1993; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993). Formulation (4.2-3) is one of the fuzzy set kinds.

2.1.2 Definition of Alpha-Level Set: The alpha−level set of fuzzy set $\widetilde{A}$ is a set $\widetilde{A}_{\alpha} = \{x \in \mathbb{R}, \mu_{\widetilde{A}}(x) \geq \alpha, 0 < \alpha \leq 1\}$. The lower and upper bounds of alpha−level set of fuzzy set $\widetilde{A}$ are finite numbers represented by $\inf(x \in \widetilde{A}_{\alpha}), \sup(x \in \widetilde{A}_{\alpha})$ respectively (Ameen, 2015; Dharani, K;Selvi, D, 2018).When Formulation (4.2-1) convert to Formulation (4.2-2) needs alpha-level set to finding efficient points via applying conditions of fuzzy set, and also used in analyzing cases for second condition of alpha-cut technique Formulation (4.1-3).

2.1.3 Definition of Convexity of Fuzzy Number: Fuzzy number is a convex fuzzy set $\widetilde{A}$ on $\mathbb{R}$ if and only if its membership function is piecewise continuous, and there exist have three intervals $[a,b],[b,c]$ and $[c,d]$ such that $\widetilde{A}$ is increasing on $[a,b]$, equal to 1 on $[b,c]$, decreasing on $[c,d]$, and equal to 0 elsewhere, $\forall a,b,c,d \in \mathbb{R}$ (Ameen, 2015; Dharani, K;Selvi, D, 2018; Mahdavi-Amiri, N;Nasseri, SH, 2006; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007; Sakawa, Fundamentals of fuzzy set theory, 1993; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993). In Figure (4.2-1) and Formulation (4.2-4) shows seven difference convex fuzzy numbers.

2.1.4 Definition of The Trapezoidal Fuzzy Number (T_pFN): A trapezoidal fuzzy number T_pFN is $\widetilde{A} = (a^L, a^U, \alpha, \beta)$, where $[a^L, a^U]$ is the modal set of $\widetilde{A}$, and $[a^L - \alpha, a^U + \beta]$ is the support part set of $\widetilde{A}$ (Ameen, 2015; Dharani, K;Selvi, D, 2018; Mahdavi-Amiri, N;Nasseri, SH, 2006; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007; Sakawa, Fundamentals of fuzzy set theory, 1993; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993). The T_pFN could be illustrates in following figure:

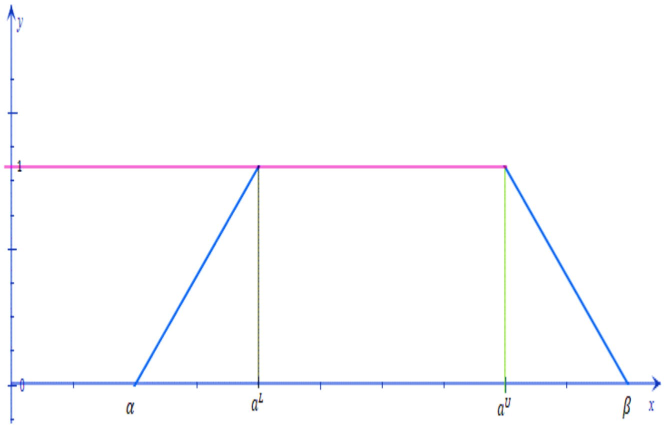


Figure: 2.1-1: Trapezoidal Fuzzy Number T_pFN

Where the linear fuzzy membership function LFMF for trapezoidal fuzzy number T_pFN is as following:

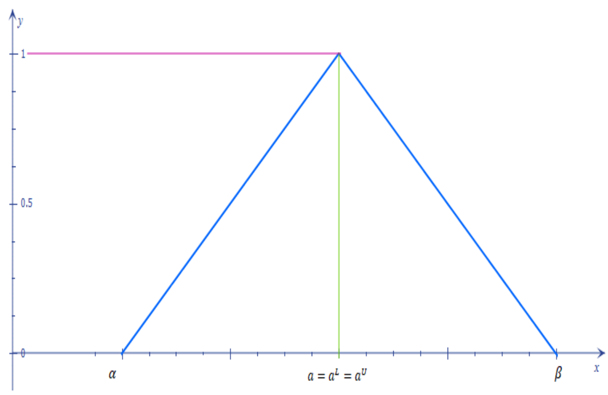
$$\mu(x) = \begin{cases} \dfrac{x-\alpha}{a^L-\alpha} & \alpha \leq x \leq a^L \\ 1 & a^L \leq x \leq a^U \\ \dfrac{\beta - x}{\beta - a^U} & a^U \leq x \leq \beta \\ 0 & Other\ Wise \end{cases}$$

Formulation 2.1-1: The Linear Fuzzy Membership Function LFMF For T_pFN

2.1.5 Definition of The Triangle Fuzzy Number (T_rFN): A trapezoidal fuzzy number T_pFN is reduced to the triangular fuzzy number T_rFN and denoted by $\widetilde{A} = (a, \alpha, \beta)$, where $a = a^L = a^U \in \widetilde{A} \subseteq F(R)$ (Ameen, 2015; Dharani, K;Selvi, D, 2018), thus $\widetilde{A} = (a, \alpha, \beta) \subset (a^L, a^U, \alpha, \beta) \subseteq F(R)$. The T_rFN could be illustrates as following figure:

Figure : 2.1-2: Triangular Fuzzy Number T_rFN

Where the linear fuzzy membership function LFMF for triangular fuzzy number T_rFN is as following:



$$\mu(x)=\begin{cases}\frac{x-\alpha}{a-\alpha} & \alpha\le x\le a\\ 1 & x=a\\ \frac{\beta-x}{\beta-a} & a\le x\le\beta\\ 0 & Other\ Wise\end{cases}$$

Formulation 2.1-2: The Linear Fuzzy Membership Function LFMF For T_rFN

2.1.6 Definition of Ranking Function $(R(F))$: A ranking function $R(F): F(R)\longrightarrow\mathbb{R}$ is a mapping that transforms each fuzzy number into its corresponding real value in real line, where a natural order exists (Ameen, 2015; Dharani, K;Selvi, D, 2018). Formulation (4.2-5) and Formulation (4.2-6) are two kinds of ranking functions.

2.2 The Relating Polyhedral Sets with Building Uncertainty Unknown Information on Probability Space

Two kinds of information on probability distribution which are fuzzy and stochastic polyhedral sets are preferred in this subsection.

2.2.1 The Fuzzy Polyhedral Set $\tilde{\pi}$: The fuzzy polyhedral set contains fuzzy uncertainty unknown information on probability distribution space $(\Omega, 2^{\Omega}, P)$ and generated by fuzzy/inexact inequalities on π which are for each probability p_i of a given event $\omega_i\in\tilde{\pi}, i=1,2,\dots,s$, and formed by:

$$\tilde{\pi}=\begin{Bmatrix} p=(p_1,p_2,\dots,p_N)^T\in\mathbb{R}^N; \\ Ap\preccurlyeq b; \sum_{i=1}^{N}p_i=1\,; \forall p_i\ge 0; i=1,\dots,N\end{Bmatrix}$$

Formulation 2.2-1: The Fuzzy Polyhedral Set $\tilde{\boldsymbol{\pi}}$

Where A and b are (s,N) and $(s,1)$ dimensions fixed fuzzy random matrices respectively, and $\preccurlyeq$ was a fuzzy/inexact inequality and crisp of P which meant that Ap was almost equal or less than b (Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2010; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Ameen, 2015; Dharani, K;Selvi, D, 2018; Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling, 2015; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993; Sakawa, Fundamentals of fuzzy set theory, 1993; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007; Mahdavi-Amiri, N;Nasseri, SH, 2006) and (Hamadameen, Abdulqader Othman;Hassan, Nasruddin, 2018; Hamadameen, Abdulqader Othman;Zainuddin, Zaitul Marlizawati, 2015).

2.2.2 The Stochastic Polyhedral Set $\boldsymbol{\pi}$: Where the information on probability distribution space $(\Omega, 2^{\Omega}, P)$ was stochastic on π and generated by stochastic inequalities on π and crisp of P which are for each probability p_i of a given events $\omega_i\in\pi, i=1,2,\dots,s$, or $(\Omega, 2^{\Omega}, P)$ converted from fuzzy to stochastic, then it is called stochastic/default/known polyhedral set and formed by:

$$\pi=\begin{Bmatrix} p=(p_1,p_2,\dots,p_N)^T\in\mathbb{R}^N; \\ Ap\le b; \sum_{i=1}^{N}p_i=1\,; \forall p_i\ge 0; i=1,\dots,N\end{Bmatrix}$$

Formulation 2.2-2: The Stochastic Polyhedral Set $\boldsymbol{\pi}$

Where A and b are (s,N) and $(s,1)$ dimensions fixed random matrices respectively (Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2010; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Ameen, 2015; Dharani, K;Selvi, D, 2018; Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling, 2015; Sakawa, Interactive multiobjective linear programming with fuzzy parameters, 1993; Sakawa, Fundamentals of fuzzy set theory, 1993; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007; Mahdavi-Amiri, N;Nasseri, SH, 2006) and (Hamadameen, Abdulqader Othman;Hassan, Nasruddin, 2018; Hamadameen, Abdulqader Othman;Zainuddin, Zaitul Marlizawati, 2015).

Therefore, an LPPs with Formulation (2.2-1) is then called linear programming problem with fuzzy uncertainty unknown information on probability distribution space LPPFI. Although, immediately every fuzzy polyhedral set $\tilde{\pi}$ Formulation (2.2-1) should be converted to stochastic polyhedral set π Formulation (2.2-2) via alpha-cut technique approach, then after converting called linear programming problem with certainty known information on probability distribution space LPP.

3. STLPPFI PROBLEM FORMULATION AND ITS MATLAB CODE PROGRAM WITH ALGORITHMS

This section discusses the organization of STLPPFI model problem mathematically and data information of STLPPFI model problem and MATLAB Code Program with its Algorithm as follows:

3.1 The Mathematical Formulation Problem of STLPPFI

The mathematical formulation of STLPPFI is shown as follows:

The Stochastic Unique-Objective Function

$$Min\ z(\omega,x)=\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}c_{ij}(\omega)$$

Subject to:

$$\sum_{j=1}^{n}x_{ij}=a_i\,; i=1,2,\dots,m$$

$$\sum_{i=1}^{m}x_{ij}=b_j\,; j=1,2,\dots,n$$

With satisfying deterministic balance condition:

$$\sum_{i=1}^{m}a_i=\sum_{j=1}^{n}b_j$$

With satisfying domain condition:

$$x_{ij}\ge 0, \forall i,j; i=1,2,\dots,m; j=1,2,\dots,n; x\in X, \omega\in\Omega$$

Formulation 3.1-1: The Mathematical Formulation of STLPPFI

Where stochastic unique-objective function is optimality condition of LPP, and it is subject to both of deterministic availability constraints and deterministic requirement constraints respectively, as well as satisfying deterministic balance condition and domain condition, where constraints are feasible solution region condition of LPP, and deterministic balance condition mean that total availability constraints satisfy total requirements constraints, and domain condition mean that non-negativity of unknown variable set and belongings of stochastic parameters to probability distribution space. Where a_i and b_j are $(m,1)$ known vectors production values of electricity in Kw/h and $(1,n)$ known vectors requirement values of using electricity in Kw/h respectively, both a_i, b_j are crisp and does not appear scholastically, and $c_{ij}(\omega)$ is probably estimated cost values of transporting electricity in IQD/Kw and it is not crisp and appears scholastically should be determined it via both transformations, and $c_{ij}(\omega)$ is (m,n) random matrix as shows in Table (3.1-1) as well as probably estimated cost values for each $c_{ij}(\omega)$ shows in Table (3.1-2) respectively, and x_{ij} is (m,n) unknown matrix should be found it via suitable methods and it is amount of transporting electricity in Kw/h. So, the Formulation (3.1-1) has fuzzy uncertainty unknown information in probability distribution space P. Depends on (Abdelaziz, Fouad Ben;Aouni, Belaid;El Fayedh, Rimeh, 2007; Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Abdelaziz, Fouad Ben;Masri, Hatem, 2010; Ameen, 2015). Formulation (3.1-1) can be defined in terms of some probability distribution space $(\Omega, 2^{\Omega}, P)$, where $\{\Omega = \{\omega_k\}; k = 1,2, \dots, N\}$ is a discrete set of events or a finite set of possible states of nature, 2^{Ω} is power set of Ω, and P is fuzzy uncertainty unknown probability distribution space that assigns to each $A \in 2^{\Omega}$ probability of occurrence $P(A)$ (i.e., P is (s, N) matrix of probabilities $p_i = P(\{\omega = \omega_i\}), i = 1,2, \dots, s, p_i \in \tilde{\pi}, \forall i$).

Although, the set X is a known polyhedral set of feasible solutions that includes deterministic constraints of STLPPFI problem, solving STLPPFI Formulation (3.1-1) first needs to be converted into DTLPP Formulation (3.2-1). Secondly, finding the set of non-negatives $x_{ij}, \forall i, j$ that minimize the objective function, satisfy constraint conditions, balanced condition and domain condition. Where the data is illustrated in Table)3.1-1) of STLPPFI Formulation (3.1-1), and note that the data in Table)3.1-1) may be vague or containing inaccurate values since there might be fuzzy information on probability distribution or there is not any previous information on probability distribution, so information and data will be distributed as follows:

Suppose that m electricity power production stations named $G_1, G_2, G_3, \dots$, and G_m with n cities need to be supplied with electricity names $K_1, K_2, K_3, \dots$, and K_n as following balanced STLPPFI table:

Table: 3.1-1: The Data Distribution Table of STLPPFI

Power Plants	Cites				Supply Million Kw/h
	K_1	K_2	...	K_n	
G_1	$c_{11}(\omega)$	$c_{12}(\omega)$	...	$c_{1n}(\omega)$	a_1
G_2	$c_{21}(\omega)$	$c_{22}(\omega)$	...	$c_{2n}(\omega)$	a_2
⋮	⋮	⋮	⋱	⋮	⋮
G_m	$c_{m1}(\omega)$	$c_{m2}(\omega)$	...	$c_{mn}(\omega)$	a_m
Demand Million Kw/h	b_1	b_2	...	b_n	Total = (v) M Kw/h Balanced

Where prices of transporting costs are stochastically, and probably estimated cost values of random matrix of $c_{ij}(\omega)$ will be as follows:

Table: 3.1-2: The Probably Estimated Cost Values of Probability Distribution Space

ω	ω_1	ω_2	...	ω_s
$c_{11}(\omega)$	$c_{11}(\omega_1)$ IQD	$c_{11}(\omega_2)$ IQD	...	$c_{11}(\omega_s)$ IQD
$c_{12}(\omega)$	$c_{12}(\omega_1)$ IQD	$c_{12}(\omega_2)$ IQD	...	$c_{12}(\omega_s)$ IQD
⋮	⋮	⋮	⋱	⋮
$c_{1n}(\omega)$	$c_{1n}(\omega_1)$ IQD	$c_{1n}(\omega_2)$ IQD	...	$c_{1n}(\omega_s)$ IQD
$c_{21}(\omega)$	$c_{21}(\omega_1)$ IQD	$c_{21}(\omega_2)$ IQD	...	$c_{21}(\omega_s)$ IQD
$c_{22}(\omega)$	$c_{22}(\omega_1)$ IQD	$c_{22}(\omega_2)$ IQD	...	$c_{22}(\omega_s)$ IQD
⋮	⋮	⋮	⋱	⋮
$c_{2n}(\omega)$	$c_{2n}(\omega_1)$ IQD	$c_{2n}(\omega_2)$ IQD	...	$c_{2n}(\omega_s)$ IQD
⋮	⋮	⋮	⋱	⋮
$c_{m1}(\omega)$	$c_{m1}(\omega_1)$ IQD	$c_{m1}(\omega_2)$ IQD	...	$c_{m1}(\omega_s)$ IQD
$c_{m2}(\omega)$	$c_{m2}(\omega_1)$ IQD	$c_{m2}(\omega_2)$ IQD	...	$c_{m2}(\omega_s)$ IQD
⋮	⋮	⋮	⋱	⋮
$c_{mn}(\omega)$	$c_{mn}(\omega_1)$ IQD	$c_{mn}(\omega_2)$ IQD	...	$c_{mn}(\omega_s)$ IQD

Where information of response cities on electricity power plants are fuzzy distributed i.e., the information probability distribution shown as fuzzy polyhedral set $\tilde{\pi}$ form (Formulation (2.2-1)). The STLPPFI Formulation (3.1-1) has stochastically uncertainty unknown expression in its objective function coefficients, and then it has fuzzily uncertainty unknown expression in its information probability distribution space $(\Omega, 2^{\Omega}, P)$. The uncertainty has randomness for parameters and fuzziness for probability distributing.

3.2 The Mathematical Formulation Problem of DTLPP

STLPPFI mathematical formulation was introduced in subsection (3.1). Now standard Deterministic Transportation Linear Programming Problems DTLPP need to be introduced. Since after defuzzifying fuzziness of information on probability distribution space of STLPPFI then STLPPFI convert to STLPP, then after derandomizing stochastic/randomness of problem formulation of STLPP then immediately STLPP convert to DTLPP. Now, DTLPP can be formulated and shown as follows:
The Deterministic Unique-Objective Function

$$Min\ z(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} c_{ij}$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = a_i \; ; i = 1,2, \dots, m$$

$$\sum_{i=1}^{m} x_{ij} = b_j \; ; j = 1,2, \dots, n$$

With satisfying deterministic balance condition:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

With satisfying domain condition:

$$x_{ij} \geq 0, \forall i, j; i = 1,2, \dots, m; j = 1,2, \dots, n; x \in X$$

Formulation 3.2-1: The Mathematical Formulation of DTLPP

Where the deterministic unique-objective function is the optimality condition of LPP, and it is subject to both deterministic availability constraints and deterministic requirement constraints respectively, as well as satisfying the deterministic balance condition and the domain condition, where constraints are the feasible solution region condition of LPP, and the deterministic balance condition mean that the total availability constraints satisfy the total requirements constraints, and the domain condition means the non-negativity of unknown variable set. Where c_{ij}, a_i and b_j are (m, n) known matrices of deterministic cost values of transporting electricity in IQD/Kw, $(m, 1)$ known vector production values of electricity in Kw/h, $(1, n)$ known vector requirement values of using electricity in Kw/h respectively, and all of them are deterministic and do not appear scholastically, and x_{ij} is (m, n) unknown matrix should be found it using a suitable method such as Dual-Simplex and VAM, and it represents the amount of transporting electricity in Kw/h.

Although the set X is a polyhedral set of feasible solutions that includes deterministic constraints of the DTLPP problem, to solve Formulation (3.2-1), we need to find a set of non-negatives $x_{ij}, \forall i, j$ that minimize the objective function, satisfy constraint conditions, the balanced condition and the domain condition. The data distributed in Table)3.2-1) of DTLPP Formulation (3.2-1), and note that data in Table)3.2-1) are most approximately equivalent values of fuzzy random estimate values before and most approximately equivalent values to exact values, since we convert fuzzy information on probability distribution space to known information or we have approximately trust information on probability distribution space now, so the information distributed as follows:

Suppose that m electricity power product stations named $G_1, G_2, G_3, \dots$, and G_m with n cities need to be supplied with electricity names $K_1, K_2, K_3, ...,$ and K_n as following balanced standard DTLPP table:

Table: 3.2-1: The Data Distribution Table of DTLPP

Power Plants	Cites				Supply Million Kw/h
	K_1	K_2	...	K_n	
G_1	c_{11}	c_{12}	...	c_{1n}	a_1
G_2	c_{21}	c_{22}	...	c_{2n}	a_2
⋮	⋮	⋮	⋱	⋮	⋮
G_m	c_{m1}	c_{m2}	...	c_{mn}	a_m
Demand Million Kw/h	b_1	b_2	...	b_n	Total = (v) M Kw/h Balanced

3.3 STLPPFI Algorithm with its MATLAB Code Program

The STLPPFI algorithm with its MATLAB code program was introduced in detail and STLPPFI MATLAB program will be applied to solve illustrate example in section 5.

3.3.1 The Algorithm Program Outline of STLPPFI

The program outline of STLPPFI will be as follows:

Input: input fuzzy information on probability distribution via three vectors b_i, d_i and α_i as credibility degree of the DM about information on probability distribution space, vagueness level vector and alpha-cut level vector respectively, then input an acceptance vector via vector p from RN matrix in each row select one value where length of vector p is 3/4/5, then input cs matrix $(mm * nn, 3)/(mm * nn, 4)/(mm * nn, 5)$ $2D$ dimensions' matrix as estimates cost values where matrix cs is (m, n) estimates distribution matrix with $(1, k)$ deterministic acceptance vector face, then input mm value and nn value respectively as length of row and column of deterministic cost values of matrix cd will be in final where should $(mm * nn)/(kk)$ i.e., $(mm * nn)$ divide over (kk) if it is not divided then the problem does not have solution, then finally input availability constraints vector avb and input requirement constraints vector req where should be total availability satisfy total requirement as applying balance condition on it.

Step 1: Form the problem as Formulation (3.1-1).

Step 2: To transform Formulation (2.2-1) into Formulation (2.2-2) use truth degrees' algorithm which contains three algorithms respectively as follows:

Step 2a: Use Formulation (4.1-3) i.e., (use alpha-cut algorithm) to obtain alpha-cut technique probability interval from three vectors b_i, d_i and α_i as credibility degree of the DM about information on probability distribution space, vagueness level vector and alpha-cut level vector respectively to cover and determine fuzzy uncertainty information on probability distribution.

Step 2b: Build the truth degrees set on probability distribution via uses dividing each obtain alpha-cut technique probability interval to ten continuous subintervals or eleven equal distance points in each row of degrees of truth of fuzzy logical value i.e., (use linspace MATLAB function).

Step 2c: Build fuzzy truth degrees polyhedral set via using linear fuzzy membership functions LFMF Formulation (4.1-1) and Formulation (4.1-2) i.e., shown fuzzy truth degrees polyhedral set via both $TrFN$ and $TpFN$ in each row as fuzzy truth degrees regions $FN(k, 1\colon 24) = [TrFN1, TpFN2, TrFN3, TpFN4, TrFN5, TpFN6, TrFN7]$, where $TrFN1$(1st 3 elements), $TpFN2$(2nd 4 elements), $TrFN3$(3rd 3 elements), $TpFN4$(4th 4 elements), $TrFN5$(5th 3 elements), $TpFN6$(6th 4 elements), $TrFN7$(7th 3 elements) of FN matrix, and where fuzzy vector forms contain $TrFN$ as $(a, alpha, betta)$, and for $TpFN$ as $(a - lower, a - upper, alpha, betta)$. Then convert fuzzy truth degrees polyhedral set to stochastic truth degrees polyhedral set or deterministic values vector or nine important efficient points in each fuzzy truth degrees regions after applying linear fuzzy ranking functions LFRF Formulation (4.2-5) and Formulation (4.2-6) (i.e., use LFRF algorithm) to deffuzzifier all seven fuzzy numbers column values for all i in each row and showing via obtain deterministic matrix values RN.

Step 3: Test second condition of alpha-cut technique probability interval Formulation (4.1-3) (i.e., use acceptance P algorithm) which is from deterministic vector p in each row, we take $p1, p2, p3 \dots pn$ values to obtain acceptance vectors to check that STLPP Formulation (3.1-1) with stochastic polyhedral set Formulation (2.2-2) are acceptable for the entire cases to go to next step or not (where n cases are passes via test 9^n cases via $sum(pi) = 1, \forall pi \geq 0$). Then create logical comparative matrix to find and select each case equal one after $sum(pi) = 1, \forall pi \geq 0$, then select locations of acceptance cases via create acceptance location vector, then collect pi as acceptances' vectors of p.

Step 4: Use Formulation (4.3-1) for deterministic acceptance vectors that passes Step 3 (i.e., use EWS algorithm) to convert STLPP Formulation (3.1-1) with stochastic polyhedral set Formulation (2.2-2) into DTLPP Formulation (3.2-1). This step starts by inputting vector p as an acceptance vector that was obtained previously from RN matrix, inputting cs $2D$ dimensions' (m, n) estimates distribution matrix as estimates cost values Table (3.1-2) with $(1, k)$ deterministic acceptance vector face, inputting mm and nn scalar values as length of row and column of final obtain deterministic cost value matrix cd in Table (3.2-1). Then cs $(m * n, k)$ $2D$ matrix converts to css (m, k, n) $3D$ matrix via applying EWS algorithm process on css (m, k, n) $3D$ matrix with $(1, k)$ dimension acceptance vector p to convert css (m, k, n) $3D$ matrix to cd (m, n) $2D$ deterministic cost value matrix in Table (3.2-1).

Step 5: Solve DTLLP Formulation (3.2-1) n times with all acceptance vectors that passes Step 3 via each step 5a & 5b separately to obtain initial feasible solution IFS, where Formulation (3.2-1) contains the obtained cd (m, n) deterministic cost matrix in Table (3.2-1) with both $(1, n)$ and $(m, 1)$ deterministic vectors by inputting availability constraints vector avb and requirement constraints vector req in Table (3.2-1) where the total availability should satisfy the total requirement.

Step 5a: Solve DTLPP via the Simplex Algorithm Method.

Step 5b: Solve DTLPP via the Vogel Approximation Algorithm Method.

Step 6: Find optimal solution among n cases of acceptance vectors that passes Step 3 for each two methods via the Modify Distribution Method (MODI).

Step 7: Select the post optimal solution via deciding from decision maker DM recommendation for a certain case.

Output: Optimal Solution that contain best basic feasible solution BFS with minimum total cost transporting electricity objective function, with selection post optimal solution value ■.

3.3.2 The STLPPFI Problem Solver Technique MATLAB Program

The MATLAB code program of STLPPFI problem solver technique is as follows:

```
% Alpha-cut technique part
format short; disp('input vector b as a credibility degree of the DM about information on probability distribution space'), b=input('b=');disp('input vector d as a vagueness level'),d=input('d=');disp('input vector a as alpha-cut level'),a=input('a=');
b=b';d=d';a=a';n=length(a);p=zeros(n,2);
if length(b)==length(d) && length(b)==length(a) && length(d)==length(a)
```

```
    disp('The problem has a solution as follows:')
    for k=1:1:n
        p(k,[1,2])=[(b(k))-((d(k))*((1-a(k)))),(b(k))+((d(k))*((1-a(k))))];
    end
else
disp('There is no solution since all vectors you inputted are not in the same dimension')
end
if length(b)==length(d) && length(b)==length(a) && length(d)==length(a)
    disp('The credibility degree vector b is')
    b=b';b
    disp('The vagueness levels vector d is')
    d=d';d
    disp('The alpha-cut levels vector a is')
    a=a';a
else
    disp('please input all vectors in same dimension')
end
% Truth Degrees technique part
b=size(p);n=b(1);l=zeros(n,11);FN=zeros(n,24);RN=zeros(n,9);
TrFN1=0;TpFN2=0;TrFN3=0;TpFN4=0;TrFN5=0;TpFN6=0;TrFN7=0;RN1=0;RN2=0;RN3=0;RN4=0;RN5=0;RN6=0;RN7=0;
if b(2)==2
    disp('The solution will be as following')
    for k=1:1:n
       l(k,1:11)=linspace(p(k,1),p(k,2),11);
TrFN1(k,1:3)=l(k,[2,1,3]);TpFN2(k,1:4)=l(k,[3,5,2,6]);TrFN3(k,1:3)=l(k,[5,3,6]);TpFN4(k,1:4)=l(k,[5,7,4,8]);TrFN5(k,1:3)=l(k,[7,6,9]);TpFN6(k,1:4)=l(k,[7,9,6,10]);TrFN7(k,1:3)=l(k,[10,9,11]);FN(k,1:24)=[TrFN1(k,:),TpFN2(k,:),TrFN3(k,:),TpFN4(k,:),TrFN5(k,:),TpFN6(k,:),TrFN7(k,:)];RN1(k,1)=(FN(k,1))+((FN(k,3)-FN(k,2))/4);RN2(k,1)=((FN(k,4)+FN(k,5))/2) +((FN(k,7)-FN(k,6))/4);RN3(k,1)=(FN(k,8))+ ((FN(k,10)-FN(k,9))/4);RN4(k,1)=((FN(k,11) +FN(k,12))/2)+((FN(k,14)-FN(k,13))/4); RN5(k,1)=(FN(k,15))+((FN(k,17)-FN(k,16))/4);RN6(k,1)=((FN(k,18)+FN(k,19))/2)+((FN(k,21)-FN(k,20))/4);RN7(k,1)=(FN(k,22))+((FN(k,24)-FN(k,23))/4);RN(k,1:9)=[p(k,1),RN1(k,:),RN2(k,:),RN3(k,:),RN4(k,:),RN5(k,:),RN6(k,:),RN7(k,:),p(k,2)];
    end
else
disp('There is no solution since p is not as a (n,2) dimension matrix')
end
if b(2)==2
    disp('The Alpha-Cut Technique probability intervals of each pi for all i in each row of (n,2) matrix p i.e., after applying Fuzzy Transformation of Probability Distribution Space via Alpha-Cut Technique')
    p
    disp('The Truth Degrees Process as follows applies: First, The Alpha-Cut Technique Probability Intervals of each pi for all i divides to eleven equal distance points in each row of following Degrees of Truth of fuzzy logical value')
    l
       disp('Second, Converting Truth Degrees to both TrFN and TpFN in each row as following fuzzy Truth Degrees region FN(k,1:24)=[TrFN1,
TpFN2,TrFN3,TpFN4,TrFN5,TpFN6,TrFN7] Where TrFN1(1st 3 elements),TpFN2(2nd 4 elements),TrFN3(3rd 3 elements), TpFN4(4th 4 elements),TrFN5(5th 3 elements),TpFN6(6th 4 elements),TrFN7(7th 3 elements) Where fuzzy vector forms contain TrFN as (a, alpha, betta), and for TpFN as (a-lower, a-upper, alpha, betta)')
    FN
       disp('The deterministic vector values or nine important power points in each fuzzy Truth Degrees regions after applying linear fuzzy ranking function LFRF to deffuzzifier fuzzy for all column values for all i as each row of the following deterministic values matrix RN')
    RN
else
    disp('please input p is a (n,2) dimension matrix')
end
% Acceptance P technique part
kkk=size(RN);
if kkk(1)==3
    disp('from deterministic vector p in each row we take p1 p2 p3 ... pn vectors then we test 9^n cases via sum(pi)=1 for all pi>0')
p1=RN(1,:);np1=length(p1);p2=RN(2,:);np2=length(p2);p3=RN(3,:);np3=length(p3);p123=zeros(np3,np1,np2);pp123=zeros(np3,np1,np2);
    if length(p1)==length(p2)&& length(p1)==length(p3)&& length(p2)==length(p3)
        for k=1:1:np3
          for m=1:1:np1
            for n=1:1:np2
          p123(m,n,k)=p1(k)+p2(m)+p3(n);p123;
            end
          end
        end
    else
        disp('There is not have solution')
    end
    p123
    disp('we find and select each case equal one after sum(pi)=1 for all pi>0 then we collect pi as acceptance vector'), pp123=logical(p123==1),[rowpp123,colpp123,volpp123]=find(pp123);rowpp123=rowpp123';colpp123=colpp123';volpp123=volpp123';rowpp123,colpp123,volpp123
else
    disp('There is not have solution since rows of RN more than 3')
end
% EWS technique part
disp('input vector p as an acceptance vector from RN matrix in each row give one value where length of vector p is 3/4/5'),p=input('p=');disp('input cs matrix (mm*nn,3)/(mm*nn,4)/(mm*nn,5) 2-D dimensions matrix as a estimates cost values where matrix cs is (m,n) distribution matrix with (1,k) stochastic vector face'),cs=input('cs='); disp('please input mm and nn as what is length of row and column of deterministic value matrix will be in final respectively Be attention should (mm*nn)/(kk) i.e., (mm*nn) divide over (kk) if not divided not have solution')
mm=input('mm=');nn=input('nn=');kk=size(cs,2);
if rem(mm*nn,kk)==0
disp('we convert cs matrix (m*n,k) 2D matrix to css (m,k,n) 3D matrix')
if length(p)==3
css=cat(3,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)));
elseif length(p)==4
css=cat(4,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)),cs((3*(mm))+1:4*(mm),1:length(p)));
elseif length(p)==5
css=cat(5,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)),cs((3*(mm))+1:4*(mm),1:length(p)),cs((4*(mm))+1:5*(mm),1:length(p)));
else
    disp('There is not have solution')
end
```

```
m=size(css,1);k=size(css,2);n=size(css,3);cd=zeros(m,k,n);css
if length(p)==k
   disp('The solution will be as following')
   cd=pagetranspose(pagemtimes(css,p'));
else
disp('There is not have solution')
end
else
disp('There is not have solution')
end
if length(p)==k && length(p)==3
  disp('The deterministic value matrix cd is'),cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3)];format long;cd
elseif length(p)==k && length(p)==4
   disp('The deterministic value matrix cd is'),cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3);cd(:,:,4)];format long;cd
elseif length(p)==k && length(p)==5
   disp('The deterministic value matrix cd is'),cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3);cd(:,:,4);cd(:,:,5)];format long;cd
else
  disp('please input (1,k) vector p that have same 3rd dimension of cs (m,n,k) 3D matrix')
end
% Solving Deterministic TLPP part
disp('Now, we have deterministic matrix cd, please input availability vector avb and input requirement vector req where should be total availability satisfy total requirement')
avb=input('avb=');req=input('req=');
if sum(avb)==sum(req)
   disp('The solution where uses dual-simplex method'),[sol,zval,exitflag,
output]=DTLPPVogel(cd,avb,req);sol=reshape(sol,size(cd,1),size(cd,2));exitflag, output, sol,format long;zval,disp('The solution where uses Vogel approximation method'),[ibfs,objCost]=VogelBeModi2(cd,avb,req)
   else
      disp('There is not have solution since total availability does not satisfy total requirement or sum(avb) not equal to sum(req)')
end
```

■.

Where three DTLPP problem solver MATLAB programs of previous works are as follows:

```
function [sol,zval,exitflag,output]= DTLPPVogel(cost,avb,req)
if sum(avb) ~= sum(req)
  ...exc=MException('tp:unbalancedProblem', ... 'Cannot solve unbalanced problem.');
      throw(exc);
end
x=optimvar('x',size(cost,1),size(cost,2),'LowerBound',0);z=sum(x.*cost,'all');numCons=size(cost,1)+size(cost,2);cons = optimconstr(numCons, 1);count = 1;
for i=1:1:size(x,1)
cons(count)=sum(x(i,:))==avb(i);count=count+1;
end
for i = 1:1:size(x, 2)
cons(count)=sum(x(:,i))==req(i);count=count+1;
end
problem=optimproblem('Objective',z,'ObjectiveSense',
'min');problem.Constraints = cons;
show(problem),problem=prob2struct(problem);
[sol,zval,exitflag,output]=linprog(problem);
end
```

(Appati, Justice Kwame;Gogovi, Gideon Kwadwo;Fosu, Gabriel Obed, 2015; Dharani, K;Selvi, D, 2018; Sengamalaselvi, 2017)■.

```
function[ibfs,objCost]=VogelModi(data)
% ===DATA PREPARATION===
cost=data(1:end-1,1:end-1);demand=data(end,1:end-1);supply=data(1:end-1,end)';ibfs=zeros(size(cost));
% ===VOGEL APPROXIMATION METHOD===
ctemp=cost; %temporal cost matrix
while
length(find(demand==0))<length(demand)||length(find(supply==0))<length(supply)
prow=sort(ctemp,1);prow=prow(2,:)-prow(1,:); %row penalty
pcol=sort(ctemp,2);pcol=pcol(:,2)-pcol(:,1); %column penalty
  [rmax,rind]=max(prow);[cmax,cind]=max(pcol);
   if rmax>cmax
[~,mind]=min(ctemp(:,rind));[amt,demand,supply,ctemp]=chkdemandsupply(demand,supply,rind,mind,ctemp);ibfs(mind,rind)=amt;
   elseif cmax>= rmax
[~,mind]=min(ctemp(cind,:));[amt,demand,supply,ctemp]=chkdemandsupply(demand,supply,mind,cind,ctemp);ibfs(cind,mind)=amt;
   end
end
objCost=sum(sum(ibfs.*cost));
%===MODIFIED DISTRIBUTION ===
val=-1;
while val<0
[prow,pcol]=find(ibfs>0);occupiedCells=[prow,pcol]';[prow,pcol]=find(ibfs==0);unoccupiedCells=[prow,pcol]';r=0;k =[];
   for i = 1:length(occupiedCells(1,:))
ri=occupiedCells(1,i);kj=occupiedCells(2,i);
[r,k]=occupiedSystemSolve(r,k,ri,kj,cost);
   end
improvementIndex=zeros(length(unoccupiedCells(1,:)),3);
   for i =1:length(unoccupiedCells(1,:))
ri=unoccupiedCells(1,i);kj=unoccupiedCells(2,i);e=cost(ri,kj)-r(ri)-k(kj);improvementIndex(i ,:)=[ri,kj,e];
   end
   [val,ind]=min(improvementIndex(:,end));
   if val< 0 %check whether improvement is required
ri=improvementIndex(ind,1);kj=improvementIndex(ind,2);disp(['Create a circuit around cell ('
num2str(ri) ',' num2str(kj) ')' ]);
circuitImproved=[ri,kj,0];n=input('Enter number of element that forms the circuit: ');
      for i = 1:n
nCells = input(['Enter the index of cell ' num2str(i) ' that forms the circuit: ']);
         if mod(i,2) == 0
circuitImproved(i+1,:)=[nCells,ibfs(nCells(1),nCells(2))];
         else
circuitImproved(i+1,:)=[nCells,-ibfs(nCells(1),nCells(2))];
         end
      end
ibfs=reallocateDemand(ibfs,circuitImproved);
disp(ibfs),objCost=sum(sum(ibfs.*cost));
   end
end
% ===OTHER REQUIRED FUNCTIONS===FUNCTION 1===
function [r,k]=occupiedSystemSolve(r,k,ri,kj,cost)
if length(r)>=ri
   k(kj)=cost(ri,kj)-r(ri);
else
   r(ri)=cost(ri,kj)-k(kj);
end
% ===FUNCTION 2===
Function
[y,demand,supply,ctemp]=chkdemandsupply(demand,supply,ded,sud,ctem)
```

```
tempd=demand;temps=supply;
if tempd(ded)>temps(sud)
temps(sud)=0;tempd(ded)=demand(ded)-
supply(sud);y=supply(sud);ctem(sud,:)=inf;
elseif tempd(ded)<temps(sud)
tempd(ded)=0;temps(sud)=supply(sud)-
demand(ded);y=demand(ded);ctem(:,ded)=inf;
elseif tempd(ded)==temps(sud)
tempd(ded)=0;temps(sud)=0;y=demand(ded);
ctem(:,ded)=inf;ctem(sud,:)=inf;
end
demand=tempd;supply=temps;ctemp = ctem;
```

(Sengamalaselvi, 2017; Appati, Justice Kwame;Gogovi, Gideon Kwadwo;Fosu, Gabriel Obed, 2015)■.

4. TECHNIQUE APPROACHES FOR SOLVING STLPPFI

This section describes the necessary technique approaches for solving STLPPFI. The algorithm program outlines of those technique approaches and MATLAB code programs of those technique approaches are stated in appendix section 7. Those algorithms and MATLAB code programs are used for solving STLPPFI and then converting it to DTLPP followed bt obtaining the g post optimal solution of STLPPFI with illustrations via application example of section 5 for each step of the solution process.

The solution process contains several stages. First stage involves fuzzy transformation on probability distribution space via alpha-cut technique approach applies, which is defuzzification of fuzzy on probability distribution space of the original STLPPFI problem. This process converts it to its corresponding equivalent to Stochastic Transportation Linear Programming Problems with Certainty Known Information on Probability Distribution Space (STLPP) by creating bounded interval with unlimited possible known values from unknown probably/stochastic values. Then the second stage uses the truth degrees technique approach for creating fuzzy probability subintervals from the obtained interval, then Linear Fuzzy Membership Functions (LFMF) will be found from fuzzy truth degrees set. This will be followed by sketching them in one combinational figure for separating various different fuzzy truth degrees regions. After that, the fuzzy truth degrees regions will be defuzzified to stochastic truth degrees regions via Linear Fuzzy Ranking Function (LFRF) to get finite discrete determined value set. Then cases via testing second condition of alpha-cut technique polyhedral set are analyzed to obtain acceptances' vectors for the preparation of applying stochastic transformation. The third stage is stochastic transformation of formulation problem, which is derandomization of probably randomness value set of random variables towards its corresponding equivalent deterministic values, where stochastic transformation of objective function via EWS technique approach applies to transforming STLPP to DTLPP. Then the fourth stage is solving obtained DTLPP via Dual-Simplex Algorithm Method and VAM, then finding optimal solution via the Modify Distribution Method (MODI), where the optimal solution of an DTLPP model problems has the minimum objective function value (i.e., have minimum objective total transportation costs). Finally, selecting post optimal solution as a final result by deciding commands from decision makers (DM) among the entire exit intervals for a certain case or analyzing results by answering what is the perfect solution of STLPPFI entirely in a certain case.

4.1 Fuzzy Transformation on Probability Distribution Space via Alpha-Cut Technique

The first transforming on an STLPPFI Formulation (3.1-1) with Formulation (2.2-1) to STLPP Formulation (3.1-1) with Formulation (2.2-2) involves transforming of the third component of probability distribution space $(\Omega, 2^{\Omega}, P)$ from fuzzy uncertainty unknown information on probability distribution space that is generated by fuzzy/inexact inequalities on $\tilde{\pi}$ and crisp of P to stochastic inequalities on π and crisp of P i.e., P is in Formulation (2.2-1) fuzzy polyhedral set $\tilde{\pi}$. Then it should be converted to Formulation (2.2-2) stochastic/default/known polyhedral set π or default probability distribution space which is the probabilities generated by stochastic inequalities on π via using alpha-cut technique approach (Abdelaziz, 2012; Abdelaziz, F Ben;Masri, Hatem, 2005; Ameen, 2015). In general alpha-cut technique works on polyhedral sets to convert them from probably estimated unknown values to bounded interval with unlimited possible determined known values to obtain STLPP. Now, all fuzzy inequalities $\sum_{j=1}^{n} a_{ij}p_j \preccurlyeq b_i, i = 1, \dots, s$ of fuzzy polyhedral set Formulation (2.2-1) could be shown as a Linear Fuzzy Membership Function (LFMF) $\mu_i, i = 1, \dots, s$ for T_rFN and T_pFN as following two LFMF's:

$$\mu_i(p) = \begin{cases} 1 & \sum_{j=1}^{N} a_{ij}p_j \le b_i \\ \dfrac{(b_i + d_i) - \sum_{j=1}^{N} a_{ij}p_j}{d_i} & b_i \le \sum_{j=1}^{N} a_{ij}p_j \le b_i + d_i \\ 0 & \sum_{j=1}^{N} a_{ij}p_j \ge b_i + d_i \end{cases}$$

Formulation 4.1-1: The $\boldsymbol{T_rFN}$ Linear Fuzzy Membership Function (RLFMF)

$$\mu_i(p) = \begin{cases} \dfrac{\sum_{j=1}^{N} a_{ij}p_j - (-b_i + d_i)}{2(b_i - d_i)} & -b_i + d_i \le \sum_{j=1}^{N} a_{ij}p_j \le b_i - d_i \\ 1 & b_i - d_i \le \sum_{j=1}^{N} a_{ij}p_j \le b_i \\ \dfrac{(b_i + d_i) - \sum_{j=1}^{N} a_{ij}p_j}{d_i} & b_i \le \sum_{j=1}^{N} a_{ij}p_j \le b_i + d_i \\ 0 & \sum_{j=1}^{N} a_{ij}p_j \ge b_i + d_i \end{cases}$$

Formulation 4.1-2: The T_pFN Linear Fuzzy Membership Function (PLFMF)

Where $p = (p_1, p_2, \dots, p_N)^T \in \mathbb{R}^N$ is a probability distribution space vector, $d = (d_1, d_2, \dots, d_N)$ is a vagueness level vector and it is used with each value of p exceeding $b + d$ should be neglected (Ameen, 2015), alpha-cut level vectors are $(\alpha_1, \alpha_2, \dots, \alpha_N)$ and the credibility degree of DM about information on probability distribution of $p_1, p_2, \dots, p_N$ are around determinated values respectively i.e., $P(p_1) \approx b_1, P(p_2) \approx b_2, \dots, and\ P(p_N) \approx b_N$ then we obtain $\big(P(p_1), P(p_2), \dots, P(p_N)\big) \approx (b_1, b_2, \dots, b_N)$. The following alpha-cut technique will be applying for each fuzzy inequalities in Formulation (2.2-1) as follows:

$$\pi^k = \left\{ \begin{matrix} p = (p_1, p_2, \dots, p_N)^T \in \mathbb{R}^N; \\ b_k - d_k(1 - \alpha_k) \le p_k \le b_k + d_k(1 - \alpha_k); \\ \sum_{k=1}^{N} p_k = 1, \forall p_k \ge 0, k = 1, \dots, N \end{matrix} \right\}$$

Formulation 4.1-3: The Alpha-Cut Technique Formula

Therefore, immediately after applying fuzzy transformation on probability distribution space by transferring polyhedral set from fuzzy polyhedral set (2.2-1) to stochastic polyhedral set (2.2-2), we could apply the truth degrees technique approach on probability distribution space, then analyzing cases and finally stochastic transformation of objective function problem formulation via EWS techniques approach (Ameen, 2015).

4.1.1 The Alpha-Cut Technique Algorithm Program Outline

The program outline of Alpha-cut technique will be stated in subsection (7.1.1).

4.1.2 The Alpha-Cut Technique MATLAB Program

The MATLAB code program of Alpha-Cut technique will be stated in subsection (7.1.2).

4.2 The Truth Degrees Technique on Probability Distribution Space

In this subsection as subsection (4.1) also focuses is on probability distribution space. After converting fuzzy uncertainty unknown information on probability distribution space $(\Omega, 2^{\Omega}, P)$ to certainty known information on probability distribution space, the obtained probability intervals could be defined as a fuzzy set of the truth degrees technique approach on probability distribution space, where truth degrees technique improves from fuzzy set of optimistic and pessimistic in probability distribution space of researcher (Ameen, 2015). In addition, continuous interval of fuzzy probability distribution polyhedral set is classified to ten equal subintervals or eleven elementary equally distanced points, from which seven efficient points are given via defuzzifying those eleven points by LFRF technique Formulation (4.2-4) and Formulation (4.2-5) with two boundary points. Then we obtain nine efficient points and mistake other unnecessary points (i.e., using LFRF technique algorithm as a part of truth degrees technique algorithm). Note that the obtained nine points from defuzzifying are different from first eleven points since those eleven points in the first time were not necessarily to be effective points but surely the obtained points are efficient ones.

The truth degrees technique obtained from logic fuzziness of human normal languages via degrees value of truth in numeric logical answering of a question. For example, when anyone is asked to give opinions on the expected result of a random subject, phenomenon, job, routine or health status, then the answers will be fuzzy logic values, as this question (Are you fine?) the answer set is {1.0 Yes/Perfect, 0.9 Excellent, 0.8 Very Good, 0.7 Good, 0.6 Well, 0.5 Moderate, 0.4 Some, 0.3 Somewhat, 0.2 Little, 0.1 Very Little, 0 No/Bad}. In general, the truth degrees technique on probability distribution space used for dividing continuous intervals to deterministic discrete values set of efficient points. So, the truth degree variable value x contain n terms $x_1, x_2, \dots, x_n$ and the series of those terms are $\{x_1, x_2, \dots, x_n\}$ (Abdelaziz, F Ben;Masri, Hatem, 2005; Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling, 2015; Ameen, 2015; Ben Abdelaziz, F;Masmoudi, Meryem, 2012; Dharani, K;Selvi, D, 2018; Hamadameen, Abdulqader Othman;Zainuddin, Zaitul Marlizawati, 2015; Hamadameen, Abdulqader Othman;Hassan, Nasruddin, 2018). Then splitting each continuous interval of the default probability distribution π in (2.2-2) into ten continuous subintervals as shown as following:

$$p = (p_1, p_2, \dots, p_N)^T \in \mathbb{R}^N : \sum_{i=1}^{N} p_i = 1,$$
$$\forall p_i \geq 0, \alpha \leq p_i \leq \beta, i = 1, \dots, N$$

Where $\alpha, \beta, \varphi \in \mathbb{R}$, with $\varphi = a_5 = b_5$, and

$$[\alpha, \beta] = [\alpha, a_1] \cup [a_1, a_2] \cup [a_2, a_3] \cup [a_3, a_4] \cup [a_4, \varphi] \cup [\varphi, b_4] \cup [b_4, b_3] \cup [b_3, b_2] \cup [b_2, b_1] \cup [b_1, \beta]$$

Formulation 4.2-1: The Splitting Original Interval of Probability Distribution Space

Such that those ten continuous subintervals classified into seven stochastic truth degrees regions:

$$A = \{\{\alpha, a_2\}, \{a_1, \varphi\}, \{a_2, \varphi\}, \{a_3, b_3\}, \{\varphi, b_2\}, \{\varphi, b_1\}, \{b_2, \beta\}\} = \left\{\begin{matrix} Little, Some, Moderate, Well, \\ Good, Very\ Good, Excellent \end{matrix}\right\} = \{Li, So, Mo, We, Go, Ve, Ex\}; \forall p_i \in \pi, i = 1, \dots, N$$

Formulation 4.2-2: Stochastic Truth Degrees Regions Set

Where $Li, So, Mo, We, Go, Ve\ and\ Ex$ are Little, Some, Moderate, Well, Good, Very Good and Excellent respectively, and also $[\alpha, \varphi]$ and $[\varphi, \beta]$ are pessimistic/fail region and optimistic/pass region probability distribution respectively, and $\varphi = a_5 = b_5$, as shown in the following Figure (4.2-1).
When $p_i \in \tilde{\pi}$ then we obtain fuzzy truth degrees set of p_i that could be shown as:

$$\tilde{A} = \{\widetilde{\{\alpha, a_2\}}, \widetilde{\{a_1, \varphi\}}, \widetilde{\{a_2, \varphi\}}, \widetilde{\{a_3, b_3\}}, \widetilde{\{\varphi, b_2\}}, \widetilde{\{\varphi, b_1\}}, \widetilde{\{b_2, \beta\}}\} = \left\{\begin{matrix} Fuzzy\ Little, Fuzzy\ Some, Fuzzy\ Moderate, \\ Fuzzy\ Well, Fuzzy\ Good, \\ Fuzzy\ Very\ Good, Fuzzy\ Excellent \end{matrix}\right\} = \{\widetilde{Li}, \widetilde{So}, \widetilde{Mo}, \widetilde{We}, \widetilde{Go}, \widetilde{Ve}, \widetilde{Ex}\};\ \forall p_i \in \tilde{\pi}, i = 1, \dots, N$$

Formulation 4.2-3: Fuzzy Truth Degrees Regions Set

Note that fuzzy truth degrees set Formulation (4.2-3) should be converted to stochastic truth degrees set Formulation (4.2-2) via linear fuzzy ranking function LFRF technique Formulation (4.2-5) and Formulation (4.2-6).

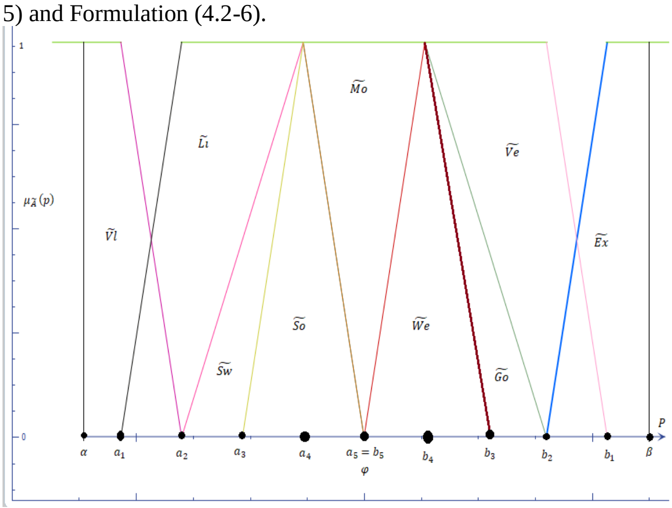


Figure: 4.2-1: Fuzzy Truth Degree Regions Set of Probability Distribution Space ***P***

The $\mu_k(p_i), k = 1,2,..,7; i = 1,2, \dots, N$ are linear fuzzy membership functions LFMFs for entire both types trapezoidal fuzzy number T_pFN regions and triangular fuzzy number T_rFN regions of fuzzy truth degrees set $\tilde{A} = \{\widetilde{Li}, \widetilde{So}, \widetilde{Mo}, \widetilde{We}, \widetilde{Go}, \widetilde{Ve}, \widetilde{Ex}\}$ of each p_i respectively on information of probability distribution space will be as follows that defined in T_rFN and T_pFN (Abdelaziz, F Ben;Masri, Hatem, 2005; Abdelaziz, Fouad Ben;Masri, Hatem, 2009; Ameen, 2015) were obtained from two Formulations (4.1-1) and (4.1-2):

$$\mu_1(p_i) = \begin{cases} 1 & \alpha \leq p_i \leq a_1 \\ \dfrac{a_2 - P}{a_2 - a_1} & a_1 \leq p_i \leq a_2 \\ 0 & Other\ wise \end{cases}$$

$$\mu_2(p_i) = \begin{cases} \dfrac{P - a_1}{a_2 - a_1} & a_1 \leq p_i \leq a_2 \\ 1 & a_2 \leq p_i \leq a_4 \\ \dfrac{\varphi - P}{\varphi - a_4} & a_4 \leq p_i \leq \varphi \\ 0 & Other\ wise \end{cases}$$

$$\mu_3(p_i)=\begin{cases}\dfrac{P-a_2}{a_4-a_2} & a_2\le p_i\le a_4\\ 1 & p_i=a_4\\ \dfrac{\varphi-P}{\varphi-a_4} & a_4\le p_i\le\varphi\\ 0 & Other\ wise\end{cases}$$

$$\mu_4(p_i)=\begin{cases}\dfrac{P-a_3}{a_4-a_3} & a_3\le p_i\le a_4\\ 1 & a_4\le p_i\le b_4\\ \dfrac{b_3-P}{b_3-b_4} & b_4\le p_i\le b_3\\ 0 & Other\ wise\end{cases}$$

$$\mu_5(p_i)=\begin{cases}\dfrac{P-\varphi}{b_4-\varphi} & \varphi\le p_i\le b_4\\ 1 & p_i=b_4\\ \dfrac{b_2-P}{b_2-b_4} & b_4\le p_i\le b_2\\ 0 & Other\ wise\end{cases}$$

$$\mu_6(p_i)=\begin{cases}\dfrac{P-\varphi}{b_4-\varphi} & \varphi\le p_i\le b_4\\ 1 & b_4\le p_i\le b_2\\ \dfrac{b_1-P}{b_1-b_2} & b_2\le p_i\le b_1\\ 0 & Other\ wise\end{cases}$$

$$\mu_7(p_i)=\begin{cases}\dfrac{P-b_2}{b_1-b_2} & b_2\le p_i\le b_1\\ 1 & b_1\le p_i\le\beta\\ 0 & Other\ wise\end{cases}$$

Formulation 4.2-4: The $\boldsymbol{\mu_k(p_i)}$ Linear Fuzzy Membership Functions

Since LFRF is used as a technical tool of fuzzy transformation of problem formulation, and also it is used as particular step process of the truth degrees technique on probability distribution to deffuzzifier fuzzy truth degrees intervals to stochastic truth degrees intervals, then via LFRF technique finite deterministic bounded discreate values set are obtained. In addition, to achieve fuzzy transformation, LFRF to defuzzify fuzzy coefficients and parameters in Fuzzy Programing Problems (FPP) is used. So, one of the most useful ranking functions proposed by various researchers is (Mahdavi-Amiri, N;Nasseri, SH, 2006; Mahdavi-Amiri;NezamNasseri;Seyed Hadi, 2007) since it could use for entire types of T_pFN because immediately after transforming fuzzy numbers to real number, its value will remain in the same interval and could be shown as average of fuzzy number components (Ameen, 2015). Therefore, LFRF uses in converting intervals of truth degrees to power useful important efficient points of its intervals.

For a $T_pFN\ \widetilde{\alpha}=(a^L,a^U,\alpha,\beta)$, we could formulate $R(F)(\widetilde{\alpha})$ as following:

$$R(F)(\widetilde{\alpha})=\frac{1}{2}\int_0^1\left(\inf(\widetilde{a_\lambda})+sup(\widetilde{a_\lambda})\right)d\lambda=\frac{a^L+a^U}{2}+\frac{\beta-\alpha}{4}$$

Formulation 4.2-5: The Linear Fuzzy Ranking Function Technique Formula For $\boldsymbol{T_pFN}$

Also, for a $T_rFN\ \widetilde{\alpha}=(a,\alpha,\beta)$, we could formulate $R(F)(\widetilde{\alpha})$ as following:

$$R(F)(\widetilde{\alpha})=\frac{1}{2}\int_0^1\left(\inf(\widetilde{a_\lambda})+sup(\widetilde{a_\lambda})\right)d\lambda=a+\frac{\beta-\alpha}{4}$$

Formulation 4.2-6: The Linear Fuzzy Ranking Function Technique Formula For $\boldsymbol{T_rFN}$

Now, by using the above linear fuzzy ranking function LFRF technique Formulation (4.2-4) and Formulation (4.2-5) we defuzzifying those $\mu_k(p_i), k=1,2,..,7; i=1,2,\dots,N$ linear fuzzy membership functions LFMF Formulation (4.2-3) for obtaining deterministic discrete values for each p_i.

4.2.1 The Ranking Function Technique Program Outline

The program outline of linear fuzzy ranking function LFRF technique approach will be stated in subsection (7.1.3).

4.2.2 The Ranking Function Technique MATLAB Program

The MATLAB Program of linear fuzzy ranking function LFRF technique approach will be stated in subsection (7.1.4).

4.2.3 The Truth Degrees Technique Program Outline

The program outline of truth degrees technique approach will be stated in subsection (7.1.5).

4.2.4 The Truth Degrees Technique MATLAB Program: The MATLAB Program of the truth degrees technique approach will be stated in subsection (7.1.6).

4.2.5 Analyzing Cases: Analyzing cases starts after all p_i of $p=(p_1,p_2,\dots,p_N)^T\in\mathbb{R}^N$ of probability distribution space vector convert to crisp values, and then each caseq that does not apply $\sum_{k=1}^{N}p_k=1, p_k\ge 0, k=1,\dots,N$ of Formulation (4.1-3) should be neglected to get acceptance to stochastic transformation.

4.2.6 The Analyzing Cases Test Program Outline

The program outline of Analyzing Cases Test will be stated in subsection (7.1.7).

4.2.7 The Analyzing Cases Test MATLAB Program

The MATLAB code program of analysing cases test will be stated in subsection (7.1.8).

4.3 Stochastic Transformation of Objective Function Problem Formulation via The Expectation Weighted Summation EWS Technique

The stochastic transformation of objective functions of Formulation (3.1-1) via expected weighted summation EWS technique approach on random objective coefficients applies to convert STLPP to DTLPP, for a probability distribution space $p=(p_1,p_2,\dots,p_N)^T\in\mathbb{R}^N$, then stochastic transformation of objective function coefficients as following applies:

$$\begin{aligned}Min\ Exp_{P\in\pi}\,z(\omega,x)&=Min\ Exp_{P\in\pi}\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}(\omega)x_{ij}\\&=Min\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\left(Exp_{P\in\pi}\,c_{ij}(\omega)\right)\\&=Min\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\left(\sum_{k=1}^{N}c_{ij}(\omega_k)p_k\right)\\&=Min\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}\,c_{ij}=Min\ z(x)\end{aligned}$$

$$\forall k\in\mathbb{N}; i=1,2,\dots,m; j=1,2,\dots,n, k=1,2,\dots,N$$

Formulation 4.3-1: The Expected Weighted Summation EWS Technique Procedure

4.3.1 The EWS Technique Program Outline

The program outline of expected weighted summation EWS technique will be stated in subsection (7.1.9).

4.3.2 The EWS Technique MATLAB Program

The MATLAB Program of the expected weighted summation EWS technique will be stated in subsection (7.1.10).

4.4 Finding Basic Feasible Solution BFS for DTLPP via Dual-Simplex and Vogel Algorithm

When STLPPFI is converted to DTLPP, its Basic Feasible Solution BFS can be found via various methods such as North-West Corner Method, MMCM, Dual-Simplex Method and VAM. So, Dual-Simplex with VAM are used, and both methods will be illustrated in the following real-problem example (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017). Where algorithms of both methods are stated in a part of STLPPFI problem solver MATLAB code program.

4.5 Finding Optimal Solution for DTLPP via Modify Distribution Method MODI

When Basic Feasible Solution BFS for DTLLP is found via Dual-Simplex and VAM, then Stepping Stone Method SSM and Modify Distribution Method MODI are used to check if BFS is optimal or not. Both methods are used to find the optimal solution when BFS is not optimal. MODI is recommended for use in such cases, and it will be illustrated in the following real-problem example as well (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017). MODI algorithms are stated in a part of STLPPFI problem solver MATLAB code program.

4.6 Selecting Post Optimal Result for DTLPP

Selecting post optimal solution as a final result via deciding commands from decision makers (DM) among the entire exit intervals for a certain case or analyzing results by answering what perfect solution is entirely in certain cases.

5. THE REAL-LIFE APPLICATION PROBLEM EXAMPLE USING MATLAB

5.1 Illustrate Example

Suppose that three electricity power production stations in Kurdistan region-Iraq namely: G1, G2 and G3 with four cities need to be supplied with electricity namely Hawler City, Duhok City, Sulaymani City and Halabja City (Government, 2020; TV, 2021) as following balanced STLPPFI problem table:

Table : 5.1-1: The Data Distribution Table of an Electricity STLPPFI Real-Life Problem

Power Plants	Cites				Supply Million Kw/h
	HR	DK	SI	HA	
G_1	$c_{11}(\omega)$	$c_{12}(\omega)$	$c_{13}(\omega)$	$c_{14}(\omega)$	36
G_2	$c_{21}(\omega)$	$c_{22}(\omega)$	$c_{23}(\omega)$	$c_{24}(\omega)$	51
G_3	$c_{31}(\omega)$	$c_{32}(\omega)$	$c_{33}(\omega)$	$c_{34}(\omega)$	42
Demand Million Kw/h	46	22	31	30	Total = 129 M Kw/h Balanced

Where prices of transporting costs are stochastically, and estimated cost values of random matrix of each $c_{ij}(\omega)$ will be (Government, 2020; TV, 2021) as follows:

Table: 5.1-2: The Probably Estimated Cost Values of Probability Distribution Space

ω	ω_1	ω_2	ω_3
$c_{11}(\omega)$	7.980 IQD	7.990 IQD	8.000 IQD
$c_{12}(\omega)$	6.000 IQD	5.990 IQD	5.980 IQD
$c_{13}(\omega)$	9.990 IQD	9.980 IQD	10.00 IQD
$c_{14}(\omega)$	8.880 IQD	8.890 IQD	9.000 IQD
$c_{21}(\omega)$	9.000 IQD	8.980 IQD	8.960 IQD
$c_{22}(\omega)$	11.98 IQD	11.96 IQD	12.00 IQD
$c_{23}(\omega)$	13.00 IQD	12.98 IQD	12.96 IQD
$c_{24}(\omega)$	6.980 IQD	6.960 IQD	7.000 IQD
$c_{31}(\omega)$	13.96 IQD	13.98 IQD	14.00 IQD
$c_{32}(\omega)$	9.000 IQD	8.980 IQD	8.960 IQD
$c_{33}(\omega)$	15.96 IQD	15.98 IQD	16.00 IQD
$c_{34}(\omega)$	4.990 IQD	5.000 IQD	4.980 IQD

Where information regarding the response of cities on electricity power plants is distributed as fuzzy polyhedral set $\tilde{\pi}$ of uncertainty unknown information on probability distribution (Government, 2020; TV, 2021) as follows:

$$\tilde{\pi} = \left\{p = (p_1, p_2, p_3)^T \in \mathbb{R}^3; Ap \preccurlyeq b; \sum_{i=1}^{3} p_i = 1; \forall p_i \geq 0; i = 1,2,3\right\}$$

Formulation 5.1-1: The Fuzzy Polyhedral Information Set $\boldsymbol{\tilde{\pi}}$ of Real-Life Problem

5.2 Solution Process

The problem could formulate as follows: Objective Function:

$$Min\ z(\omega,x) = x_{11}c_{11}(\omega) + x_{12}c_{12}(\omega) + x_{13}c_{13}(\omega) + x_{14}c_{14}(\omega) + x_{21}c_{21}(\omega) + x_{22}c_{22}(\omega) + x_{23}c_{23}(\omega) + x_{24}c_{24}(\omega) + x_{31}c_{31}(\omega) + x_{32}c_{32}(\omega) + x_{33}c_{33}(\omega) + x_{34}c_{34}(\omega)$$

Subject to both availability and requirement constraints

$$x_{11} + x_{12} + x_{13} + x_{14} = 36, x_{21} + x_{22} + x_{23} + x_{24} = 51$$
$$x_{31} + x_{32} + x_{33} + x_{34} = 42$$
$$x_{11} + x_{21} + x_{31} = 46, x_{12} + x_{22} + x_{32} = 22$$
$$x_{13} + x_{23} + x_{33} = 31, x_{14} + x_{24} + x_{34} = 30$$

Where balanced condition with domain condition satisfies respectively

$$\sum_{i=1}^{3} a_i = 36 + 51 + 42 = \sum_{j=1}^{4} b_j = 46 + 22 + 31 + 30 = 129$$
$$x_{ij} \geq 0, \forall i = 1,2,3,; \forall j = 1,2,3,4;\ x \in X, \omega \in \Omega$$

Formulation 5.2-1: The STLPPFI Real-Life Application Problem

The problem has uncertainty unknown expression in both randomness for objective function coefficients and fuzziness for information probability distribution space $(\Omega, 2^{\Omega}, P)$. The STLPPFI solution's process starts from deffuzzifier fuzziness and derandomizing randomness of it respectively via two main transformations followed by analyzing cases, solving using Dual-Simplex and VAM, finding the optimal solution via MODI and selecting post optimal solution. Where a_i, b_j are crisp and do not appear scholastically and they are (3,1) and (1,4) known vectors as shown in Table (5.1-1) respectively, and c_{ij} is not crisp and appears scholastically which needs to be determined via suitable transformations, and $c_{ij}(\omega)$ is (3,4) random matrix as shown in Table (5.1-1) with estimated probably values for each $c_{ij}(\omega)$ shown in Table (5.1-2)The matrix x_{ij} is (3,4) unknown matrix should be determined via Dual-Simplex and VAM, under fuzzy uncertainty unknown information on probability distribution P. The Formulation (5.2-1) is defined in terms of some probability distribution space $(\Omega, 2^{\Omega}, P)$, where $\Omega = \{\omega_1, \omega_2, \omega_3\} = \{\omega_i\}, i = 1,2,3$ is a discrete set of events or a finite set of possible states of nature, 2^{Ω} is power set of Ω, and P is fuzzy uncertainty unknown information on probability distribution space. Where that P assigns to each $A \in 2^{\Omega}$ is the probability of occurrence $P(A)$ (i.e., P is the $(12,3)$ matrix of probabilities $p_i = P(\{\omega = \omega_i\}), i = 1,2, \dots, 12, p_i \in \tilde{\pi}, \forall i$), with fuzzy distribution information of response cities on electricity power plants are as fuzzy polyhedral set $\tilde{\pi}$ Formulation (5.1-1). Also, the set X is a polyhedral set of feasible solutions that includes determined constraints of problem on probability distribution space $(\Omega, 2^{\Omega}, P)$. To solve Formulation (5.2-1) we need to find a set of non-negatives $x_{ij}, \forall i, j$ that minimize objective function, satisfies constraints, balances condition and domain conditions.

Now, the solution process classified over steps/stations as follows:

5.2.1 Station 1: Suppositions of Stochastics: Suppose that the credibility degree of DM about information on probability distribution of p_1, p_2, p_3 are around $\frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ respectively, where $P(p_1) \cong b_1 = \frac{1}{2}, P(p_2) \cong b_2 = \frac{1}{5}, P(p_3) \cong b_3 = \frac{1}{10}$ which is mean $(b_1, b_2, b_3) = \left(\frac{1}{2}, \frac{1}{5}, \frac{1}{10}\right)$, since the information on probability distribution are fuzzy. So, the suppositions have been started, where the vagueness levels are $(d_1, d_2, d_3) = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$ and the alpha-cut levels are $(\alpha_1, \alpha_2, \alpha_3) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. Note that the vagueness levels and the alpha-cut levels control the length of interval in each p_i allowing for their reuse k-times as necessary to obtain most close approximate values from exact values till we discover which values have efficient on solution process results.

5.2.2 Station 2: Fuzzy Transformation on Probability Distribution Space: Fuzzy transformation on probability distribution will be applied via alpha-cut technique Formulation (4.1-3) for each fuzzy inequalities polyhedral set Formulation (5.1-1) as follows:

$$\pi^{1,2,3} = \left\{ \begin{array}{c} p = (p_1, p_2, p_3)^T \in \mathbb{R}^3; \\ b_i - d_i(1 - \alpha_i) \le p_i \le b_i + d_i(1 - \alpha_i); \\ \sum_{i=1}^{3} p_i = 1, \forall p_i \ge 0, i = 1,2,3 \end{array} \right\}$$

Then convert Formulation (5.2-1) from STLPPFI to STLPP via apply above alpha-cut formula for each $p_{1,2,3}$, we get:

$$b_1 - d_1(1 - \alpha_1) \le p_1 \le b_1 + d_1(1 - \alpha_1)$$
$$\Rightarrow \frac{1}{2} - \frac{1}{6}\left(1 - \frac{1}{2}\right) \le p_1 \le \frac{1}{2} + \frac{1}{6}\left(1 - \frac{1}{2}\right) \Rightarrow \frac{25}{60} \le p_1 \le \frac{35}{60}$$

$$b_2 - d_2(1 - \alpha_2) \le p_2 \le b_2 + d_2(1 - \alpha_2)$$
$$\Rightarrow \frac{1}{5} - \frac{1}{6}\left(1 - \frac{1}{2}\right) \le p_2 \le \frac{1}{5} + \frac{1}{6}\left(1 - \frac{1}{2}\right) \Rightarrow \frac{7}{60} \le p_2 \le \frac{17}{60}$$

$$b_3 - d_3(1 - \alpha_3) \le p_3 \le b_3 + d_3(1 - \alpha_3)$$
$$\Rightarrow \frac{1}{10} - \frac{1}{6}\left(1 - \frac{1}{2}\right) \le p_3 \le \frac{1}{10} + \frac{1}{6}\left(1 - \frac{1}{2}\right) \Rightarrow \frac{1}{60} \le p_3 \le \frac{11}{60}$$

Now, the fuzzy transformation on the probability distribution space via alpha-cut technique is applied, in which the fuzziness of probability distribution of STLPPFI is removed, and STLPPFI transfers to STLPP by creating bounded interval with unlimited possible known values from unknown probably/stochastic value i.e., $b_i; \forall i = 1,2,3$ converted from stochastic values $b_{1,2,3} = \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ to bounded interval with unlimited possible known values $p_{i=1,2,3}$ as $\frac{25}{60} \le p_1 \le \frac{35}{60}, \frac{7}{60} \le p_2 \le \frac{17}{60}, \frac{1}{60} \le p_1 \le \frac{11}{60}$. Then fuzzy polyhedral set Formulation (5.1-1) of STLPPFI Formulation (5.2-1) convert to stochastic polyhedral version set as follows:

$$\pi = \left\{ \begin{array}{c} p = (p_1, p_2, p_3)^T \in \mathbb{R}^3; Ap \le b; \sum_{i=1}^{3} p_i = 1; \\ \forall p_i \ge 0; i = 1,2,3 \end{array} \right\}$$

5.2.3 Station 3: The Truth Degrees Technique on Probability Distribution Intervals: The current step will be converting each bounded interval with unlimited possible known values to bounded discrete finite possible known values set via truth degrees technique approach, to choose the best effective values for each $p_{1,2,3}$ in each interval we use truth degrees logical values. First, we divide each interval into ten parts as follows:

$$\frac{25}{60} \le p_1 \le \frac{35}{60} \Rightarrow \left\{\left[\frac{25}{60}, \frac{35}{60}\right]\right\}$$
$$= \left\{\frac{25}{60}, \frac{26}{60}, \frac{27}{60}, \frac{28}{60}, \frac{29}{60}, \frac{30}{60}, \frac{31}{60}, \frac{32}{60}, \frac{33}{60}, \frac{34}{60}, \frac{35}{60}\right\}$$
$$\frac{7}{60} \le p_2 \le \frac{17}{60} \Rightarrow \left\{\left[\frac{7}{60}, \frac{17}{60}\right]\right\}$$
$$= \left\{\frac{7}{60}, \frac{8}{60}, \frac{9}{60}, \frac{10}{60}, \frac{11}{60}, \frac{12}{60}, \frac{13}{60}, \frac{14}{60}, \frac{15}{60}, \frac{16}{60}, \frac{17}{60}\right\}$$
$$\frac{1}{60} \le p_3 \le \frac{11}{60} \Rightarrow \left\{\left[\frac{1}{60}, \frac{11}{60}\right]\right\}$$
$$= \left\{\frac{1}{60}, \frac{2}{60}, \frac{3}{60}, \frac{4}{60}, \frac{5}{60}, \frac{6}{60}, \frac{7}{60}, \frac{8}{60}, \frac{9}{60}, \frac{10}{60}, \frac{11}{60}\right\}$$

5.2.4 Station 4: Finding Modulus/Membership Function of Truth Degrees Set: Formulation (4.2-3) with Figure (4.2-1) will be used to fuzzifier truth degrees set and to find seven LFMF for each $p_{1,2,3}$ as follows:

$$\mu_1(p_1) = \begin{cases} 1 & \frac{25}{60} \le p_1 \le \frac{26}{60} \\ \dfrac{\frac{27}{60} - P}{\frac{27}{60} - \frac{26}{60}} & \frac{26}{60} \le p_1 \le \frac{27}{60} \\ 0 & Other\ wise \end{cases}$$

$$\mu_2(p_1) = \begin{cases} \dfrac{P - \frac{26}{60}}{\frac{27}{60} - \frac{26}{60}}; \frac{26}{60} \le p_1 \le \frac{27}{60} \\ 1; \frac{27}{60} \le p_1 \le \frac{30}{60} \\ \dfrac{\frac{30}{60} - P}{\frac{30}{60} - \frac{29}{60}}; \frac{29}{60} \le p_1 \le \frac{30}{60} \\ 0; Other\ wise \end{cases}$$

$$\mu_3(p_1) = \begin{cases} \dfrac{P - \frac{27}{60}}{\frac{29}{60} - \frac{27}{60}}; \frac{27}{60} \le p_1 \le \frac{29}{60} \\ 1; p_1 = \frac{29}{60} \\ \dfrac{\frac{30}{60} - P}{\frac{30}{60} - \frac{29}{60}}; \frac{29}{60} \le p_1 \le \frac{30}{60} \\ 0; Other\ wise \end{cases}$$

$$\mu_4(p_1) = \begin{cases} \dfrac{P - \frac{28}{60}}{\frac{29}{60} - \frac{28}{60}}; \frac{28}{60} \le p_1 \le \frac{29}{60} \\ 1; \frac{29}{60} \le p_1 \le \frac{31}{60} \\ \dfrac{\frac{33}{60} - P}{\frac{33}{60} - \frac{31}{60}}; \frac{31}{60} \le p_1 \le \frac{33}{60} \\ 0; Other\ wise \end{cases}$$

$$\mu_5(p_1) = \begin{cases} \dfrac{P - \frac{30}{60}}{\frac{31}{60} - \frac{30}{60}}; \frac{30}{60} \le p_1 \le \frac{31}{60} \\ 1; p_1 = \frac{31}{60} \\ \dfrac{\frac{33}{60} - P}{\frac{33}{60} - \frac{31}{60}}; \frac{31}{60} \le p_1 \le \frac{33}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_6(p_1) = \begin{cases} \dfrac{P - \frac{30}{60}}{\frac{31}{60} - \frac{30}{60}}; \frac{30}{60} \le p_1 \le \frac{31}{60} \\ 1; \frac{31}{60} \le p_1 \le \frac{33}{60} \\ \dfrac{\frac{34}{60} - P}{\frac{34}{60} - \frac{33}{60}}; \frac{33}{60} \le p_1 \le \frac{34}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_7(p_1) = \begin{cases} \dfrac{P - \frac{33}{60}}{\frac{34}{60} - \frac{33}{60}}; \frac{33}{60} \le p_1 \le \frac{34}{60} \\ 1; \frac{34}{60} \le p_1 \le \frac{35}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_1(p_2) = \begin{cases} 1; \frac{7}{60} \le p_2 \le \frac{8}{60} \\ \dfrac{\frac{9}{60} - P}{\frac{9}{60} - \frac{8}{60}}; \frac{8}{60} \le p_2 \le \frac{9}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_2(p_2) = \begin{cases} \dfrac{P - \frac{8}{60}}{\frac{9}{60} - \frac{8}{60}}; \frac{8}{60} \le p_2 \le \frac{9}{60} \\ 1; \frac{9}{60} \le p_2 \le \frac{11}{60} \\ \dfrac{\frac{12}{60} - P}{\frac{12}{60} - \frac{11}{60}}; \frac{11}{60} \le p_2 \le \frac{12}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_3(p_2) = \begin{cases} \dfrac{P - \frac{9}{60}}{\frac{11}{60} - \frac{9}{60}}; \frac{9}{60} \le p_2 \le \frac{11}{60} \\ 1; p_2 = \frac{11}{60} \\ \dfrac{\frac{12}{60} - P}{\frac{12}{60} - \frac{11}{60}}; \frac{11}{60} \le p_2 \le \frac{12}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_4(p_2) = \begin{cases} \dfrac{P - \frac{10}{60}}{\frac{11}{60} - \frac{10}{60}}; \frac{10}{60} \le p_2 \le \frac{11}{60} \\ 1; \frac{11}{60} \le p_2 \le \frac{13}{60} \\ \dfrac{\frac{14}{60} - P}{\frac{14}{60} - \frac{13}{60}}; \frac{13}{60} \le p_2 \le \frac{14}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_5(p_2) = \begin{cases} \dfrac{P - \frac{12}{60}}{\frac{13}{60} - \frac{12}{60}}; \frac{12}{60} \le p_2 \le \frac{13}{60} \\ 1; p_2 = \frac{13}{60} \\ \dfrac{\frac{15}{60} - P}{\frac{15}{60} - \frac{13}{60}}; \frac{13}{60} \le p_2 \le \frac{15}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_6(p_2) = \begin{cases} \dfrac{P - \frac{12}{60}}{\frac{13}{60} - \frac{12}{60}}; \frac{12}{60} \le p_2 \le \frac{13}{60} \\ 1; \frac{13}{60} \le p_2 \le \frac{15}{60} \\ \dfrac{\frac{16}{60} - P}{\frac{16}{60} - \frac{15}{60}}; \frac{15}{60} \le p_2 \le \frac{16}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_7(p_2) = \begin{cases} \dfrac{P - \frac{15}{60}}{\frac{16}{60} - \frac{15}{60}}; \frac{15}{60} \le p_2 \le \frac{16}{60} \\ 1; \frac{16}{60} \le p_2 \le \frac{17}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_1(p_3) = \begin{cases} 1; \frac{1}{60} \le p_3 \le \frac{2}{60} \\ \dfrac{\frac{3}{60} - P}{\frac{3}{60} - \frac{2}{60}}; \frac{2}{60} \le p_3 \le \frac{3}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_2(p_3) = \begin{cases} \dfrac{P - \frac{2}{60}}{\frac{3}{60} - \frac{2}{60}}; \frac{2}{60} \le p_3 \le \frac{3}{60} \\ 1; \frac{3}{60} \le p_3 \le \frac{5}{60} \\ \dfrac{\frac{6}{60} - P}{\frac{6}{60} - \frac{5}{60}}; \frac{5}{60} \le p_3 \le \frac{6}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_3(p_3) = \begin{cases} \dfrac{P - \frac{3}{60}}{\frac{5}{60} - \frac{3}{60}}; \frac{3}{60} \le p_3 \le \frac{5}{60} \\ 1; p_3 = \frac{5}{60} \\ \dfrac{\frac{6}{60} - P}{\frac{6}{60} - \frac{5}{60}}; \frac{5}{60} \le p_3 \le \frac{6}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_4(p_3) = \begin{cases} \dfrac{P - \frac{4}{60}}{\frac{5}{60} - \frac{4}{60}}; \frac{4}{60} \le p_3 \le \frac{5}{60} \\ 1; \frac{5}{60} \le p_3 \le \frac{7}{60} \\ \dfrac{\frac{8}{60} - P}{\frac{8}{60} - \frac{7}{60}}; \frac{7}{60} \le p_3 \le \frac{8}{60} \\ 0;\ Other\ wise \end{cases}$$

$$\mu_5(p_3)=\begin{cases}\dfrac{P-\frac{6}{60}}{\frac{7}{60}-\frac{6}{60}};\frac{6}{60}\le p_3\le\frac{7}{60}\\ 1;p_3=\frac{7}{60}\\ \dfrac{\frac{9}{60}-P}{\frac{9}{60}-\frac{7}{60}};\frac{7}{60}\le p_3\le\frac{9}{60}\\ 0;\ Other\ wise\end{cases}$$

$$\mu_6(p_3)=\begin{cases}\dfrac{P-\frac{6}{60}}{\frac{7}{60}-\frac{6}{60}};\frac{6}{60}\le p_3\le\frac{7}{60}\\ 1;\frac{7}{60}\le p_3\le\frac{9}{60}\\ \dfrac{\frac{10}{60}-P}{\frac{10}{60}-\frac{9}{60}};\frac{9}{60}\le p_3\le\frac{10}{60}\\ 0;\ Other\ wise\end{cases}$$

$$\mu_7(p_3)=\begin{cases}\dfrac{P-\frac{9}{60}}{\frac{10}{60}-\frac{9}{60}};\frac{9}{60}\le p_3\le\frac{10}{60}\\ 1;\frac{10}{60}\le p_3\le\frac{11}{60}\\ 0;\ Other\ wise\end{cases}$$

Now, both kinds of fuzzy numbers are obtaining from above LFMFs' of truth degrees set as follows:
$T_rFN\ \tilde{\alpha}\left(\mu_1(p_1)\right)=(a,\alpha,\beta)=\left(\frac{26}{60},\frac{25}{60},\frac{27}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_2(p_1)\right)=(a^L,a^U,\alpha,\beta)=\left(\frac{27}{60},\frac{29}{60},\frac{26}{60},\frac{30}{60}\right)$,
$T_rFN\ \tilde{\alpha}\left(\mu_3(p_1)\right)=\left(\frac{29}{60},\frac{27}{60},\frac{30}{60}\right)$,$T_pFN\ \tilde{\alpha}\left(\mu_4(p_1)\right)=\left(\frac{29}{60},\frac{31}{60},\frac{28}{60},\frac{32}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_5(p_1)\right)=\left(\frac{31}{60},\frac{30}{60},\frac{33}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_6(p_1)\right)=\left(\frac{31}{60},\frac{33}{60},\frac{30}{60},\frac{34}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_7(p_1)\right)=\left(\frac{34}{60},\frac{33}{60},\frac{35}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_1(p_2)\right)=(a,\alpha,\beta)=\left(\frac{8}{60},\frac{7}{60},\frac{9}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_2(p_2)\right)=(a^L,a^U,\alpha,\beta)=\left(\frac{9}{60},\frac{11}{60},\frac{8}{60},\frac{12}{60}\right)$,
$T_rFN\ \tilde{\alpha}\left(\mu_3(p_2)\right)=\left(\frac{11}{60},\frac{9}{60},\frac{12}{60}\right)$,$T_pFN\ \tilde{\alpha}\left(\mu_4(p_2)\right)=\left(\frac{11}{60},\frac{13}{60},\frac{10}{60},\frac{14}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_5(p_2)\right)=\left(\frac{13}{60},\frac{12}{60},\frac{15}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_6(p_2)\right)=\left(\frac{13}{60},\frac{15}{60},\frac{12}{60},\frac{16}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_7(p_2)\right)=\left(\frac{16}{60},\frac{15}{60},\frac{17}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_1(p_3)\right)=(a,\alpha,\beta)=\left(\frac{2}{60},\frac{1}{60},\frac{3}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_2(p_3)\right)=(a^L,a^U,\alpha,\beta)=\left(\frac{3}{60},\frac{5}{60},\frac{2}{60},\frac{6}{60}\right)$,
$T_rFN\ \tilde{\alpha}\left(\mu_3(p_3)\right)=\left(\frac{5}{60},\frac{3}{60},\frac{6}{60}\right)$,$T_pFN\ \tilde{\alpha}\left(\mu_4(p_3)\right)=\left(\frac{5}{60},\frac{7}{60},\frac{4}{60},\frac{8}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_5(p_3)\right)=\left(\frac{7}{60},\frac{6}{60},\frac{9}{60}\right)$,
$T_pFN\ \tilde{\alpha}\left(\mu_6(p_3)\right)=\left(\frac{7}{60},\frac{9}{60},\frac{6}{60},\frac{10}{60}\right)$,$T_rFN\ \tilde{\alpha}\left(\mu_7(p_3)\right)=\left(\frac{10}{60},\frac{9}{60},\frac{11}{60}\right)$

Now, Linear Fuzzy Ranking Function LFRF Formulation (4.2-4) and Formulation (4.2-5) will be used to defuzzify fuzzily above fuzzy numbers of LFMFs' of truth degrees set as follows:

$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_1(p_1)\right)\right)=a+\frac{\beta-\alpha}{4}$$
$$=\frac{26}{60}+\frac{\frac{27}{60}-\frac{25}{60}}{4}=\frac{26.5}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_2(p_1)\right)\right)=\frac{a^L+a^U}{2}+\frac{\beta-\alpha}{4}$$
$$=\frac{\frac{27}{60}+\frac{29}{60}}{2}+\frac{\left(\frac{30}{60}-\frac{26}{60}\right)}{4}=\frac{29}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_3(p_1)\right)\right)=\frac{29}{60}+\frac{\left(\frac{30}{60}-\frac{27}{60}\right)}{4}=\frac{29.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_4(p_1)\right)\right)=\frac{\frac{29}{60}+\frac{31}{60}}{2}+\frac{\left(\frac{32}{60}-\frac{28}{60}\right)}{4}=\frac{31}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_5(p_1)\right)\right)=\frac{31}{60}+\frac{\left(\frac{33}{60}-\frac{30}{60}\right)}{4}=\frac{31.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_6(p_1)\right)\right)=\frac{\frac{31}{60}+\frac{33}{60}}{2}+\frac{\left(\frac{34}{60}-\frac{30}{60}\right)}{4}=\frac{33}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_7(p_1)\right)\right)=\frac{34}{60}+\frac{\left(\frac{35}{60}-\frac{33}{60}\right)}{4}=\frac{34.5}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_1(p_2)\right)\right)=\frac{8}{60}+\frac{\left(\frac{9}{60}-\frac{7}{60}\right)}{4}=\frac{8.5}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_2(p_2)\right)\right)=\frac{\frac{9}{60}+\frac{11}{60}}{2}+\frac{\left(\frac{12}{60}-\frac{8}{60}\right)}{4}=\frac{11}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_3(p_2)\right)\right)=\frac{11}{60}+\frac{\left(\frac{12}{60}-\frac{9}{60}\right)}{4}=\frac{11.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_4(p_2)\right)\right)=\frac{\frac{11}{60}+\frac{13}{60}}{2}+\frac{\left(\frac{14}{60}-\frac{10}{60}\right)}{4}=\frac{13}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_5(p_2)\right)\right)=\frac{13}{60}+\frac{\left(\frac{15}{60}-\frac{12}{60}\right)}{4}=\frac{13.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_6(p_2)\right)\right)=\frac{\frac{13}{60}+\frac{15}{60}}{2}+\frac{\left(\frac{16}{60}-\frac{12}{60}\right)}{4}=\frac{15}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_7(p_2)\right)\right)=\frac{16}{60}+\frac{\left(\frac{17}{60}-\frac{15}{60}\right)}{4}=\frac{16.5}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_1(p_3)\right)\right)=\frac{2}{60}+\frac{\frac{3}{60}-\frac{1}{60}}{4}=\frac{2.5}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_2(p_3)\right)\right)=\frac{\frac{3}{60}+\frac{5}{60}}{2}+\frac{\left(\frac{6}{60}-\frac{2}{60}\right)}{4}=\frac{5}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_3(p_3)\right)\right)=\frac{5}{60}+\frac{\left(\frac{6}{60}-\frac{3}{60}\right)}{4}=\frac{5.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_4(p_3)\right)\right)=\frac{\frac{5}{60}+\frac{7}{60}}{2}+\frac{\left(\frac{8}{60}-\frac{4}{60}\right)}{4}=\frac{7}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_5(p_3)\right)\right)=\frac{7}{60}+\frac{\left(\frac{9}{60}-\frac{6}{60}\right)}{4}=\frac{7.75}{60}$$
$$R(T_pFN)\left(\tilde{\alpha}\left(\mu_6(p_3)\right)\right)=\frac{\frac{7}{60}+\frac{9}{60}}{2}+\frac{\left(\frac{10}{60}-\frac{6}{60}\right)}{4}=\frac{9}{60}$$
$$R(T_rFN)\left(\tilde{\alpha}\left(\mu_7(p_3)\right)\right)=\frac{10}{60}+\frac{\left(\frac{11}{60}-\frac{9}{60}\right)}{4}=\frac{10.5}{60}$$

Finally, nine efficient points are obtained for each $p_{1,2,3}$ in each interval as follows:

$$\frac{25}{60}\le p_1\le\frac{35}{60}\Rightarrow$$
$$p_1\in\left\{\frac{25}{60},\frac{26.5}{60},\frac{29}{60},\frac{29.75}{60},\frac{31}{60},\frac{31.75}{60},\frac{33}{60},\frac{34.5}{60},\frac{35}{60}\right\}$$
$$\frac{7}{60}\le p_2\le\frac{17}{60}\Rightarrow$$
$$p_2\in\left\{\frac{7}{60},\frac{8.5}{60},\frac{11}{60},\frac{11.75}{60},\frac{13}{60},\frac{13.75}{60},\frac{15}{60},\frac{16.5}{60},\frac{17}{60}\right\}$$
$$\frac{1}{60}\le p_3\le\frac{11}{60}\Rightarrow$$
$$p_3\in\left\{\frac{1}{60},\frac{2.5}{60},\frac{5}{60},\frac{5.75}{60},\frac{7}{60},\frac{7.75}{60},\frac{9}{60},\frac{10.5}{60},\frac{11}{60}\right\}$$

5.2.5 Station 5: Analyzing Cases: We have 9 cases for each p_i, then totally obtain $9^i; i \in \mathbb{N}$ cases to check. So, for the current example $9^3 = 729$ cases are existed. Now, to check all cases to state condition $p = (p_{1h}, p_{2h}, p_{3h})^T \in \mathbb{R}^3, \sum_{i=1}^{N} p_{ih} = 1, p_{ih} \geq 0, i = 1,2,3, h = 1,2, \ldots, 9$ of alpha-cut technique as follows:

For case (1,1,1): let $(p_{11}, p_{21}, p_{31}) = \left(\frac{25}{60}, \frac{7}{60}, \frac{1}{60}\right)$ then $\forall p_1, p_2, p_3 \geq 0$ and $p_1 + p_2 + p_3 = \frac{25+7+1}{60} = \frac{33}{60} \neq 1$, that is failed.

For case (7,8,8): let $(p_{17}, p_{28}, p_{38}) = \left(\frac{33}{60}, \frac{16.5}{60}, \frac{10.5}{60}\right)$ then $\forall p_1, p_2, p_3 \geq 0$ and $p_1 + p_2 + p_3 = \frac{33+16.5+10.5}{60} = \frac{60}{60} = 1$, that is pass and so on for entire other cases. So, all of them failed except 3 cases which are (p_{17}, p_{28}, p_{38}), (p_{18}, p_{27}, p_{38}) and (p_{18}, p_{28}, p_{37}) are passes as follows:

$$(p_{17}, p_{28}, p_{38}) = \left(\frac{33}{60}, \frac{16.5}{60}, \frac{10.5}{60}\right)$$
$$(p_{18}, p_{27}, p_{38}) = \left(\frac{34.5}{60}, \frac{15}{60}, \frac{10.5}{60}\right)$$
$$(p_{18}, p_{28}, p_{37}) = \left(\frac{34.5}{60}, \frac{16.5}{60}, \frac{9}{60}\right)$$

5.2.6 Station 6: Stochastic Transformation of Objective Coefficients via Expected Weighted Summation Technique Approach: We have 3 acceptance cases to apply stochastic transformation of objective coefficients on it to convert from STLPP to DTLPP by applying Formulation (4.3-1) on Table (5.1-2) with entire 3 accepted cases of $p = (p_{17}, p_{28}, p_{38}) = \left(\frac{33}{60}, \frac{16.5}{60}, \frac{10.5}{60}\right) \in \mathbb{R}^3, p = (p_{18}, p_{27}, p_{38}) = \left(\frac{34.5}{60}, \frac{15}{60}, \frac{10.5}{60}\right) \in \mathbb{R}^3, p = (p_{18}, p_{28}, p_{37}) = \left(\frac{34.5}{60}, \frac{16.5}{60}, \frac{9}{60}\right) \in \mathbb{R}^3$. The expectation weighted summation Formulation (4.3-1) for each three above probability spaces $p = (p_1, p_2, p_3) \in \mathbb{R}^3$, then transformation of objective coefficients will be as follows:

$$Exp\ c_{ij}(\omega) = \sum_{k=1}^{3} c_{ij}(\omega_k) p_k = c_{ij}\ ; i = 1,2,3; j = 1,2,3,4$$

$$Exp\ c_{11}(\omega) = c_{11}(\omega_1)p_1 + c_{11}(\omega_2)p_2 + c_{11}(\omega_3)p_3 = c_{11}$$
$$= (7.98)\left(\frac{33}{60}\right) + (7.99)\left(\frac{16.5}{60}\right) + (8)\left(\frac{10.5}{60}\right) = 7.98625$$

$$Exp\ c_{12}(\omega) = c_{12}(\omega_1)p_1 + c_{12}(\omega_2)p_2 + c_{12}(\omega_3)p_3 = c_{12}$$
$$= (6)\left(\frac{33}{60}\right) + (5.99)\left(\frac{16.5}{60}\right) + (5.98)\left(\frac{10.5}{60}\right) = 5.99375$$

$$Exp\ c_{13}(\omega) = c_{13}(\omega_1)p_1 + c_{13}(\omega_2)p_2 + c_{13}(\omega_3)p_3 = c_{13}$$
$$= (9.99)\left(\frac{33}{60}\right) + (9.98)\left(\frac{16.5}{60}\right) + (10)\left(\frac{10.5}{60}\right) = 9.989$$

$$Exp\ c_{14}(\omega) = c_{14}(\omega_1)p_1 + c_{14}(\omega_2)p_2 + c_{14}(\omega_3)p_3 = c_{14}$$
$$= (8.88)\left(\frac{33}{60}\right) + (8.89)\left(\frac{16.5}{60}\right) + (9)\left(\frac{10.5}{60}\right) = 8.90375$$

$$Exp\ c_{21}(\omega) = c_{21}(\omega_1)p_1 + c_{21}(\omega_2)p_2 + c_{21}(\omega_3)p_3 = c_{21}$$
$$= (9)\left(\frac{33}{60}\right) + (8.98)\left(\frac{16.5}{60}\right) + (8.96)\left(\frac{10.5}{60}\right) = 8.9875$$

$$Exp\ c_{22}(\omega) = c_{22}(\omega_1)p_1 + c_{22}(\omega_2)p_2 + c_{22}(\omega_3)p_3 = c_{22}$$
$$= (11.98)\left(\frac{33}{60}\right) + (11.96)\left(\frac{16.5}{60}\right) + (12)\left(\frac{10.5}{60}\right) = 11.978$$

$$Exp\ c_{23}(\omega) = c_{23}(\omega_1)p_1 + c_{23}(\omega_2)p_2 + c_{23}(\omega_3)p_3 = c_{23}$$
$$= (13)\left(\frac{33}{60}\right) + (12.98)\left(\frac{16.5}{60}\right) + (12.96)\left(\frac{10.5}{60}\right) = 12.9875$$

$$Exp\ c_{24}(\omega) = c_{24}(\omega_1)p_1 + c_{24}(\omega_2)p_2 + c_{24}(\omega_3)p_3 = c_{24}$$
$$= (6.98)\left(\frac{33}{60}\right) + (6.96)\left(\frac{16.5}{60}\right) + (7)\left(\frac{10.5}{60}\right) = 6.978$$

$$Exp\ c_{31}(\omega) = c_{31}(\omega_1)p_1 + c_{31}(\omega_2)p_2 + c_{31}(\omega_3)p_3 = c_{31}$$
$$= (13.96)\left(\frac{33}{60}\right) + (13.98)\left(\frac{16.5}{60}\right) + (14)\left(\frac{10.5}{60}\right) = 13.9725$$

$$Exp\ c_{32}(\omega) = c_{32}(\omega_1)p_1 + c_{32}(\omega_2)p_2 + c_{32}(\omega_3)p_3 = c_{32}$$
$$= (9)\left(\frac{33}{60}\right) + (8.98)\left(\frac{16.5}{60}\right) + (8.96)\left(\frac{10.5}{60}\right) = 8.9875$$

$$Exp\ c_{33}(\omega) = c_{33}(\omega_1)p_1 + c_{33}(\omega_2)p_2 + c_{33}(\omega_3)p_3 = c_{33}$$
$$= (15.96)\left(\frac{33}{60}\right) + (15.98)\left(\frac{16.5}{60}\right) + (16)\left(\frac{10.5}{60}\right) = 15.9725$$

$$Exp\ c_{34}(\omega) = c_{34}(\omega_1)p_1 + c_{34}(\omega_2)p_2 + c_{34}(\omega_3)p_3 = c_{34}$$
$$= (4.99)\left(\frac{33}{60}\right) + (5)\left(\frac{16.5}{60}\right) + (4.98)\left(\frac{10.5}{60}\right) = 4.991$$

5.2.7 Station 7: Form Problem as DTLPP Standard Version: DTLPP is obtained for *case* (7,8,8) as following table:

Table 5.2-1: The Obtain DTLPP Data Distribution Table of an Electricity STLPPFI Real-Life Problem for Case (7,8,8)

Power Plants	Cites				Supply Million Kw/h
	HR	DK	SI	HA	
G_1	7.98625	5.99375	9.989	8.90375	36
G_2	8.9875	11.978	12.9875	6.978	51
G_3	13.9725	8.9875	15.9725	4.991	42
Demand Million Kw/h	46	22	31	30	Total = 129 M Kw/h Balanced

5.2.8 Station 8: Finding BFS for DTLPP via VAM Method: VAM will be applied to find BFS for the above DTLPP via the difference of both least minimum cost in each row and column of matrix c_{ij} as penalty values, then enter maximum penalty in row or column, then select minimum cost value cell for this maximum penalty value in row or column reduce the same value for other costs that is not enter, repeat this till the end to get BFS (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017; Karagul, Kenan;Sahin, Yusuf, 2020) as follows:

Table 5.2-2: The DLPP Solution Steps via VAM to Get BFS for Case (7,8,8)

7.98625	5.99375&10	9.989 &26	8.90375	36//26//0	1.9925//2.00275
8.9875 &46	11.978	12.9875&5	6.978	51//5//0	2.0095//2.9905 //4
13.9725	8.9875&12	15.9725	4.991 &30	42//12//0	3.9965//4.985//0
46//0	22//10 //0	31//5//0	30//0	129	----
1.00125//0	2.99375//5.98425 //0	2.9985	1.987//0	-----	Penalty values

Therefore, BFS is obtained based on (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017) as follows:

Table 5.2-3: The BFS Table of An Electricity STLPPFI Real-Life Problem for Case (7,8,8)

0	10	26	0
46	0	5	0
0	12	0	30

5.2.9 Station 9: Finding Optimal Solution for DTLPP via MODI Method: The current solution is optimal, so the total transporting cost is as follows:

$$(8.9875 * 46) + (5.99375 * 10) + (8.9875 * 12)$$
$$+(9.989 * 26) + (12.9875 * 5) + (4.991 * 30)$$
$$= 1055.594\ IQD$$

Now, MODI is not necessary to be used to find the optimal solution since immediately BFS is optimal solution based on (Reeb, James Edmund;Leavengood, Scott A, 2002; Winston, Wayne L;Goldberg, Jeffrey B, 2004; Sharma, 1974; Sengamalaselvi, 2017).

Now, via using MATLAB code program to solve the above example this process is explained in MATLAB as follows:

- Run STLPPFI problem solver MATLAB program in Editor window of MATLAB Application.
- In command window: input fuzzy information on probability distribution via three vectors b_i, d_i and α_i as credibility degree of the DM about information on probability distribution space, vagueness level vector and alpha-cut level vector respectively, then input an acceptance vector via vector p from RN matrix in each row select one value where length of vector p is 3/4/5, then input cs matrix $(mm * nn, 3)/(mm * nn, 4)/(mm * nn, 5)$ $2D$ dimensions' matrix as estimates cost values where matrix cs is (m, n) estimates distribution matrix with $(1, k)$ deterministic acceptance vector face, then input mm value and nn value respectively as length of row and column of deterministic cost values of matrix cd will be in final where should $(mm * nn)/(kk)$ i.e., $(mm * nn)$ divide over (kk) if it is not divided then the problem does not have solution, then finally input availability constraints vector avb and input requirement constraints vector req where should be total availability satisfy total requirement.
- In command window: The solution will be shown optimal solution that contains the best BFS with minimum total cost transporting electricity objective function, with selection post optimal solution value.

Command Window

input vector b as a credibility degree of the DM about information on probability distribution space, input vector d as a vagueness level, input vector a as alpha-cut level respectively

b=[1/2 1/5 1/10];d=[1/6 1/6 1/6];a=[1/2 1/2 1/2];

The problem has solution as follows: The Alpha-Cut Technique probability intervals of each pi for all i in each row of (n,2) matrix p after applying Fuzzy Transformation of Probability Distribution Space via Alpha-Cut Technique

p=

0.4167 0.5833
0.1167 0.2833
0.0167 0.1833

The Truth Degrees process is applied as follows: First, Alpha-Cut technique probability intervals of each pi divides to eleven equals distance points in each row of following Degrees of Truth of fuzzy logical value

l=

0.4167 0.4333 0.4500 0.4667 0.4833 0.5000 0.5167 0.5333 0.5500 0.5667 0.5833

0.1167 0.1333 0.1500 0.1667 0.1833 0.2000 0.2167 0.2333 0.2500 0.2667 0.2833

0.0167 0.0333 0.0500 0.0667 0.0833 0.1000 0.1167 0.1333 0.1500 0.1667 0.1833

Second, converting Truth Degrees to both TrFN and TpFN in each row as follow: fuzzy Truth Degrees region FN(k,1:24)=[TrFN1, TpFN2, TrFN3, TpFN4, TrFN5, TpFN6, TrFN7] Where TrFN1(1st 3 elements), TpFN2(2nd 4 elements), TrFN3(3rd 3 elements), TpFN4(4th 4 elements), TrFN5(5th 3 elements), TpFN6(6th 4 elements), TrFN7(7th 3 elements). Where fuzzy vector forms contain TrFN as (a, alpha, betta), and for TpFN as (a-lower, a-upper, alpha, betta)

FN=

Columns 1 through 12

0.4333 0.4167 0.4500 0.4500 0.4833 0.4333 0.5000 0.4833 0.4500 0.5000 0.4833 0.5167

0.1333 0.1167 0.1500 0.1500 0.1833 0.1333 0.2000 0.1833 0.1500 0.2000 0.1833 0.2167

0.0333 0.0167 0.0500 0.0500 0.0833 0.0333 0.1000 0.0833 0.0500 0.1000 0.0833 0.1167

Columns 13 through 24

0.4667 0.5333 0.5167 0.5000 0.5500 0.5167 0.5500 0.5000 0.5667 0.5667 0.5500 0.5833

0.1667 0.2333 0.2167 0.2000 0.2500 0.2167 0.2500 0.2000 0.2667 0.2667 0.2500 0.2833

0.0667 0.1333 0.1167 0.1000 0.1500 0.1167 0.1500 0.1000 0.1667 0.1667 0.1500 0.1833

The deterministic vector values or nine importance power points in each fuzzy truth degrees regions after applying LFRF to deffuzzifier fuzzy for all column values for all i as each row of following deterministic values RN matrix

RN=

0.4167 0.4417 0.4833 0.4958 0.5167 0.5292 0.5500 0.5750 0.5833

0.1167 0.1417 0.1833 0.1958 0.2167 0.2292 0.2500 0.2750 0.2833

0.0167 0.0417 0.0833 0.0958 0.1167 0.1292 0.1500 0.1750 0.1833

from deterministic vector p in each row, we take p1 p2 p3 ... pn vectors then we test 9^n cases via sum(pi)=1 for all pi>0

p123(:,:,1)=

0.5500 0.5750 0.6167 0.6292 0.6500 0.6625 0.6833 0.7083 0.7167
0.5750 0.6000 0.6417 0.6542 0.6750 0.6875 0.7083 0.7333 0.7417
0.6167 0.6417 0.6833 0.6958 0.7167 0.7292 0.7500 0.7750 0.7833
0.6292 0.6542 0.6958 0.7083 0.7292 0.7417 0.7625 0.7875 0.7958
0.6500 0.6750 0.7167 0.7292 0.7500 0.7625 0.7833 0.8083 0.8167
0.6625 0.6875 0.7292 0.7417 0.7625 0.7750 0.7958 0.8208 0.8292
0.6833 0.7083 0.7500 0.7625 0.7833 0.7958 0.8167 0.8417 0.8500
0.7083 0.7333 0.7750 0.7875 0.8083 0.8208 0.8417 0.8667 0.8750
0.7167 0.7417 0.7833 0.7958 0.8167 0.8292 0.8500 0.8750 0.8833

p123(:,:,2)=

0.5750 0.6000 0.6417 0.6542 0.6750 0.6875 0.7083 0.7333 0.7417
0.6000 0.6250 0.6667 0.6792 0.7000 0.7125 0.7333 0.7583 0.7667
0.6417 0.6667 0.7083 0.7208 0.7417 0.7542 0.7750 0.8000 0.8083

0.6542 0.6792 0.7208 0.7333 0.7542 0.7667 0.7875 0.8125 0.8208
0.6750 0.7000 0.7417 0.7542 0.7750 0.7875 0.8083 0.8333 0.8417
0.6875 0.7125 0.7542 0.7667 0.7875 0.8000 0.8208 0.8458 0.8542
0.7083 0.7333 0.7750 0.7875 0.8083 0.8208 0.8417 0.8667 0.8750
0.7333 0.7583 0.8000 0.8125 0.8333 0.8458 0.8667 0.8917 0.9000
0.7417 0.7667 0.8083 0.8208 0.8417 0.8542 0.8750 0.9000 0.9083

p123(:,:,3)=
0.6167 0.6417 0.6833 0.6958 0.7167 0.7292 0.7500 0.7750 0.7833
0.6417 0.6667 0.7083 0.7208 0.7417 0.7542 0.7750 0.8000 0.8083
0.6833 0.7083 0.7500 0.7625 0.7833 0.7958 0.8167 0.8417 0.8500
0.6958 0.7208 0.7625 0.7750 0.7958 0.8083 0.8292 0.8542 0.8625
0.7167 0.7417 0.7833 0.7958 0.8167 0.8292 0.8500 0.8750 0.8833
0.7292 0.7542 0.7958 0.8083 0.8292 0.8417 0.8625 0.8875 0.8958
0.7500 0.7750 0.8167 0.8292 0.8500 0.8625 0.8833 0.9083 0.9167
0.7750 0.8000 0.8417 0.8542 0.8750 0.8875 0.9083 0.9333 0.9417
0.7833 0.8083 0.8500 0.8625 0.8833 0.8958 0.9167 0.9417 0.9500

p123(:,:,4)=
0.6292 0.6542 0.6958 0.7083 0.7292 0.7417 0.7625 0.7875 0.7958
0.6542 0.6792 0.7208 0.7333 0.7542 0.7667 0.7875 0.8125 0.8208
0.6958 0.7208 0.7625 0.7750 0.7958 0.8083 0.8292 0.8542 0.8625
0.7083 0.7333 0.7750 0.7875 0.8083 0.8208 0.8417 0.8667 0.8750
0.7292 0.7542 0.7958 0.8083 0.8292 0.8417 0.8625 0.8875 0.8958
0.7417 0.7667 0.8083 0.8208 0.8417 0.8542 0.8750 0.9000 0.9083
0.7625 0.7875 0.8292 0.8417 0.8625 0.8750 0.8958 0.9208 0.9292
0.7875 0.8125 0.8542 0.8667 0.8875 0.9000 0.9208 0.9458 0.9542
0.7958 0.8208 0.8625 0.8750 0.8958 0.9083 0.9292 0.9542 0.9625

p123(:,:,5)=
0.6500 0.6750 0.7167 0.7292 0.7500 0.7625 0.7833 0.8083 0.8167
0.6750 0.7000 0.7417 0.7542 0.7750 0.7875 0.8083 0.8333 0.8417
0.7167 0.7417 0.7833 0.7958 0.8167 0.8292 0.8500 0.8750 0.8833
0.7292 0.7542 0.7958 0.8083 0.8292 0.8417 0.8625 0.8875 0.8958
0.7500 0.7750 0.8167 0.8292 0.8500 0.8625 0.8833 0.9083 0.9167
0.7625 0.7875 0.8292 0.8417 0.8625 0.8750 0.8958 0.9208 0.9292
0.7833 0.8083 0.8500 0.8625 0.8833 0.8958 0.9167 0.9417 0.9500
0.8083 0.8333 0.8750 0.8875 0.9083 0.9208 0.9417 0.9667 0.9750
0.8167 0.8417 0.8833 0.8958 0.9167 0.9292 0.9500 0.9750 0.9833

p123(:,:,6)=
0.6625 0.6875 0.7292 0.7417 0.7625 0.7750 0.7958 0.8208 0.8292
0.6875 0.7125 0.7542 0.7667 0.7875 0.8000 0.8208 0.8458 0.8542
0.7292 0.7542 0.7958 0.8083 0.8292 0.8417 0.8625 0.8875 0.8958
0.7417 0.7667 0.8083 0.8208 0.8417 0.8542 0.8750 0.9000 0.9083
0.7625 0.7875 0.8292 0.8417 0.8625 0.8750 0.8958 0.9208 0.9292
0.7750 0.8000 0.8417 0.8542 0.8750 0.8875 0.9083 0.9333 0.9417
0.7958 0.8208 0.8625 0.8750 0.8958 0.9083 0.9292 0.9542 0.9625
0.8208 0.8458 0.8875 0.9000 0.9208 0.9333 0.9542 0.9792 0.9875
0.8292 0.8542 0.8958 0.9083 0.9292 0.9417 0.9625 0.9875 0.9958

p123(:,:,7)=
0.6833 0.7083 0.7500 0.7625 0.7833 0.7958 0.8167 0.8417 0.8500
0.7083 0.7333 0.7750 0.7875 0.8083 0.8208 0.8417 0.8667 0.8750
0.7500 0.7750 0.8167 0.8292 0.8500 0.8625 0.8833 0.9083 0.9167
0.7625 0.7875 0.8292 0.8417 0.8625 0.8750 0.8958 0.9208 0.9292
0.7833 0.8083 0.8500 0.8625 0.8833 0.8958 0.9167 0.9417 0.9500
0.7958 0.8208 0.8625 0.8750 0.8958 0.9083 0.9292 0.9542 0.9625
0.8167 0.8417 0.8833 0.8958 0.9167 0.9292 0.9500 0.9750 0.9833
0.8417 0.8667 0.9083 0.9208 0.9417 0.9542 0.9750 1.0000 1.0083
0.8500 0.8750 0.9167 0.9292 0.9500 0.9625 0.9833 1.0083 1.0167

p123(:,:,8)=
0.7083 0.7333 0.7750 0.7875 0.8083 0.8208 0.8417 0.8667 0.8750
0.7333 0.7583 0.8000 0.8125 0.8333 0.8458 0.8667 0.8917 0.9000
0.7750 0.8000 0.8417 0.8542 0.8750 0.8875 0.9083 0.9333 0.9417
0.7875 0.8125 0.8542 0.8667 0.8875 0.9000 0.9208 0.9458 0.9542
0.8083 0.8333 0.8750 0.8875 0.9083 0.9208 0.9417 0.9667 0.9750
0.8208 0.8458 0.8875 0.9000 0.9208 0.9333 0.9542 0.9792 0.9875
0.8417 0.8667 0.9083 0.9208 0.9417 0.9542 0.9750 1.0000 1.0083
0.8667 0.8917 0.9333 0.9458 0.9667 0.9792 1.0000 1.0250 1.0333
0.8750 0.9000 0.9417 0.9542 0.9750 0.9875 1.0083 1.0333 1.0417

p123(:9) =
0.7167 0.7417 0.7833 0.7958 0.8167 0.8292 0.8500 0.8750 0.8833

0.7417 0.7667 0.8083 0.8208 0.8417 0.8542 0.8750 0.9000 0.9083
0.7833 0.8083 0.8500 0.8625 0.8833 0.8958 0.9167 0.9417 0.9500
0.7958 0.8208 0.8625 0.8750 0.8958 0.9083 0.9292 0.9542 0.9625
0.8167 0.8417 0.8833 0.8958 0.9167 0.9292 0.9500 0.9750 0.9833
0.8292 0.8542 0.8958 0.9083 0.9292 0.9417 0.9625 0.9875 0.9958
0.8500 0.8750 0.9167 0.9292 0.9500 0.9625 0.9833 1.0083 1.0167
0.8750 0.9000 0.9417 0.9542 0.9750 0.9875 1.0083 1.0333 1.0417
0.8833 0.9083 0.9500 0.9625 0.9833 0.9958 1.0167 1.0417 1.0500

Now, we need to find and select each case equal one after sum(pi)=1 for all pi>0 then we collect pi as acceptance vector 9×9×9 logical array

pp123(:,:,7)= pp123(:,:,8)=
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

rowpp123= colpp123= volpp123=
8 8 7 62 70 71 1 1 1

input vector p as deterministic acceptance vector from RN matrix in each row give one value where length of vector p is 3/4/5
p=[0.55,0.275,0.175]

input cs matrix (mm*nn,3)/(mm*nn,4)/(mm*nn,5) 2-D dimensions matrix as estimates cost values where matrix cs is (m,n) distribution matrix with (1,k) deterministic acceptance vector face.

cs=[7.980,7.990,8.000;6.000,5.990,5.980;9.990,9.980,10.00;8.880,8.890,9.000;9.000,8.980,8.960;11.98,11.96,12.00;13.00,12.98,12.96;6.980,6.960,7.000;13.96,13.98,14.00;9.000,8.980,8.960;15.96,15.98,16.00;4.990,5.000,4.980]

please input mm and nn as length of obtain row and column of deterministic value matrix must be in final respectively. Be attention should (mm*nn)/(kk) since if (mm*nn) does not divide over (kk) then we do not have solution.

mm=4, nn=3, we convert cs (m*n,k) 2D matrix to css (m,k,n) 3D matrix

css(:,:,1)= css(:,:,2)=
7.9800 7.9900 8.0000 9.0000 8.9800 8.9600
6.0000 5.9900 5.9800 11.980 11.960 12.000
9.9900 9.9800 10.000 13.000 12.980 12.960
8.8800 8.8900 9.0000 6.9800 6.9600 7.0000

css(:,:,3)=
13.960 13.980 14.000
9.0000 8.9800 8.9600
15.960 15.980 16.000
4.9900 5.0000 4.9800

The deterministic value of cd matrix well be
cd=
7.98625 5.99375 9.98900 8.90375
8.98750 11.9780 12.9875 6.97800
13.9725 8.98750 15.9725 4.99100

Now, we have deterministic cd matrix, please input availability vector avb and input requirement vector req where should be total availability satisfy total requirement.
avb=[36 51 42], req=[46 22 31 30]

The solution where uses Vogel method
Optimization Problem: Solve for: x
minimize:7.98625*x(1,1)+8.9875*x(2,1)+13.9725*x(3,1)+5.99375*x(1,2)+11.978*x(2,2)+8.9875*x(3,2)+9.989*x(1,3)+12.9875*x(2,3)+15.9725*x(3,3)+8.90375*x(1,4)+6.978*x(2,4)+4.991*x(3,4)
subject to:
x(1,1)+x(1,2)+x(1,3)+x(1,4)==36;x(2,1)+x(2,2)+x(2,3)+x(2,4)==51;x(3,1)+x(3,2)+x(3,3)+x(3,4)==42;x(1,1)+x(2,1)+x(3,1)==46;x(1,2)+x(2,2)+x(3,2)==22;x(1,3)+x(2,3)+x(3,3)==31;x(1,4)+x(2,4)+x(3,4)==30;
variable bounds:
0<=x(1,1);0<=x(2,1);0<=x(3,1);0<=x(1,2);0<=x(2,2);0<=x(3,2);0<=x(1,3);0<=x(2,3);0<=x(3,3);0<=x(1,4);0<=x(2,4);0<=x(3,4);
Optimal solution found:
BFS=
0 10 26 0
46 0 5 0
0 12 0 30
Zval ObjCost Optimal= 1055.594

5.2.10 Station 10: Selecting Post Optimal Solution for DTLPP via Deciding from DM: By repeating steps from station 6 tell getting optimal solution for $p = (p_{18}, p_{27}, p_{38}) = \left(\frac{34.5}{60}, \frac{15}{60}, \frac{10.5}{60}\right)$, then optimal solution is obtained, where the total transporting cost is as follows:

$$(8.988 * 46) + (5.994 * 10) + (8.988 * 12) + (9.9893 * 26) + (12.988 * 5) + (4.9908 * 30) = 1055.627\ IQD$$

Also, by repeating steps from station 6 tell getting optimal solution for $p = (p_{18}, p_{28}, p_{37}) = \left(\frac{34.5}{60}, \frac{16.5}{60}, \frac{9}{60}\right)$, then optimal solution is obtained, where the total transporting cost is as follows:

$$(8.9885 * 46) + (5.9943 * 10) + (8.9885 * 12) + (9.9887 * 26) + (12.9885 * 5) + (4.9912 * 30) = 1055.663\ IQD$$

Finally, we have the optimal solution among three cases which are $cases$ (7,8,8), where the total transporting cost is $1055.594\ IQD$, where analytic optimal solution for the above problem via crisp values version is $1057\ IQD$.This solution shows efficiency of STLPPFI algorithm via closing numerical solution from analytic/exact solution.

Now, to selecting post optimal solution for a certain amount among obtained solutions will be as follows:

1. If importance rank of probability problem is p_2 then p_3 then p_1 from $p = (p_1, p_2, p_3) = \in \mathbb{R}^3$, then DM should select $case$ (7,8,8) as a post optimal solution for certain problem.
2. If importance rank of probability problem is p_1 then p_3 then p_2 from $p = (p_1, p_2, p_3) = \in \mathbb{R}^3$, then DM should select $case$ (8,7,8) as a post optimal solution for certain problem with little additional penalty cost which is $0.033\ IQD$.
3. If importance rank of probability problem is p_2 then p_1 then p_3 from $p = (p_1, p_2, p_3) = \in \mathbb{R}^3$, then DM should select $case$ (8,8,7) as a post optimal solution for certain problem with little additional penalty cost which is $0.069\ IQD$.

Note that comparative comments on both methods Dual-Simplex and VAM is that the Dual-Simplex Method immediately gives the optimal solution without using MODI method after finding IBFS with more elapsed time and more iterations and it is a more difficult way in great problems, where VAM sometimes does not give optimal solution without using MODI method with less elapsed time and less iterations and it is easy way in great problems.

CONCLUSION

In this study, the MATLAB code program with its algorithm program of solving STLPPFI model problems are proposed. The proposed method to solve the STLPPFI model problems was based on several stages of solution process. The study considered STLPPFI models. The method utilized concepts such as LFMF, $TpFN$, $TrFN$, LFRF, Alpha-Cut technique on probability distribution, Truth Degrees technique on probability distribution, EWS technique and analyzing cases via second condition test of alpha-cut technique. The study showed that STLPPFI model problem solver is more efficient and has so close solution to analytic/exact solution. The STLPPFI problem solver is used to convert STLPPFI into its corresponding equivalent DTLPP via defuzzifying the probability distribution and derandomization randomness of problem formulation respectively. The solution of numerical example in electricity field was shown to find the optimal solution of it and selecting post optimality solution for it. The result is evaluated and supports the entire stated technique approaches, algorithms, and it supports MATLAB code program of STLPPFI problem solver. The study showed that the theory was paralell with the algorithm and its MATLAB program. The MATLAB code programs of both transformations with followed technique approaches as Alpha-Cut technique, Truth Degrees technique, $TpFN$, $TrFN$, LFMF, LFRF, EWS technique and analyzing/filtering cases test were simplified process of finding the optimal solution to STLPPFI, and it issued the emerge of the entire actual real-life problem situations. The study illustrates that the proposed method implemented via MATLAB programming is practical, easy and applicable to practical applications in the fields of energy and industry. Moreover, the proposed method takes solution of the problem at the lowest cost, least time running, maximum transporting amounts and maximum profits.

REFERENCES

A. Edward Samuel;M. Venkatachalapathy. (2011). Modified Vogel’s Approximation Method for Fuzzy Transportation Problems. *Applied Mathematical Sciences, 5*(28), 1367 - 1372.

Abdelaziz, F Ben;Masri, Hatem. (2005). Stochastic programming with fuzzy linear partial information on probability distribution. *European Journal of Operational Research, 162*(3), 619-629.

Abdelaziz, F. B. (2012). Solution approaches for the multiobjective stochastic programming. *European Journal of Operational Research, 216*(1), 1-16.

Abdelaziz, Fouad Ben;Aouni, Belaid;El Fayedh, Rimeh. (2007). Multi-objective stochastic programming for portfolio selection. *European Journal of Operational Research, 177*(3), 1811-1823.

Abdelaziz, Fouad Ben;Masri, Hatem. (2009). Multistage stochastic programming with fuzzy probability distribution. *Fuzzy Sets and Systems, 160*(22), 3239-3249.

Abdelaziz, Fouad Ben;Masri, Hatem. (2010). A compromise solution for the multiobjective stochastic linear programming under partial uncertainty. *European Journal of Operational Research, 202*(1), 55-59.

Ameen, A. O. (2015). *Improved Two-phase Solution Strategy for Multiobjective Fuzzy Stochastic Linear Programming Problems with Uncertain Probability Distribution* (1st ed.). Malaysia: Universiti Teknologi Malaysia.

Appati, Justice Kwame;Gogovi, Gideon Kwadwo;Fosu, Gabriel Obed. (2015). MATLAB Implementation of Vogel's Approximation and the Modified Distribution Methods. *Compusoft, 4*(1), 1449.

Ben Abdelaziz, F;Masmoudi, Meryem. (2012). A multiobjective stochastic program for hospital bed planning. *Journal of the Operational Research Society, 63*(4), 530-538.

Dharani, K;Selvi, D. (2018). Solving intuitionistic fuzzy transportationproblem with ranking method using matlab code. *Applied Scienceand Computations, 5*(12), 20-26.

Government, K. R. (2020). *The Masterplan of Generation, Distribution and Controling Electricity Power with Existing and Proposed (400&132) kV System in Kurdistan Region.* Erbil - Iraq: Media center - KRG Ministry of Electricity.

Guo, Haiying;Wang, Xiaosheng;Zhou, Shaoling. (2015). A transportation problem with uncertain costs and random supplies. *International Journal of e-Navigation and Maritime Economy, 2*(1), 1-11.

Hamadameen, Abdulqader Othman;Hassan, Nasruddin. (2018). Pareto optimal solution for multiobjective stochastic linear programming problems with partial uncertainty. *International Journal of Mathematics in Operational Research, 12*(2), 139-166.

Hamadameen, Abdulqader Othman;Zainuddin, Zaitul Marlizawati. (2015). A reciprocated result using an approach of multiobjective stochastic linear programming models with partial uncertainty. *International Journal of Mathematics in Operational Research, 7*(4), 395-414.

Karagul, Kenan;Sahin, Yusuf. (2020). A novel approximation method to obtain initial basic feasible solution of transportation problem. *Journal of King Saud University-Engineering Sciences, 32*(3), 211-218.

Mahdavi-Amiri, N;Nasseri, SH. (2006). Duality in fuzzy number linear programming by use of a certain linear ranking function. *Applied mathematics and computation, 180*(1), 206-216.

Mahdavi-Amiri;NezamNasseri;Seyed Hadi. (2007). Duality results and a dual simplex method for linear programming problems with trapezoidal fuzzy variables. *Fuzzy sets and systems, 158*(17), 1961-1978.

Reeb, James Edmund;Leavengood, Scott A. (2002). *Transportation problem: a special case for linear programming problems* (1st ed.). Washington: Oregon State University .

Sakawa, M. (1993). Fundamentals of fuzzy set theory. *Fuzzy sets and interactive multiobjective optimization*, 7-35.

Sakawa, M. (1993). Interactive multiobjective linear programming with fuzzy parameters. *Fuzzy sets and interactive multiobjective optimization*, 149-173.

Sengamalaselvi, J. (2017). Solving transportation problem by using Matlab. *International Journal of Engineering Sciences & Research Technology, 6*(1), 374-381.

Sharma, S. (1974). *Operations Research for Hons. & Post-graduate Students* (1st ed.). Kedar Nath Ram Nath.

TV, K. (2021, 10 28). *Kurdistan24 TV Production.* Retrieved 10 30, 2021, from Kurdistan24.NET: https://www.youtube.com/watch?v=7xVykfknVPI

Winston, Wayne L;Goldberg, Jeffrey B. (2004). *Operations research: applications and algorithms* (3rd ed.). Thomson Brooks/Cole Belmont.

6. APPENDEX

Technique approach algorithm program outlines with its MATLAB code programs of them:

6.1.1 The Alpha-Cut Technique Algorithm Program Outline

The program outline of Alpha-cut technique will be as follows:

Input: Fuzzy Probability Polyhedral Set $\widetilde{p}$
Step 1: For $k = 1,2, \ldots N$, Do Steps 2-6
Step 2: $\widetilde{p} \in \widetilde{\pi}$, then define $\widetilde{p}$ as a $F(R)$
Step 3: Fuzzifier $\widetilde{p}$ in F(R) via Triangle or Trapezoid Linear Fuzzy Membership Function $\mu_k(p)$
Step 4: Use Alpha-cut technique formula

$$b - d(1 - \alpha) \le p \le b + d(1 - \alpha)$$

Step 5: Defuzzifier p which p is stochastic now i.e., $p \in \pi$
Step 6: $k = k + 1$
Step 7: If $(k \ne N)$, then (Return to Step 1)
Step 8: If $(k = N)$, then (End of Process)
Output: Stochastic Probability Polyhedral Set p ■.

6.1.2 The Alpha-Cut Technique MATLAB Code Program

The MATLAB Program of the alpha-cut technique is as following:

```
format short;disp('input vector b as credibility degree of DM about information on probability distribution space'),b=input ('b=');disp('input vector d as vagueness level'),d=input('d='); disp(' input vector a as alpha-cut level'), a=input('a=');b=b';d=d';a=a';n=length(a);p=zeros(n,2);
if length(b)==length(d) && length(b)==length(a) && length(d)==length(a)
   disp('The solution will be as following')
   for k=1:1:n
p(k,[1,2])=[(b(k))-((d(k))*((1-a(k)))),(b(k))+((d(k))*((1-a(k))))];
   end
else
disp('There is not have solution since all vectors are you inputted are not in same dimension')
end
if length(b)==length(d) && length(b)==length(a) && length(d)==length(a)
disp('The credibility degree vector b is'), b=b';b,disp('The vagueness levels vector d is'),
d=d';d,disp('The alpha-cut levels vector a is'),
a=a';a,disp('The probability intervals of each pi for all i as each row of following probability interval matrix p i.e., applying Fuzzy Transformation of Probability Distribution Space via Alpha-Cut Technique'),p
else
   disp('please input all vectors in same dimension')
end
```

6.1.3 The Ranking Function Technique Program Outline

The program outline of linear fuzzy ranking function LFRF technique approach will be as follows:
Input: Fuzzy Numbers $\widetilde{a} \in F(R)$
Step 1: For $k = 1,2, \ldots N$, Do Steps 2-7
Step 2: Define Fuzzy $\widetilde{a} \in F(R)$
Step 3: Fuzzifier $\widetilde{a}$ in F(R) via Parametric Form then Ranking Function $R(F)(\widetilde{a})$ respectively
Step 4: Use Parametric Form for Trapezoidal $\widetilde{a}$

$$[\widetilde{a}]\lambda = [\alpha(\lambda - 1) + a^L,\ a^U + (1 - \lambda)\beta]$$

Or Use Parametric Form for Triangular $\widetilde{a}$

$$[\widetilde{a}]\lambda = [\alpha(\lambda - 1) + a,\ a + (1 - \lambda)\beta]$$

Step 5: Use LFRF Technique for Trapezoidal $\widetilde{a}$

$$R(F)(\widetilde{a}) = R(F)(a^L, a^U, \alpha, \beta) = \frac{a^L + a^U}{2} + \frac{\beta - \alpha}{4}$$

Or Use LFRF Technique for Triangular $\widetilde{a}$

$$R(F)(\widetilde{a}) = R(F)(a, \alpha, \beta) = a + \frac{\beta - \alpha}{4}$$

Step 6: Deffuzzifier a which is a is deterministic now i.e., $a \in \mathbb{R}$
Step 7: $k = k + 1$
Step 8: If $k \ne N$, then (Return to Step 1)
Step 9: If $k = N$, then (End of the Process)
Output: Real values of a ■.

6.1.4 The Ranking Function Technique MATLAB Program

The MATLAB Program of Linear Fuzzy Ranking Function technique is as following:

```
format short;disp('input n * 3 or n * 4 matrix (FN) as a vectors of TpFN or TrFN in each row where each row of n * 3 matrix (FN) contains (a, alpha, betta) as a vector of TrFN or each row of n * 4 matrix (FN) contains (aL, aU, alpha, betta) as a vector of TpFN'),FN=input ('FN=');b=size(FN);n=b(1);RN=zeros(n,1);
if b(2)==3
   disp('it is TrFN')
   for k=1:1:n
   RN(k,1)=(FN(k,1))+((FN(k,3)-FN(k,2))/4);
   end
elseif b(2)==4
   disp('it is TpFN')
   for k=1:1:n
   RN(k,1)=((FN(k,1)+FN(k,2))/2)+((FN(k,4)-FN(k,3))/4);
   end
else
   disp('There is not have solution since FN is not TpFN nor TrFN')
end
if b(2)==3 || b(2)==4
   disp('The fuzzy number values before converting will be as follows vector/matrix'),FN,disp('The deterministic real values after converting from fuzzy numbers will be as follows vector'),RN
else
   disp('please input RN vector/matrix as a TpFN or TrFN, or correct your input matrix into n * 3 or n * 4 matrix (FN) as a vector of TpFN or TrFN')
end
```

6.1.5 The Truth Degrees Technique Program Outline:

The program outline of truth degrees technique will be as follows:

Input: Stochastic Probability Polyhedral Set p
Step 1: For $k = 1,2, \ldots N$, Do Steps 2-8
Step 2: Define each $p \in \pi$ as a continuous interval $[\alpha, \beta]$
Step 3: Divide $[\alpha, \beta]$ of each $p \in \pi$ as ten continuous subintervals $[\alpha, \beta] = [\alpha, a_1] \ldots [b_1, \beta]$
Step 4: Replace p at each ten subintervals as fuzzy truth degrees regions membership functions $\mu_k(p_i)$
Step 5: p is fuzzy number i.e., $p \in \widetilde{A}$
Step 6: Deffuzzifier $\widetilde{p}$ via linear fuzzy ranking function technique formula to get stochastic truth degrees regions from fuzzy truth degrees regions i.e., $p \in A$
Step 7: Deffuzzifier p i.e., $p \in \mathbb{R}$ or p is deterministic discrete value now
Step 8: $k = k + 1$
Step 9: If $k \ne N$, then (Return to Step 1)
Step 10: If $k = N$, then (End of Process)
Output: Real values of p ■.

6.1.6 The Truth Degrees Technique MATLAB Program

The MATLAB code program of the truth degrees technique is as follows:

```
format short;disp('input (n,2) matrix p as Alpha-Cut Technique probability intervals of each pi for all i in each row of matrix p after applying Fuzzy Transformation of Probability Distribution Space via Alpha-Cut Technique'),p=input('p=');b=size(p);n=b(1);l=zeros(n,11);FN=z
```

```
eros(n,24);RN=zeros(n,9);TrFN1=0;TpFN2=0;TrFN3=0;TpFN4=0;TrFN5=0;TpFN6=0;TrFN7=0;RN1=0;RN2=0;RN3=0;RN4=0;RN5=0;RN6=0;RN7=0;
if b(2)==2
   disp('The solution will be as following')
   for k=1:1:n
l(k,1:11)=linspace(p(k,1),p(k,2),11);TrFN1(k,1:3)=l(k,[2,1,3]);TpFN2(k,1:4)=l(k,[3,5,2,6]);TrFN3(k,1:3)=l(k,[5,3,6]);TpFN4(k,1:4)=l(k,[5,7,4,8]);TrFN5(k,1:3)=l(k,[7,6,9]);TpFN6(k,1:4)=l(k,[7,9,6,10]);TrFN7(k,1:3)=l(k,[10,9,11]);FN(k,1:24)=[TrFN1(k,:),TpFN2(k,:),TrFN3(k,:),TpFN4(k,:),TrFN5(k,:),TpFN6(k,:),TrFN7(k,:)];RN1(k,1)=(FN(k,1))+((FN(k,3)-FN(k,2))/4);RN2(k,1)= ((FN(k,4)+FN(k,5))/2)+((FN(k,7)-FN(k,6))/4); RN3(k,1)=(FN(k,8))+((FN(k,10)-FN(k,9))/4);RN4(k,1)=((FN(k,11)+FN(k,12))/2)+((FN(k,14)-FN(k,13))/4);RN5(k,1)=(FN(k,15))+((FN(k,17)-FN(k,16))/4);RN6(k,1)=((FN(k,18)+FN(k,19))/2)+((FN(k,21)-FN(k,20))/4);RN7(k,1)=(FN(k,22)) +((FN(k,24)-FN(k,23))/4);RN(k,1:9)=[p(k,1), RN1(k,:),RN2(k,:),RN3(k,:),RN4(k,:),RN5(k,:),RN6(k,:),RN7(k,:),p(k,2)];
   end
else
disp('There is not have solution since p is not as a (n,2) dimension matrix')
end
if b(2)==2
disp('The Alpha-Cut Technique probability intervals of each pi for all i in each row of matrix p'),p,disp('The Truth Degrees Process as follows applies: First, The Alpha-Cut Technique Probability Intervals of each pi for all i divides to 11 points in each row of following Degrees of Truth of fuzzy logical value'),l, disp('Second, Converting Truth Degrees to both TrFN and TpFN in each row as following fuzzy Truth Degrees region FN(k,1:24)=[TrFN1,TpFN2, TrFN3,TpFN4,TrFN5,TpFN6,TrFN7] Where TrFN1(1st 3 elements),TpFN2(2nd 4 elements), TrFN3(3rd 3 elements), TpFN4(4th 4 elements), TrFN5(5th 3 elements), TpFN6(6th 4 elements), TrFN7(7th 3 elements). Where fuzzy vector forms for TrFN is (a, alpha, betta),for TpFN is (a-lower, a-upper, alpha, betta)'),FN,disp('The deterministic vector values or importance power points in each fuzzy Truth Degrees regions after applying linear fuzzy ranking function LFRF to deffuzzifier fuzzy for all column values for all i as each row of the following deterministic values matrix rf'),RN
else
disp('please input p is a (n,2) dimension matrix')
end
```

6.1.7 The Analyzing Cases Test Program Outline

The program outline of Analyzing Cases Test will be as follows:

Input: Deterministic matrix RN values or importance efficient points that divides as each row of matrix RN are deterministic vector values Probability Polyhedral Set p.

Step 1: Define $RN = [P_1; P_2; P_3]$, and each $p_k \in P_1 \subseteq \mathbb{R}^N, p_m \in P_2 \subseteq \mathbb{R}^N, p_n \in P_3 \subseteq \mathbb{R}^N$; $\forall k, m, n = 1,2, \dots N$.

Step 2: For $k = 1,2, \dots N$, for $m = 1,2, \dots N$, for $n = 1,2, \dots N$, Do Steps 3-8.

Step 3: Do $p_{kmn} = p_k + p_m + p_n$, $\forall m, n, k = 1,2, \dots N$ of entire location (m, n, k) of $3D$ matrix P.

Step 4: Replace p_{kmn} at place (m, n, k) of $3D$ matrix P.

Step 5: Create comparative (m, n, k) $3D$ matrix PP as logical answer of question is $p_{kmn} = 1$ or not, if yes take logical value 1 and if no take logical value 0.

Step 6: Find entire locations of logical matrix PP that have value 1, to getting and finding location of p_{kmn} of $3D$ matrix P, then separate location p_{kmn} to p_k, p_m and p_n.

Step 7: Create entire acceptance vectors $P1, P2, \dots Pi$, $i \in \mathbb{N}$, where each $Pi = (p_{ki}, p_{mi}, p_{ni}) \subseteq \mathbb{R}^{\mathbb{N}}$.

Step 8: $k = k + 1$ if loop k applies, or $m = m + 1$ if loop m applies, or $n = n + 1$ if loop n applies.

Step 9: If $k \neq N$ and $m \neq N$ and $n \neq N$, then (Return to Step 2).

Step 10: If $k = N$ and $m = N$ and $n = N$, then (End of Process).

Output: Real values of acceptance vectors $Pi = (p_{ki}, p_{mi}, p_{ni}) \subseteq \mathbb{R}^{\mathbb{N}}$ ■.

6.1.8 The Analyzing Cases Test MATLAB Program

The MATLAB code program of analyzing cases test is as follows:

```
format short;disp('input (3,n) matrix RN as 3 obtained deterministic vectors p in each row take p1 p2 p3 ... pn as n-dimensional vectors'),RN=input('RN=');kkk=size(RN);
if kkk(1)==3
disp('from deterministic vector p in each row take p1 p2 p3 ... pn vectors then we test 9^n cases via sum(pi)=1 for all pi>0'),p1=RN(1,:);np1=length(p1);p2=RN(2,:);np2=length(p2);p3=RN(3,:);np3=length(p3);p123=zeros(np3,np1,np2);pp123=zeros(np3,np1,np2);
  if length(p1)==length(p2)&&length(p1)==length(p3)&&length(p2)==length(p3)
    for k=1:1:np3
      for m=1:1:np1
        for n=1:1:np2
     p123(m,n,k)=p1(k)+p2(m)+p3(n);p123;
        end
      end
    end
  else
    disp('There is not have solution')
  end
p123,disp('we find and select each case equal one after sum(pi)=1 for all pi>0 then we collect pi as acceptance vector'),pp123=logical (p123==1),[rowpp123,colpp123,volpp123]=find(pp123);rowpp123=rowpp123';colpp123=colpp123';volpp123=volpp123';rowpp123,colpp123,volpp123
else
  disp('There is not have solution since rows of RN more than 3 or less than 3')
end
```

6.1.9 The EWS Technique Program Outline

The program outline of expected weighted summation EWS technique will be as follows:

Input: Stochastic objective function coefficients equation

Step 1: For $k = 1,2, \dots N$, Do Steps 2-5

Step 2: Define stochastic objective coefficients

$$Min\, z(x, \omega) = Min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}(\omega)\, x_{ij}$$

Step 3: Fuzzifier stochastic objective coefficients via the EWS approach $\forall P = (p_1, p_2, \dots, p_N)^T$

$$Min\, Exp_{p\in\pi}\, z(x, \omega) = Min\, Exp_{p\in\pi} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}(\omega)\, x_{ij}$$

$$= Min \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \left(Exp_{p\in\pi} \sum_{k=1}^{N} c_{ij}(\omega_k)\, p_k \right);\ \forall p \in \pi$$

Step 4: Deffuzzifier stochastic objective coefficients to deterministic

$$Min\, z(x) = Min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Step 5: $k = k + 1$

Step 6: If $k \neq N$, then (Return to Step 1)

Step 7: If $k = N$, then (End of Process)

Output: Deterministic objective function coefficients equation ■.

6.1.10 The EWS Technique MATLAB Program

The MATLAB Program of the expected weighted summation EWS technique is as following:

```
format short;disp('input (1,3)/(1,4)/(1,5) vector p as acceptance probability distribution vector'),p=input('p=');disp(' input cs matrix (mm*nn,3)/(mm*nn,4)/(mm*nn,5) 2D matrix as estimate cost values where cs matrix contains (m,n) distribution matrix with (1,k) acceptance weighted vector face'),cs=input('cs='); disp('please input mm and nn as what is length of row and column of deterministic value matrix will be in final respectively Be attention should (mm*nn)/(kk) i.e., (mm*nn) divided over (kk) if not divided not have solution'),mm=input ('mm=');nn=input('nn=');kk=size(cs,2);
if rem(mm*nn,kk)==0
disp('we convert cs matrix (m*n,k) 2D matrix to css (m,k,n) 3D matrix')
if length(p)==3
css=cat(3,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)));
elseif length(p)==4
css=cat(4,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)),cs((3*(mm))+1:4*(mm),1:length(p)));
elseif length(p)==5
css=cat(5,cs(1:mm,1:length(p)),cs((mm)+1:2*(mm),1:length(p)),cs((2*(mm))+1:3*(mm),1:length(p)),cs((3*(mm))+1:4*(mm),1:length(p)),cs((4*(mm))+1:5*(mm),1:length(p)));
else
   disp('There is not have solution')
end
m=size(css,1);k=size(css,2);n=size(css,3);cd=zeros(m,k,n);css
if length(p)==k
   disp('The solution will be as following')
   cd=pagetranspose(pagemtimes(css,p'));
else
disp('There is not have solution')
end
else
disp('There is not have solution')
end
if length(p)==k && length(p)==3
disp('The deterministic value matrix cd is'), cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3)];cd
elseif length(p)==k && length(p)==4
disp('The deterministic value matrix cd is'),
cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3);cd(:,:,4)];cd
elseif length(p)==k && length(p)==5
disp('The deterministic value matrix cd is'),
cd=[cd(:,:,1);cd(:,:,2);cd(:,:,3);cd(:,:,4);cd(:,:,5)];cd
else
disp('please input (1,k) vector p that have same 3rd dimension of cs (m,n,k) 3D matrix')
end
```