Neutrosophic Sets and Systems, Vol. 49, 2022
University of New Mexico

Neutrosophic Sociogram Approach to Neutrosophic Cognitive Maps in Swift Language

R.Priya[1], Nivetha Martin[2]

[1]Department of Mathematics,PKN Arts and Science College,Madurai,India.Email: iampriyaravi@gmail.com

[2]Department of Mathematics,Arul Anandar College(Autonomous), Karumathur,India.

Email: nivetha.martin710@gmail.com

Abstract: Neutrosophic Cognitive Maps (NCM) decision-making system encompasses sequential tasks of resolving the confrontations in finding the ideal solution. This paper introduces neutrosophic sociogram (NS) approach as an alternative to the conventional approach of NCM in finding the fixed point of the dynamical NCM system. A comparative analysis is made between two approaches and the efficiency of the proposed approach is validated with an application to find the interrelationship between the persuading eleven factors of Technopreneurship. It was observed that the results obtained using the proposed approach is more compatible, feasible and simple than the conventional approach in making decisions on the implications between the factors. The two approaches are programmed using SWIFT language to make the computations easier and the results are in consensus with the manual calculations.

Keywords: FCM; Neutrosophic sociogram; neutrosophic cognitive maps; Technopreneurship; SWIFT;

1. Introduction

Sociogram tools are significant in exploring the relationship between the members of the group. Jacob Levy Moreno made the first attempt in developing the sociogram techniques to investigate more on the interrelationship between the members and the factors persuading it. The extension of this classical sociogram to fuzzy sociogram was initiated to resolve the uncertainties prevailing in the social relationship. On profound study, it was found that the relationship involves indeterminacies inaddition to uncertainties and Gustavo et al [1] proposed the approach of the neutrosophic sociogram in group analysis. It was observed that neutrosophic approach has yielded more accurate results in enhancing the group dynamics. These sociogram approaches are predominantly applied to find the interrelationship between the members (m_i, i = 1 to n) of the group say G, where G = {m_1,m_2,..m_n}. The information on the interrelationship between the members is obtained using a questionnaire. In the neutrosophic sociogram (NS) approach, the most influential member is found and addressed as the leader of the group and the chances of strengthening the relationship between the members are also determined. The modalities of these approaches are in coherence with Fuzzy Cognitive Maps and Neutrosophic Cognitive Maps. Kosko [2] introduced Fuzzy Cognitive maps (FCM) by incorporating the elements of fuzzy to the theory of Cognitive maps. FCM has extensive applications in various fields. To mention a few, Jason et al [3]developed FCM model to draw inferences on the impacts of learner's comprehension. Senniappan et al [4]proposed an FCM model to categorize concrete forms. Chrispen [5] applied FCM models to study the social aspects of livelihood. Abdollah [6] presented a review of different FCM models used in medicine.FCM decision-making models were constructed for making optimal decisions on agriculture by Makrinos et al [7] . Papageorgiou et al[8] developed a model for making decisions on environmental aspects and cotton yield. Song et al [9] Katarzyna et al [10] extended FCM models to make predictions, Antonie et al [11]made an attempt to extend FCM for further explorations. Papageorgiou et al [12] discussed the various methods and algorithms of FCM models. Felix et al [13] described the various software used for the computational purpose of FCM models.

FCM is a directed graph with nodes and arcs representing the decisive factors of the problems and their relationship respectively. The edge weights belong to {-1,0,1} states the influencing nature of the relationship between the factors. The value 1 denotes the positive kind of influence, -1 represents the negative kind of influence and the value 0 symbolizes null influence. The connection matrix of order n×n is derived from the FCM graphical representation of n factors. A general representation of FCM has nodes of the form $C_1, C_2...Cn$ and the instantaneous vector V of the form V= $(V_1,V_2,..V_n)$, where V_i takes the value 0 or 1 signifying ON or OFF position of the factors. For instance a decision-making problem on finding the factors causing gestational diabetics is considered with five factors say C1,C2,C3,C4,C5, here V = $(V_1\ V_2\ V_3\ V_4\ V_5)$ where vi takes the value 0 or 1 signifying the ON or OFF position of the factors if V= (1 0 0 0 0) then it signifies that the first factor is in ON position and other factors are in OFF position. FCM with a directed cycle formed by the edges has feedback and it becomes a dynamical system. The passing of this vector into the respective connection matrix M of order 5 × 5 results in another vector and the threshold function is used to update the values by assigning 1 to the values of the vector greater than 1, -1 to the values of the vector lesser than -1. The equilibrium state of the dynamical system is called the hidden pattern and it occurs when the vector with any of the factors kept in ON position is repeated after successive passing and threshold of the vectors. If the equilibrium state is a unique vector then it is called the fixed point and if it settles in the form of V1 →V2 →…V5 →V1 then it is termed as a limit cycle. Here Vi denotes the ON position of the i^{th} factor respectively and the OFF position of the remaining other factors. These are the underlying concepts of FCM involved in making optimal decisions.

Neutrosophic Cognitive maps (NCM) also involves the same aspects with the inclusion of indeterminacy. The edge weight set of NCM is {-1,0,*I*,1} and the elements of the instantaneous vector assume the values 0,1, *I*. NCM was first developed by Smarandache and Vasantha Kandasamy [14].NCM models are extended to combined overlap models and neutrosophic relational maps models for decision-making. Gaurav et al [15] used genetic algorithm in NCM models. NCM has extensive applications as FCM models in diagnosis [16], medicine[17], situational analysis[18,19], pandemic causative factors [20], decision-making [21-25],impact of imaginative play on children[26], religious impacts[27].Banerjee et al [28] compared FCM and NCM models and suggested NCM models be more compatible.NCM decision-making models are also developed to make optimal decisions on various dimensions of society, science and technology. On profound analysis, FCM and NCM model approaches appear to be similar to the approaches of Fuzzy and Neutrosophic Sociogram. FCM & NCM models determine the most influential factors and their interrelational impacts in which the latter considers indeterminacy, whereas Fuzzy and neutrosophic sociogram approaches determine the most influential person of the group and the extent of relational compatibility between the members of the group. The approach of Fuzzy sociogram was used in developing a new genre of FCM model by Jegan et al [29] to study the emotional intelligence of the students. FCM models with and without fuzzy sociogram approach were compared and the ranking of the factors in both the cases was the same and it was suggested to incorporate fuzzy sociogram approach in FCM model development. Based on this new sociogram approach in FCM models, this paper proposes the integration of the neutrosophic sociogram approach in the NCM model. In a NCM model, the indeterminacy between the factors is considered and the relational impacts between the factors are determined. The positive, negative or indeterminate influential status between the factors can be determined but there is no space to make a prediction on the extent of resolving indeterminate relational impacts between the factors. Also the indeterminacy between the factors on subjecting to computations gets retained as indeterminate values itself at certain instances. These are some of the constraints of decision making using NCM models. To overcome this shortcoming of NCM models, the neutrosophic sociogram approach shall be used as an alternative approach to the NCM model. The proposed approach is compared with the conventional NCM model. The efficiency of the two model approaches is tested by applying to the factors influencing Technopreneurship. SWIFT language is used to write coding for the two approaches to ease the computation and to draw the results instantly. The paper is structured as follows: Section 2 presents the origin and the development of the proposed NS integrated NCM approach; section 3 consists of the application of the proposed approach; section 4 discusses the results and the last section concludes the work.

2. Origin and Development of the Proposed Approach

This section presents the origin and the development of the proposedneutrosophic sociogram alternative approach to NCM.

A neutrosophic cognitive map is a neutrosophic directed graph (a directed graph with atleast one edge of indeterminate nature) with factors as nodes and their interrelationship as edges. The edge weight assume the values belonging to the set {-1,I,0,1}, where -1 indicated the negative interrelational impacts, I denotes the indeterminacy relation between the factors, 0 signifies void interrelational impacts and 1 represents the positive interrelational impacts.

Let us consider the factors contributing to environmental catastrophe, say E1,E2,E3,E4,E5. It is assumed that the following connection matrix (say M) is determined based on the questionnaire given to the experts.

M =

	E1	E2	E3	E4	E5
E1	0	1	0	I	0
E2	1	0	1	0	0
E3	I	1	0	1	0
E4	1	1	I	0	1
E5	0	0	1	1	0

Let us consider an instantaneous vector of the form V = (1 0 0 0 0), which states that the first factor E1 is in ON position and the other factors are in OFF position. The vector V is passed onto M and the resultant vector obtained is (0 1 0 I 0) and on updating the values the vector V, the new vector V1 = (1 1 0 I 0) is obtained.

$V \rightarrow V_1$, where $\rightarrow$ denotes the threshold of values (vi, i = 1,2..5) in a vector.

The threshold function T (x) is represented as

$$\begin{cases} 1 \ if \ v_i \geq 0 \\ 0 \ otherwise \end{cases}$$

On passing the vector V1 onto M again and repeating the steps as above the final vector obtained is (11111). The final vector thus obtained signifies the influence of the first factor over the other and it shows that the factor 1 is related to all other factors. The lastly obtained vector (11111) is called the fixed point or the limit cycle.

Suppose if the final vector obtained is of the form (1 1 0 I 0), it signifies that the first factor has positive influence over the second factor, null influences on third and fifth factors and indeterminate influence on the fourth factor. If such kinds of fixed points are obtained on keeping the factors in ON position, then the holistic decision on the influences between the factors cannot be made. The indeterminate influences remains as such and there is no scope for the possibilities of alleviating such indeterminacy.

Let us consider the same with two experts

Expert-I

	E1	E2	E3	E4	E5
E1	0	1	0	I	0
E2	1	0	1	0	0
E3	I	1	0	1	0
E4	1	1	I	0	1
E5	0	0	1	1	0

Expert II

	F1	F2	F3	F4	F5
F1	0	1	1	I	0
F2	1	0	I	0	0
F3	I	1	0	0	0
F4	I	1	I	0	1
F5	0	I	1	1	0

The combined connection matrix is

	F1	F2	F3	F4	F5
F1	0	1	0.5	2I	0
F2	1	0	0.5+I	0	I
F3	2I	1	0	0.5	0
F4	0.5+I	1	2I	0	1
F5	0	2I	1	1	0

On repeating the same NCM procedure to the above matrix with the initial step of keeping the first factor E1 in ON position we obtain the same fixed point (11111). This is one of the shortcomings of NCM . To handle the limitations of NCM, the neutrosophic sociogram approach shall be used as an alternative approach to the conventional method of NCM.

Neutrosophic sociogram approach aims in determining the social dynamics of the group. It considers the members of the group and the interest of the members in working or getting along with other members are determined based on questionnaire. Sometimes the responses are deterministic in nature and sometimes they are indeterminate. The neutrosophic sociogram approach enables to arrive at a conclusion of finding the possibilities of alleviating the indeterminacy along with the numerical range of extent. It also facilitates to find the leader of the group (the most preferred person). The neutrosophic sociogram as discussed by [] presents vividly the sequential steps and mathematical formulation of the decision-making model with a hypothetical example. On intense study NCM approach can be aligned in line with NS approach as both intends to find the associational impacts.

In NCM with NS approach, the factors of the group are considered as the members and based on the expert's opinion the deterministic and the indeterminate associational impact or influence between the factors is determined. The generic tabular representation is as follows.

	E1	**E2**	**…………….**	**Em**
Factors				
F1	$F^D_{11};F^N_{11}$	$F^D_{12};F^N_{12}$		$F^D_{1m};F^N_{1m}$
F2	$F^D_{21};F^N_{21}$	$F^D_{22};F^N_{22}$		$F^D_{2m};F^N_{2m}$
.	……….	………..	…………………..	………

.	……….	………..	…………………..	……….
Fn	$F^D_{n1};F^N_{n1}$	$F^D_{n2};F^N_{n2}$		$F^D_{nm};F^N_{nm}$

$F^D ij \subset F$, where F is the set of factors (i = 1,2,..n), (j = 1,2,..m), the response represents the factors (Fi) of deterministic association in expert's (Ej) point of view where $F^N ij$ represents the indeterminate associational response from the expert. Let us apply the proposed procedure to the above example of NCM with Factors Fi, i = 1,2,..5.

Let us consider the connection matrix given by two experts

Expert-I

	F1	F2	F3	F4	F5
F1	0	1	0	I	0
F2	1	0	1	0	I
F3	I	1	0	1	0
F4	1	1	I	0	1
F5	0	I	1	1	0

Expert II

	F1	F2	F3	F4	F5
F1	0	1	1	I	0
F2	1	0	I	0	0
F3	I	1	0	0	0
F4	I	1	I	0	1
F5	0	I	1	1	0

The fuzzy amicable degree f_{ij} and the neutrosophic amicable degree n_{ij} are determined by the below equations respectively together with the consideration of weight of the experts and the evaluation matrices of the experts.

$$\frac{2}{f_{ij}} = \frac{1}{d_{ij}} + \frac{1}{d_{ji}}$$

$$\frac{2}{n_{ij}} = \frac{1}{h_{ij}} + \frac{1}{h_{ji}}$$

here d_{ij} and h_{ij} denotes the deterministic and neutrosophic associations between the factors i and respectively.

The significance of the factor with fuzzy amicable degree shall be determined by using the following index

$$S\ (F_i) = \frac{\sum_j f_{ij}}{\sum_i \sum_j f_{ij}}$$

The competency of the factor together with neutrosophic amicable degree shall be determined by using the following index

$$C\ (F_i) = \frac{\sum_j n_{ij}}{\sum_i \sum_j n_{ij}}$$

The representation of the above two connection matrices in the newly proposed NS integrated NCM approach.

	Expert I	**Expert II**
F1	**F2;F4**	**F2,F3;F4**

F2	**F1,F3;F5**	**F1;F3**
F3	**F2,F4;F1**	**F2;F1**
F4	**F1,F2,F5;F3**	**F2,F5;F1,F3**
F5	**F3,F4;F2**	**F3,F4;F2**

Fuzzy Amicable degree Matrix

	F1	F2	F3	F4	F5
F1	0	1	0	0	0
F2	1	0	0.67	0	0
F3	0	0.67	0	0	0
F4	0	0	0	0	1
F5	0	0	0	1	0

Neutrosophic Amicable Degree Matrix

	F1	F2	F3	F4	F5
F1	0	1	0.67	1	0
F2	1	0	1	0	0.67
F3	0.5	1	0	0.67	0
F4	1	0	0.67	0	1
F5	0	0.67	0	1	0

F1-F2	[1,1]	F2-F3	[0.67,1]	F3-F4	[0,0.67]	F4-F5	[1,1]
F1-F3	[0,0.67]	F2-F4	[0,0]	F3-F5	[0,0]		
F1-F4	[0,1]	F2-F5	[0,0.67]				
F1-F5	[0,0]						

The values [1,1] and [0,0] indicates the strong influence and void influence between the factors and the other range of values signifies the possibilities of increasing the association between the factors. For instance the first factor has no influence on fifth factor, strong influence on second factor and the extent of influence over the factors third and fourth factor is determined by the given range of values. [Also F1-F2 is same as F2-F1]. This is a better result than the fixed point obtained from the conventional procedure of NCM.

3. Application of NS integrated NCM approach

This section presents the validation of the proposed approach to the decision-making on the factors influencing Technopreneurship. The decision-making environment is characterized by eleven factors say F1,F2,….F11.These factors are obtained from the experts in the field of Business administration and the respective stakeholders through a questionnaire. The initial input is presented in the table below

		Expert –I	Expert-II	Expert-III	Expert-IV
F1	Individual characteristic	F2,F3,F5,F6,F8,F10; F4	F2,F3,F4,F5,F8; F6,F10	F2,F3,F5,F6,F8,F10; F4,F7	F2,F3,F4,F5,F6,F7,F8,F9; F10

	factor				
F2	Motivation factor	F1,F6,F9,F10,F11; F3.	F1,F3,F8,F9,F10,F11; F5,F6	F1,F3,F6,F8,F10; F11	F3,F4,F5,F6,F8,F9; F7,F11
F3	Situational factors	F5,F6,F7,F9; F2,F4	F4,F6,F9,F10,F11; F7,F8	F10,F11; F1,F4	F1,F2,F5,F7,F8,F9,F10; F4
F4	Exogenous factors	F1; F3,F9	F1,F2,F3,F6,F9; F8,F5	F1,F5,F11;F3,F7	F1,F2,F3,F5,F8,F9,F10,F11; F6
F5	Social Factors	F9,F10,F11; F3,F6,F7	F7,F9; F10,F11	F10,F11; F1,F9	F1,F3,F4,F8,F9,F10,F11; F2,F6
F6	Financial factor	F10,F11; F7,F9	F2,F4,F10,F11; F3,F8	F3,F9,F10,F11; F1,F8	F1,F2,F7,F8,F10,F11; F3,F9
F7	Non-Financial Assistance factor	F2,F8,F9; F6,F10	F3,F6,F10,F11; F2,F9	F1,F2,F10; F6,F11	F2,F5,F8,F10; F1,F9
F8	Entrepreneurial and business skills factor	F1,F2,F3,F9; F4.	F1,F2,F3,F4,F5,F6, F8,F9,F10,F11; F7	F1,F2,F3,F5,F6,F7 F8,F9,F10,F11; F4	F1,F2,F3,F4,F6,F7,F9,F10,F11; F5
F9	Cultural factors	F1,F2,F4; F6,F8	F1,F2,F8; F4,F7	F1,F3,F5,F6; F4,F10	F1,F2,F3,F4,F5,F7,F8; F10,F11
F10	Socioeconomic conditions factor	F3,F6,F11; F1,F9	F1,F2,F3,F11; F5,F8	F1,F4,F5,F6,F11; F3.	F1,F2,F3,F4,F5,F7,F8,F11; F6
F11	Government policies and procedures factor	F2,F10; F1,F9	F2,F3,F7,F10; F6,F8	F1,F2,F4,F6,F10; F8,F9	F2,F3,F5,F6,F7,F8,F10; F1,F4

The evaluation matrix of each experts is as follows

Expert I	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**
F1	0	1	1	0	1	1	0	1	0	1	0
F2	1	0	0	0	0	1	0	0	1	1	1
F3	0	0	0	0	1	1	1	0	1	0	0
F4	1	0	0	0	0	0	0	0	0	0	0
F5	0	0	0	0	0	0	0	0	1	1	1
F6	0	0	0	0	0	0	0	0	0	1	1

Expert II	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**
F1	0	1	1	1	1	0	0	1	0	0	0
F2	1	0	1	0	0	0	0	1	1	1	1
F3	0	0	0	1	0	1	0	0	1	1	1
F4	1	1	1	0	0	1	0	0	1	0	0
F5	0	0	0	0	0	0	1	0	1	0	0
F6	0	1	0	1	0	0	0	0	0	1	1

F7	0	1	0	0	0	0	0	1	1	0	0
F8	1	1	1	0	0	0	0	0	1	0	0
F9	1	1	0	1	0	0	0	0	0	0	0
F10	0	0	1	0	0	1	0	0	0	0	1
F11	0	1	0	0	0	0	0	0	0	1	0

F7	0	0	1	0	0	1	0	0	0	1	1
F8	1	1	1	1	1	1	0	0	1	1	1
F9	1	1	0	0	0	0	0	1	0	0	0
F10	1	1	1	0	0	0	0	0	0	0	1
F11	0	1	1	0	0	0	1	0	0	1	0

Expert III	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
F1	0	1	1	0	1	1	0	1	0	1	0
F2	1	0	1	0	0	1	0	1	0	1	0
F3	0	0	0	0	0	0	0	0	0	1	1
F4	1	0	0	0	1	0	0	0	0	0	1
F5	0	0	0	0	0	0	0	0	0	1	1
F6	0	0	1	0	0	0	0	0	1	1	1
F7	1	1	0	0	0	0	0	0	0	1	0
F8	1	1	1	0	1	1	1	0	1	1	1
F9	1	0	1	0	1	1	0	0	0	0	0
F10	1	0	0	1	1	1	0	0	0	0	1
F11	1	1	0	1	0	1	0	0	0	1	0

Expert IV	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
F1	0	1	1	1	1	1	1	1	1	0	0
F2	0	0	1	1	1	1	0	1	1	0	0
F3	1	1	0	0	1	0	1	1	1	1	0
F4	1	1	1	0	1	0	0	1	1	1	1
F5	1	0	1	1	0	0	0	1	1	1	1
F6	1	1	0	0	0	0	1	1	0	1	1
F7	0	1	0	0	1	0	0	1	0	1	0
F8	1	1	1	1	0	1	1	0	1	1	1
F9	1	1	1	1	1	0	1	1	0	0	0
F10	1	1	1	1	1	0	1	1	0	0	1
F11	0	1	1	0	1	1	1	1	0	1	0

The final fuzzy amicable degrees representing the associational impacts between the factors

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
F1	0	.86	.4	.67	.4	.38	.25	1	.4	.6	0
F2	.86	0	.38	0.33	0	0.6	0	.86	.75	.5	.67
F3	.4	.38	0	0.33	.33	.33	.33	.4	0.6	.75	.5
F4	.67	0.33	.33	0	.33	.25	0	.33	.5	.33	.33
F5	.4	0	.33	.33	0	0	.25	.33	.6	.6	.38
F6	.38	0.6	.33	.25	0	0	.25	.38	.25	.67	.67
F7	.25	0	.33	0	.25	.25	0	.5	.25	.38	.33
F8	1	.86	.4	.33	.33	.38	.5	0	.67	0.38	.38
F9	.4	.75	.6	.5	.6	.25	.25	.67	0	0	0

F10	.6	.5	.75	.33	.6	.67	.38	.38	0	0	1
F11	0	.67	.5	.33	.38	.67	.33	.38	0	1	0

The final matrix representing the neutrosophic amicable degrees

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
F1	0	.86	.67	1	.67	.67	.5	1	.4	1	0
F2	.86	0	.67	.33	.33	.67	.4	.86	.75	.6	.86
F3	.67	.67	0	1	.5	.6	.38	.67	.6	.86	.5
F4	1	.33	1	0	.38	.25	0	.67	.86	.33	.5
F5	.67	.33	.5	.38	0	0	0.25	.38	.67	.86	.4
F6	.67	.67	.6	.25	0	0	.6	.75	.6	.86	.86
F7	.5	.4	.38	0	0.25	.6	0	.6	.6	.67	.6
F8	1	.86	.67	.67	.38	.75	.6	0	.86	.6	.6
F9	.4	.75	.6	.86	.67	.6	.6	.86	0	.5	.5
F10	1	.6	.86	.33	.86	.86	.67	.6	.5	0	1
F11	0	.86	.5	.5	.4	.86	.6	.6	.5	1	0

Associational Impact between the factors

F1-F2	[0.86,0.86]	F2-F3	[.38,.67]	F3-F4	[.33,1]	F4-F5	[.33,.38]	F5-F6	[0,0]
F1-F3	[.4,.67]	F2-F4	[.33,.33]	F3-F5	[.33,.5]	F4-F6	[.25,.25]	F5-F7	[0.25,.25]
F1-F4	[.67,1]	F2-F5	[0,.33]	F3-F6	[.33,.6]	F4-F7	[0,0]	F5-F8	[.33,.38]
F1-F5	[.4,.67]	F2-F6	[.6,.67]	F3-F7	[.33,.38]	F4-F8	[.33,.67]	F5-F9	[.6,.67]
F1-F6	[.38,.67]	F2-F7	[0,.4]	F3-F8	[.4,.67]	F4-F9	[.5,.6]	F5-F10	[.6,.86]
F1-F7	[.25,.5]	F2-F8	[.86,.86]	F3-F9	[.6,.6]	F4-F10	[.33,.33]	F5-F11	[.38,.4]
F1-F8	[1,1]	F2-F9	[.75,.75]	F3-F10	[.75,.86]	F4-F11	[.33,.5]		
F1-F9	[.4,.4]	F2-F10	[.5,.6]	F3-F11	[.5,.5]				
F1-F10	[.6,1]	F2-F11	[.67,.86]						

F1-F11	[0,0]

F6-F7	[.25,.6]	F7-F8	[.5,.6]	F8-F9	[.67,.86]	F9-F10	[0,.5]	F10-F11	[1,1]
F6-F8	[.38,.75]	F7-F9	[.25,.6]	F8-F10	[.38,.6]	F9-F11	[0,.5]		
F6-F9	[.25,.6]	F7-F10	[.38,.67]	F8-F11	[.38,.6]				
F6-F10	[.67,.86]	F7-F11	[.33,.6]						
F6-F11	[.67,.86]								

To ease the computation SWIFT language is used for programming the NS integrated NCM approach and the coding is presented in Appendix. The following figures represent the input data and output data.

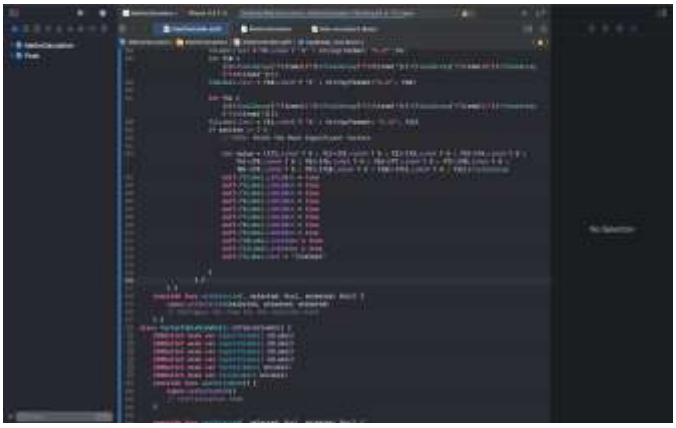


Fig 3a

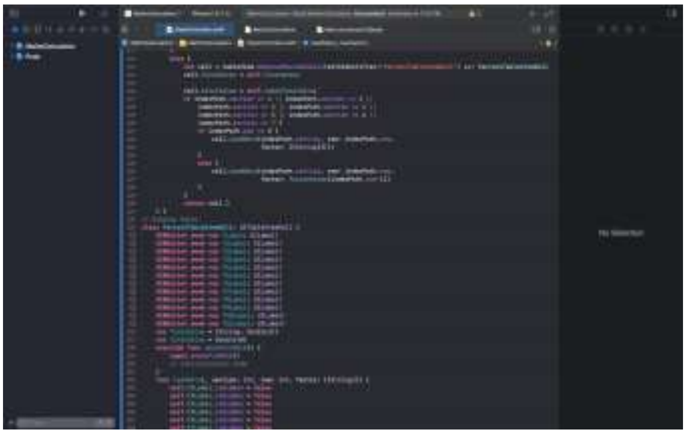


Fig 3b

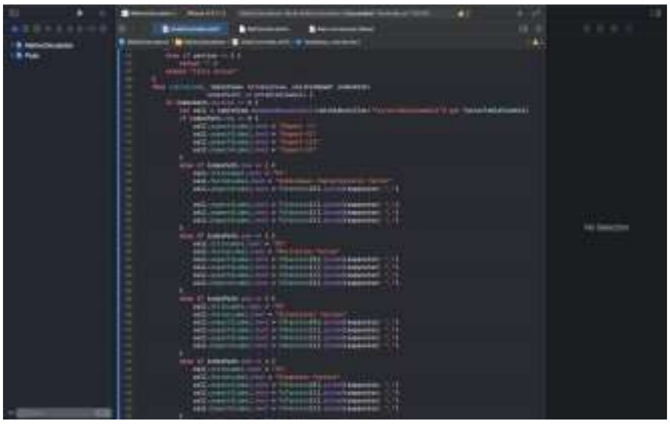


Fig 3c

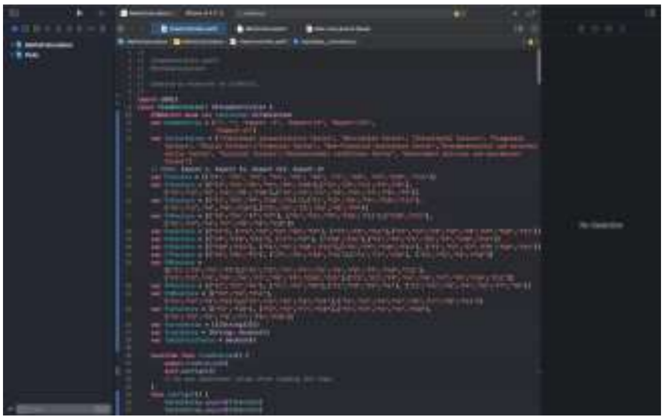
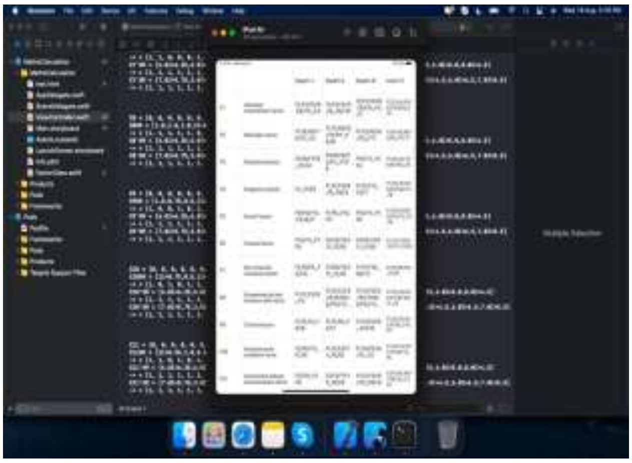


Fig 3d

Fig 3a, 3b, 3c and 3d represents the input data in SWIFT language.



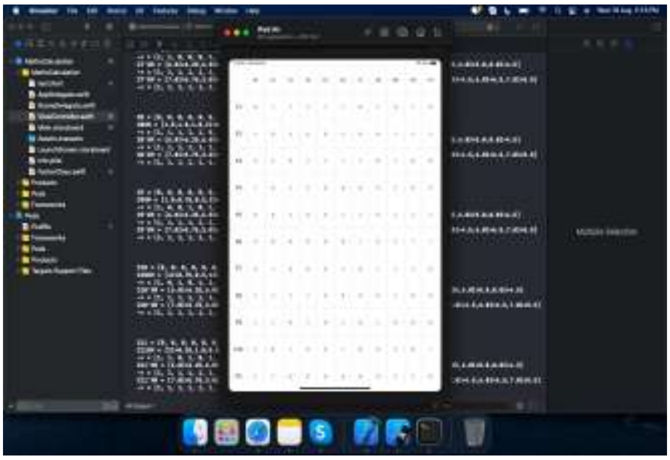
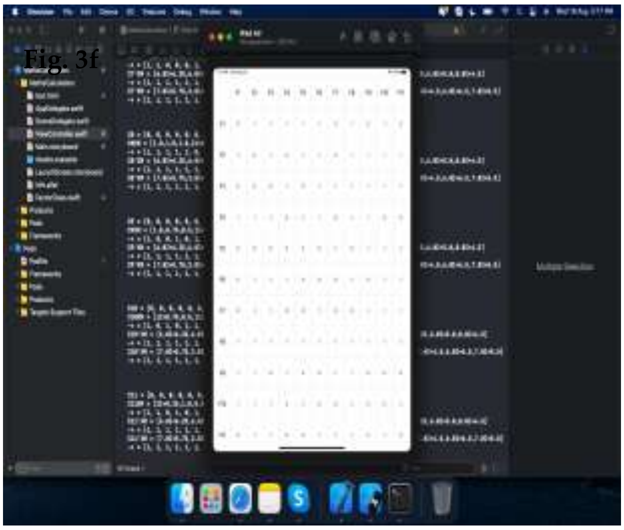


Fig. 3e **Fig. 3f**



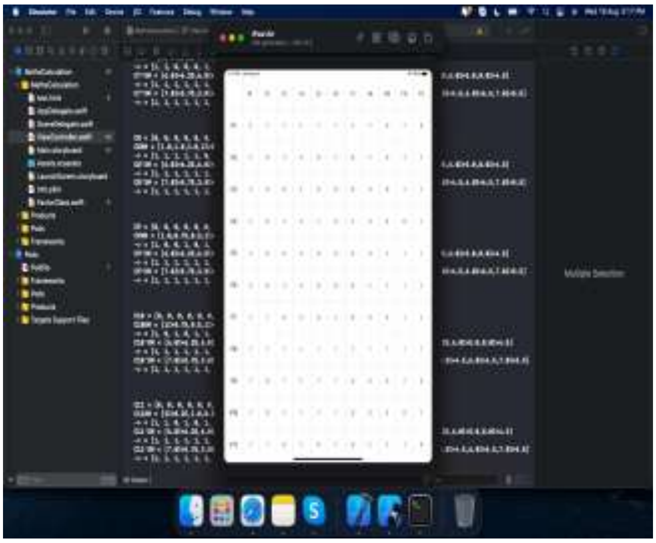


Fig. 3g **Fig. 3h**

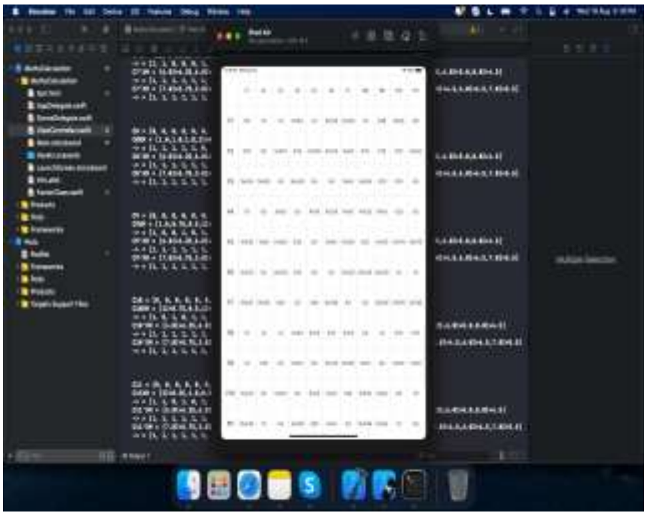


Fig 3i

Fig 3e, 3f, 3g, 3h and 3i represents the output of the associational impact values.

4. Discussion

The associational impact values between the factors clearly presents the extent of influence of one factor over the other , some of the values are deterministic in sense a real number that indicates the exact numerical value of influence and the range of values present the extent of influence between the factors and also it shows the existing possibilities of enhancing the influences between the factors for the holistic growth and development of the Technopreneurs and Technopreneurship. The results vividly shows that NS integrated NCM approach is highly accommodative in nature.

On applying the conventional NCM approach

Table 4.1. NCM Conventional approach results

On Position of the Factors	Fixed Point
(10000000000)	(11111111111)
(01000000000)	(11111111111)
(00100000000)	(11111111111)
(00010000000)	(11111111111)
(00001000000)	(11111111111)
(00000100000)	(11111111111)
(00000010000)	(11111111111)
(00000001000)	(11111111111)
(00000000100)	(11111111111)
(00000000010)	(11111111111)
(00000000001)	(11111111111)

The results obtained from conventional NCM approach (Table 4.1) has no provision for making specific analysis, the results shows that each factor influences others in a more general manner, but the actual extent of influence is not explored from the above obtained fixed points. For instance the fixed point obtained on keeping the first factor at ON position (10000000000) signifies that the first factor has impact on all the factors, but whereas the proposed approach gives the specific range of the associational impact between the factor F1 and all other factors say F2,F3,F4,F5,F6,F7,F8,F9,F10,F11. Thus the proposed approach is more efficient than the existing approach.

5. Conclusion

This paper has proposed Neutrosophic Sociogram integrated NMC approach as an alternative to the conventional NCM. The proposed approach facilitates to determine the specific associational impact between the factors rather in general manner as in the conventional method. The proposed decision-making alternative approach is highly compatible, flexible and simple in comparison with the existing conventional approach. This research work is an initiative to develop new approaches of finding the impact between the factors and this same approach shall be extended to Plithogenic Sociogram and Plithogenic cognitive maps (PCM). Just as FCM, NCM, PCM models shall be integrated with plithogenic sociogram approach.

Acknowledgement

The authors express their sincere gratitude to Dr.S.Jegan Karuppiah, Assistant Professor of Rural Development Science and Dr.P.Jerlin Rupa, Head Department of Business Administration of Arul Anandar College, Karumathur for assisting in collecting data from Technopreneurs during the pandemic period.

References

[1]. Gustavo Alvarez Gomez, Jorge Fernando GoyesGarcia, Sharon DinarzaAlvarez Gomez.,Florentin Smarandache (2020)., Neutrosophic Sociogram for Group Analysis, Neutrosophic Sets and systems {Special Issue: Impact of neutrosophy in solving the Latin American's social problems }, Vol.37.

[2]. Kosko, B. (1986), Fuzzy cognitive maps, International journal of man-machine studies, vol-24(1), pp.65-75.

[3]. Jason R.C.,Persichitte.A.K.,(2000),Fuzzy Cognitive Mapping:Applications in Education,International Journal Of Intelligent Systems,vol-15,pp-1-25.

[4]. Senniappan, V., Subramanian, J., Papageorgiou, E.I,suji mohan,(2017), Application of fuzzy cognitive maps for crack categorization in columns of reinforced concrete structures. Neural Computing & Application,vol- 28, pp.107-117.

[5]. Chrispen Murungweni., Mark T Van Wijk., Jens A Andersson., Eric MA Smaling., Ken E Giller.,(2011), Application of fuzzy cognitive mapping in livelihood vulnerability analysis, Ecology and Society,vol-16(4).

[6]. AbdollahAmirkhani,Elpiniki.I.Papegeorgiou.,AkramMohseni.,Mohammad R.Mosavi., (2017),A review of fuzzy cognitive maps in medicine: Taxonomy, methods, and applications,Computer Methods and Programs in Biomedicine, Vol-142, pp-129-145.

[7]. Makrinos, A.,Papageorgiou Elpinik Stylios., ChrysostomosGemtos T., (2007),Introducing Fuzzy Cognitive Maps for decision making in precision agriculture, Papers Presented at the 6th European Conference on Precision Agriculture, ECPA 2007,223-231.

[8]. Elpiniki I.Papageorgiou, AthanasiosMarkinos, TheofanisGemptos, (2009),Application of fuzzy cognitive maps for cotton yield management in precision farming, Expert Systems with Applications,Volume 36(10), pp- 12399-12413.

[9]. song, Hengjie.,Miao, Chunyan.,Roel, Wuyts., Shen, Zhiqi.,Catthoor, Francky, (2010).,Implementation of Fuzzy Cognitive Maps Using Fuzzy Neural Network and Application in Prediction of Time Series, IEEE Transactions on Fuzzy Systems,vol-18(2),pp.233-250.

[10]. Katarzyna Poczeta,Lukasz Kubus, Alexander Yastrebov (2019) ., Structure Optimization and Learning of Fuzzy Cognitive Map with the Use of Evolutionary Algorithm and Graph Thory, Metrics.131-147.

[11]. Antonie J.Jetter.,Kasperkok.,(2014),Fuzzy Cognitive Maps for futures studies—A methodological assessment of concepts and methods,Futures,Vol-61, pp. 45-57

[12]. Papageorgiou.E.I., Salmeron.J.L.,(2014), Methods and Algorithms for Fuzzy Cognitive Map-based Modeling,Fuzzy Cognitive Maps for Applied Sciences and Engineering,Vol-54,pp 1-28.

[13]. Felix, G., Napoles, G., Falcon, R. (2019), A review on methods and software for fuzzy cognitive maps, Artif icial intelligence Review , vol-52,pp-1707–1737 .

[14]. Smarandache, Florentin, Kandasamy, W.,(2014)., Fuzzy Cognitive Maps And Neutrosophic Cognitive Maps, Phoenix.

[15]. Gaurav, M. Kumar, K. Bhutani and S. Aggarwal.,(2015)., Hybrid model for medical diagnosis using Neutrosophic Cognitive Maps with Genetic Algorithms, 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-7.

[16]. Shadrach, F.D., Kandasamy, G. (2021),Neutrosophic Cognitive Maps (NCM) based feature selection approach for early leaf disease diagnosis. journal of Ambient Intelligence and Humanized Computingvolume 12, pages 5627–5638 .

[17]. Papageorgiou.E.I.,Spyridonos.P.P .,Stylios.C.D., Nikiforidis.G.C.,Groumpos.P.P .,(2003), Grading Urinary Bladder Tumors Using Unsupervised Hebbian Algorithm for Fuzzy Cognitive Maps,International journal of biomedical soft computing and human sciences,Vol-9,pp.33-39.

[18]. Aasim Zafar ,.Mohd Anas , (2019),Neutrosophic Cognitive Maps for Situation Analysis, Neutrosophic Sets and Systems,vol-29,pp-78-88.

[19]. Zafar, A., & Wajid, M. A. (2020). Neutrosophic cognitive maps for situation analysis. Infinite Study.

[20].Zafar, A., & Wajid, M. A. (2020). A Mathematical Model to Analyze the Role of Uncertain and Indeterminate Factors in the Spread of Pandemics like COVID-19 Using Neutrosophy: A Case Study of India (Vol. 38). Infinite Study.

[21].Wajid, M. A., & Zafar, A. (2021). Multimodal Fusion: A Review, Taxonomy, Open Challenges, Research Roadmap and Future Directions. Neutrosophic Sets and Systems, 45(1), 8.

[22].Wajid, M. A., & Zafar, A. (2021). PESTEL Analysis to Identify Key Barriers to Smart Cities Development in India. Neutrosophic Sets and Systems, 42, 39-48.

[23].Abdel-Basset, M., El-Hoseny, M., Gamal, A., & Smarandache, F. (2019). A novel model for evaluation Hospital medical care systems based on plithogenic sets. Artificial intelligence in medicine, 100, 101710.

[24].Abdel-Basset, M., Manogaran, G., Gamal, A., & Chang, V. (2019). A novel intelligent medical decision support model based on soft computing and IoT. IEEE Internet of Things Journal, 7(5), 4160-4170.

[25]. Makrinos, A.Papageorgiou, Elpinik,Stylios, Chrysostomos,Gemtos, T. (2007), Introducing Fuzzy Cognitive Maps for decision making in precision agriculture, Papers Presented at the 6th European Conference on Precision Agriculture, ECPA 2007,223-231.

[26]. Vasantha W.B. Kandasamy I. Devvrat V. Ghildiyal S. , (2019),Study of Imaginative Play in Children Using Neutrosophic Cognitive Maps Model, Neutrosophic Sets and Systems,vol-30,pp-241-252

[27]. Victor Devadoss.A, A.Rajkumar, N.Jose Parvin Praveena,(2013), A Study on Miracles through Holy Bible using Neutrosophic Cognitive Maps (NCMS), International Journal of Computer Applications ,Volume 69(3),pp-22-27.

[28]. Banerjee, Goutam,(2008)., Adaptive fuzzy cognitive maps vs neutrosophic cognitive maps: decision support tool for knowledge based institutions, 665-673, Vol-67(09).

[29]. S. Jegan Karuppiah , S.Abirami , Nivetha Martin , R.Priya, (2021),Fuzzy Sociogram Approach in Cognitive Maps to explore the factors fostering the Emotional Intelligence of the learners of present age, 2021-International Conference on Multidisciplinary approaches in Humanities, Management and Sciences,pp.57-63

Appendix

NS INTEGRATED NCM CODING INPUT

```
//
//  ViewController.swift
//  MathsCalculation
//
//  Created by Hitasoft on 14/06/21.
//

import UIKit
import PDFGenerator
class ViewController: UIViewController { @IBOutlet weak var tableView: UITableView!
var headerArray = ["", "", "Expert –I", "Expert-II", "Expert-III", "Expert-IV"]
var factorsArray = ["Individual characteristic factor", "Motivation factor", "Situational factors",
"Exogenous factors", "Social Factors", "Financial factor", "Non-Financial Assistance factor",
"Entrepreneurial and business skills factor", "Cultural factors", "Socioeconomic conditions factor",
"Government policies and procedures factor"]
// MARK: Expert I, Expert II, Expert III, Expert IV
var FFactors = [["F1", "F2", "F3", "F4", "F5", "F6", "F7", "F8", "F9", "F10", "F11"]]
var F1Factors = [["F2","F3","F5","F6","F8","F10", ";", "F4"],
["F2","F3","F4","F5","F8", ";", "F6", "F10"],
["F2","F3","F5","F6","F8","F10", ";", "F4","F7"],
["F2","F3","F4","F5","F6","F7","F8","F9",";","F10"]]
```

```
var F2Factors = [["F1","F6","F9","F10","F11",";","F3"],
["F1","F3","F8","F9","F10","F11",";","F5","F6"],
["F1","F3","F6","F8","F10",";","F11"],
["F3","F4","F5","F6","F8","F9",";","F7","F11"]]
var F3Factors = [["F5","F6","F7","F9",";","F2","F4"],
["F4","F6","F9","F10","F11",";","F7","F8"], ["F10","F11",";","F1","F4"],
["F1","F2","F5","F7","F8","F9","F10",";","F4"]]
var F4Factors = [["F1",";","F3","F9"], ["F1","F2","F3","F6","F9",";","F8","F5"],
["F1","F5","F11",";","F3","F7"], ["F1","F2","F3","F5","F8","F9","F10","F11",";", "F6"]]
var F5Factors = [["F9","F10","F11",";","F3","F6","F7"], ["F7","F9",";","F10","F11"],
["F10","F11",";","F1","F9"],
["F1","F3","F4","F8","F9","F10","F11",";" ,"F2","F6"]] var F6Factors = [["F10","F11",";","F7","F9"],
["F2","F4","F10","F11",";","F3","F8"],
["F3","F9","F10","F11",";","F1","F8"],
["F1","F2","F7","F8","F10","F11",";","F3","F9"]]
var F7Factors = [["F2","F8","F9",";","F6","F10"],
["F3","F6","F10","F11",";","F2","F9"], ["F1","F2","F10",";","F6","F11"], ["F2","F5","F8","F10",";","F1","F9"]]
var F8Factors = [["F1","F2","F3","F9",";","F4"],
["F1","F2","F3","F4","F5","F6", "F8","F9","F10","F11",";","F7"],["F1","F2","F3","F5","F6","F7",
"F8","F9","F10","F11",";","F4"], ["F1","F2","F3","F4","F6","F7","F9","F10","F11",";","F5"]]
var F9Factors = [["F1","F2","F4",";","F6","F8"], ["F1","F2","F8",";","F4","F7"],
["F1","F3","F5","F6",";","F4","F10"],
["F1","F2","F3","F4","F5","F7","F8",";", "F10","F11"]] var F10Factors = [["F3","F6","F11",";","F1","F9"],
["F1","F2","F3","F11",";","F5","F8"],
["F1","F4","F5","F6","F11",";","F3"],
["F1","F2","F3","F4","F5","F7","F8","F11",";","F6"]] var F11Factors = [["F2","F10",";","F1","F9"],
["F2","F3","F7","F10",";","F6","F8"],
["F1","F2","F4","F6","F10",";","F8","F9"],
["F2","F3","F5","F6","F7","F8","F10",";","F1","F4"]]

var factorArray = [[[String]]]() var finalArray = [String: String]() var tableTotalValue = Double(0)
override func viewDidLoad() { super.viewDidLoad() self.configUI()
// Do any additional setup after loading the view.
}
func configUI() {
```

```
factorArray.append(F1Factors) factorArray.append(F2Factors) factorArray.append(F3Factors)
factorArray.append(F4Factors) factorArray.append(F5Factors) factorArray.append(F6Factors)
factorArray.append(F7Factors) factorArray.append(F8Factors) factorArray.append(F9Factors)
factorArray.append(F10Factors) factorArray.append(F11Factors)
self.tableView.rowHeight = UITableView.automaticDimension
self.tableView.estimatedRowHeight = 45 self.tableView.sectionHeaderHeight =
UITableView.automaticDimension self.tableView.estimatedSectionHeaderHeight = 45
//    for row in 0..<11 {
//    for factor in 0..<factorArray.count {
//    // MARK: Calculate Expert I, Expert II, Expert III, Expert IV Value
//    finalArray["F\(row+1)\(factor+1)"] = (factorArray[row][0].contains("F\(factor+1)") ? 0.25 :
0)+(factorArray[row][1].contains("F\(factor+1)") ? 0.25 :
0)+(factorArray[row][2].contains("F\(factor+1)") ? 0.25 :
0)+(factorArray[row][3].contains("F\(factor+1)") ? 0.25 : 0)
//    }
//    }
for row in 0..<11 {
for factor in 0..<factorArray.count {
let factor1String = factorArray[row][0].split(separator: ";")
let factor2String = factorArray[row][1].split(separator: ";")
let factor3String = factorArray[row][2].split(separator: ";")
let factor4String = factorArray[row][3].split(separator: ";")
// MARK: Calculate Expert I, Expert II, Expert III, Expert IV Value
let FFirstVal = (((factor1String.first?.contains("F\(factor+1)") ?? false)
? 0.25 : 0) +
((factor2String.first?.contains("F\(factor+1)") ?? false)
? 0.25 : 0) +
((factor3String.first?.contains("F\(factor+1)") ?? false)
? 0.25 : 0) +
((factor4String.first?.contains("F\(factor+1)") ?? false)
? 0.25 : 0))
let FLastVal = ((factor1String.last?.contains("F\(factor+1)") ?? false) ?
1 : 0) + ((factor2String.last?.contains("F\(factor+1)") ?? false) ? 1 : 0) +
((factor3String.last?.contains("F\(factor+1)") ?? false) ?
1 : 0) +    ((factor4String.last?.contains("F\(factor+1)")
?? false) ? 1 : 0)
```

```
finalArray["F\(row+1)\(factor+1)"] = FLastVal == 0 ? "\(FFirstVal)" : "\(FLastVal)I\(FFirstVal == 0 ?
"" : "+\(FFirstVal)")"
//    print(finalArray["F\(row+1)\(factor+1)"] ?? "")
}
//    print("\n")
}
var totalRowArray = [String]() var totalColumnArray = [String]() for row in 0..<11 {
var c = [0,0,0,0,0,0,0,0,0,0,0]
c[row] = 1

var RowITotal = Double(0) var RowTotal = Double(0)
var ColumnITotal = Double(0) var ColumnTotal = Double(0)
for factor in 0..<factorArray.count {
// MARK: The fuzzy comatibility degree calculation
var rowTot = Double(0) var columnTot = Double(0)
if finalArray["F\(row+1)\(factor+1)"]?.contains("I") ?? false {
let component = finalArray
["F\(row+1)\(factor+1)"]?.components(separatedBy: "+") RowITotal = RowITotal+1
//          print("Component = \(component)") if (component?.count ?? 0) > 1 {
rowTot = (Double(component?.last ?? "0") ?? 0)
}
}
else {
//    print("Component =
\(Double(finalArray["F\(row+1)\(factor+1)"] ?? "0") ?? 0)")
rowTot = (Double(finalArray["F\(row+1)\(factor+1)"] ?? "0") ?? 0)
}

if finalArray["F\(factor+1)\(row+1)"]?.contains("I") ?? false {
let component = finalArray
["F\(factor+1)\(row+1)"]?.components(separatedBy: "+")
ColumnITotal = ColumnITotal+1 if (component?.count ?? 0) > 1 {
columnTot = (Double(component?.last ?? "0") ?? 0)
}
}
else {
```

```
columnTot = (Double(finalArray["F\(factor+1)\(row+1)"]
?? "0") ?? 0)
}
if !rowTot.isNaN {
RowTotal = RowTotal+rowTot
}
if !columnTot.isNaN {
ColumnTotal = ColumnTotal+columnTot
}
}

let totalRow = "\(RowITotal > Double(0) ? "\(RowITotal)I" : "")\(RowTotal > Double(0) ?
"+\(RowTotal)" : "")"
totalRowArray.append(totalRow)

let totalColumn = "\(ColumnITotal > Double(0) ? "\(ColumnITotal)I" : "")\(ColumnTotal >
Double(0) ? "+\(ColumnTotal)" : "")"
totalColumnArray.append(totalColumn)
}

for row in 0..<11 {
var totalRow = [Int]() var totalColumn = [Int]() var fArray = [String]() var totArray = [Int]()
print("\n\n")

var c = [0,0,0,0,0,0,0,0,0,0,0]
c[row] = 1 print("\tC\(row+1) = \(c)")
for factor in 0..<factorArray.count { if row == 0 && factor == 10 {
fArray.append("0")
totArray.append(0)
}
else {
fArray.append(finalArray["F\(row+1)\(factor+1)"] ?? "") let text =
((finalArray["F\(row+1)\(factor+1)"] ??
"").contains("I")) ? 1 : ((Double(finalArray["F\(row+1)\(factor+1)"] ?? "") ?? 0)>=1 ? 1 : 0)
let finalText = c[factor]+text totArray.append(finalText)
```

```
}
}
print(("\tC\(row+1)XM = [\(fArray.joined(separator: ","))]")) print(("\t-> = \(totArray)"))
for factor in 0..<totalRowArray.count {
let text = (totalRowArray[factor].contains("I")) ? 1 : ((Double(totalRowArray[factor]) ?? 0)>0 ? 1 : 0)
let finalText = c[factor]+text>=1 ? 1 : 0
totalRow.append(finalText)
}
print(("\tC\(row+1)'XM = [\(totalRowArray.joined(separator: ","))]"))
print(("\t-> = \(totalRow)"))
for factor in 0..<totalColumnArray.count {
let text = (totalColumnArray[factor].contains("I")) ? 1 : ((Double(totalColumnArray[factor]) ?? 0)>0 ?
1 : 0)
let finalText = (c[factor]+text)>=1 ? 1 : 0
totalColumn.append(finalText)
}
print(("\tC\(row+1)'XM = [\(totalColumnArray.joined(separator: ","))]"))
print(("\t-> = \(totalColumn)"))


}
}
func generatePDF() {
guard let v1 = self.tableView else { return }
//          v1.contentSize = CGSize(width: 100.0, height: 200.0) let dst = URL(fileURLWithPath:
NSTemporaryDirectory().appending("FinalFactorCalculation.pdf"))
// writes to Disk directly. do {
try PDFGenerator.generate(v1, to: dst)
} catch (let error) { print(error)
}
}
}
extension ViewController: UITableViewDelegate, UITableViewDataSource { func tableView(_
tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
return 12
}
```

```
func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
return self.view.frame.height/12
}
func numberOfSections(in tableView: UITableView) -> Int { return 8
}
func tableView(_ tableView: UITableView, heightForHeaderInSection section: Int) -> CGFloat {
return 45
}
func tableView(_ tableView: UITableView, titleForHeaderInSection section: Int) -> String? {
if section == 0 {
return ""
}
else if section == 1 { return "Expert - I"
}
else if section == 2 { return "Expert - II"
}
else if section == 3 { return "Expert - III"
}
else if section == 4 { return "Expert - IV"
}
else if section == 5 { return "Table 5"
}
else if section == 6 { return ""
}
return "Final Scores"
}
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
if indexPath.section == 0 {
let cell = tableView.dequeueReusableCell(withIdentifier: "FactorTableViewCell") as!
FactorTableViewCell
if indexPath.row == 0 { cell.expert1Label.text = "Expert –I" cell.expert2Label.text = "Expert-II"
cell.expert3Label.text = "Expert-III" cell.Expert4Label.text = "Expert-IV"
}
else if indexPath.row == 1 {
cell.titleLabel.text = "F1"
```

```
cell.factorLabel.text = "Individual characteristic factor" cell.expert1Label.text =
F1Factors[0].joined(separator: ",") cell.expert2Label.text = F1Factors[1].joined(separator: ",")
cell.expert3Label.text = F1Factors[2].joined(separator: ",") cell.Expert4Label.text =
F1Factors[3].joined(separator: ",")
}
else if indexPath.row == 2 { cell.titleLabel.text = "F2" cell.factorLabel.text = "Motivation factor"
cell.expert1Label.text = F2Factors[0].joined(separator: ",") cell.expert2Label.text =
F2Factors[1].joined(separator: ",") cell.expert3Label.text = F2Factors[2].joined(separator: ",")
cell.Expert4Label.text = F2Factors[3].joined(separator: ",")


}
else if indexPath.row == 3 { cell.titleLabel.text = "F3" cell.factorLabel.text = "Situational factors"
cell.expert1Label.text = F3Factors[0].joined(separator: ",") cell.expert2Label.text =
F3Factors[1].joined(separator: ",") cell.expert3Label.text = F3Factors[2].joined(separator: ",")
cell.Expert4Label.text = F3Factors[3].joined(separator: ",")


}
else if indexPath.row == 4 { cell.titleLabel.text = "F4" cell.factorLabel.text = "Exogenous factors"
cell.expert1Label.text = F4Factors[0].joined(separator: ",") cell.expert2Label.text =
F4Factors[1].joined(separator: ",") cell.expert3Label.text = F4Factors[2].joined(separator: ",")
cell.Expert4Label.text = F4Factors[3].joined(separator: ",")


}
else if indexPath.row == 5 { cell.titleLabel.text = "F5" cell.factorLabel.text = "Social Factors"
cell.expert1Label.text = F5Factors[0].joined(separator: ",") cell.expert2Label.text =
F5Factors[1].joined(separator: ",") cell.expert3Label.text = F5Factors[2].joined(separator: ",")
cell.Expert4Label.text = F5Factors[3].joined(separator: ",")


}
else if indexPath.row == 6 { cell.titleLabel.text = "F6" cell.factorLabel.text = "Financial factor"
cell.expert1Label.text = F6Factors[0].joined(separator: ",") cell.expert2Label.text =
F6Factors[1].joined(separator: ",") cell.expert3Label.text = F6Factors[2].joined(separator: ",")
cell.Expert4Label.text = F6Factors[3].joined(separator: ",")


}
else if indexPath.row == 7 { cell.titleLabel.text = "F7"
cell.factorLabel.text = "Non-Financial Assistance factor" cell.expert1Label.text =
F7Factors[0].joined(separator: ",") cell.expert2Label.text = F7Factors[1].joined(separator: ",")
cell.expert3Label.text = F7Factors[2].joined(separator: ",") cell.Expert4Label.text =
```

```
F7Factors[3].joined(separator: ",")
}
else if indexPath.row == 8 { cell.titleLabel.text = "F8"
cell.factorLabel.text = "Entrepreneurial and business skills factor"
cell.expert1Label.text = F8Factors[0].joined(separator: ",") cell.expert2Label.text =
F8Factors[1].joined(separator: ",") cell.expert3Label.text = F8Factors[2].joined(separator: ",")
cell.Expert4Label.text = F8Factors[3].joined(separator: ",")


}
else if indexPath.row == 9 { cell.titleLabel.text = "F9" cell.factorLabel.text = "Cultural factors"
cell.expert1Label.text = F9Factors[0].joined(separator: ",") cell.expert2Label.text =
F9Factors[1].joined(separator: ",") cell.expert3Label.text = F9Factors[2].joined(separator: ",")
cell.Expert4Label.text = F9Factors[3].joined(separator: ",")


}
else if indexPath.row == 10 { cell.titleLabel.text = "F10"
cell.factorLabel.text = "Socioeconomic conditions factor" cell.expert1Label.text =
F10Factors[0].joined(separator:
",")
cell.expert2Label.text = F10Factors[1].joined(separator: ",")
cell.expert3Label.text = F10Factors[2].joined(separator: ",")
cell.Expert4Label.text = F10Factors[3].joined(separator: ",")


}
else if indexPath.row == 11 { cell.titleLabel.text = "F11"
cell.factorLabel.text = "Government policies and procedures factor"
cell.expert1Label.text = F11Factors[0].joined(separator: ",")
cell.expert2Label.text = F11Factors[1].joined(separator: ",")
cell.expert3Label.text = F11Factors[2].joined(separator: ",")
cell.Expert4Label.text = F11Factors[3].joined(separator:
",")
}
return cell
}
else {
let cell = tableView.dequeueReusableCell(withIdentifier: "Factor1TableViewCell") as!
Factor1TableViewCell
```

```
//          cell.finalArray = self.finalArray cell.totalValue = self.tableTotalValue
if indexPath.section == 1 || indexPath.section == 2 || indexPath.section == 3 || indexPath.section ==
4 || indexPath.section == 5 || indexPath.section == 6 || indexPath.section == 7 {
if indexPath.row == 0 {
cell.loadData(indexPath.section, row: indexPath.row, factor: [[String]]())
}
else {
cell.loadData(indexPath.section, row: indexPath.row, factor: factorArray[indexPath.row-1])


}
}
//
return cell
}


}
}


// Display Value
class Factor1TableViewCell: UITableViewCell {


@IBOutlet weak var fLabel: UILabel! @IBOutlet weak var f1Label: UILabel! @IBOutlet weak var
f2Label: UILabel! @IBOutlet weak var f3Label: UILabel! @IBOutlet weak var f4Label: UILabel!
@IBOutlet weak var f5Label: UILabel! @IBOutlet weak var f6Label: UILabel! @IBOutlet weak var
f7Label: UILabel! @IBOutlet weak var f8Label: UILabel! @IBOutlet weak var f9Label: UILabel!
@IBOutlet weak var f10Label: UILabel! @IBOutlet weak var f11Label: UILabel! var finalArray =
[String: Double]() var totalValue = Double(0)
override func awakeFromNib() { super.awakeFromNib()
// Initialization code
}


func loadData(_ section: Int, row: Int, factor: [[String]]) { self.f2Label.isHidden = false
self.f3Label.isHidden = false self.f4Label.isHidden = false self.f5Label.isHidden = false
self.f6Label.isHidden = false self.f7Label.isHidden = false
self.f8Label.isHidden = false self.f9Label.isHidden = false self.f10Label.isHidden = false
self.f11Label.isHidden = false fLabel.text = ""
f1Label.text = "" f2Label.text = "" f3Label.text = "" f4Label.text = "" f5Label.text = "" f6Label.text = ""
```

```
f7Label.text = "" f8Label.text = "" f9Label.text = "" f10Label.text = "" f11Label.text = "" if row == 0 {
if section == 7 { fLabel.text = "Factors" f1Label.text = "Score"
self.f2Label.isHidden = true self.f3Label.isHidden = true self.f4Label.isHidden = true
self.f5Label.isHidden = true self.f6Label.isHidden = true self.f7Label.isHidden = true
self.f8Label.isHidden = true self.f9Label.isHidden = true self.f10Label.isHidden = true
self.f11Label.isHidden = true
}
else {
fLabel.text = "" f1Label.text = "F1" f2Label.text = "F2" f3Label.text = "F3" f4Label.text = "F4"
f5Label.text = "F5" f6Label.text = "F6" f7Label.text = "F7" f8Label.text = "F8" f9Label.text = "F9"
f10Label.text = "F10" f11Label.text = "F11"
}
}
else {
fLabel.text = "F\(row)" f1Label.text = "0"
f2Label.text = "0"
f3Label.text = "0"
f4Label.text = "0"
f5Label.text = "0"
f6Label.text = "0"
f7Label.text = "0"
f8Label.text = "0"
f9Label.text = "0"
f10Label.text = "0"
f11Label.text = "0"
f1Label.font = UIFont.systemFont(ofSize: 15) f2Label.font = UIFont.systemFont(ofSize: 15)
f3Label.font = UIFont.systemFont(ofSize: 15) f4Label.font = UIFont.systemFont(ofSize: 15)
f5Label.font = UIFont.systemFont(ofSize: 15) f6Label.font = UIFont.systemFont(ofSize: 15)
f7Label.font = UIFont.systemFont(ofSize: 15) f8Label.font = UIFont.systemFont(ofSize: 15)
f9Label.font = UIFont.systemFont(ofSize: 15) f10Label.font = UIFont.systemFont(ofSize: 15)
f11Label.font = UIFont.systemFont(ofSize: 15)
if section == 1 || section == 2 || section == 3 || section == 4{ if factor.count > (section-1) {
let factorString = factor[section-1].split(separator: ";")
if    factor[section-1].contains("F1") {


f1Label.text = "I"
f1Label.font = UIFont.boldSystemFont(ofSize:
```

```
15)



}     else {


}
if    factor[section-1].contains("F2") {


f2Label.text = "I"
      f2Label.font = UIFont.boldSystemFont(ofSize:    15)



}     else {


}
if    factor[section-1].contains("F3") {


f3Label.text = "I"
      f3Label.font = UIFont.boldSystemFont(ofSize:    15)



}     else {


}
if    factor[section-1].contains("F4") {


f4Label.text = "I"
      f4Label.font = UIFont.boldSystemFont(ofSize:    15)
      else {


}



if factorString.last?.contains("F1") ?? false {
```

```
}

f1Label.text = "1"

if factorString.last?.contains("F2") ?? false {

}

f2Label.text = "1"

if factorString.last?.contains("F3") ?? false {

}

f3Label.text = "1"

if factorString.last?.contains("F4") ?? false {

}

f4Label.text = "1"
}
if factor[section-1].contains("F5") {
if factorString.last?.contains("F5") ?? false { f5Label.text = "I"
```

```
f5Label.font = UIFont.boldSystemFont(ofSize: 15)
}
else {
f5Label.text = "1"
}
}
if factor[section-1].contains("F6") {
if factorString.last?.contains("F6") ?? false { f6Label.text = "I"
f6Label.font = UIFont.boldSystemFont(ofSize: 15)
}
else {
f6Label.text = "1"
}
}
if factor[section-1].contains("F7") {
if factorString.last?.contains("F7") ?? false { f7Label.text = "I"
f7Label.font = UIFont.boldSystemFont(ofSize: 15)
}
else {
f7Label.text = "1"
}
}
if factor[section-1].contains("F8") {
if factorString.last?.contains("F8") ?? false { f8Label.text = "I"
f8Label.font = UIFont.boldSystemFont(ofSize: 15)
}
else {
f8Label.text = "1"
}
}
if factor[section-1].contains("F9") {
if factorString.last?.contains("F9") ?? false { f9Label.text = "I"
f9Label.font = UIFont.boldSystemFont(ofSize: 15)
}
else {
```

```
f9Label.text = "1"
}
}
if factor[section-1].contains("F10") {
if factorString.last?.contains("F10") ?? false { f10Label.text = "I"
f10Label.font = UIFont.boldSystemFont(ofSize:
15)
}
else {
f10Label.text = "1"
}
}
if factor[section-1].contains("F11") {
if factorString.last?.contains("F11") ?? false { f11Label.text = "I"
f11Label.font = UIFont.boldSystemFont(ofSize:
15)
}
else {
f11Label.text = "1"
}
}
}
}
else if section == 5 || section == 6 || section == 7 { let factor1String = factor[0].split(separator: ";") let
factor2String = factor[1].split(separator: ";") let factor3String = factor[2].split(separator: ";") let
factor4String = factor[3].split(separator: ";")
let F1FirstVal = "\(((factor1String.first?.contains("F1")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F1") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F1") ?? false) ?
0.25 : 0) +       ((factor4String.first?.contains("F1") ?? false) ? 0.25 : 0))"
let F1LastVal = ((factor1String.last?.contains("F1") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F1") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F1") ?? false) ? 1 :
0) +((factor4String.last?.contains("F1") ?? false) ? 1 : 0)
self.f1Label.text = F1LastVal == 0 ? F1FirstVal : "\(F1LastVal)I+\(F1FirstVal)"
```

```
let F2FirstVal = "\(((factor1String.first?.contains("F2")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F2") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F2") ?? false) ?
0.25 : 0) +       ((factor4String.first?.contains("F2") ?? false) ? 0.25 : 0))"
let F2LastVal = ((factor1String.last?.contains("F2") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F2") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F2") ?? false) ? 1 :
0) +((factor4String.last?.contains("F2") ?? false) ? 1 : 0)
self.f2Label.text = F2LastVal == 0 ? F2FirstVal : "\(F2LastVal)I+\(F2FirstVal)"
let F3FirstVal = "\(((factor1String.first?.contains("F3")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F3") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F3") ?? false) ?
0.25 : 0) +       ((factor4String.first?.contains("F3") ?? false) ? 0.25 : 0))"
let F3LastVal = ((factor1String.last?.contains("F3") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F3") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F3") ?? false) ? 1 :
0) +((factor4String.last?.contains("F3") ?? false) ? 1 : 0)
self.f3Label.text = F3LastVal == 0 ? F3FirstVal : "\(F3LastVal)I+\(F3FirstVal)"
let F4FirstVal = "\(((factor1String.first?.contains("F4")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F4") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F4") ?? false) ?
0.25 : 0) +       ((factor4String.first?.contains("F4") ?? false) ? 0.25 : 0))"
let F4LastVal = ((factor1String.last?.contains("F4") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F4") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F4") ?? false) ? 1 :
0) +((factor4String.last?.contains("F4") ?? false) ? 1 : 0)
self.f4Label.text = F4LastVal == 0 ? F4FirstVal : "\(F4LastVal)I+\(F4FirstVal)"

let F5FirstVal = "\(((factor1String.first?.contains("F5")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F5") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F5") ?? false) ?
0.25 : 0) +       ((factor4String.first?.contains("F5") ?? false) ? 0.25 : 0))"
let F5LastVal = ((factor1String.last?.contains("F5") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F5") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F5") ?? false) ? 1 :
0) +((factor4String.last?.contains("F5") ?? false) ? 1 : 0)
self.f5Label.text = F5LastVal == 0 ? F5FirstVal : "\(F5LastVal)I+\(F5FirstVal)"
```

```
let F6FirstVal = "\(((factor1String.first?.contains("F6")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F6") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F6") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F6") ?? false) ? 0.25 : 0))"
let F6LastVal = ((factor1String.last?.contains("F6") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F6") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F6") ?? false) ? 1 :
0) +((factor4String.last?.contains("F6") ?? false) ? 1 : 0)
self.f6Label.text = F6LastVal == 0 ? F6FirstVal : "\(F6LastVal)I+\(F6FirstVal)"
let F7FirstVal = "\(((factor1String.first?.contains("F7")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F7") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F7") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F7") ?? false) ? 0.25 : 0))"
let F7LastVal = ((factor1String.last?.contains("F7") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F7") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F7") ?? false) ? 1 :
0) +((factor4String.last?.contains("F7") ?? false) ? 1 : 0)
self.f7Label.text = F7LastVal == 0 ? F7FirstVal : "\(F7LastVal)I+\(F7FirstVal)"
let F8FirstVal = "\(((factor1String.first?.contains("F8")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F8") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F8") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F8") ?? false) ? 0.25 : 0))"
let F8LastVal = ((factor1String.last?.contains("F8") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F8") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F8") ?? false) ? 1 :
0) +((factor4String.last?.contains("F8") ?? false) ? 1 : 0)
self.f8Label.text = F8LastVal == 0 ? F8FirstVal : "\(F8LastVal)I+\(F8FirstVal)"
let F9FirstVal = "\(((factor1String.first?.contains("F9")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F9") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F9") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F9") ?? false) ? 0.25 : 0))"
let F9LastVal = ((factor1String.last?.contains("F9") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F9") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F9") ?? false) ? 1 :
0) +((factor4String.last?.contains("F9") ?? false) ? 1 : 0)
self.f9Label.text = F9LastVal == 0 ? F9FirstVal : "\(F9LastVal)I+\(F9FirstVal)"
```

```
let F10FirstVal = "\(((factor1String.first?.contains("F10")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F10") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F10") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F10") ?? false) ? 0.25 : 0))"
let F10LastVal = ((factor1String.last?.contains("F10") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F10") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F10") ?? false) ? 1
: 0) +     ((factor4String.last?.contains("F10")
?? false) ? 1 : 0)
self.f10Label.text = F10LastVal == 0 ? F10FirstVal : "\(F10LastVal)I+\(F10FirstVal)"
let F11FirstVal = "\(((factor1String.first?.contains("F11")
?? false) ? 0.25 : 0) +
((factor2String.first?.contains("F11") ?? false) ? 0.25 :
0) + ((factor3String.first?.contains("F11") ?? false) ?
0.25 : 0) +     ((factor4String.first?.contains("F11") ?? false) ? 0.25 : 0))"
let F11LastVal = ((factor1String.last?.contains("F11") ?? false) ? 1 : 0) +
((factor2String.last?.contains("F11") ?? false) ? 1 : 0) + ((factor3String.last?.contains("F11") ?? false) ? 1
: 0) +     ((factor4String.last?.contains("F11")
?? false) ? 1 : 0)
self.f11Label.text = F11LastVal == 0 ? F11FirstVal : "\(F11LastVal)I+\(F11FirstVal)"
}


}
}


override func setSelected(_ selected: Bool, animated: Bool) { super.setSelected(selected, animated:
animated)
// Configure the view for the selected state
}


}
class FactorTableViewCell: UITableViewCell {


@IBOutlet weak var Expert4Label: UILabel! @IBOutlet weak var expert3Label: UILabel! @IBOutlet
weak var expert2Label: UILabel! @IBOutlet weak var expert1Label: UILabel! @IBOutlet weak var
factorLabel: UILabel! @IBOutlet weak var titleLabel: UILabel!
override func awakeFromNib() { super.awakeFromNib()
```

```
// Initialization code
}

override func setSelected(_ selected: Bool, animated: Bool) { super.setSelected(selected, animated: animated)
// Configure the view for the selected state
}

}
```

Received: Dec. 13, 2021. Accepted: April 5, 2022.