# Text Mining of Disneyland reviews using various techniques


## Abstract:

In this paper, different topic modelling techniques such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Non-Negative Matrix Factorization (NMF), Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) are implemented on the reviews given by visitors to three Disneyland Branches- Hong Kong, Paris, California. The aim of the model is to form groups of similar patterns in the reviews. This study will help Disneyland to identify reviews based on particular features such as food, price, facilities, etc leading them to improve themselves. The behaviour of all the implemented techniques will be compared to choose the best suitable method.



**Keywords**: Topic modelling, Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA), Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM), Disneyland, reviews.


## Introduction:

Topic modelling techniques based on Latent Semantic analysis and Latent Dirichlet Allocation have been intensively used for a large number of applications over recent years [1]. Topic modelling is an unsupervised technique which identifies latent patterns of word occurrence using the distribution of words in a collection of documents[2]. This technique is generally used for automatically organizing, understanding, searching, and summarizing large corpus [3].

In this model, each document consists of multiple topics. The "topics" are the latent variable relations which are needed to be estimated; relations link words in a vocabulary and their occurrence in documents[3]. The outcome of the model is a cluster of words that occur in the documents according to certain pattern for each topic which facilitates in exploring the data on the basis of topics. This task has been made easy by the technique otherwise human beings have to scan large heaps of documents to analyse the text and identify relation between them consume large amount of time. There are humongous applications of topic modelling such as spam filtering, chatbots, topic tracking, summarization, etc.

There are many algorithms to apply topic modelling such as LDA, LSA, pLSA, NMF, etc. In this paper, four techniques will be implemeted- LDA, LSA, NMF, GSDMM. The textual data is generally unstructured and manual analysis of this data is not practically feasible. As a result, text mining has been used to analyse the unstructured data.

## Problem and Dataset:

The link to the dataset is: https://www.kaggle.com/arushchillar/disneyland-reviews

The dataset has 42,000 reviews of 3 Disneyland branches - Paris, California and Hong Kong. These reviews are posted by visitors on Trip Advisor. It has 6 columns: Review_ID is the unique identification number given to each review, Rating column shows the ratings given by customers. Its value ranges from 1 (bad review) to 5 (good review). 'Year_Month' column shows when the reviewer visited the theme park and Reviewer_Location signifies the country from which visitor belongs. Review_Text has the comments given by the visitor. Disneyland_Branch tells the branch name to which review belongs to.

The branches want to improve themselves depending upon the reviews received by the visitors. This can be achieved if they will be able to identify some major things about which review has been made. They need to analyse the reviews thoroghly for each facility provided by them.

Topic modelling is a very popular technique to analyse the text. It churns out collections of expressions and words that model thinks are related. This technique will be useful for this problem as the models will group similar word patterns to infer topics which will help company to analyse reviews based on particular feature. For eg: Company wants to analyse reviews about price, it will not have to search from a lot of feedbacks and spending long hours. Instead, reviews can be grouped by the modelling algorithms and company can go through the group of interest.

## Methods:

## Latent Dirichlet Allocation:

LDA is used for topic modelling. It works on two assumptions that every document consists of multiple topics and every topic consists of multiple words. It forms groups of the text by identifying latents meaning and relation between different words in a document. It builds topic per document model and words per topic model : P( topics | documents) and P( word | topics) using probabilistic approach. According to these models, LDA model forms clusters of words and each word is assigned based on these probability model values. LDA model uses distributional hypothesis which states that words having similar meaning should be present in a similar topic. To form clusters, LDA model assigns topic to each word or group of words formed by n-grams. It does not consider syntactic meaning of document. There can be overlapping in clusters formed by LDA model also as one word can be assigned more than one topic. In this context, it is quite different from traditional clustering where each item is assigned only one cluster. The demonstration of LDA is shown below in Fig1 where LDA model is forming clusters from documents depending upon number of topics given to model.

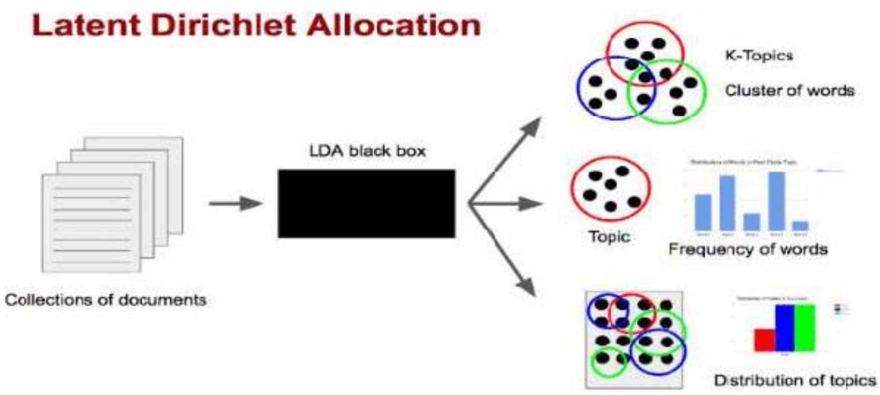


Fig1 LDA mechanism

(Source: https://www.researchgate.net/publication/331669603/figure/fig1/AS:735488403070977@1552365508288/The-flow-of-LDA-Retrieved-from-https-toolboxkuriocoid-topic-modeling-696d7ba2592f.png)

## Parameters of LDA:

LDA model is based on three parameters which are given below:

Number of topics: This parameter tells the model the number of topics among which documents can be categorised. It should be given by the user while implementing the model.

Alpha: It is a Dirichlet prior concentration parameter representing document-topic density. The large value of alpha shows that documents consist of more topics, hence, resulting in more specific topic distribution per document[4]. On the other hand, lower value of alpha shows documents consist of less number of topics.

Beta: It is a same prior concentration parameter representing topic-word density. The large value of beta shows topics consists of many words, hence, resulting in more specific word distribution per topic. The smaller value of beta shows there are less number of words in a topic.

## Latent Semantic Analysis:

LSA is another useful technique for topic modelling. This model accepts matrix of documents and terms and further decompose into two other matrices- document-topic matrix and topic-term matrix[6]. LSA is also based on distributional hypothesis similar to LDA. The key difference is that LSA computes word and document frequency using CountVectoriser or TF-IDF vectoriser. After the creation of document-term matrix, dimensionality reduction is done on the matrix known as SVD as the document-term matrix is very sparse, so that latent topics can be identified. Truncated SVD will reduce dimensionality of matrix as it selects only some largest singular values; the number of values to be selected is defined explicitly. The results of this step is a new latent semantic space containing only important values which can further be used to compute similiarities between words and documents. Though LSA is quick, but it has some cons such as it has less efiicient representation and it is generally used when dataset is really huge.

## Non-negative Matrix Factorization:

NMF is a statistical method which is useful for changing the high-dimensional vectors into lower dimensions of the input corpora. It is a Linear-algebraic model that is used to identify the latent structure of data in the document[7]. It gives less weightage to the words that are having less coherence. Its results are non-negative vectors. It takes term-document matrix as input and decompose it into two two matrices where first matrix is of the original n-words by k topics and other one is those same k topics by the m original documents[7].The NMF mechanism is shown below in Fig2.

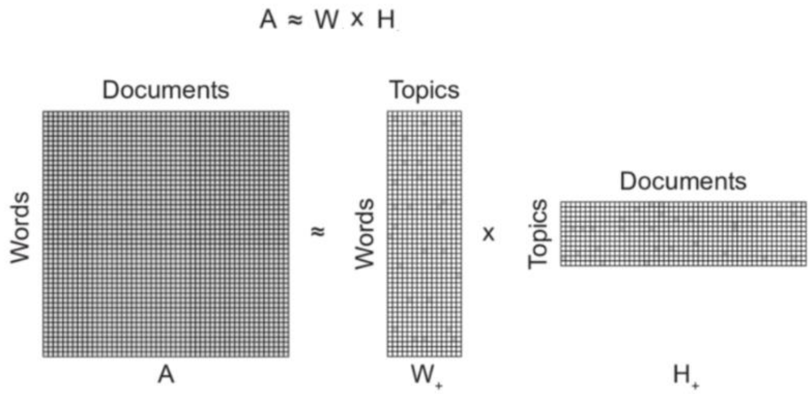



Fig2 NMF mechanism

(Source: https://lh4.googleusercontent.com/FvUGnhGjYVFUY-K1pZxIrgPJKElEKxdnaX1GfDg3suLoMY5KUzgXKRsqcs-pDso-NXGyGdkq9r6S81G2Nc5zRaEQtQ88OfBSgMzRBk3RiZTXANxTYFzTLeWEoEZTOhxPH1UYCwqd)

## Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM):

Reviews can be considered as short texts as sometimes customers do not give detailed feedbacks. So, it is highly possible that entire document may consist of single topic. In this, the traditional technique of LDA does not prove to be much efficient. Some other topic modelling techniques such as GSDMM can be used which is specially designed for short text clustering. This technique can also be considered as extended LDA algorithm but there are some key differences in both the approaches such as LDA is based on assumption that each document contains multiple topics and it calculates the contribution of each topic to the document[8]. On the contrary, GSDMM assumes only one topic per document, hence, suitable for micro-blogging content such as tweets, reviews, etc. Moreover, the number of topics are specified in the beginning for the LDA model whlist in GSDMM, the maximum bounds of model is given representing the maximum number of clusters model can form. It selects the number of topics itself. It accepts two input variables- preprocessed text in the numpy array form and the length of created dictionary corpus.

## Experimental Setup:

### Importing required libraries:

Initially all the libraries required for loading dataset, handling dataframe operations, preprocessing textual data, building models are imported. Some packages that are required for Natural Language processing and visualisation of model results are installed. The code is written in Google colab so that minimal packages should be installed as it has various built-in packages. Moreover, Google Colab provides GPU runtime which speeds up the execution process.

### Loading Dataset:

After importing all the necessary libraries, the dataset was loaded. The dataset has 42656 rows and 6 columns. The column of interest is 'Review_Text' containing reviews given by customers. It contains textual data in unstructured format which is very difficult to analyse manually. So, Text mining techniques are used.

**Preprocessing:**

The next step after loading the dataset is to clean the textual data so that it can be used for analysis. Initially, sentences are tokenized using gensim's simple_preprocess( ) function from which all the unicodes, mentions, double spaces are removed. Punctuations are removed by setting the option deacc=True. The preprocessed textual data is then converted to a list.

Afterwards, bigrams and trigrams are formed in which two and three words sequences are formed which is then passed to Phraser for speedy execution. Lemmatization is done using spacy module in which only nouns, adjectives, verbs, adverbs are preserved; stop words are also removed.

**Building Topic Models:**

In order to build the LDA model, initially dictionary containing preprocessed data and corpus containing Term Document frequency should be built. Initially, the model was provided 4 as number of topics to work upon.

The next model to be implemented was Non-Negative matrix factorisation which accepts inputs as tf-idf vectors in normalised form. Hence, tfidf vectors were created and normalised and passed to the model in order to extract topics from the text. Number of topics are given as 4.

The third model to be implemented was Latent Semantic Analysis (LSA). Some Exploratory data Analysis was done to get insights of the data such as most occurring words, tags, etc. From Fig3, it can be observed that park, disney, disneyland, etc are the most frequently occurring words in a document. There are total 5,603,254 words in a document and mean of number of words per review is 131.

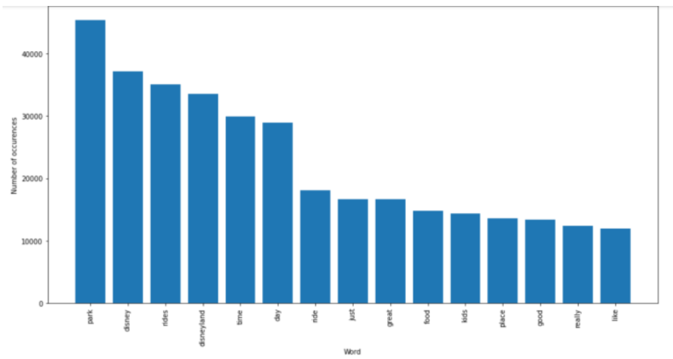



Fig3 Most frequently occurring words

Afterwards, Part of Speech was also analysed for the reviews. It can be observed from Fig4 that the most common part of speech tag is NN which represents singular noun. It can be disney, disneyland, park, ride, etc.

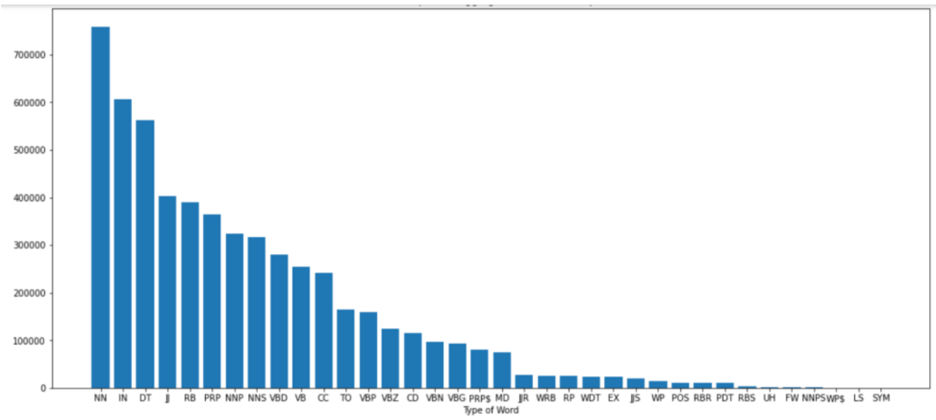



Fig4 Part of speech tags for reviews corpus

The next model to be built was Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM). It is an extended LDA algorithm which is especially designed for short length text clustering. The package can be installed using github link:

```
pip install git+https://github.com/rwalk/gsdmm.git
```

The selected value of hyperparameters are: number of topics are given as 4, alpha is 0.1, beta is 0.3 and number of iterations are 15.

Results of all the models can be found in results section and link to code is given in appendix.

# Results and Discussions:

## Latent Dirichlet Allocation (LDA):

The coherence score came out to be 0.3785 for LDA model. Results are compared better when they are visualised with figures. So, some visualisations are shown below. The figure below shows topics in top 20 documents.

| | Topic # 01 | Topic # 02 | Topic # 03 | Topic # 04 |
|---|---|---|---|---|
| 0 | ride | hotel | food | kid |
| 1 | queue | stay | eat | time |
| 2 | time | park | price | child |
| 3 | day | staff | restaurant | ride |
| 4 | park | people | bit | great |
| 5 | wait | experience | pay | character |
| 6 | hour | disney | drink | day |
| 7 | minute | studio | money | really |
| 8 | expensive | room | french | year |
| 9 | long | breakfast | area | place |
| 10 | ticket | last | cost | show |
| 11 | pass | leave | bad | love |
| 12 | fast | mountain | close | old |
| 13 | walk | theme | amazing | parade |
| 14 | meal | look | village | enjoy |
| 15 | way | little | cold | much |
| 16 | book | service | bring | visit |
| 17 | find | clean | high | family |

Fig5 Topics by LDA model

In LDA models, there are multiple topics in each document. However, there is only one topic which is quite dominant. The figure below shows the topic which is dominant along with weight and keywords of the topic.

| | Document_No | Dominant_Topic | Topic_Perc_Contrib | Keywords | Text |
|---|---|---|---|---|---|
| 0 | 0 | 4.0 | 0.7712 | ride, time, day, park, queue, great, kid, real... | If you've ever been to Disneyland anywhere you... |
| 1 | 1 | 1.0 | 0.6927 | hotel, stay, staff, disney, experience, people... | Its been a while since d last time we visit HK... |
| 2 | 2 | 4.0 | 0.8927 | ride, time, day, park, queue, great, kid, real... | Thanks God it wasn t too hot or too humid wh... |
| 3 | 3 | 4.0 | 0.5128 | ride, time, day, park, queue, great, kid, real... | HK Disneyland is a great compact park. Unfortu... |
| 4 | 4 | 4.0 | 0.9268 | ride, time, day, park, queue, great, kid, real... | the location is not in the city, took around 1... |
| 5 | 5 | 0.0 | 0.6931 | expensive, ticket, park, little, find, food, r... | Have been to Disney World, Disneyland Anaheim ... |
| 6 | 6 | 2.0 | 0.5000 | year, food, old, price, pay, room, drink, brea... | Great place! Your day will go by and you won't... |
| 7 | 7 | 0.0 | 0.6504 | expensive, ticket, park, little, find, food, r... | Think of it as an intro to Disney magic for th... |
| 8 | 8 | 4.0 | 0.5890 | ride, time, day, park, queue, great, kid, real... | Feel so let down with this place,the Disneylan... |
| 9 | 9 | 0.0 | 0.4791 | expensive, ticket, park, little, find, food, r... | I can go on talking about Disneyland. Whatever... |

Fig6 Dominant Topic in each document

In the above figure, Document_No shows the document, dominant_topic shows the topic which is dominant in each topic, topic_perc_contrib shows the percentage of topic contribution for the given document, keywords show the keywords selected in each document and text represents the actual text column – reviews given by customer. For eg: in first document, the dominant topic is 4 whose contribution is 77% to this document.

| | Topic_Num | Topic_Perc_Contrib | Keywords | Representative Text |
|---|---|---|---|---|
| 0 | 0.0 | 0.9576 | ride, queue, time, day, park, wait, hour, minute, expensive, long | Went there on a public holiday but the crowds weren't too big. Longest wait for a ride was prob... |
| 1 | 1.0 | 0.9307 | hotel, stay, park, staff, people, experience, disney, studio, room, breakfast | Disneyland may not be the biggest park or even the best park, but it is where Walt's dream began... |
| 2 | 2.0 | 0.8741 | food, eat, price, restaurant, bit, pay, drink, money, french, area | the best every to go to the best every the best every one shold go there it the best every the b... |
| 3 | 3.0 | 0.9752 | kid, time, child, ride, great, character, day, really, year, place | Smallest of all Disneylands across the world, this one is worth spending time if you are travell... |

Fig7 Displaying most relevant document for each topic

The above figure shows the dominant sentence for each topic. The word clouds are also shown in fig8 for each topic where most occurring word for each topic is displayed.


Fig8 Word Cloud for each topic by LDA model

From topic0, it can be assumed that reviews are quite negative in that showing people had to wait in queues, long waiting-hours; some people find park expensive, etc. Topic1 might contain information about experience visitors had with staff, hotel room stay, etc. Topic 2 might contain reviews about food whilst topic3 can contain positive experience of visitors.

Fig 9 explains the relation between number of documents and dominant topic and topic weightage. It can be observed that Topic3 contains the maximum number of documents and it is dominant in large number of documents whilst topic 2 contains the least number of documents and is dominant in less number of documents.

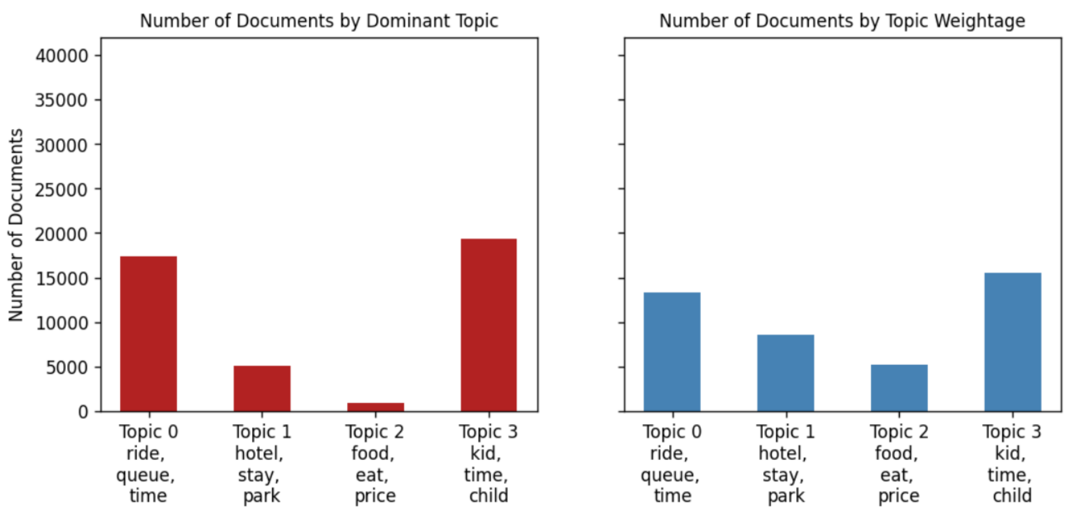



Fig9 Number of documents by (a)Dominant Topic (b) Topic weightage

The results of LDA can be visualised using two clustering techniques: T-SNE clustering and PyLDAvis. Fig 10 shows the results of PyLDAvis which allows to interpret the topics in a topic model. Each bubble represents topic.

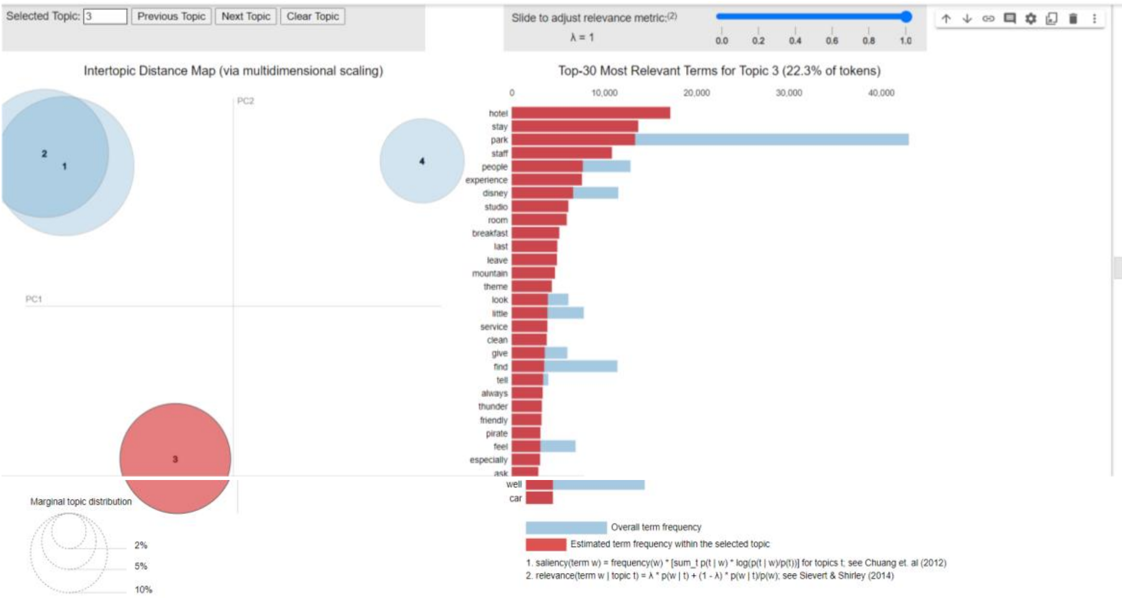



Fig10 Results of PyLDAvis notebook

In this figure, the overall frequency of each word in the corpus is represented by blue bars represent the overall frequency of each word in the corpus. Red bars give the estimated number of times a given term was generated by a given topic. The word with the longest red bar is the word that is used the most by the tweets belonging to that topic[9]. From fig10, it is clear that term hotel has been used maximum times in reviews within topic 3. There is overlapping in clusters 1 and 2. All other clusters are well spaced. Similar pattern can be observed in T-SNE clustering results as well which is shown in Fig11.

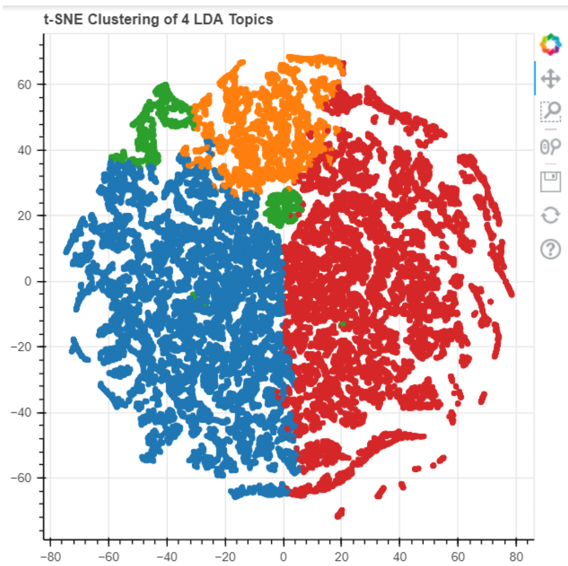



Fig11 T-SNE clustering

## Non-Negative matrix Factorisation:

| | Topic # 01 | Topic # 02 | Topic # 03 | Topic # 04 |
|---|---|---|---|---|
| 0 | ride | place | love | great |
| 1 | day | kid | always | time |
| 2 | park | happy | year | always |
| 3 | time | earth | back | fun |
| 4 | visit | visit | old | family |
| 5 | wait | family | kid | food |
| 6 | long | enjoy | absolutely | crowd |
| 7 | pass | magical | amazing | age |
| 8 | small | adult | disneyland | ride |
| 9 | queue | must | much | day |

Fig12 Topics made by NMF model

Fig 12 represents the topics formed by NMF model. The results of the NMF model can not be visualised. Hence, only topics for top 10 documents are displayed in Fig 10. The results are not satisfactory as relation between words in a topic is not identifiable as all words in all topics show enjoyment; positive experience. No different meaning can not be predicted.

## Latent Semantic Analysis (LSA):

The top 5 words in each topic formed by LSA model is displayed in Fig13. It can be seen that words like 'disney', 'disneyland', 'rides' are coming in more than one topics. This is the difference between Clustering and Topic modelling that one word can be assigned more than one cluster or topic in topic modelling.

```
Top 5 words in topic 1 :  ['park', 'disney', 'disneyland', 'rides', 'day']
Top 5 words in topic 2 :  ['disney', 'paris', 'hotel', 'staff', 'room']
Top 5 words in topic 3 :  ['time', 'disneyland', 'birthday', 'son', 'went']
Top 5 words in topic 4 :  ['disney', 'time', 'ride', 'mountain', 'rides']
```

Fig13 Topics made by LSA model

Each topic can be viewed in detail using word clouds. Hence, wordclouds for all 4 topics are shown in Fig14.

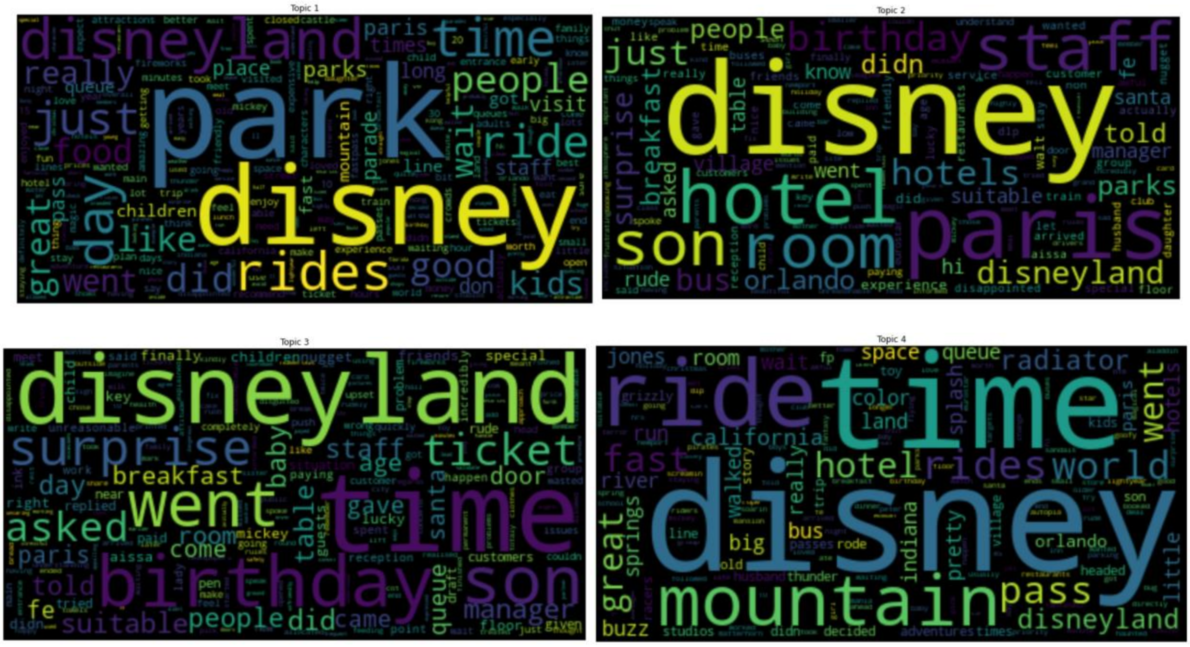



Fig14 Word clouds for each topic formed by LSA model

The results of topics can also be visualised using T-SNE clustering. The main aim of drawing t-sne clustering plot is to compare the results of both LSA and LDA technique.

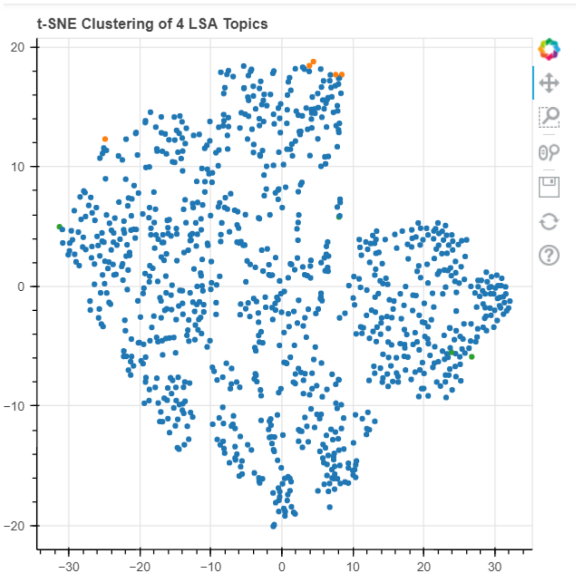



Fig15 T-SNE clustering for LSA topics

The results of T-sne clustering for LSA is highly unexpected. It considers one topic as highly dominant and negligible words in other topics. There is high degree of overlapping in clusters formed which shows the failure of model. The results of LDA model is better than LSA model.

## Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM):

The last model specifically implemented for short-text clustering has made 4 topics. The results are shown in Fig16. It can be observed that Topic0 is the most dominant topic as it contains the maximum number of topics: 16095. Some words of each topic are also shown in Fig16.

```
Number of documents per topic : [16095  6784 11406  8371]
Most important clusters (by number of docs inside): [0 2 3 1]

Cluster 0 : [('ride', 10809), ('time', 8027), ('park', 7291), ('day', 6857), ('place', 6090), ('visit', 5948), ('great', 5518), ('love', 5364), ('kid', 5335), ('show', 3798), ('

Cluster 2 : [('ride', 14477), ('time', 10605), ('day', 10602), ('park', 8698), ('pass', 5802), ('wait', 5452), ('fast', 4534), ('visit', 4136), ('great', 4068), ('long', 3972),

Cluster 3 : [('ride', 19337), ('park', 14895), ('time', 13939), ('day', 11843), ('queue', 7106), ('wait', 6056), ('food', 5967), ('people', 5558), ('show', 5232), ('kid', 5207),

Cluster 1 : [('ride', 7455), ('park', 5921), ('time', 4513), ('day', 3574), ('people', 2703), ('visit', 2611), ('food', 2562), ('queue', 2433), ('wait', 2278), ('staff', 2106),
```

Fig16 Topics formed by GSDMM model

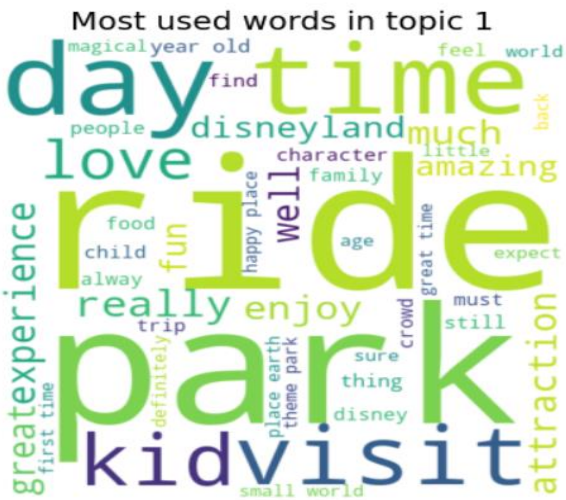
The topic for top10 and last 10 documents can be seen from Fig17 where Text shows the original reviews received by Disneyland. It can be observed from figures topic0 (type1) is the most dominant topic.

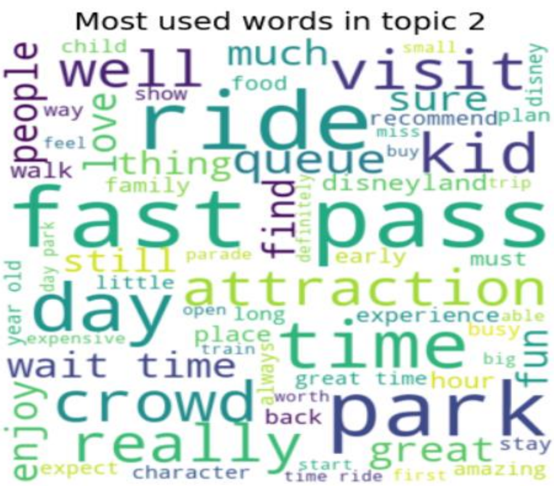
| | Text | Topic | Lemma-text |
|---|---|---|---|
| 0 | If you've ever been to Disneyland anywhere you'll find Disneyland Hong Kong very similar in the ... | type 1 | [ever, anywhere, find, similar, walk, main_street, familiar, feel, ride, small, world, absolutel... |
| 1 | Its been a while since d last time we visit HK Disneyland .. Yet, this time we only stay in Tomo... | type 1 | [last, time, visit, time, stay, aka, marvel, land, experience, newly, open, great, feature, exci... |
| 2 | Thanks God it wasn t too hot or too humid when I was visiting the park otherwise it would be... | type 2 | [hot, visit, park, otherwise, big, issue, shade, arrive, around, left, pm, unfortunately, last, ... |
| 3 | HK Disneyland is a great compact park. Unfortunately there is quite a bit of maintenance work go... | type 4 | [park, unfortunately, quite, bit, maintenance, work, present, number, area, close, include, famo... |
| 4 | the location is not in the city, took around 1 hour from Kowlon, my kids like disneyland so much... | type 1 | [hour, kowlon, kid, disneyland, much, fine, really, crowd, hot, hong_kong] |
| 5 | Have been to Disney World, Disneyland Anaheim and Tokyo Disneyland but I feel that Disneyland Ho... | type 4 | [really, small, call, souvenir, food, entrance, ticket, slightly, expensive, basically, park, sm... |
| 6 | Great place! Your day will go by and you won't even know it. Obviously went there for my daughte... | type 1 | [obviously, daughter, absolutely, love, bad, parade, cancel] |
| 7 | Think of it as an intro to Disney magic for the little ones. Almost all of the attractions can b... | type 2 | [little, one, almost, attraction, complete, day, drawback, timing, example, disney, close, thu, ... |
| 8 | Feel so let down with this place,the Disneyland train was fantastic until you get past the stati... | type 4 | [feel, let, place, disneyland, train, fantastic, past, station, bad, signage, terrible, staff, r... |
| 9 | I can go on talking about Disneyland. Whatever I say about it, is less. Disneyland is all about ... | type 2 | [talk, less, childhood, dream, true, start, entrance, environment, popcorn, show, mtr, train, st... |

| | Text | Topic | Lemma-text |
|---|---|---|---|
| 42646 | It's my 4th. visit to disneyland paris and all the times I had rain.A point to improve is langua... | type 3 | [rain, point, improve, languaje, handling, kid, french, show, great, part, interest, specially, ... |
| 42647 | I would not recommend going if you have been to other Disney resorts. This one falls short and o... | type 1 | [resort, fall, short, child, terribly, disappoint, shopping, store, ride, true, pitty, looking_f... |
| 42648 | I've been to Disney in California 5 times. I've been to Disney in Florida 4 times. My best Disne... | type 2 | [time, trip, vacation, pleasant, mid, trip, break, museum, history, wrong, love, museum, history... |
| 42649 | Disneyland Park Paris is the rolls royce of theme parks. The park blatantly cost a fortune to bu... | type 1 | [theme, park, blatantly, cost, fortune, build, detail, everywhere, themed, man, hole, cover, wel... |
| 42650 | Did the kids free trip last week in March no crowds great weather. Stayed at Hotel Cheyenne grea... | type 3 | [kid, free, trip, last, week, crowd, stay, hotel, cheyenne, great, value, feel, magic, bigtime, ... |
| 42651 | i went to disneyland paris in july 03 and thought it was brilliant. i visited all the hotels and... | type 3 | [visit, stay, really, hotel, park, big, hotel, kyriad, away, still, really, hotel, saw, room, ho... |
| 42652 | 2 adults and 1 child of 11 visited Disneyland Paris beginning of Feb 04 and had an absolute fant... | type 2 | [visit, absolute, fantastic, time, night, queue, people, arrive, stay, really, drink, hotel, pre... |
| 42653 | My eleven year old daughter and myself went to visit my son in London and we decided to go on to... | type 2 | [year, old, daughter, visit, son, decide, stay, night, day, du, great, value, money, close, high... |
| 42654 | This hotel, part of the Disneyland Paris complex, is a wonderful place for families. Since we ha... | type 3 | [complex, wonderful, place, family, kid, room, crew, area, consist, separate, toilet, area, show... |
| 42655 | I went to the Disneyparis resort, in 1996, with a small child. The minute you enter, the transpo... | type 3 | [resort, small, child, minute, enter, transport, treat, special, guest, compnay, really, treat, ... |

Fig17 Topics for documents by GSDMM

Word clouds for each topic is shown below in Fig18.



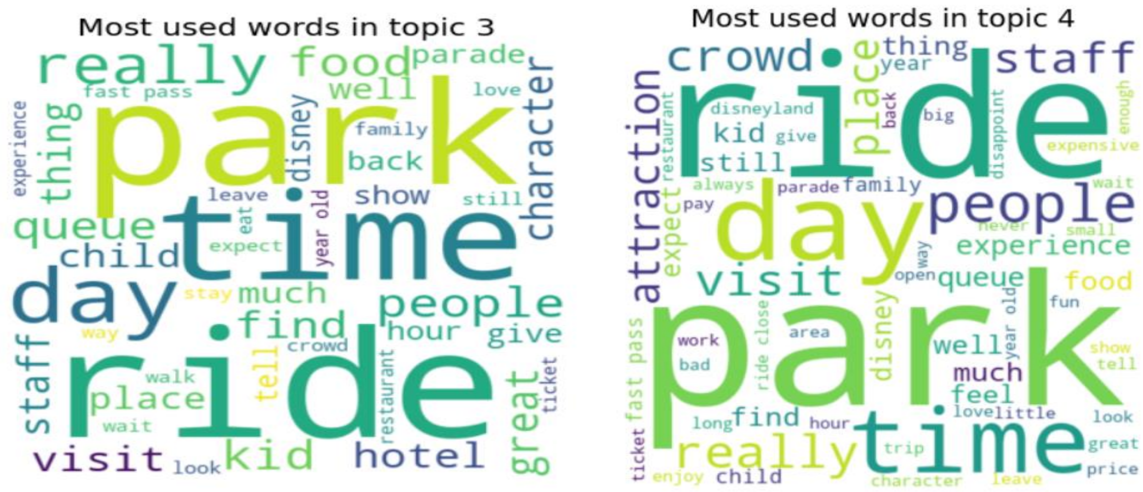



Fig18 Wordclouds for GSDMM model

## Social, ethical, legal and professional considerations:

The dataset use for topic modelling was available freely on Kaggle. No personal data of any person was present on dataset. Moreover, attributes like country of visitor, userid, etc have not been used. So, all social, ethical, legal rules were followed while implementing the project.

## Conclusions:

After analysing all the results of all the 4 models, it can be concluded that LDA outperformed all the models as the topics which were made by the model has words which have same lemantic meaning; they show relation with each other whilst no such major pattern can be observed in the results of all other models. GSDMM also provided detailed results. Some of the pattern can be observed but not very precise as there is large extent of overlapping between topics which is clearly visible from keywords for each topic as well as word clouds. The results of LSA can be achieved better if there is large sets of data. However, all the techniques were suitable for topic modelling to get estimate about what the unstructured textual data contains.

# Evolutionary and Fuzzy Systems

**Bharti Tayal** **Gurpreet Kaur**

**Sid-10895366** **Sid-11439853**

**tayalb@coventry.ac.uk** **gurpree18@uni.coventry.ac.uk**

## Part-1: Design and Implementation of the FLC

In this task, the aim is to design a Smart home for disabled residence. It can be easily automated using Fuzzy Logic Controller. MATLAB Fuzzy Logic Toolbox can be used to design a FLC. Fuzzy logic produces output based on assumptions. It takes input from the sensors which is then passed to Fuzzy logic inference where some rules are designed in order to predict output. It works on the principle of if-else-then, i.e. if A and B, then C. Output is produced after defuzzification.

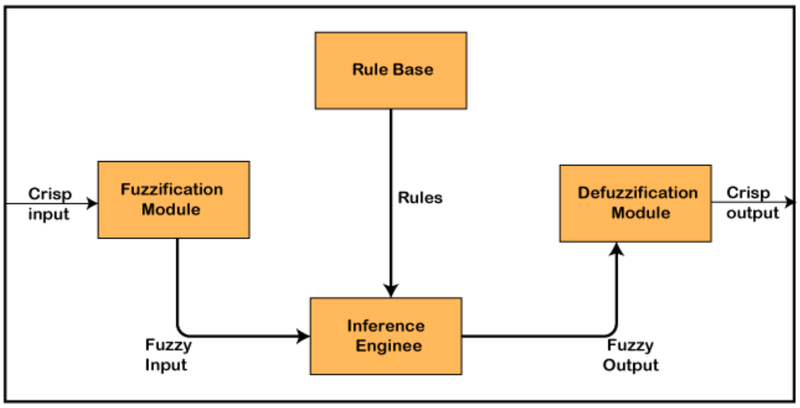



Fig19 Structure of Fuzzy Logic Controller

(Source: https://www.section.io/engineering-education/an-overview-of-fuzzy-logic-system/architecture-of-fuzzy-logic-system.png)

## Working of FLC:

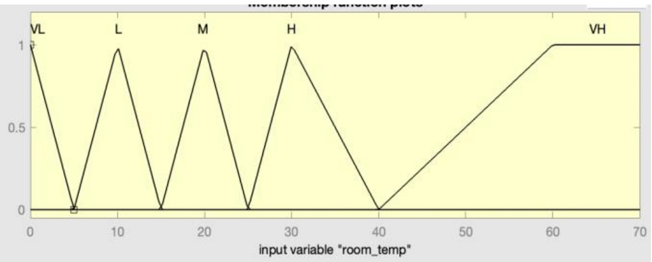
The above picture shows the basic components of FLC. Initially some inputs are provided along with their membership functions. The next step is to pass these inputs to the Inference Engine where rules are implemented so that fuzzy inputs can be converted to Fuzzy Output. Afterwards, this fuzzy output is further defuzzied based on membership functions of output and crisp output is obtained. These components are explained below:

## A. Input and its membership functions:

There are eight major inputs considered in this task to implement Assistive Smart Home which are: Room Temperature, Light level, Time of Day, $CO_2$ concentration, dirt, Gate Sensor, Surveillance, Groceries.

### (i) Room Temperature:

The ambient temperature in houses makes everyone feel comfortable. It should be neither too low or too high. The room temperature is measured in Celsius and can be controlled using Cooling fans and heaters/boilers. The membership function chosen is Triangular. The values taken for room_temp are VL, L, M, H, VH. The range for input variable room_temp is -10°C to 100°C.

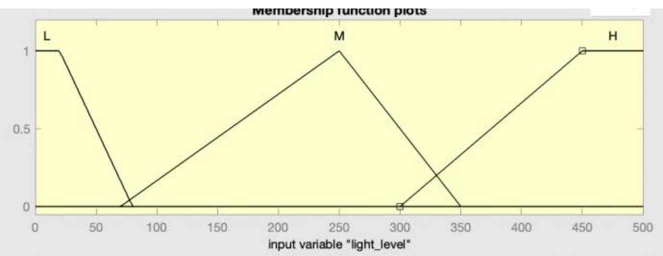



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | VL | -10°C to 5°C | Trapezoidal |
| 2 | L | 5°C to 15°C | Triangular |
| 3 | M | 15°C to 25°C | Triangular |
| 4 | H | 25°C to 40°C | Triangular |
| 5 | VH | 40°C to 100°C | Trapezoidal |

Fig20 Room Temperature Membership Function

### (ii) Light Level:

Light level is measured in Lux. Its value ranges from 0 to 500 LUX. Ideally, the house should be well lit so that tasks can be done easily. Light level can be managed by dimmers and blinds. For the sake of less output variables, here, light level is being managed by blinds only. The light level can be low, medium and high.

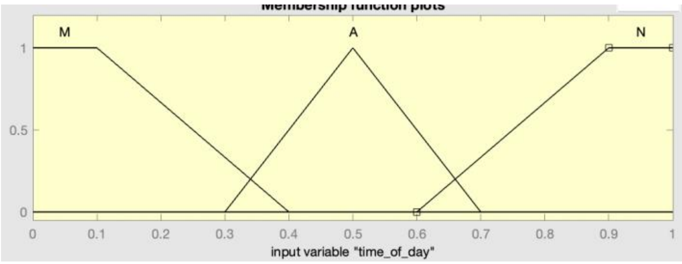



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | L | 0 to 80 LUX | Trapezoidal |
| 2 | M | 70 to 350 LUX | Triangular |
| 3 | H | 300 to 600 LUX | Trapezoidal |

Fig21 Light level Membership Function

### (iii) Time of Day:

Time of day plays crucial role in deciding if the blinds should be opened or not in order to increase light level in house. So, It is taken into consideration along with light level. It can take three values: Morning, Afternoon, Night.

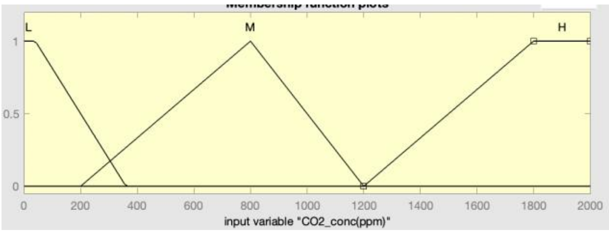



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | M | 0 to 0.4 | Trapezoidal |
| 2 | A | 0.3 to 0.7 | Triangular |
| 3 | N | 0.6 to 2 | Trapezoidal |

Fig22 Time_of_day Membership Function

### (iv) $CO_2$ concentration:

The air quality should always be good for any house which means Carbon dioxide concentration should always be maintained. Its range is 0 to 2700 ppm. The $CO_2$ concentration can be managed by opening windows.

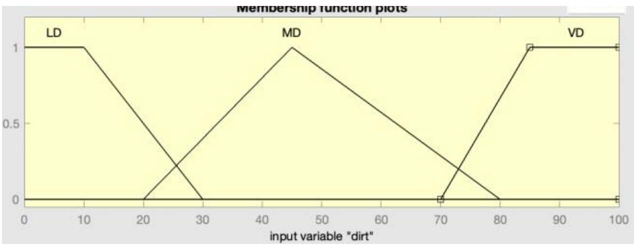



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | L | 0 to 360 ppm | Trapezoidal |
| 2 | M | 200 to 1200 ppm | Triangular |
| 3 | H | 1200 to 2700 ppm | Trapezoidal |

Fig23 $CO_2$ conc Membership Function

### (v) Degree of dirt:

Cleaning robots are nowadays used to clean the house automatically. The degree of dirt can be passed to it. Depending upon the degree of dirt, it can get the output time for which it should be operated. Its membership functions are: Less Dirty, Medium Dirty, Very Dirty.

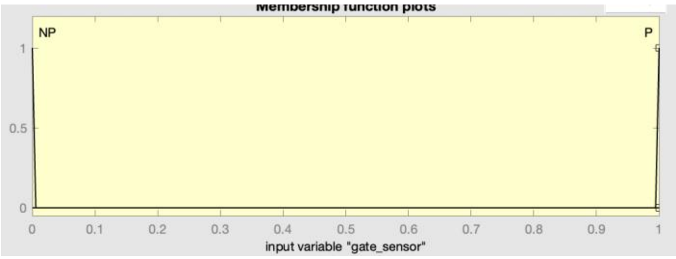



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | LD | 0 to 30% | Trapezoidal |
| 2 | MD | 20 to 80% | Triangular |
| 3 | VD | 70 to 100% | Trapezoidal |

Fig24 Dirt Membership Function

### (vi) Gate Sensor:

Gate Sensor will detect if there is anyone on door. Its value can be either True or False. NP shows it does not detect any person which can be signified by value 0 and P represents there is person on door, shown as 1.

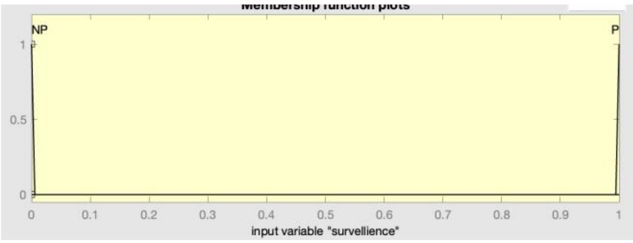



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | NP | 0 | Trapezoidal |
| 2 | P | 1 | Trapezoidal |

Fig25 Gate_Sensor Membership Function

### (vii) Surveillance:

Surveillance will show if the person is known or not for whom door should be opened. So, Surveillance is also like Boolean variable which can be No person or Person.

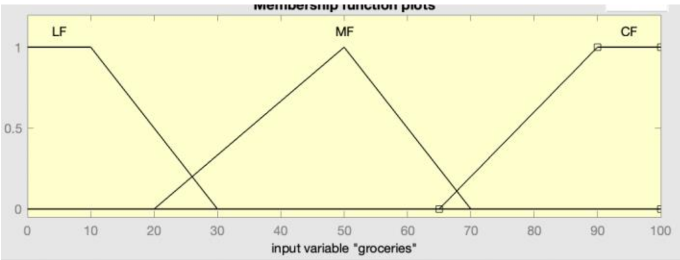



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | NP | 0 | Trapezoidal |
| 2 | P | 1 | Trapezoidal |

Fig26 Surveillance Membership Function

### (viii) Groceries:

It will tell the level of groceries in refrigerator so that cooling in refrigerator can be controlled.




| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | LF | 0 to 30% | Trapezoidal |
| 2 | MF | 20 to 70% | Triangular |
| 3 | CF | 65 to 100% | Trapezoidal |

Fig27 Groceries Membership Function

### B. Fuzzy Inference System:

In this step, the input is mapped to an output using fuzzy logic. This step involves all the information such as fuzzy logic operators, membership functions, if-then rules. There are mainly two types of Fuzzy inference system: Mamdani-type and Sugeno-type. The methods of calculating output are different from each other. Here, Mamdani Inference System is used in which output of each rule is a fuzzy set. This fuzzy set can be obtained using output membership function and the implication method which are then combined into single set in Aggregation step. The final crisp output value is obtained after defuzzification of aggregated fuzzy sets.

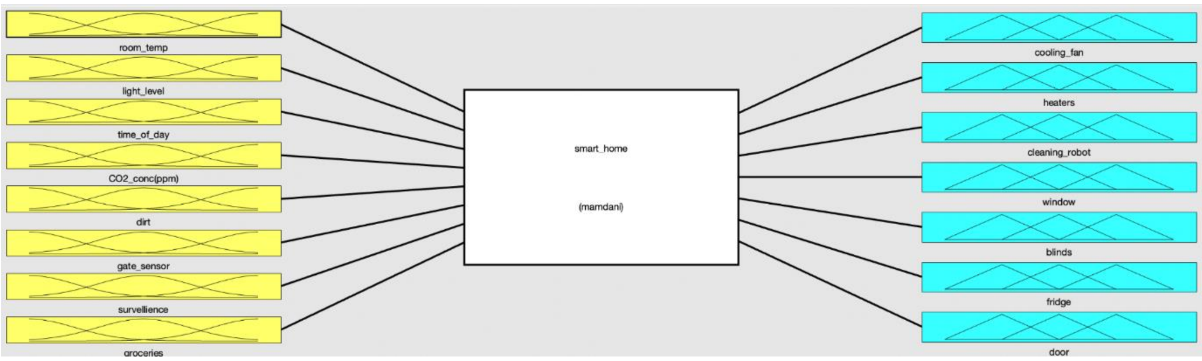


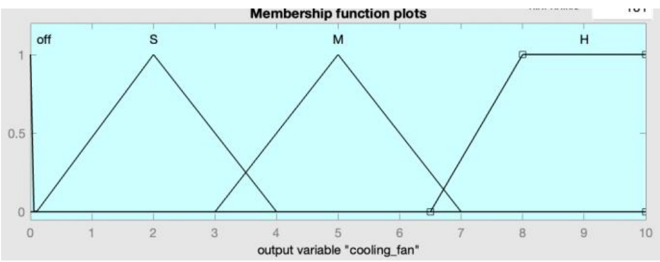

Fig28 Mamdani Inference GUI

## C. Output and Membership Functions:

Depending upon the input received, seven outputs were designed along with their membership functions.

### (i) Cooling Fan:

It is used to cool down the room temperature if it is too high in order to achieve ambient temperature. Its value ranges from 0 to 10.

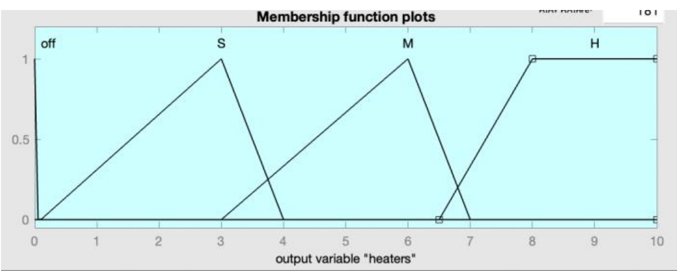



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | Off | 0 | Trapezoidal |
| 2 | S | 0.1 to 4 | Triangular |
| 3 | M | 3 to 7 | Triangular |
| 4 | H | 6.5 to 10 | Trapezoidal |

Fig29 Cooling fan Output Membership Function

### (ii) Heaters:

It is used to make the room temperature warm if it is too high in order to achieve ambient temperature. Its value ranges from 0 to 10.

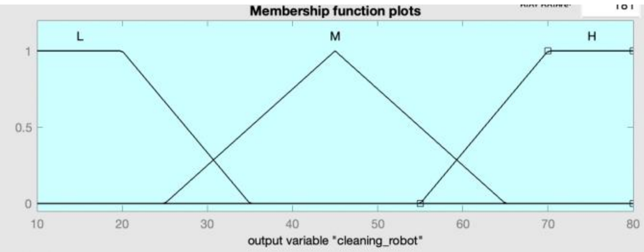



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | Off | 0 | Trapezoidal |
| 2 | S | 0.1 to 4 | Triangular |
| 3 | M | 3 to 7 | Triangular |
| 4 | H | 6.5 to 10 | Trapezoidal |

Fig30 Heaters Output Membership Function

### (iii) Cleaning robot:

It is used to clean the house by taking the amount of dirt in the house. It value lies between 10 min to 80 min.

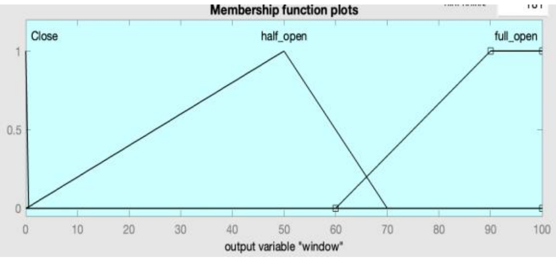



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | L | 10 min to 35 min | Trapezoidal |
| 2 | M | 25 min to 65 min | Triangular |
| 3 | H | 55 min to 80 min | Trapezoidal |

Fig31 Cleaning Robot Output Membership Function

**(iv) Window:**

This output variable tells the extent upto which window can be opened for air ventilation so that air quality can be good inside the home.

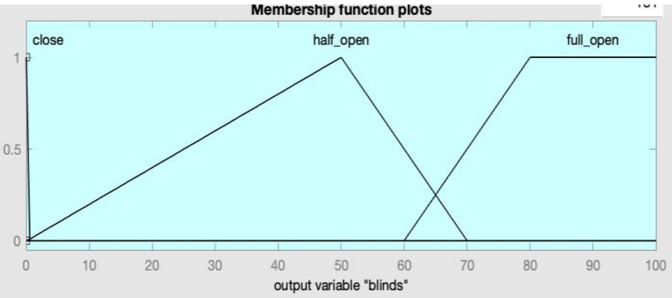



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | Close | 0% | Trapezoidal |
| 2 | Half_open | 0.1 to 70% | Triangular |
| 3 | Full_open | 60 to 100% | Trapezoidal |

Fig32 Window Output Membership Function

**(v) Blinds:**

This will tell the level of open of Blinds so that house can be well lit depending upon the time of day and light level in the house.

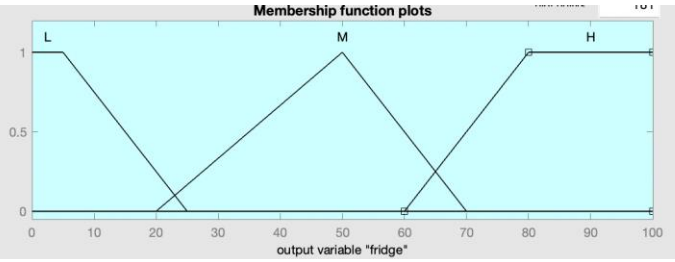



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | Close | 0% | Trapezoidal |
| 2 | Half_open | 0.1 to 70% | Triangular |
| 3 | Full_open | 60 to 100% | Trapezoidal |

Fig33 Blinds Output Membership Function

**(vi) Fridge:**

It tells the level of cooling in fridge. If there are large groceries in fridge, it should produce more cooling and less if there are less groceries.

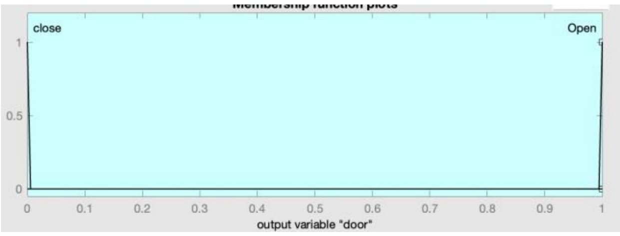



| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | L | 0 to 25% | Trapezoidal |
| 2 | M | 20 to 70% | Triangular |
| 3 | H | 60 to 100% | Trapezoidal |

Fig34 Fridge Output Membership Function

### (vii) Door:

The result of this output is that door should be opened or closed. It determines the result by considering surveillance and gate sensor inputs.




| Sr No. | Membership Function | Range | Type |
|---|---|---|---|
| 1 | Close | 0 | Trapezoidal |
| 2 | Open | 1 | Trapezoidal |

Fig35 Door Output Membership Function

## D. Fuzzy rules:

Fuzzy rules play crucial role in order to predict output from input. These rules are responsible for the efficiency of fuzzy logic controller. Some expertise knowledge is required about the system to make it efficient such as relation between inputs and outputs. Multiple variables can be combined using fuzzy rules. For this system, rules are designed with basic knowledge of smart homes. Fuzzy rules are of two types: t-norm which use AND operator to join variables whilst t-conorms use OR operator for joining variables. Here, rules are designed using t-norm.

| Room Temprature | VL | L | M | H | VH |
|---|---|---|---|---|---|
| Cooling Fan | Off | Off | S | M | H |
| Heater | H | M | Off | Off | Off |

Table: Input Room Temperature vs Cooling Fan and Heater as Output

| Light Level / Time of Day | L | M | H |
|---|---|---|---|
| M | Full Open | Half Open | Close |
| A | Full Open | Half Open | Close |
| N | Close | Close | Close |

Table: Input Light Level and Time of Day vs Blinds as Output

| CO2 Conc. (ppm) | L | M | H |
|---|---|---|---|
| Window | Close | Half Open | Full Open |

Table: Input $CO_2$ conc vs Window as Output

| Dirt | LD | MD | VD |
|---|---|---|---|
| Cleaning Robot | L | M | H |

Table: Input Dirt vs Cleaning Robot as Output

| Groceries | LF | MF | CF |
|---|---|---|---|
| Fridge | L | M | H |

Table: Input Groceries vs Fridge as Output

| Gate Sensor / Surveillance | NP | P |
|---|---|---|
| NP | Close | Close |
| P | Close | Open |

Table: Input Gate Sensor and Surveillance vs Door as Output

## E. Aggregation:

In this step, fuzzy sets representing the outputs of each rule are joined to form a single fuzzy set. It occurs only once for each output variable. It accepts list of truncated output functions for each rule and produces one fuzzy set for each output variable. It always accepts maximum value if it gets membership function at two different values.

### F. Defuzzification:

The aggregated output fuzzy set is defuzzied resulting into a single number. There are 5 built-in defuzzification methods: centroid, bisector, middle of maximum, smallest of minimum, and largest of minimum. Centroid calculation is the most popular defuzzification method. In this, center of the area under the aggregate fuzzy set is returned. This method is also used in this FLC. It can be calculated as:

$$\text{xCentroid} = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)}$$

Where $\mu(x_i)$ is the membership value for point $x_i$ in the universe of discourse.

After this step, rules can be viewed as:

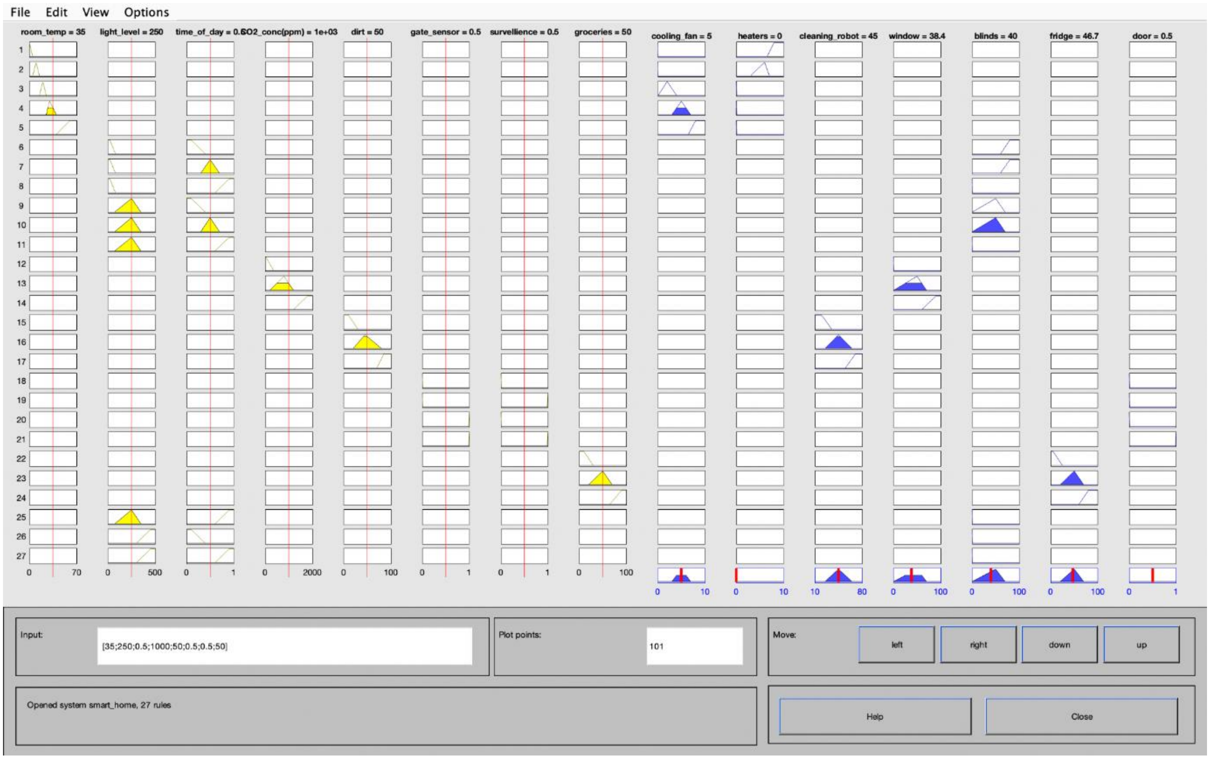



Fig37: Fuzzy Rules

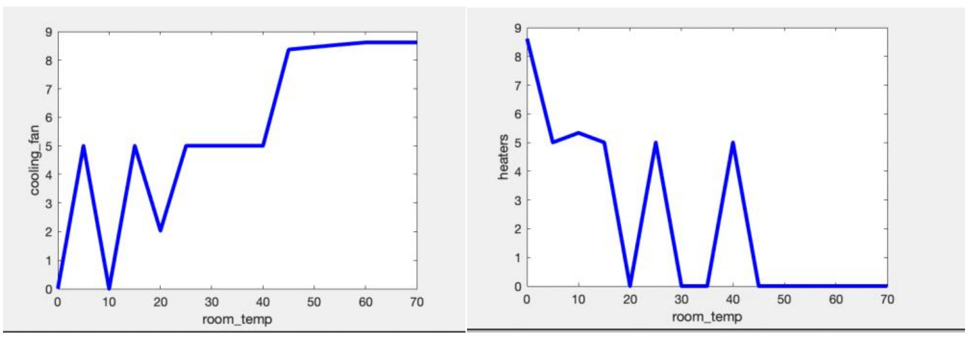



Fig38: Surface Graph of Input Room Temperature vs (a) Cooling Fan and (b) Heater as Output

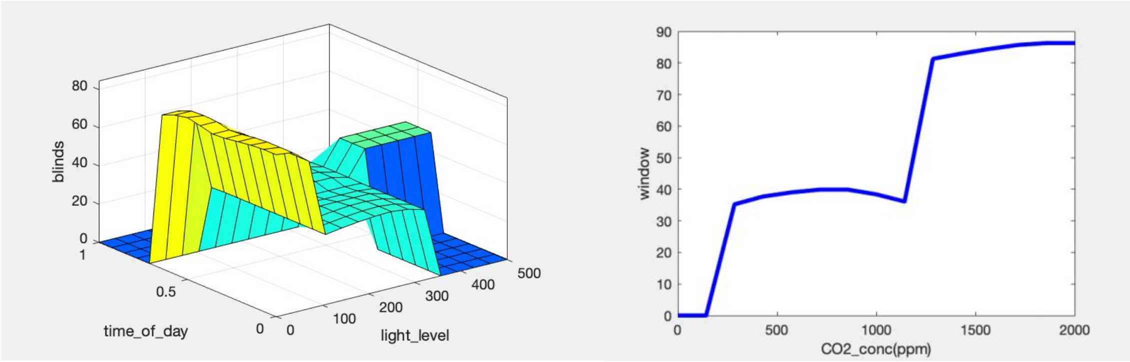



Fig39: Surface Graph of Input (a) Time of Day and Light_level vs Blinds (b) $CO_2$ conc vs window as Output

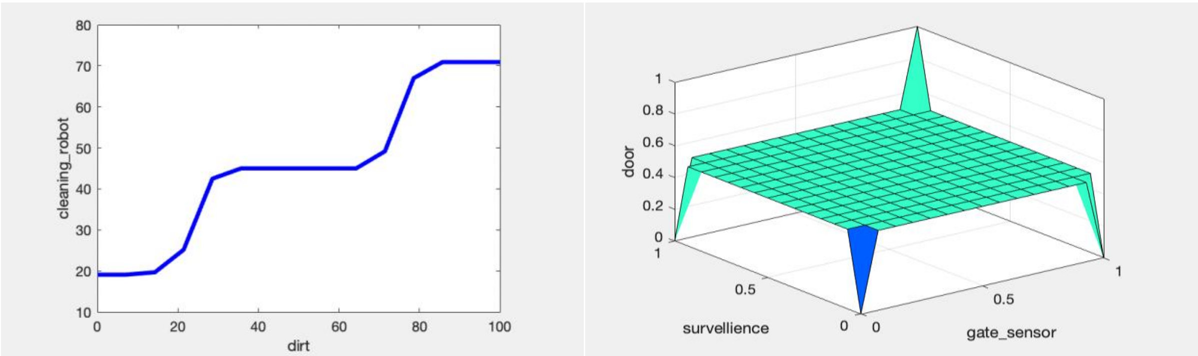



Fig40: Surface Graph of Input (a) Dirt vs Cleaning Robot (b) GateSensor and Surveillance vs Door as Output

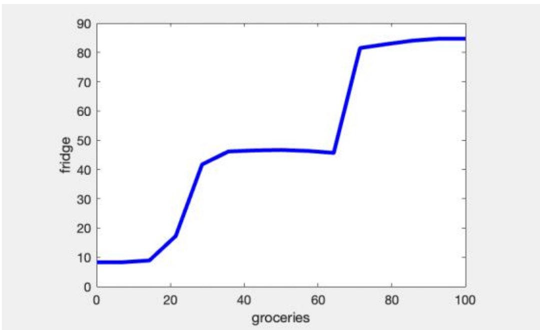



Fig41: Surface Graph of Input Groceries vs fridge as output

## Part-2: Optimizing the FLC developed for Part 1:

To optimize the FLC, there are various algorithms present such as: Genetic Algorithms, Swarm Intelligence Algorithms, etc. Here, genetic algorithm is used to optimize the FLC. The steps of execution for Genetic Algorithms are:

### (a) Initial Population:

Genetic algorithms can be applied to FLC if each membership functions' parameters are in genes so that entire FLC can be considered as Chromosome. These genes are represented in binary format where each bit represents a gene.

The FLC developed has two types of membership functions: Trapezoidal and Triangular, each having four and three parameters for optimization respectively.

For input membership functions, there will be total [(4+3+3+3+4) + (4+3+4) + (4+3+4) + (4+3+4) + (4+3+4) + (4+3+4) + (4+4) + (4+4) + (4+3+4)] = 88 chromosomes in the population.

For output membership functions, there will be total [(4+3+3+4) + (4+3+3+4) + (4+3+4) + (4+3+4) + (4+3+4) + (4+3+4) + (4+4)] = 80 chromosomes in the population.

**(b) Fitness Function:**

Fitness function is used to find the fitness of the individual so that it can be selected for reproduction. It accepts input as each chromosome and assigns fitness score to it.

**(c) Selection:**

Depending upon the fitness score, two chromosomes will be selected for mating. Selection can be done using various methods such as: Uniform method, Stochastic Uniform method, Roulette Wheel, etc. Roulette Wheel is the most popular method used for selection due to its efficiency.

**(d) Crossover:**

Crossover is the step in which parts of 2 selected chromosomes are exchanged with some crossover rate. This results in 2 new solutions. Crossover rate can be selected randomly, generally between 0.5 and 1. There is not any rule defined to select this value; mostly 0.7 produces good results. It is very crucial method as in this step, offspring of the chromosomes are established.

**(e) Mutation:**

In this step, any gene in a chromosome is selected and changed randomly. Mutation rate is generally kept low typically around 0.001- 0.01 of total number of genes. Sometimes, mutation might not take place due to low mutation rate.

**(f) Stopping Criterion:**

The last step is to check if number of chromosomes are exceeded or not. If there is no change in offspring , the steps are repeated. These steps are repeated until stopping criteria is satisfied which is defined explicitly. If optimisation results are not satisfactory, algorithm is started again.

## Difference between Sugeno and Mamdani:

Both the inference methods are somewhat similar. The initial two steps are same for both- fuzzifying inputs and applying the fuzzy operator.

The difference between them is in their outputs. The output received in designed FLC using mamdani is **fuzzy set** which is intuitive and is well-suited to human inputs. On the other hand, the output of Sugeno-type fuzzy is **linear or constant** which is computationally efficient.

A fuzzy rule used in first-order Sugeno model has the form:

**If x is A and y is B, then Z= px + qy + r**

A ,B - fuzzy sets in the antecedent and p,q, r are constants.

If Sugeno has been used instead of Mamdani, the membership functions of all the variables will result in linear or constant values instead of sets. Due to its linear dependency, it is well suited for linear techniques such as PID control; optimisation and adaptive techniques, mathematical analysis, etc.

Sugeno uses sum aggregation method whilst FLC developed using Mamdani used max aggregation method.

**Part-3: Comparison of different optimization techniques on CEC’2005 functions**

There are many optimisation algorithms available such as: Genetic Algorithms, Particle Swarm Optimisation Algorithms, Simulated Annealing algorithms, and so on. In this task, two different optimization techniques are applied on two different functions chosen from CEC’2005. CEC’2005 gave 25 benchmark functions; among those 2 basic functions- F10: Shifted Rotated Rastrigin’s Function and F11- Shifted Rotated Weierstrass Function are selected. Optimisation techniques are applied on both the functions using both algorithms and results are compared.

**(i) Shifted Rotated Rastrigin’s Function:** It can be given by the formula:

$$F_{10}(\mathbf{x}) = \sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias_{10}, \quad \mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}, \ \mathbf{x} = [x_1, x_2, ..., x_D]$$

*D:* dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$ : the shifted global optimum

**M**: linear transformation matrix, condition number=2

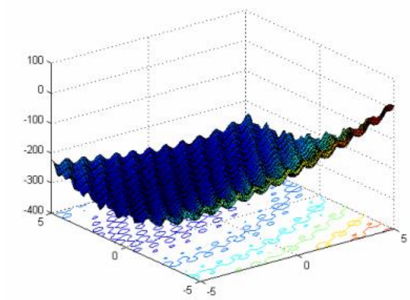



Fig41: 3D map for F10 function

Properties:

- Multi-modal
- Shifted
- Rotated
- Non-separable
- Scalable
- Local optima’s number is huge
- x∈ $[-5,5]^D$ , Global optimum x* = o , $F_{10}(x^*)$ =$f\_bias_{10}$= - 330

**(ii) Shifted Rotated Weierstrass Function:** It can be given by the formula:

$$F_{11}(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}[a^k \cos(2\pi b^k (z_i + 0.5))]\right) - D\sum_{k=0}^{k\max}[a^k \cos(2\pi b^k \cdot 0.5)] + f\_bias_{11},$$

a=0.5, b=3, $k_{max}$=20, $\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$ , $\mathbf{x} = [x_1, x_2, ..., x_D]$

*D:* dimensions

$\mathbf{o} = [o_1, o_2, ..., o_D]$ : the shifted global optimum

**M**: linear transformation matrix, condition number=5

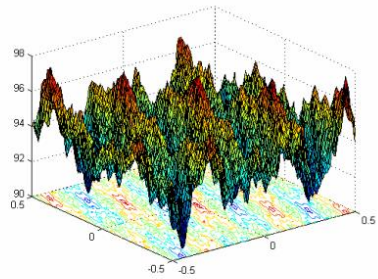


Fig42: 3D map for F11 function

Properties:

- Multi-modal
- Shifted
- Rotated
- Non-separable
- Scalable
- Continuous but differentiable only on a set of points
- x∈ $[-0.5,0.5]^D$ , Global optimum x* = o , $F_{11}(x^*)$ =$f\_bias_{11}$= 90

Genetic Algorithm and Particle Swarm Optimisation Techniques are applied to optimise two functions mentioned above. 15 iterations are run for D=2 (Two dimensions) and D=10 (Ten dimensions).

**Results:**

**(a) Shifted Rotated Rastrigin's Function:**

The results using both algorithms for D=2 and D=10 are shown below in tables:

| Two Dimensions | | | | | |
|---|---|---|---|---|---|
| **Algorithm** | **Best Performance** | **Worst Performance** | **Mean** | **Target Global Minima** | **Standard Deviation** |
| **Genetic Algorithm** | **-329.3025** | **-320.2820** | **-327.0233** | **-330** | **2.1948** |
| **Particle Swarm Optimisation** | **-330.0000** | **-328.9743** | **-329.5302** | **-330** | **0.5129** |

Table: Shifted Rotated Rastrigin's Function Two Dimension Results Summary

| Ten Dimensions | | | | | |
|---|---|---|---|---|---|
| **Algorithm** | **Best Performance** | **Worst Performance** | **Mean** | **Target Global Minima** | **Standard Deviation** |
| **Genetic Algorithm** | **-302.9415** | **-244.8916** | **-280.7450** | **-330** | **16.2211** |
| **Particle Swarm Optimisation** | **-323.0353** | **-283.3498** | **-310.8615** | **-330** | **11.0887** |

Table: Shifted Rotated Rastrigin's Function Ten Dimension Results Summary

From the table , it can be seen that PSO outweighs Genetic Algorithm. The mean value obtained for Global minima is very close to Target Global minima; especially in case of two dimensions. Standard Deviation is also less for PSO. The difference between results of both optimization techniques is very negligible.

**(b) Shifted Rotated Weierstrass Function:**

The results using both algorithms for D=2 and D=10 are shown below in tables:

| Two Dimensions | | | | | |
|---|---|---|---|---|---|
| **Algorithm** | **Best Performance** | **Worst Performance** | **Mean** | **Target Global Minima** | **Standard Deviation** |
| **Genetic Algorithm** | **91.0384** | **90.0853** | **90.5656** | **90** | **0.2605** |
| **Particle Swarm Optimisation** | **90.5992** | **90.0883** | **90.3575** | **90** | **0.1657** |

Table: Shifted Rotated Weierstrass Function Two Dimension Results Summary

| Ten Dimensions | | | | | |
|---|---|---|---|---|---|
| **Algorithm** | **Best Performance** | **Worst Performance** | **Mean** | **Target Global Minima** | **Standard Deviation** |
| **Genetic Algorithm** | **101.0709** | **98.7298** | **100.0526** | **90** | **0.6551** |
| **Particle Swarm Optimisation** | **101.9029** | **98.1188** | **100.7039** | **90** | **1.0825** |

Table: Shifted Rotated Weierstrass Function Ten Dimension Results Summary

From above tables, it is evident that Particle Swam Optimisation works good for Shifted Rotated Weierstrass function also. The results are very close to each other. Genetic algorithm performed little bit better in 10 dimensions. This proves that optimization algorithms perform differently with different problems. In order to find best suitable approach, it is recommended to run multiple algorithms. Moreover, results obtained are very close to target which means both of the algorithms are good and can be tried for various problems.

# Appendix:

## Task-1:

The link for code is: https://colab.research.google.com/drive/1LNf-VzCwrg2i0FYKoBJcEPTZI0u3N_Hv?usp=sharing

The link for dataset is: https://www.kaggle.com/arushchillar/disneyland-reviews

## Task-2:

Matlab code of Part-1: Designing Fuzzy Logic

```
[System]

Name='smart_home'

Type='mamdani'

Version=2.0

NumInputs=8

NumOutputs=7

NumRules=27

AndMethod='min'

OrMethod='max'

ImpMethod='min'

AggMethod='max'

DefuzzMethod='centroid'

[Input1]

Name='room_temp'

Range=[0 70]

NumMFs=5

MF1='VL':'trapmf',[-10 -1 0 5]

MF2='L':'trimf',[5 10 15]

MF3='M':'trimf',[15 20 25]

MF4='H':'trimf',[25 30 40]

MF5='VH':'trapmf',[40 60 70 100]

[Input2]

Name='light_level'

Range=[0 500]

NumMFs=3

MF1='L':'trapmf',[0 0 20 80]

MF2='M':'trimf',[70 250 350]

MF3='H':'trapmf',[300 450 520 600]
```

```
[Input3]
Name='time_of_day'
Range=[0 1]
NumMFs=3
MF1='M':'trapmf',[0 0 0.1 0.4]
MF2='A':'trimf',[0.3 0.5 0.7]
MF3='N':'trapmf',[0.6 0.9 1 2]
[Input4]
Name='CO2_conc(ppm)'
Range=[0 2000]
NumMFs=3
MF1='L':'trapmf',[0 0 40 360]
MF2='M':'trimf',[200 800 1200]
MF3='H':'trapmf',[1200 1800 2000 2700]
[Input5]
Name='dirt'
Range=[0 100]
NumMFs=3
MF1='LD':'trapmf',[0 0 10 30]
MF2='MD':'trimf',[20 45 80]
MF3='VD':'trapmf',[70 85 100 100]
[Input6]
Name='gate_sensor'
Range=[0 1]
NumMFs=2
MF1='NP':'trapmf',[0 0 0 0]
MF2='P':'trapmf',[1 1 1 1]
[Input7]
Name='survellience'
Range=[0 1]
NumMFs=2
MF1='NP':'trapmf',[0 0 0 0]
MF2='P':'trapmf',[1 1 1 1]
[Input8]
Name='groceries'
Range=[0 100]
NumMFs=3
```

```
MF1='LF':'trapmf',[0 0 10 30]
MF2='MF':'trimf',[20 50 70]
MF3='CF':'trapmf',[65 90 100 100]
[Output1]
Name='cooling_fan'
Range=[0 10]
NumMFs=4
MF1='off':'trapmf',[0 0 0 0]
MF2='H':'trapmf',[6.5 8 10 10]
MF3='S':'trimf',[0.1 2 4]
MF4='M':'trimf',[3 5 7]
[Output2]
Name='heaters'
Range=[0 10]
NumMFs=4
MF1='off':'trapmf',[0 0 0 0]
MF2='S':'trimf',[0.1 3 4]
MF3='M':'trimf',[3 6 7]
MF4='H':'trapmf',[6.5 8 10 10]
[Output3]
Name='cleaning_robot'
Range=[10 80]
NumMFs=3
MF1='L':'trapmf',[10 10 20 35]
MF2='M':'trimf',[25 45 65]
MF3='H':'trapmf',[55 70 80 80]
[Output4]
Name='window'
Range=[0 100]
NumMFs=3
MF1='Close':'trapmf',[0 0 0 0]
MF2='half_open':'trimf',[0.1 50 70]
MF3='full_open':'trapmf',[60 90 100 100]
[Output5]
Name='blinds'
Range=[0 100]
NumMFs=3
```

```
MF1='close':'trapmf',[0 0 0 0]
MF2='half_open':'trimf',[0.1 50 70]
MF3='full_open':'trapmf',[60 80 100 100]
[Output6]
Name='fridge'
Range=[0 100]
NumMFs=3
MF1='L':'trapmf',[0 0 5 25]
MF2='M':'trimf',[20 50 70]
MF3='H':'trapmf',[60 80 100 100]
[Output7]
Name='door'
Range=[0 1]
NumMFs=2
MF1='Open':'trapmf',[1 1 1 1]
MF2='close':'trapmf',[0 0 0 0]
[Rules]
1 0 0 0 0 0 0 0, 1 4 0 0 0 0 0 (1) : 1
2 0 0 0 0 0 0 0, 1 3 0 0 0 0 0 (1) : 1
3 0 0 0 0 0 0 0, 3 1 0 0 0 0 0 (1) : 1
4 0 0 0 0 0 0 0, 4 1 0 0 0 0 0 (1) : 1
5 0 0 0 0 0 0 0, 2 1 0 0 0 0 0 (1) : 1
0 1 1 0 0 0 0 0, 0 0 0 0 3 0 0 (1) : 1
0 1 2 0 0 0 0 0, 0 0 0 0 3 0 0 (1) : 1
0 1 3 0 0 0 0 0, 0 0 0 0 1 0 0 (1) : 1
0 2 1 0 0 0 0 0, 0 0 0 0 2 0 0 (1) : 1
0 2 2 0 0 0 0 0, 0 0 0 0 2 0 0 (1) : 1
0 2 3 0 0 0 0 0, 0 0 0 0 1 0 0 (1) : 1
0 0 0 1 0 0 0 0, 0 0 0 1 0 0 0 (1) : 1
0 0 0 2 0 0 0 0, 0 0 0 2 0 0 0 (1) : 1
0 0 0 3 0 0 0 0, 0 0 0 3 0 0 0 (1) : 1
0 0 0 0 1 0 0 0, 0 0 1 0 0 0 0 (1) : 1
0 0 0 0 2 0 0 0, 0 0 2 0 0 0 0 (1) : 1
0 0 0 0 3 0 0 0, 0 0 3 0 0 0 0 (1) : 1
0 0 0 0 0 1 1 0, 0 0 0 0 0 0 2 (1) : 1
0 0 0 0 0 1 2 0, 0 0 0 0 0 0 2 (1) : 1
0 0 0 0 0 2 1 0, 0 0 0 0 0 0 2 (1) : 1
```

```
0 0 0 0 0 2 2 0, 0 0 0 0 0 0 1 (1) : 1
0 0 0 0 0 0 0 1, 0 0 0 0 0 1 0 (1) : 1
0 0 0 0 0 0 0 2, 0 0 0 0 0 2 0 (1) : 1
0 0 0 0 0 0 0 3, 0 0 0 0 0 3 0 (1) : 1
0 2 3 0 0 0 0 0, 0 0 0 0 1 0 0 (1) : 1
0 3 1 0 0 0 0 0, 0 0 0 0 1 0 0 (1) : 1
0 3 3 0 0 0 0 0, 0 0 0 0 1 0 0 (1) : 1
```

### Snippet of Fuzzy Rules

1. If (room_temp is VL) then (cooling_fan is off)(heaters is H) (1)
2. If (room_temp is L) then (cooling_fan is off)(heaters is M) (1)
3. If (room_temp is M) then (cooling_fan is S)(heaters is off) (1)
4. If (room_temp is H) then (cooling_fan is M)(heaters is off) (1)
5. If (room_temp is VH) then (cooling_fan is H)(heaters is off) (1)
6. If (light_level is L) and (time_of_day is M) then (blinds is full_open) (1)
7. If (light_level is L) and (time_of_day is A) then (blinds is full_open) (1)
8. If (light_level is L) and (time_of_day is N) then (blinds is close) (1)
9. If (light_level is M) and (time_of_day is M) then (blinds is half_open) (1)
10. If (light_level is M) and (time_of_day is A) then (blinds is half_open) (1)
11. If (light_level is M) and (time_of_day is N) then (blinds is close) (1)
12. If (CO2_conc(ppm) is L) then (window is Close) (1)
13. If (CO2_conc(ppm) is M) then (window is half_open) (1)
14. If (CO2_conc(ppm) is H) then (window is full_open) (1)
15. If (dirt is LD) then (cleaning_robot is L) (1)
16. If (dirt is MD) then (cleaning_robot is M) (1)
17. If (dirt is VD) then (cleaning_robot is H) (1)
18. If (gate_sensor is NP) and (survellience is NP) then (door is close) (1)
19. If (gate_sensor is NP) and (survellience is P) then (door is close) (1)
20. If (gate_sensor is P) and (survellience is NP) then (door is close) (1)
21. If (gate_sensor is P) and (survellience is P) then (door is Open) (1)

21. If (gate_sensor is P) and (survellience is P) then (door is Open) (1)

22. If (groceries is LF) then (fridge is L) (1)
23. If (groceries is MF) then (fridge is M) (1)
24. If (groceries is CF) then (fridge is H) (1)
25. If (light_level is M) and (time_of_day is N) then (blinds is close) (1)
26. If (light_level is H) and (time_of_day is M) then (blinds is close) (1)
27. If (light_level is H) and (time_of_day is N) then (blinds is close) (1)

### Rule Evaluation:

| Parameter | Value |
|---|---|
| **Room Temperature** | **30 C** |
| **Light Level** | **200 Lux** |
| **Time_of_Day** | **Afternoon (A) - 0.5** |
| **CO2 conc** | **300 ppm** |
| **dirt** | **30 %** |

| | |
|---|---|
| **Gate_Sensor** | **1** |
| **Surveillance** | **1** |
| **Groceries** | **20 %** |
| Cooling Fan | 5 |
| Heater | 0 |
| Blinds | 39.4 |
| Window | 35.6 |
| Cleaning_robot | 45 min |
| Door | 1 |
| Fridge | 9.95 |

## Part-3: Results with Genetic Algorithm:

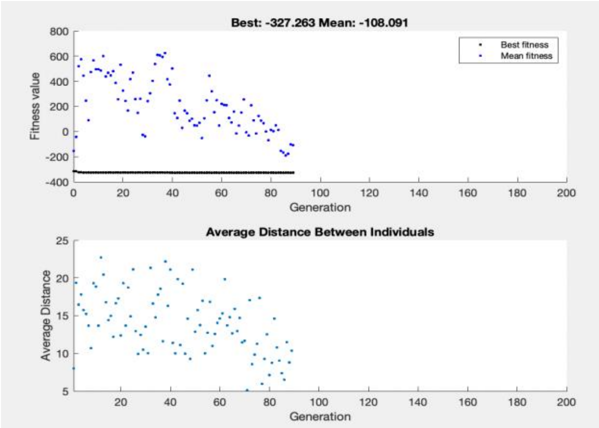
| | CEC Function=10 | | CEC Function=11 | |
|---|---|---|---|---|
| **No.** | **Local Minima (D=2)** | **Local Minima (D=10)** | **Local Minima (D=2)** | **Local Minima (D=10)** |
| 1 | -327.9473 | -244.8916 | 90.4661 | 100.2491 |
| 2 | -328.9320 | -296.4951 | 91.0384 | 99.8267 |
| 3 | -329.3025 | -292.8514 | 90.7802 | 100.3924 |
| 4 | -327.3298 | -271.3307 | 90.4535 | 101.0709 |
| 5 | -326.2487 | -271.5613 | 90.0853 | 99.3020 |
| 6 | -328.3183 | -284.4126 | 90.4215 | 99.8109 |
| 7 | -320.2820 | -291.0264 | 90.4844 | 100.7370 |
| 8 | -326.6554 | -251.4698 | 90.8686 | 98.9969 |
| 9 | -328.2626 | -302.9415 | 91.0066 | 100.6435 |
| 10 | -328.6632 | -294.4664 | 90.5695 | 99.9360 |
| 11 | -328.3107 | -271.4746 | 90.4951 | 100.4600 |
| 12 | -325.9939 | -282.8292 | 90.4241 | 98.7298 |
| 13 | -325.4924 | -286.6287 | 90.6565 | 100.0841 |
| 14 | -326.3479 | -287.7069 | 90.3886 | 100.5769 |
| 15 | -327.2634 | -281.0889 | 90.3448 | 99.9722 |
| **Max_value** | **-329.3025** | **-302.9415** | **91.0384** | **101.0709** |
| **Min_value** | **-320.2820** | **-244.8916** | **90.0853** | **98.7298** |
| **Mean value** | **-327.0233** | **-280.7450** | **90.5656** | **100.0526** |
| **Standard Deviation** | **2.1948** | **16.2211** | **0.2605** | **0.6551** |

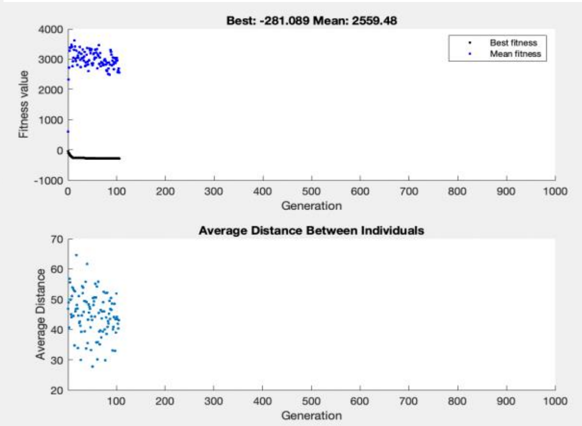
## Results with Particle Swarm Optimisation Algorithm:

| | CEC Function=10 | | CEC Function=11 | |
|---|---|---|---|---|
| **No.** | **Local Minima (D=2)** | **Local Minima (D=10)** | **Local Minima (D=2)** | **Local Minima (D=10)** |
| 1 | -329.9477 | -322.0403 | 90.3149 | 101.4359 |
| 2 | -330.0000 | -314.0807 | 90.5590 | 100.2602 |
| 3 | -329.0050 | -318.0605 | 90.1082 | 101.7809 |
| 4 | -329.0050 | -310.1009 | 90.1951 | 101.7052 |
| 5 | -328.9743 | -305.1261 | 90.5196 | 100.1453 |
| 6 | -329.0050 | -306.6347 | 90.5136 | 98.1188 |
| 7 | -330.0000 | -321.0454 | 90.3280 | 99.5010 |
| 8 | -330.0000 | -314.0807 | 90.2887 | 100.3330 |
| 9 | -329.0050 | -302.1412 | 90.5992 | 99.8430 |
| 10 | -330.0000 | -317.0655 | 90.2805 | 101.8162 |
| 11 | -330.0000 | -283.3498 | 90.4542 | 101.9029 |
| 12 | -330.0000 | -296.1715 | 90.3628 | 100.3721 |
| 13 | -329.0050 | -323.0353 | 90.5345 | 101.7726 |
| 14 | -330.0000 | -307.9492 | 90.0883 | 100.4332 |
| 15 | -329.0050 | -322.0403 | 90.2164 | 101.1387 |
| **Max_value** | **-330.0000** | **-283.3498** | **90.5992** | **101.9029** |
| **Min_value** | **-328.9743** | **-323.0353** | **90.0883** | **98.1188** |
| **Mean value** | **-329.5302** | **-310.8615** | **90.3575** | **100.7039** |
| **Standard Deviation** | **0.5129** | **11.0887** | **0.1657** | **1.0825** |

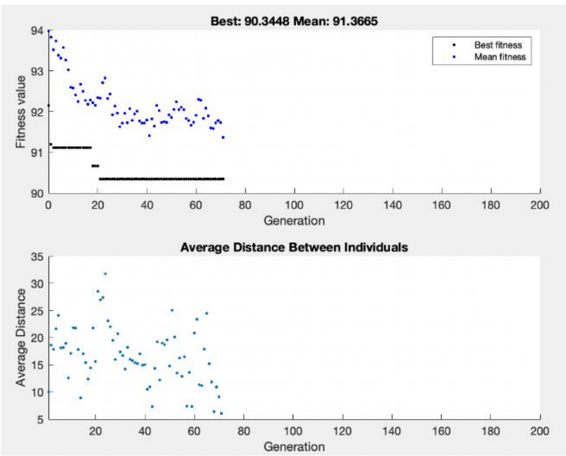
## Plot of CEC function for Genetic algorithm Tool:

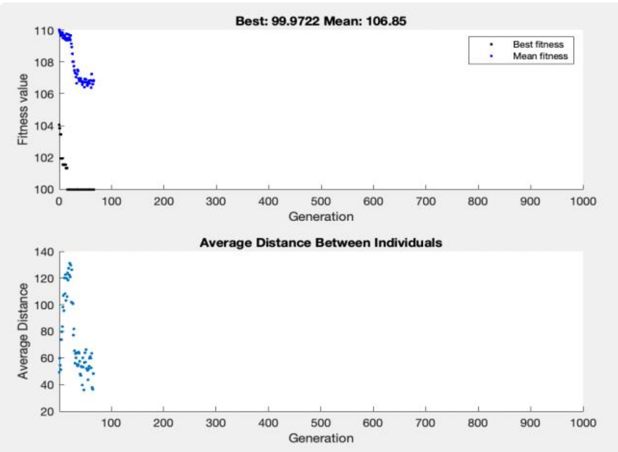
Function 10: D=2 D= 10



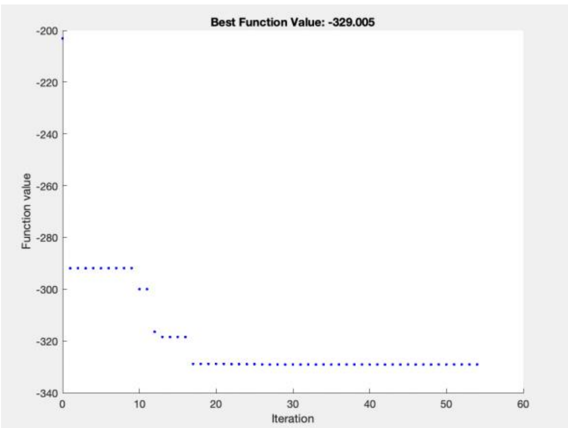

Function 11: D=2 D= 10

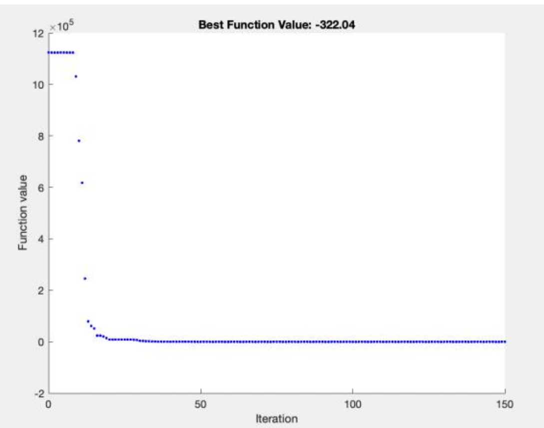


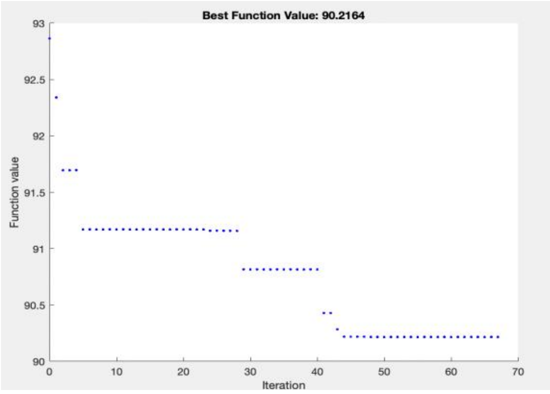

**Plot of CEC function for Particle Swarm Optimisation Tool:**

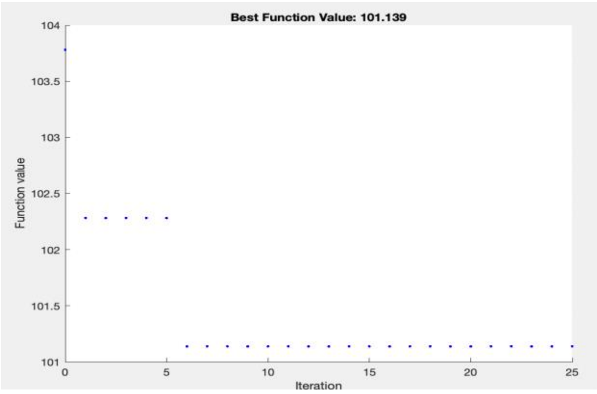
Function 10: D=2 D= 10




Function 11: D=2 D= 10




**CEC Function Code for Genetic Algorithm**

```
clc, clear, close all
rng default
global initial_flag
```

## Genetic Algorithm Optimization 15 iterations

```
initial_flag = 0;
options = optimoptions('ga','PlotFcn',{@gaplotbestf,@gaplotdistance});
for i=1:15
    initial_flag = 0;
    % use the below for 2D function 10
    %[ga_x,ga_val,ga_exit_flag,ga_output] =ga(@(x)benchmark_func(x,10),2,options)
    % use the below for 10D function 10
    [ga_x,ga_val,ga_exit_flag,ga_output] =ga(@(x)benchmark_func(x,10),10,options)
    % use the below for 2D function 11
    %[ga_x,ga_val,ga_exit_flag,ga_output] =ga(@(x)benchmark_func(x,11),2,options)
    % use the below for 10D function 11
    %[ga_x,ga_val,ga_exit_flag,ga_output] =ga(@(x)benchmark_func(x,11),10,options)
    ga_main_x(i,:) = ga_x
    ga_main_val(i) = ga_val
    ga_main_exit_flag(i) = ga_exit_flag
    ga_main_output(i) = ga_output
    % save visualizations to file
    fname = sprintf('filename(%d).fig', i) ;
    savefig(fname)
end
```

## GA 15 iteration measures

```
ga_val_max = max(ga_main_val)
ga_val_min = min(ga_main_val)
```

```
ga_val_mean = mean(ga_main_val)
ga_val_std = std(ga_main_val)
```

**CEC Function Code for Particle Swarm Optimisation Algorithm**

```
clc, clear, close all
rng default
global initial_flag
%% Particle Swarm Optimization 15 Iterations
options = optimoptions('particleswarm','PlotFcn',{@pswplotbestf}); %rng default
initial_flag = 0;
for i=1:15
  %use the below for 10D function 10
%[swarm_x,swarm_val,swarm_exit_flag,swarm_output] =
particleswarm(@(x)benchmark_func(x,10),10,[-100,-100],[100,100],options)
  % use the below for 2D function 10
%[swarm_x,swarm_val,swarm_exit_flag,swarm_output] =
particleswarm(@(x)benchmark_func(x,10),2,[-100,-100],[100,100],options)
  % use the below for 10D function 11
[swarm_x,swarm_val,swarm_exit_flag,swarm_output] =
particleswarm(@(x)benchmark_func(x,11),10,[-100,-100],[100,100],options)
  % use the below for 2D function 11
%[swarm_x,swarm_val,swarm_exit_flag,swarm_output] =
particleswarm(@(x)benchmark_func(x,11),2,[-100,-100],[100,100],options)
swarm_main_x(i,:) = swarm_x
swarm_main_val(i) = swarm_val
swarm_main_exit_flag(i) = swarm_exit_flag
swarm_main_output(i) = swarm_output
  % save visualizations to file
fname = sprintf('filename(%d).fig', i) ;
  savefig(fname)
end
%% Particle Swarm calculations
swarm_val_max = max(swarm_main_val)
swarm_val_min = min(swarm_main_val)
swarm_val_mean = mean(swarm_main_val)
swarm_val_std = std(swarm_main_val)
```

## References:

[1].Nikolenko, S. I., Koltcov, S., & Koltsova, O. (2017). Topic modelling for qualitative studies. *Journal of Information Science*, *43*(1), 88-102.

[2]. Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital journalism*, *4*(1), 89-106.

[3]. Tong, Z., & Zhang, H. (2016, May). A text mining research based on LDA topic modelling. In *International Conference on Computer Science, Engineering and Information Technology* (pp. 201-210).

[4]. Kapadia, S. (2019). *Topic Modeling in Python: Latent Dirichlet Allocation (LDA)*. Medium. Retrieved 13 December 2021, from https://towardsdatascience.com/end-to-end-topic-modeling-in-python-latent-dirichlet-allocation-lda-35ce4ed6b3e0.

[5]. https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

[6]. *Topic Modeling with LSA, PSLA, LDA & lda2Vec*. Medium. (2018). Retrieved 14 December 2021, from https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05.

[7]. Goyal, C. (2021). *Topic Modelling using NMF | Guide to Master NLP (Part 14)*. Analytics Vidhya. Retrieved 15 December 2021, from https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/.

[8]. *Short-Text Topic Modelling: LDA vs GSDMM*. Medium. (2021). Retrieved 15 December 2021, from https://towardsdatascience.com/short-text-topic-modelling-lda-vs-gsdmm-20f1db742e14.

[9]. *pyLDAvis: Topic Modelling Exploration Tool That Every NLP Data Scientist Should Know - neptune.ai*. neptune.ai. (2021). Retrieved 15 December 2021, from https://neptune.ai/blog/pyldavis-topic-modelling-exploration-tool-that-every-nlp-data-scientist-should-know.

[10]. (2021). Retrieved 11 December 2021, from https://uk.mathworks.com/help/fuzzy/defuzzification-methods.html.

[11]. (2021). Retrieved 11 December 2021, from https://uk.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html.