# ASSIGNMENT

| | |
|---|---|
| **Course Code** | CSC401A |
| **Course Name** | Computational Intelligence |
| **Programme** | B.Tech |
| **Department** | Computer Science |
| **Faculty** | FET |


| | |
|---|---|
| **Name of the Student** | SHUBHAM AGARWAL |
| **Reg. No** | 17ETCS002175 |
| **Semester/Year** | 7$^{th}$ /2017 |
| **Course Leader/s** | Mr. Sagar U |

<table>
<tr><th colspan="6"><u>Declaration Sheet</u></th></tr>
<tr><td>Student Name</td><td colspan="5">SHUBHAM AGARWAL</td></tr>
<tr><td>Reg. No</td><td colspan="5">17ETCS002175</td></tr>
<tr><td>Programme</td><td colspan="2">B.Tech</td><td>Semester/Year</td><td colspan="2">7/2017</td></tr>
<tr><td>Course Code</td><td colspan="5">CSC401A</td></tr>
<tr><td>Course Title</td><td colspan="5">Computational Intelligence</td></tr>
<tr><td>Course Date</td><td>***</td><td>to</td><td colspan="3">****</td></tr>
<tr><td>Course Leader</td><td colspan="5">Mr Sagar U</td></tr>
<tr><td colspan="6">Declaration<br>The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.</td></tr>
<tr><td>Signature of the Student</td><td colspan="3"></td><td>Date</td><td></td></tr>
<tr><td>Submission date stamp<br>(by Examination & Assessment Section)</td><td colspan="5"></td></tr>
<tr><td colspan="2">Signature of the Course Leader and date</td><td colspan="4">Signature of the Reviewer and date</td></tr>
<tr><td colspan="2"></td><td colspan="4"></td></tr>
</table>

| Faculty of Engineering and Technology | | | |
|---|---|---|---|
| **Ramaiah University of Applied Sciences** | | | |
| Department | Computer Science and Engineering | Programme | B. Tech. in CSE |
| Semester/Batch | 7/2017 | | |
| Course Code | CSC401A | Course Title | Computational Intelligence |
| Course Leader | Dr. Vaishali R. Kulkarni/Prof. Prabhakar/Mr. Sagar U. | | |

## Assignment-02

| **Reg.No.** | **17ETCS002175** | **Name of Student** | **SHUBHAM AGARWAL** |
|---|---|---|---|

| Sections | | Marking Scheme | Mark | | |
|---|---|---|---|---|---|
| | | | **Max Marks** | **First Examiner Marks** | **Moderator** |
| **Part A** | | | | | |
| | **A.1.1** | **A critical review of capabilities of Artificial Neural Networks (ANNs)** | **04** | | |
| | **A.1.2** | **The application areas of ANNs** | **04** | | |
| | **A.1.3** | **Conclusion** | **02** | | |
| | | **Part-A Max Marks** | **10** | | |
| **Part B.1** | | | | | |
| | **B.1.1** | **A model of generalised fuzzy decision system** | **05** | | |
| | **B.1.2** | **Fuzzy sets with attributes** | **05** | | |
| | **B.1.3** | **Python program demonstration** | **05** | | |
| | | **B.1 Max Marks** | **15** | | |
| | | **Total Assignment Marks** | **25** | | |

## Course Marks Tabulation

| Component-1 (B) Assignment | First Examiner | Remarks | Moderator | Remarks |
|---|---|---|---|---|
| A | | | | |
| B.1 | | | | |
| B.2 | | | | |
| **Marks (out of 25)** | | | | |

# Contents

## 1. Part A

**Solution to Part A Question No 1:**

**A.1.1 A critical review of cognitive capabilities of ANNs**

Artificial neural networks (ANNs) have become popular and helpful model for classification, clustering, pattern recognition and prediction in many disciplines. ANNs are one type of model for machine learning (ML) and has become relatively competitive to conventional regression and statistical models regarding usefulness. Currently, artificial intelligence, information security, big data, cloud computing, internet, and forensic science are all hotspots and exciting topics of information and communication technology (ICT).
ANNs full applications can be evaluated with respect to data analysis factors such as accuracy, processing speed, latency, performance, fault tolerance, volume, scalability and convergence. The great potential of ANNs is the high-speed processing provided in a massive parallel implementation and this has heightened the need for research in this domain. ANNs can be developed and used for image recognition, natural language processing and so on. Nowadays, ANNs are mostly used for universal function approximation in numerical paradigms because of their excellent properties of self-learning, adaptively, fault tolerance, nonlinearity, and advancement in input to an output mapping.

These data analysis factors give more reason why ANNs are effective, efficient and successful in providing a high level of capability in handling complex and noncomplex problems in many spheres of life. ANNs are capable of handling problems in agriculture, science, medical science, education, finance, management, security, engineering, trading commodity and art. Including problems in manufacturing, transportation, computer security, banking, insurance, properties management, marketing, energy, and those challenges that cannot be solve by the computational ability of traditional procedures and conventional mathematics. Despite these extensive e applications of ANNs, there is an increasing need to address the problem of adopting a systematic approach in ANNs development phase to improve its performance. For instance, an approach to address major factors and topics in a choice of data sets, the accuracy of data, data instrument, data standardization, type of data inputs, data division, and data preprocessing, validations, processing and output techniques.

The other key challenges or issues that are common with ANN modeling which have received interest and require further investigation in future. Including developmental techniques that can improve designing of robust models, improving pattern transparency and allowing useful knowledge from trained ANNs. More also are the challenges of improving extrapolation ability, new approaches to uncertainty and improving convergences. More also, there is continuous gradient enigma and quantization of variable problems and noise.
Furthermore, there is a need to address the traversal of the error surface by utilizing quantization of variable and time-consuming convergence problems common to most artificial neural systems (ANS) that use supervised training. Some of these problems are highlights as follows:

1. Improve designing of robust models: model robustness means predictive capability of ANN kinds in generalizing range of data like those used for training. An example is using of textual data or information to improve modeling prediction of the financial market.
2. Some experts believe if that ANNs become globally accepted and reach apex potentiality, they will not only provide a good fit to calibration and validation of data. But will enable predictions that will be plausible regarding model's correlation and robustness in any range of conditions. ANNs validated of error can give accurate predictions for conditions like those found in trained data.
3. Improving of model transparency and the enabling of knowledge extraction from trained ANNs: means the possibility of interpreting ANN models in a way that provides a deep understanding of the effect of model inputs to outputs.
4. Improving extrapolation ability: extrapolation of ANN models is the capability of the model to predict accurately outward range of data used for ANN model calibration. ANNs perform best if they do not extrapolate above the range of data used for design or model calibration.
5. New approaches to uncertainty: another limitation of ANNs including uncertainty in predictions which may not be taken to account. When uncertainty is not accounted, it becomes difficult to measure ANN predictions quality, which can critically limit or reduces their efficacy. Although ANNs has had their issues, new approaches like cognitive computing and deep learning have significantly raised the support in these fields. A synthetic machine might still be out of reach, but systems like ANNs that help improve people's lives are here today.

### A.1.2 The application areas in which ANNs have been successful in delivering human-like performance

Artificial Neural Networks (ANNs) application have become popular in various area of human needs. Many organizations are investing in the neural networks to solve problems in various fields and the economic sector which traditionally fall under the responsibility of operations research. What makes artificial intelligence unique is that it is mostly proposed for data analyses by academics in the fields of social science and arts apart from its usefulness in science and engineering, because of its wide applications. For example, in recent times, artificial intelligence (AI) has been extensively applied to optimization issues in diverse areas like industrial production and petroleum exploration and business setting.

A good advantage of ANNs application is that it can make models easy to use and more accurate from complex natural systems with large inputs. The ANN is found to be a very novel and useful model applied to problem-solving and machine learning. ANN is an information manager model that is similar to biological nervous systems function of the man brain.

Recently, research interest in brain functionality has rapidly increased globally. According to Haykin, an ANN can be comparable machine produced to function the same way the human brain performs a given task of interest. For example, "the human brain is big and highly efficient. The man brain is like an information-processing machine that has a variety of complex signal computing operations" that can be easily coordinated to perform a task.

The main element of this brain is the unique design of their information processing capability. It constitutes many complexes interconnected "neurons" in the form of elements working together to solve specific problems on daily basis. A typical example of a neural network function is the human brain that is connected to send and receive signals for human action.

Many artificial neural network techniques have been adopted in the academia and industries to address the challenges in computer vision, speech and pattern recognitions, face alignment, and detection.

**ANNs has seen massive use in specific domains.**

- Diagnosis of hepatitis.
- Speech recognition.
- Recovery of data in telecommunications from faulty software.
- Interpretation of multi-language messages.
- Three-dimensional object recognition.
- Texture analysis.
- Facial recognition.
- Hand-written word recognition.

1. **Speech recognition:**

The application of ANNs has become divergence and understood in the capability of its successes in speech or communication recognition. In the past decades, ML algorithms have applied widely in areas like acoustic modeling and ASR (automatic speech recognition).

2. **Computer vision.**

Artificial Neural Networks play a vital role in Computer vision that aims at making computers to accurately understand and process visual data efficiently like videos and images. Main goal of computer vision is to provide computers with the kind of ability of man brain functionality. Theoretically, computer vision alludes to the logical control which studies how to separate data from images in artificial frameworks. Sub domains of computer vision include object detection and object recognition, object estimation, object position, event detection, scene reconstruction, image restoration, image editing, video enhancement, and statistical learning. Hence, in computer vision, ANN models are very useful.

3. **Pattern recognition.**

Artificial Neural Networks play a vital role in Pattern recognition. The recent improvement in deep learning models has given novel ways to deal with the issue in recognition of a pattern or pattern recognition (PR). PR is a scientific area that focus in identification of sequence in each input. PR is a general concept that surrounds various subdomains such as speech tagging, regression, sequence labeling and classification. There are rapidly increasing needs for information processing and output, due to industrial development, that has new trend and challenges to PR.

4. **Face alignment.**

Artificial Neural Networks play a vital role in face alignment. Face alignment plays a role that is significant in diverse visual applications. In recent times ANNs has claimed successes in face alignment and face recognition and other models have shown successes. Interestingly DL techniques can be applying to explain genetic variants to identify pathogenic variants. Usually, combined annotation dependent depletion algorithm is popularly applied to interpret the coding and non-coding variants.

5. **Detection**

Detection in medical diagnosis, security, image objects, financial irregularity, a fault in a system, are being enhanced through ANNs application. Thus, ANN plays an essential role in the detection, particularly when applied to breast cancer. The performance of ANN can be relatively compared with other approaches in crime detection such as DNA and activity profiling and the use of big data for financial crime detection. Despite the many publications in the utilization of ANN in different medical challenges, but there are few reviews study available that explain the architecture in improving the detection methods regarding performance, accuracy, sensitivity, and specificity.

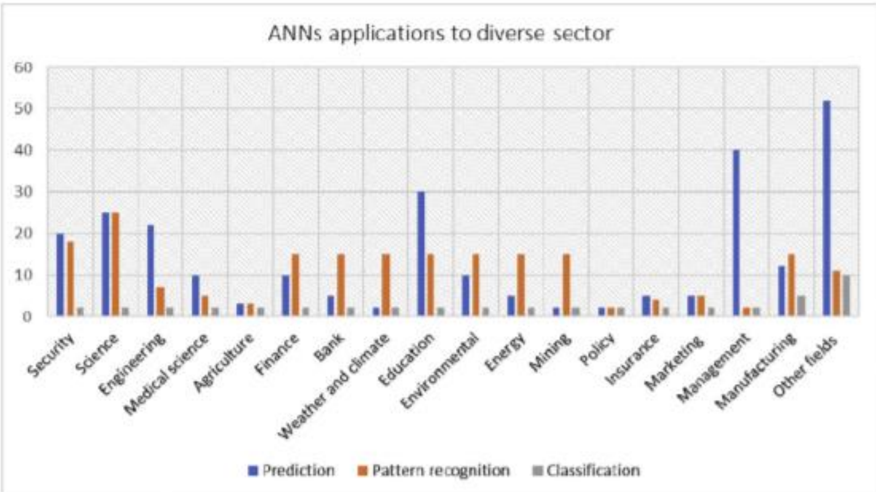



**Figure 1: ANN Application Histogram**

The above indicates the result of ANNs application to different areas of lives and disciplines. The correlation among the distinct fields further reveals that ANN can apply to any areas of studies, industries, and profession. The histogram reveals the areas of application of ANN in security, science, engineering, medical science, agriculture, finance, banking, weather and climate, education, environmental, energy, mining, insurance, marketing etc. Therefore, interested researchers can explore the ANN application in these areas or many other emerging areas for future research for better solution to problems in their fields. Since there is always an algorithm, model, scheme, and framework for any problem.

In the recent times various successful used of ANNs emerged in catalysis, meteorology, biology, chemistry, physics, nuclear physics, high-energy physics, and other areas of science. Nowadays, ANN has found uses in a new area such as in catalyzing especially in the chemical industrial sector. Catalysis is term as the significant energy in the modernization process of chemical industries. It ensures effective, efficient and successful use of finite natural resources, it prevent waste and air pollution, and provides safety for the industrial sector. Catalysis become the foundation of large-scale operations regarding size in chemistry and petrochemistry environment. However, as demand changes, new environmental challenges now require new catalytic solutions. For example, changed in the energy economy has driven an increasing demand for coal and gas, hence given room for new challenges for catalytic technology in the areas like liquefaction in material science.

Thus, ANNs can learn by example like people. In some cases, ANNs can be designed for a specific application like data classification or pattern recognition through the learning process. Learning in the human brain requires adjustments to the synaptic relationship between and among neurons, likewise the learning in ANNs. Generally, an ANN function like an imitation of the man brain.

## **A.1.3 Conclusion**

The above comprehensive discussion on how ANN could apply to address human needs. ANNs has many names connectionism models, adaptive systems, parallel distributed processing models, self-organizing systems, neuromorphic and neurocomputing systems.

The ANNs application areas considered in the survey includes, computer security, medical science, business, finance, bank, insurance, the stock market, electricity generation, management, nuclear industry, mineral exploration, mining, crude oil fractions quality prediction, crops yield prediction, water treatment, and policy. It is interesting to know that neural network data analysis adds accuracy, processing speed, fault tolerance, latency, performance, volume, and scalability. Many new and enhanced data management and data analysis approaches help in the management of ANN. Creating analytics from the available data that aid in largely prioritizing information and provide its human business value. The ANN analytics in turn help in combating challenges and mitigate any possible risks.

Therefore, based on data analysis factors such as accuracy, processing speed, latency, performance, fault tolerance, volume, and scalability, an evaluation was made of the ANN techniques. Then, proposes that neural-networks models such as FFBP and hybrid model using neural networks are performing better for implementation of human problems when compared to other approaches currently in practice. Also, the study proposes hybrid neural networks models and genetic algorithms (GA) for a better performance regarding effectiveness and efficiency. ANN are new computational model with rapid and large uses for handling various complex real-world issues. ANNs popularity lies in information processing characteristics to learning power, high parallelism, fault tolerance, nonlinearity, noise tolerance, and capabilities of generalization.

## 2. <u>Part B</u>

**Solution to Part B Question No 1:**

### B.1.1. A model of generalised fuzzy decision system for project evaluation

Fuzzy Logic is defined as a many-valued logic form which may have truth values of variables in any real number between 0 and 1. It is the handle concept of partial truth. In real life, we may come across a situation where we can't decide whether the statement is true or false. At that time, fuzzy logic offers very valuable flexibility for reasoning.

Fuzzy logic algorithm helps to solve a problem after considering all available data. Then it takes the best possible decision for the given the input. The FL method imitates the way of decision making in a human which consider all the possibilities between digital values T and F.

Fuzzy Logic Architecture contains four parts :

- RULE BASE: It contains the set of rules and the IF-THEN conditions provided by the experts to govern the decision making system, on the basis of linguistic information. Recent developments in fuzzy theory offer several effective methods for the design and tuning of fuzzy controllers. Most of these developments reduce the number of fuzzy rules.
- FUZZIFICATION: It is used to convert inputs i.e. crisp numbers into fuzzy sets. Crisp inputs are basically the exact inputs measured by sensors and passed into the control system for processing, such as temperature, pressure, rpm’s, etc.
- INFERENCE ENGINE: It determines the matching degree of the current fuzzy input with respect to each rule and decides which rules are to be fired according to the input field. Next, the fired rules are combined to form the control actions.
- DEFUZZIFICATION: It is used to convert the fuzzy sets obtained by inference engine into a crisp value. There are several defuzzification methods available and the best suited one is used with a specific expert system to reduce the error.

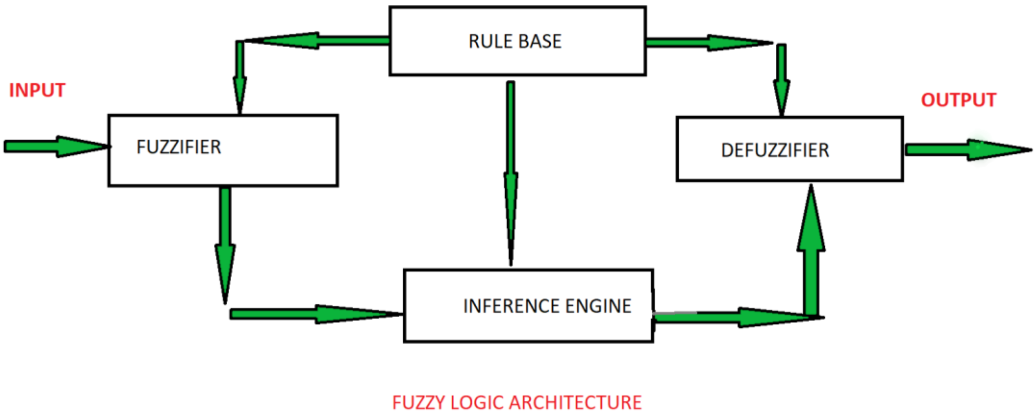



**Figure 2: Generalized Fuzzy System.**

A **fuzzy variable** has a **crisp value** which takes on some number over a pre-defined domain (in fuzzy logic terms, called a **universe**). The crisp value is how we think of the variable using normal mathematics. A fuzzy variable also has several **terms** that are used to describe the variable. The terms taken together are the **fuzzy set** which can be used to describe the "fuzzy value" of a fuzzy variable. These terms are usually adjectives like "poor," "mediocre," and "good." Each term has a **membership function** that defines how a crisp value maps to the term on a scale of 0 to 1.

The problem stated in this question is modeled as follows – two judges rate projects based on criteria such as demonstration, presentation, etc., between one to twenty-five. These will be the crisp inputs to the decision system as depicted in the diagram. The fuzzifier will convert these inputs into fuzzy sets using membership functions (explained in the next section, and the graphs of the membership functions are in section B1.3). These fuzzy sets will be used in the inference engine, and along with rules (a subset of the rules, not all $5^2 = 25$ possible rules) will product the fuzzy output set. This result is then defuzzified to get crisp output, i.e., the students grade between zero and fifty.

The Linguistic Variable/ fuzzy set for the Project Evaluation Problem are selected as Outstanding, Excellent, Very Good, Good, Above Average, Average, Pass, Fail.

## B.1.2 Fuzzy sets with attributes:

Before talking about fuzzy sets, topics such as universe of discourse and membership functions must be touched upon.

The term "universe of discourse" generally refers to the collection of objects being discussed in a specific discourse. In other words, universe of discourse can be thought of as the universal set in traditional set theory. In the case of this question, the universe of discourse would be the various types of opinions each judge can give, i.e.,

$$\Omega = \{\text{bad, average, good, very good, excellent}\}$$

Since fuzzy logic deals with degrees of truth, each element in the universe of discourse has partial membership in various sets that are created using elements from the universe of discourse. This contrasts with regular sets where each element in the universal set has either a membership or a non-membership in the sets created using the elements from it. As such, the function that defines the degree of membership an element has in a set is called membership function, denoted by $\mu(a)$ or $m(a)$.

Now, a fuzzy set is essentially a set whose elements have degrees of membership between 0 and 1. In other words,

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in \Omega\}$$

For example, a Examiner can give an evaluation which borders between excellent, `very good` and `good`. In this case, the fuzzy set of the Examiner evaluation can be represented by:

$$\tilde{A} = \{(\text{bad}, 0), (\text{average}, 0), (\text{good}, 0.57), (\text{very good}, 0.80), (\text{excellent}, 0.57)\}$$

When the universe of discourse is discrete and finite, the fuzzy set $\tilde{A}$ is given by:

$$\tilde{A} = \sum_{i=1}^{n} \frac{\mu_{\tilde{A}}(x_i)}{x_i}\text{s}$$

If it is continuous, it is given by:

$$\tilde{A} = \int \frac{\mu_{\tilde{A}}(x)}{x}$$

The projects are judged based on three aspects which are project goal and area, functionality, project report documentation. The judges take these three aspects into consideration and give scores for each aspect to each project. These three scores are then taken into deciding the final score for each project and these scores are used to calculate grades for each student.

The projects are then given grades based on the Examiner evaluation. The grades are O, A, B, P and F, where O is the best grade and F is the worst.

$$\Omega = \{\text{O}, \text{A}, \text{B}, \text{P}, \text{F}\}$$

Membership function also known as characteristic function can take value between 0 and 1 and indicates degree of membership. Since there are infinite numbers between 0 and 1, infinite degrees of membership are possible.

So the possible Rules for Evaluation System are:

# if one Examiner says bad and the other says bad, then the grade is F

# if one Examiner says bad and the other says average, then the grade is F

# if one Examiner says bad and the other says good, then the grade is P

# if one Examiner says bad and the other says very good, then the grade is P

# if one Examiner says bad and the other says excellent, then the grade is B

# if one Examiner says average and the other says average, then the grade is P

# if one Examiner says average and the other says good, then the grade is B

# if one Examiner says average and the other says very good, then the grade is B

# if one Examiner says average and the other says excellent, then the grade is A

# if one Examiner says good and the other says good, then the grade is B

# if one Examiner says good and the other says very good, then the grade is A

# if one Examiner says good and the other says excellent, then the grade is A

# if one Examiner says very good and the other says very good, then the grade is A

# if one Examiner says very good and the other says excellent, then the grade is O

# if one Examiner says excellent and the other says excellent, then the grade is O

**Advantages of Fuzzy Logic System**

- The structure of Fuzzy Logic Systems is easy and understandable
- Fuzzy logic is widely used for commercial and practical purposes
- It helps you to control machines and consumer products
- It may not offer accurate reasoning, but the only acceptable reasoning
- It helps you to deal with the uncertainty in engineering
- Mostly robust as no precise inputs required
- It can be programmed to in the situation when feedback sensor stops working
- It can easily be modified to improve or alter system performance
- inexpensive sensors can be used which helps you to keep the overall system cost and complexity low
- It provides a most effective solution to complex issues

**Disadvantages of Fuzzy Logic Systems**

- Fuzzy logic is not always accurate, so the results are perceived based on assumption, so it may not be widely accepted.
- Fuzzy systems don't have the capability of machine learning as-well as neural network type pattern recognition
- Validation and Verification of a fuzzy knowledge-based system needs extensive testing with hardware
- Setting exact, fuzzy rules and, membership functions is a difficult task
- Some fuzzy time logic is confused with probability theory and the term

## B.1.3 Python program demonstration for at least 50 test cases.

```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
```

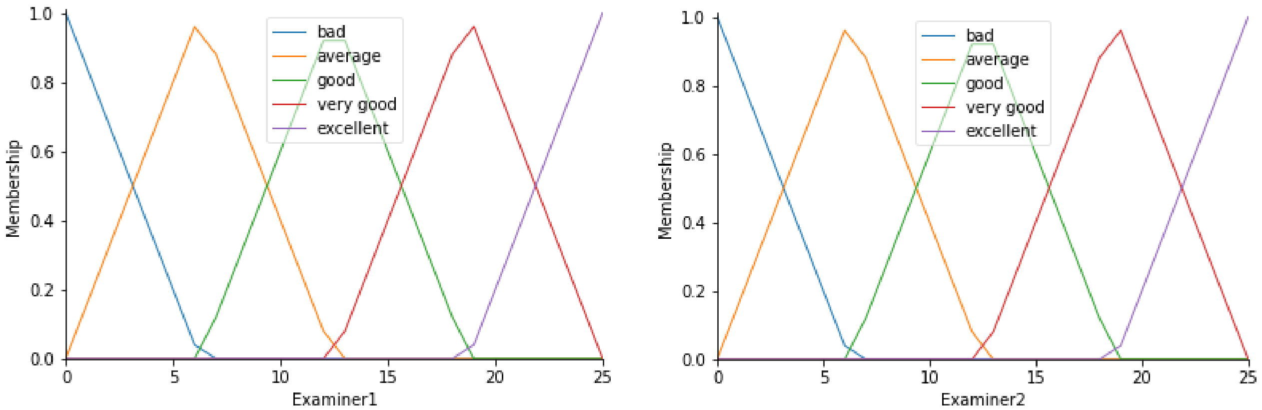
Creating the inputs (antecedents) and outputs (consequents)

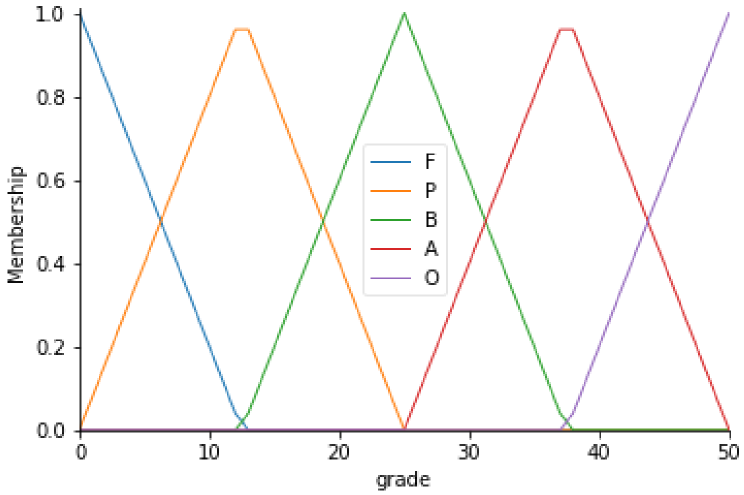
```python
# Examiner can score between 1-25
examiner1 = ctrl.Antecedent(np.arange(0, 26, 1), 'Examiner1')
examiner2 = ctrl.Antecedent(np.arange(0, 26, 1), 'Examiner2')

# final grade is between 1-50
grade = ctrl.Consequent(np.arange(0, 51, 1), 'grade')
```

Setting the membership function

```python
# Consider A system of 2 Examiner and these examiner can grade each student from 5 different avaliable grade.

examiner1.automf(5, names=["bad", "average", "good", "very good", "excellent"])
examiner2.automf(5, names=["bad", "average", "good", "very good", "excellent"])
#Grade code
grade.automf(5, names=["F", "P", "B", "A", "O"])
```

```python
examiner1.view()
examiner2.view()
grade.view()
```






Now, to make these triangles useful, we define the fuzzy relationship between input and output variables.

```
rules = [] #dictonary which will contain  different rules.

rules.append(ctrl.Rule(examiner1['bad'] & examiner2['bad'], grade['F']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['average']) | (examiner1['average'] & examiner2['bad']), grade['F']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['good']) | (examiner1['good'] & examiner2['bad']), grade['P']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['very good']) | (examiner1['very good'] & examiner2['bad']), grade['P']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['bad']), grade['B']))
rules.append(ctrl.Rule(examiner1['average'] & examiner2['average'], grade['P']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['good']) | (examiner1['good'] & examiner2['average']), grade['B']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['very good']) | (examiner1['very good'] & examiner2['average']), grade['B']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['average']), grade['A']))
rules.append(ctrl.Rule(examiner1['good'] & examiner2['good'], grade['B']))
rules.append(ctrl.Rule((examiner1['good'] & examiner2['very good']) | (examiner1['very good'] & examiner2['good']), grade['A']))
rules.append(ctrl.Rule((examiner1['good'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['good']), grade['A']))
rules.append(ctrl.Rule(examiner1['very good'] & examiner2['very good'], grade['A']))
rules.append(ctrl.Rule((examiner1['very good'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['very good']), grade['O']))
rules.append(ctrl.Rule(examiner1['excellent'] & examiner2['excellent'], grade['O']))
```

Creating a control system and simulation

```
[7] gradeControl = ctrl.ControlSystem(rules)
    gradingSystem = ctrl.ControlSystemSimulation(gradeControl)
```

In order to simulate this control system, we will create a ControlSystemSimulation. Think of this object representing our controller applied to a specific set of circumstances.

We can now simulate our control system by simply specifying the inputs and calling the compute method

```
for i in range(50):
    examiner1 = np.random.uniform(1, 25)
    examiner2 = np.random.uniform(1, 25)

    gradingSystem.inputs({
        "Examiner1": examiner1,
        "Examiner2": examiner2
    })

    gradingSystem.compute()
    print("Test Case No: {} - 1st Examiner Score: {:.3f}, 2nd Examiner Score: {:.3f},  Final result: {:.2f} / 50".format(i+1, examiner1, examiner2, gradingSystem.output["grade"]))
```

```
Test Case No: 1 - 1st Examiner Score: 6.745, 2nd Examiner Score: 1.948,  Final result: 11.76 / 50
Test Case No: 2 - 1st Examiner Score: 11.566, 2nd Examiner Score: 3.348,  Final result: 18.95 / 50
Test Case No: 3 - 1st Examiner Score: 24.168, 2nd Examiner Score: 13.658,  Final result: 37.71 / 50
Test Case No: 4 - 1st Examiner Score: 4.452, 2nd Examiner Score: 18.386,  Final result: 21.25 / 50
Test Case No: 5 - 1st Examiner Score: 15.713, 2nd Examiner Score: 6.754,  Final result: 26.69 / 50
Test Case No: 6 - 1st Examiner Score: 22.968, 2nd Examiner Score: 16.926,  Final result: 40.36 / 50
Test Case No: 7 - 1st Examiner Score: 21.366, 2nd Examiner Score: 17.228,  Final result: 38.54 / 50
Test Case No: 8 - 1st Examiner Score: 12.918, 2nd Examiner Score: 11.338,  Final result: 26.16 / 50
Test Case No: 9 - 1st Examiner Score: 16.964, 2nd Examiner Score: 21.364,  Final result: 38.53 / 50
Test Case No: 10 - 1st Examiner Score: 18.761, 2nd Examiner Score: 21.636,  Final result: 38.77 / 50
Test Case No: 11 - 1st Examiner Score: 15.409, 2nd Examiner Score: 15.311,  Final result: 30.89 / 50
Test Case No: 12 - 1st Examiner Score: 24.481, 2nd Examiner Score: 19.430,  Final result: 43.76 / 50
```

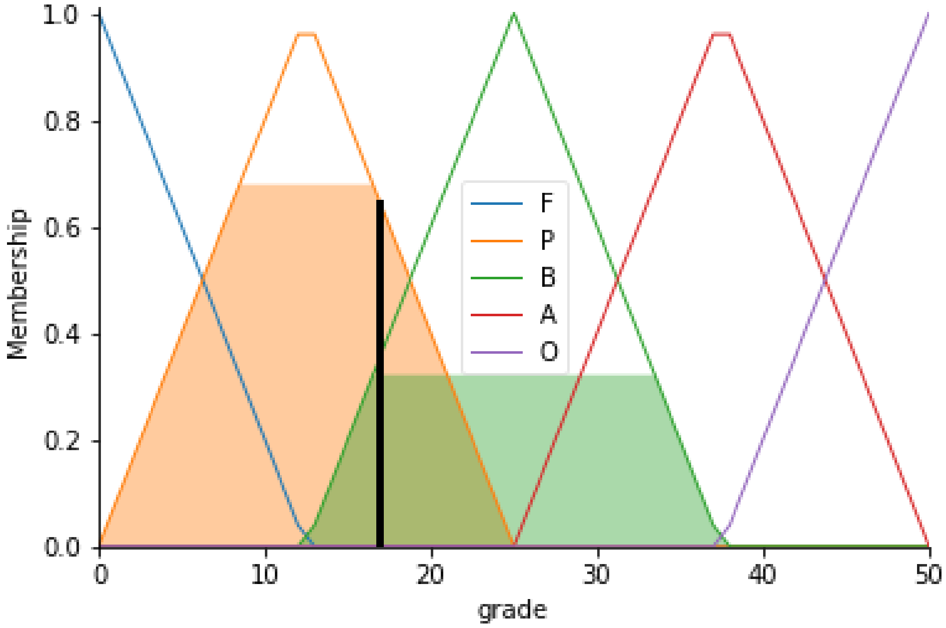



**Figure 3: grades of students**

## 3. Bibliography

---


- https://www.computerworld.com/article/2591759/artificial-neural-networks.html
- https://medium.com/@camirosso/are-artificial-neural-networks-the-holy-grail-3626a41e35
- https://www.xenonstack.com/blog/artificial-neural-network-applications/
- ttps://www.guru99.com/what-is-fuzzy-logic.html
- https://www.investopedia.com/terms/f/fuzzy-logic.asp
- https://www.scientificamerican.com/article/what-is-fuzzy-logic-are-t/
- https://www.geeksforgeeks.org/fuzzy-logic-introduction/
- https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_systems.htm
- https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html


The Colab Notebook can be found :

https://colab.research.google.com/drive/1k8i6f9hAUJ6MwrblGEa-4i7yxVsGtVrT?usp=sharing

# Fuzzy Logic

In [ ]:

```
!pip install -U scikit-fuzzy
```

```
Requirement already up-to-date: scikit-fuzzy in /usr/local/lib/python3.6/dist-packages (0
.4.2)
Requirement already satisfied, skipping upgrade: numpy>=1.6.0 in /usr/local/lib/python3.6
/dist-packages (from scikit-fuzzy) (1.19.5)
Requirement already satisfied, skipping upgrade: scipy>=0.9.0 in /usr/local/lib/python3.6
/dist-packages (from scikit-fuzzy) (1.4.1)
Requirement already satisfied, skipping upgrade: networkx>=1.9.0 in /usr/local/lib/python
3.6/dist-packages (from scikit-fuzzy) (2.5)
Requirement already satisfied, skipping upgrade: decorator>=4.3.0 in /usr/local/lib/pytho
n3.6/dist-packages (from networkx>=1.9.0->scikit-fuzzy) (4.4.2)
```

In [ ]:

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
```

#### Creating the inputs (antecedents) and outputs (consequents)

In [ ]:

```
# Examiner can score between 1-25
examiner1 = ctrl.Antecedent(np.arange(0, 26, 1), 'Examiner1')
examiner2 = ctrl.Antecedent(np.arange(0, 26, 1), 'Examiner2')

# final grade is between 1-50
grade = ctrl.Consequent(np.arange(0, 51, 1), 'grade')
```

#### Setting the membership function

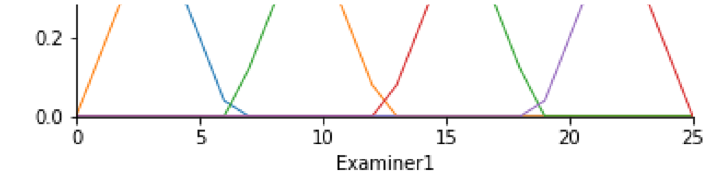
In [ ]:

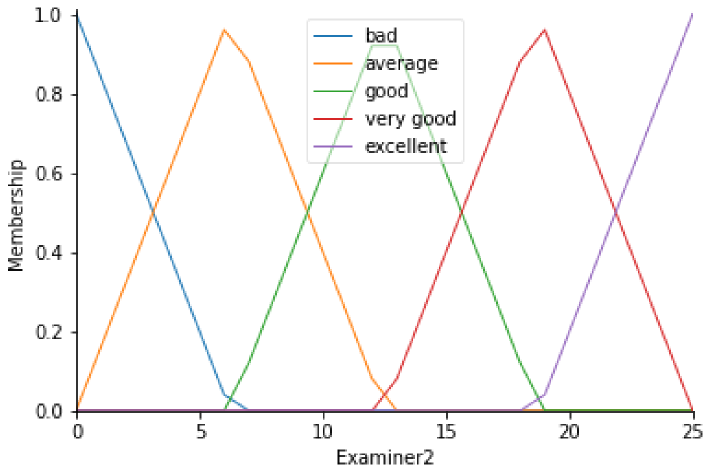
```
# Consider A system of 2 Examiner and these examiner can grade each student from 5 differ
ent avaliable grade.

examiner1.automf(5, names=["bad", "average", "good", "very good", "excellent"])
examiner2.automf(5, names=["bad", "average", "good", "very good", "excellent"])
#Grade code
grade.automf(5, names=["F", "P", "B", "A", "O"])
```

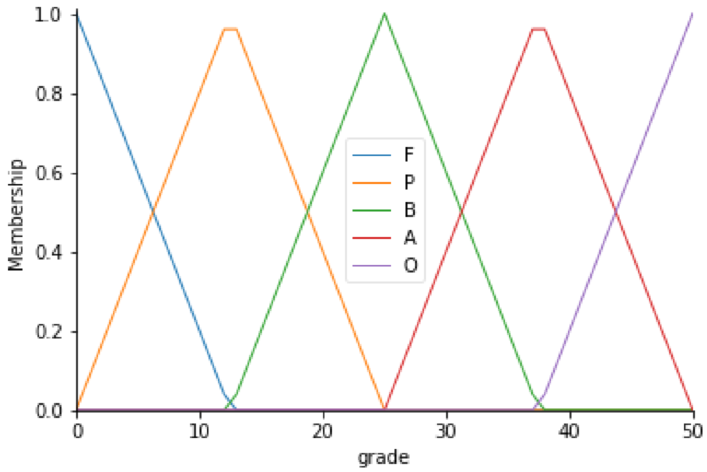
#### Plotting the membership functions of the inputs and outputs

In [ ]:

```
examiner1.view()
examiner2.view()
grade.view()
```




#### Creating the inference rules

In [ ]:

```
rules = [] #dictonary which will contain  different rules.

rules.append(ctrl.Rule(examiner1['bad'] & examiner2['bad'], grade['F']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['average']) | (examiner1['average'] & examiner2['bad']), grade['F']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['good']) | (examiner1['good'] & examiner2['bad']), grade['P']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['very good']) | (examiner1['very good'] & examiner2['bad']), grade['P']))
rules.append(ctrl.Rule((examiner1['bad'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['bad']), grade['B']))
rules.append(ctrl.Rule(examiner1['average'] & examiner2['average'], grade['P']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['good']) | (examiner1['good'] & examiner2['average']), grade['B']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['very good']) | (examiner1['very good'] & examiner2['average']), grade['B']))
rules.append(ctrl.Rule((examiner1['average'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['average']), grade['A']))
rules.append(ctrl.Rule(examiner1['good'] & examiner2['good'], grade['B']))
rules.append(ctrl.Rule((examiner1['good'] & examiner2['very good']) | (examiner1['very good'] & examiner2['good']), grade['A']))
rules.append(ctrl.Rule((examiner1['good'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['good']), grade['A']))
rules.append(ctrl.Rule(examiner1['very good'] & examiner2['very good'], grade['A']))
rules.append(ctrl.Rule((examiner1['very good'] & examiner2['excellent']) | (examiner1['excellent'] & examiner2['very good']), grade['O']))
rules.append(ctrl.Rule(examiner1['excellent'] & examiner2['excellent'], grade['O']))
```

### Creating a control system and simulation

In [ ]:

```
gradeControl = ctrl.ControlSystem(rules)
gradingSystem = ctrl.ControlSystemSimulation(gradeControl)
```

### Running 50 simulations

In [ ]:

```
for i in range(50):
    examiner1 = np.random.uniform(1, 25)
    examiner2 = np.random.uniform(1, 25)

    gradingSystem.inputs({
        "Examiner1": examiner1,
        "Examiner2": examiner2
    })

    gradingSystem.compute()
    print("Test Case No: {} - 1st Examiner Score: {:.3f}, 2nd Examiner Score: {:.3f},  F
inal result: {:.2f} / 50".format(i+1, examiner1, examiner2, gradingSystem.output["grade"
]))
```

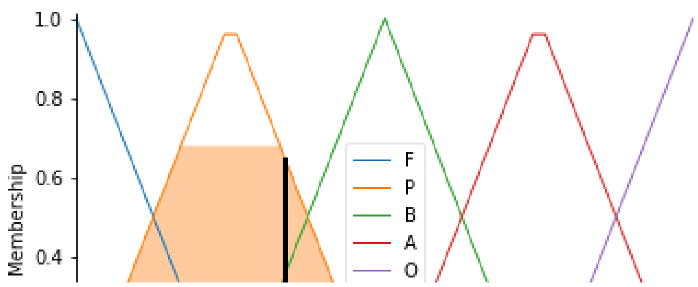
```
Test Case No: 1 - 1st Examiner Score: 6.745, 2nd Examiner Score: 1.948,  Final result: 11
.76 / 50
Test Case No: 2 - 1st Examiner Score: 11.566, 2nd Examiner Score: 3.348,  Final result: 1
8.95 / 50
Test Case No: 3 - 1st Examiner Score: 24.168, 2nd Examiner Score: 13.658,  Final result:
37.71 / 50
Test Case No: 4 - 1st Examiner Score: 4.452, 2nd Examiner Score: 18.386,  Final result: 2
1.25 / 50
Test Case No: 5 - 1st Examiner Score: 15.713, 2nd Examiner Score: 6.754,  Final result: 2
6.69 / 50
Test Case No: 6 - 1st Examiner Score: 22.968, 2nd Examiner Score: 16.926,  Final result:
40.36 / 50
Test Case No: 7 - 1st Examiner Score: 21.366, 2nd Examiner Score: 17.228,  Final result:
38.54 / 50
Test Case No: 8 - 1st Examiner Score: 12.918, 2nd Examiner Score: 11.338,  Final result:
26.16 / 50
Test Case No: 9 - 1st Examiner Score: 16.964, 2nd Examiner Score: 21.364,  Final result:
38.53 / 50
Test Case No: 10 - 1st Examiner Score: 18.761, 2nd Examiner Score: 21.636,  Final result:
38.77 / 50
Test Case No: 11 - 1st Examiner Score: 15.409, 2nd Examiner Score: 15.311,  Final result:
30.89 / 50
Test Case No: 12 - 1st Examiner Score: 24.481, 2nd Examiner Score: 19.430,  Final result:
43.76 / 50
Test Case No: 13 - 1st Examiner Score: 22.407, 2nd Examiner Score: 13.258,  Final result:
37.62 / 50
Test Case No: 14 - 1st Examiner Score: 17.027, 2nd Examiner Score: 21.569,  Final result:
38.71 / 50
Test Case No: 15 - 1st Examiner Score: 19.763, 2nd Examiner Score: 1.116,  Final result:
18.66 / 50
Test Case No: 16 - 1st Examiner Score: 9.899, 2nd Examiner Score: 16.378,  Final result:
32.08 / 50
Test Case No: 17 - 1st Examiner Score: 11.639, 2nd Examiner Score: 19.191,  Final result:
35.28 / 50
Test Case No: 18 - 1st Examiner Score: 6.050, 2nd Examiner Score: 16.987,  Final result:
24.40 / 50
Test Case No: 19 - 1st Examiner Score: 13.391, 2nd Examiner Score: 13.035,  Final result:
27.26 / 50
Test Case No: 20 - 1st Examiner Score: 7.982, 2nd Examiner Score: 20.059,  Final result:
28.92 / 50
Test Case No: 21 - 1st Examiner Score: 2.695, 2nd Examiner Score: 3.488,  Final result: 1
0.59 / 50
Test Case No: 22 - 1st Examiner Score: 23.234, 2nd Examiner Score: 13.942,  Final result:
37.83 / 50
Test Case No: 23 - 1st Examiner Score: 12.844, 2nd Examiner Score: 21.199,  Final result:
```

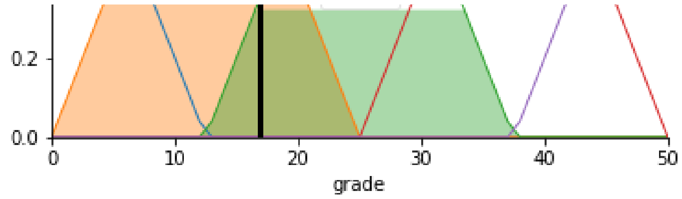
```
37.25 / 50
Test Case No: 24 - 1st Examiner Score: 19.710, 2nd Examiner Score: 14.414,  Final result:
37.66 / 50
Test Case No: 25 - 1st Examiner Score: 1.080, 2nd Examiner Score: 18.383,  Final result:
15.56 / 50
Test Case No: 26 - 1st Examiner Score: 7.236, 2nd Examiner Score: 12.584,  Final result:
25.23 / 50
Test Case No: 27 - 1st Examiner Score: 10.522, 2nd Examiner Score: 2.198,  Final result:
16.63 / 50
Test Case No: 28 - 1st Examiner Score: 9.245, 2nd Examiner Score: 17.001,  Final result:
31.03 / 50
Test Case No: 29 - 1st Examiner Score: 18.169, 2nd Examiner Score: 13.235,  Final result:
35.82 / 50
Test Case No: 30 - 1st Examiner Score: 20.983, 2nd Examiner Score: 19.498,  Final result:
38.26 / 50
Test Case No: 31 - 1st Examiner Score: 14.446, 2nd Examiner Score: 13.869,  Final result:
29.30 / 50
Test Case No: 32 - 1st Examiner Score: 19.714, 2nd Examiner Score: 7.428,  Final result:
27.87 / 50
Test Case No: 33 - 1st Examiner Score: 18.206, 2nd Examiner Score: 15.219,  Final result:
35.57 / 50
Test Case No: 34 - 1st Examiner Score: 24.387, 2nd Examiner Score: 12.894,  Final result:
37.37 / 50
Test Case No: 35 - 1st Examiner Score: 24.668, 2nd Examiner Score: 4.011,  Final result:
31.60 / 50
Test Case No: 36 - 1st Examiner Score: 9.258, 2nd Examiner Score: 7.637,  Final result: 1
8.58 / 50
Test Case No: 37 - 1st Examiner Score: 10.731, 2nd Examiner Score: 16.788,  Final result:
33.15 / 50
Test Case No: 38 - 1st Examiner Score: 12.985, 2nd Examiner Score: 12.192,  Final result:
26.32 / 50
Test Case No: 39 - 1st Examiner Score: 15.732, 2nd Examiner Score: 6.643,  Final result:
26.33 / 50
Test Case No: 40 - 1st Examiner Score: 5.472, 2nd Examiner Score: 13.100,  Final result:
22.99 / 50
Test Case No: 41 - 1st Examiner Score: 4.291, 2nd Examiner Score: 3.148,  Final result: 1
1.03 / 50
Test Case No: 42 - 1st Examiner Score: 14.807, 2nd Examiner Score: 15.981,  Final result:
31.81 / 50
Test Case No: 43 - 1st Examiner Score: 16.074, 2nd Examiner Score: 22.922,  Final result:
39.48 / 50
Test Case No: 44 - 1st Examiner Score: 13.680, 2nd Examiner Score: 22.907,  Final result:
37.74 / 50
Test Case No: 45 - 1st Examiner Score: 12.577, 2nd Examiner Score: 24.164,  Final result:
36.88 / 50
Test Case No: 46 - 1st Examiner Score: 14.811, 2nd Examiner Score: 1.304,  Final result:
15.92 / 50
Test Case No: 47 - 1st Examiner Score: 7.595, 2nd Examiner Score: 10.440,  Final result:
20.50 / 50
Test Case No: 48 - 1st Examiner Score: 17.211, 2nd Examiner Score: 23.772,  Final result:
41.21 / 50
Test Case No: 49 - 1st Examiner Score: 8.037, 2nd Examiner Score: 9.864,  Final result: 1
9.55 / 50
Test Case No: 50 - 1st Examiner Score: 13.072, 2nd Examiner Score: 2.007,  Final result:
16.93 / 50
```

### Viewing the graph for the 50th simulation

In [ ]:

```
grade.view(sim=gradingSystem)
```




0.2
0.0
0
10
20
30
40
50
grade