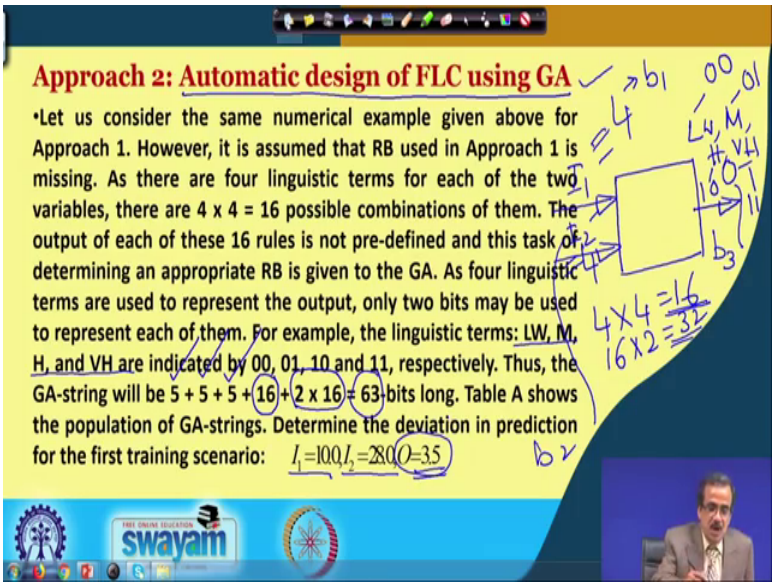
Fuzzy Logic and Neural Networks
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 19
Optimization of Fuzzy Reasoning and Clustering Tool (Contd.)

Our aim is to design optimal knowledge base of a fuzzy reasoning tool. Now, what we do is, in approach 1, which we have already discussed, the designer based on his own experience of the problem to be modeled, he or she designs the knowledge base.

(Refer Slide Time: 00:37)



That is the rule base and data base of the fuzzy reasoning tool. Now, after that, we use one optimizer say one nature-inspired optimization tool like genetic algorithm to tune its database and a rule base. Now, this genetic algorithm through a large number of iterations, we will try to find out what should be the optimal knowledge base for this fuzzy reasoning tool.

Now, supposing that we are going to model a very complicated process, a real-world problem. And, it is bit difficult for the designer to determine the knowledge base and the rule base of the fuzzy reasoning tool beforehand. Now, in that case, actually we cannot go for approach 1, that is the genetic algorithm-based tuning of the knowledgebase of

fuzzy reasoning tool. And, there we will have to go for another approach and that is your approach 2 that is nothing but automatic design of FLC using a genetic algorithm.

Now, here, we do not determine the rule base of the fuzzy reasoning tool or fuzzy logic controller beforehand, because we do not have sufficient information of the process to be controlled. And, here, the whole task of designing the rule base is given to the genetic algorithm. Now, genetic algorithm through a large number of iterations will try to evolve, what should be the optimal database, and what should be the optimal rule base of the fuzzy reasoning tool, so that it can make the prediction as accurately as possible.

Now, this approach, that is approach 2, so we are going to discuss with the help of one numerical example, the same numerical example which I consider for approach 1, so I am just going to consider it once again. So, our aim is to design and develop the knowledge base of one fuzzy reasoning tool or fuzzy logic controller, whose aim is to model a process having two inputs: I_1 and I_2, and it has got only one output that is O.

Now, we have already discussed that four linguistic terms are used to represent I_1, I_2, and the output O. And, the linguistic terms are like very low, low, medium or high, or it could be your say low, medium, high and very high. So, we are going to consider for linguistic term, that is low, medium, high and very high. So, I have got four such linguistic terms here for representing I_1; I have got four linguistic term for representing I_2. So, I have got 4 multiplied by 4, there are 16 rules, that mean 16 possible combinations for the input parameters.

And, once again, we are going to use four linguistic terms to represent the output, that is, we are going to use low, medium, then comes your high and very high. Now, let us see how can you implement this particular approach that is automatic design of FLC using a genetic algorithm. Now, here what you do is, so there are four linguistic terms for the outputs. So, what we do, we try to represent, say if the output is low, that is represented by say 0 0; then medium can be represented by 0 1; high can be represented by 1 0; and very high can be represented by your 1 1. So, to represent the output of a particular rule, we are going to use, in fact, 2-bits. And, there are 16 rules, so I will have to use 16 multiplied by 2, that is your 32 bits to represent what should be the output of these rules.

Now, here, so we have got the variables like the way I discussed in the first approach. So, we have got the variables to represent the shape of the membership function

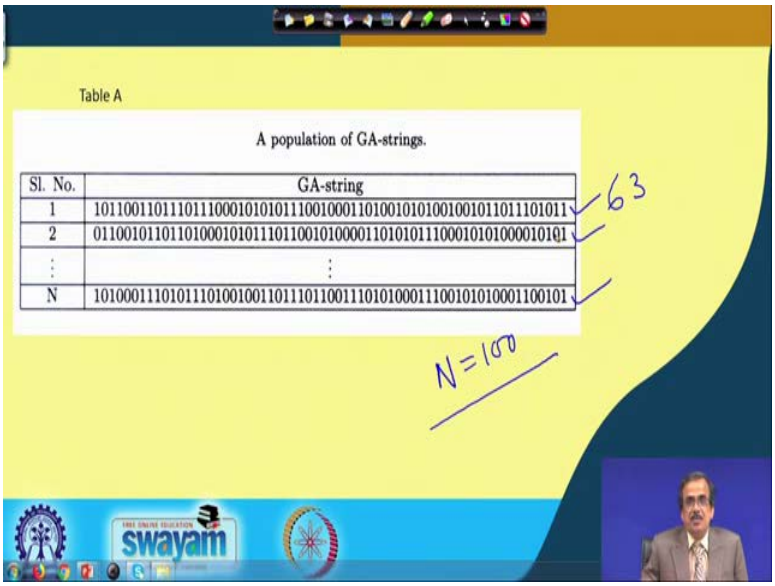
distribution for input one that is nothing but b_1 if we remember. For I_2, we have got another variable that is b_2; and for this output, we have got another variable that is nothing but your b_3. So, we have got b_1, b_2 and b_3, so these three real variables. And, then, we have got 16 rules just to represent the presence or absence of the rules. And, we have got 16 multiplied by 2, that is, 32 bits to represent what should be the output for the 16 rules.

Now, if you see, we are assigning 5-bits to represent b_1; 5-more bits to represent b_2, and 5 more bits to represent b_3, then there will be 16 bits to represent the 16 rules like the presence or absence of the rule. And, 2 multiplied by 16, that is 32 bits to represent what should be the output for the 16 rules. So, we have got $5+5+5+16+2\times16=63$. So, in total we have got 63 bits. So, the GA-string will be 63 bits long. And, a particular GA-string will carry information of the data base and rule base, and the output of the rules for this the fuzzy logic controller.

Now, our aim is to pass one set of inputs like I_1 is equal to 10 and I_2 is 28.0. And, we have got actually the target output, that is 3.5. And, we will try to find out what should be the calculated output and that particular calculated output will be compared with the target output just to find out the deviation. And, this particular deviation, we will have to minimize. So, this is actually the problem description. The same numerical example, we are going to solve using approach 2.

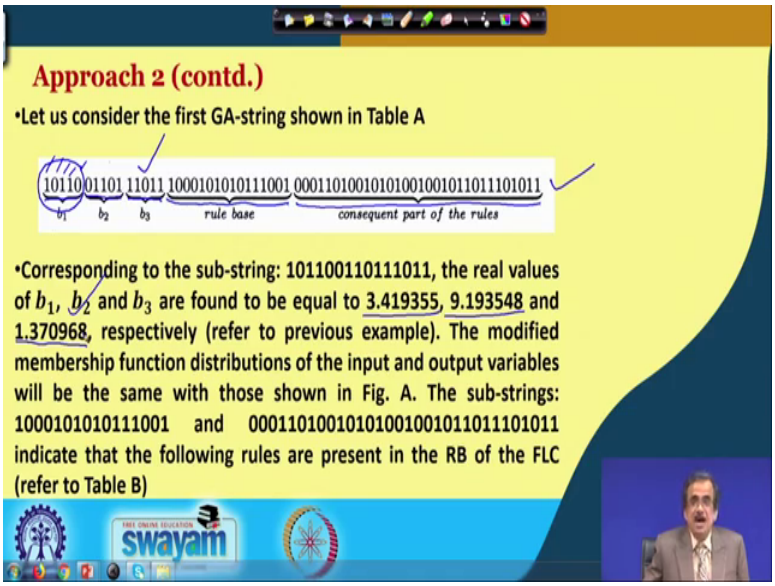
The only thing, the only difference here, is in earlier approach, that is, approach 1, we have predetermined set of rules; and here, we have got the combination of the input parameters, but their corresponding outputs are not known. Now, here, actually we are going to use genetic algorithm to find out what should be the output for each of these 16 rules. So, this is actually approach 2. Now, I am just going to solve this numerical example in details.

(Refer Slide Time: 08:24)



Now, if you see, the population of the solution or the GA strings, so a particular GA-string will look like this. So, as I mentioned that say it has got 63 bits. So, this is nothing but the first GA-string. Similarly, we have got second GA-string and we have got capital N number of GA-strings, because the population size is nothing but your capital N. And, generally we consider N is equal to say 100. So, we have got 100 such GA-strings in the population. And, these particular GA-strings are generated at random using the random number generator. Now, if I concentrate on a particular GA-string, for example, the first GA string here. Now, let us see, how to determine the output corresponding to this particular the GA-string.

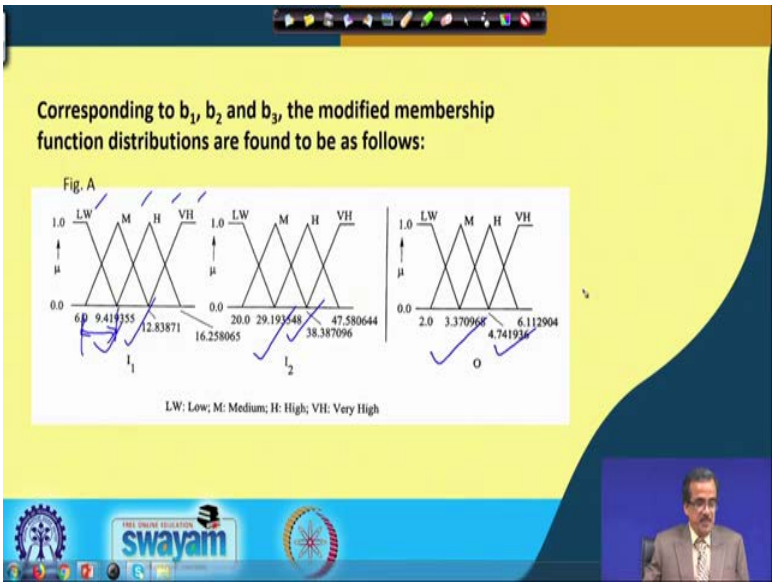
(Refer Slide Time: 09:21)



Now, here this shows, in fact, the first GA-string. So, this is, in fact, the first GA-string. So, the first 5 bits represent the b_1 that is 1, 2, 3, 4, 5. So, 5 bits represent b_1. The next 5 bits represent b_2; and b_3 is represented by the next 5 bits. Now, 16 bits are going to represent actually the presence or absence of the rules, and the output of the rules are represented by these 32-bits, this is 16 multiplied by 2, that is, 32-bits. So, one complete GA-string is going to represent actually the database and the rule base for this particular fuzzy logic controller.

Now, let us see, how to find out the output for a set of inputs. Now, corresponding to these particular 5-bits used to represent b_1, we can find out what should be their real values knowing the lower limit and upper limit for this particular b_1. Now, if you determine the real values for this particular b_1, so we will be getting 3.419355. So, this is the real value for b 1. And by following the same principle, I can find out the real value for b_2, and that is nothing but 9.193548. Then, corresponding to this, I can find out the real value for this particular b_3, and that is nothing but 1.370968. And, once we have got the real values for this particular b_1, b_2 and b_3. So now, we are in a position to find out what should be the membership function distribution or the modified membership function distribution for this b_1, b_2, and b_3.

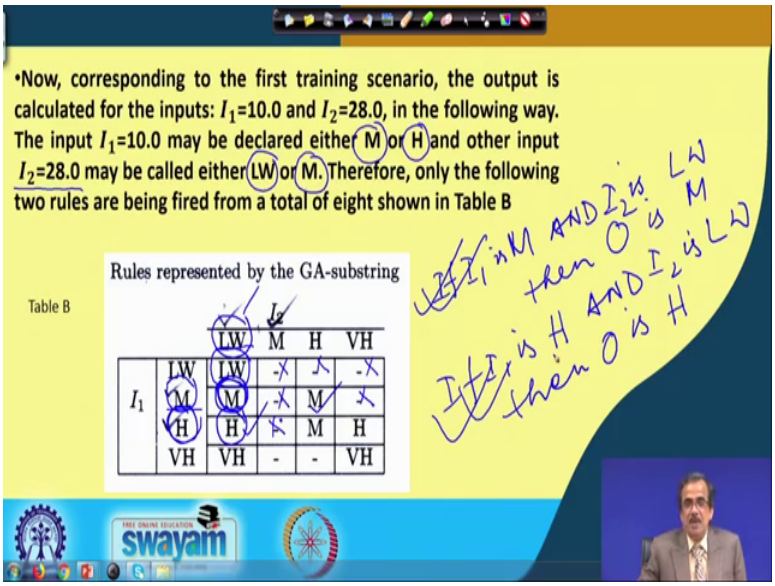
(Refer Slide Time: 11:25)



Now, if you see the modified membership function distribution, this is actually the modified membership function distribution for I_1, this is for I_2 and this is for the output O. And as we discussed that we have got four linguistic terms: low, medium, high and very high, for I_1, I_2 and this particular the output. The only thing is, we have optimized what should be the base width for the right angle triangle used to represent low or the half base-width for the isosceles triangle used to represent medium and high, and so on.

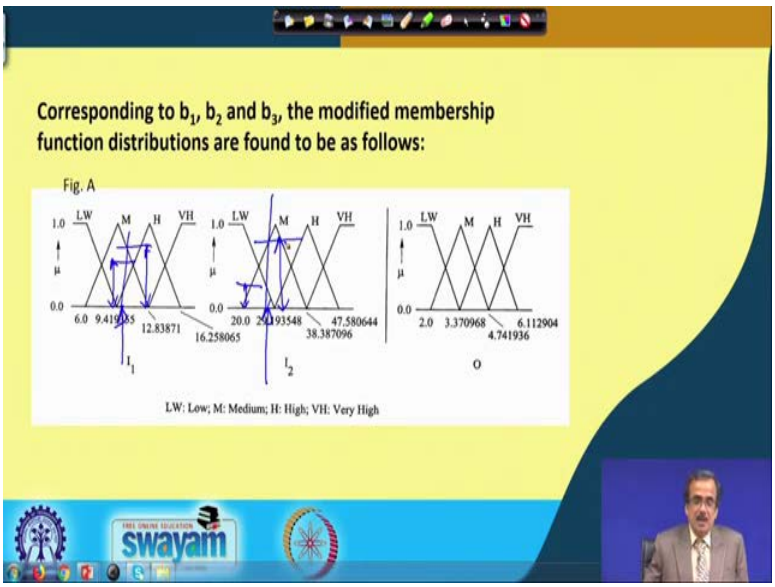
And for simplicity, we consider the symmetric triangles. So, this is the modified membership function distribution for I_1, modified membership function distribution for I_2, and modified membership function distribution for your the output. Now, actually what we will have to do is, we will have to represent or we will have to find out what should be the rule base, and the rule base, which is represented by this particular GA-string.

(Refer Slide Time: 12:42)



Now, before that let me just try to say that, if I put I_1 equals to 10, and I_2 equals to 28, so this will be as follows:

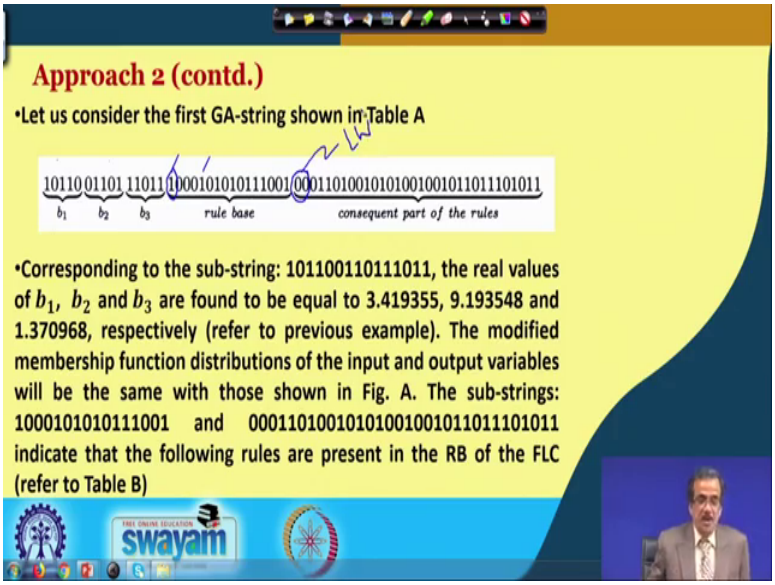
(Refer Slide Time: 12:57)



If I put I_1 equals to 10 here, so I_1 equals to 10 means I am here. So, this particular I_1 can be called your medium with this much of membership function value. And, it can be called high with this much of membership function value. Similarly, if I just write here, I_2 equals to 28 and corresponding to 28, we can find out the membership function value

corresponding to low and membership function value corresponding to this particular the medium.

(Refer Slide Time: 13:37)



Now, let us see, what happens to the rule base. Now, to find out the rule base corresponding to this particular substring, I should say, for example, say these 16-bits will represent the presence and absence of the rule, that means, if I start from here, 1 indicates that the first rule will be present, then second, third, fourth will be absent, then the fifth rule will be present, and so on. Now, if I just implement, so very easily we can find out what are the rules to be there in the rule base. Now, if I consider that this first rule is present and what should be its output, that is decided by these two bits.

Now, this is 0 0. Now, if it is 0 0, so this is nothing but your so this is nothing but the first option that is your low; so the first option that is going to be indicated by this 0 0. So, this is nothing but the low. So, if the first rule is present, so its output will be low. Now, if I just see this, so on this particular table, I can find out, according to that particular substring, the rules which are present, which are found to be present are as follows.

Like if I_1 is low and I_2 is low, then the output is low. So, this particular rule is present, but the second rule is absent, the third rule is absent, the fourth rule is absent. Then the fifth rule is present, which states if I_1 is medium and I_2 is low then the output is medium. Then sixth is absent; seventh is present; eighth is absent; the ninth is present;

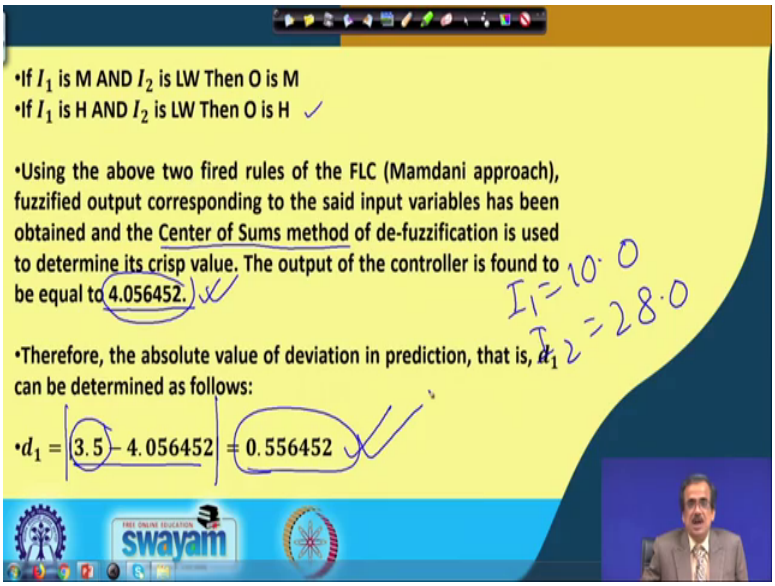
tenth is absent, and so on. Now, as I told that this particular I_1, which is equal to 10.0, it can be called either medium or it could be high. Similarly, the I_2, which is equal to 28.0 can be called either low or it can be called medium. So, there is a maximum of four fired rules. And, let us see out of those four fired rules, which one or which two or which three or whether all four are present or not.

Now, let me try to find out, whether the first combination is present or not. So, if I_1 is medium, so I am here, if I_1 is medium and I_2 is low, so I_2 is low, then output is actually the medium. So, this particular the rule is present. So, this is present. Next is if I 1 is medium, and I_2 is medium, the rule is absent here. The next is, if I_1 is H, so I am here. And I_2 is low, that means, I am here. So, this particular rule is present whose output is actually high. The next is if I_1 is H, so I am here; and I_2 is M, so I am here. So, that particular rule is absent.

So, out of the four maximum rules, only two are found to be present here. And the rules are as follows: like if I_1 is medium, so if I_1 is medium, and your I_2 is low, then the output O is nothing but the medium. So, this is one present fired rule, and another present fired rule is if I_1 is say high, so this one, and I_2 is low, then the output o is nothing but is your high. So, out of these four, only these two rules are found to be present here. Now, we will have to find out, what should be the output corresponding to these two fired rules. And, then, I will have to combine just to find out what should be the control the combined output or the combined control action.

Now, let us see how to find out.

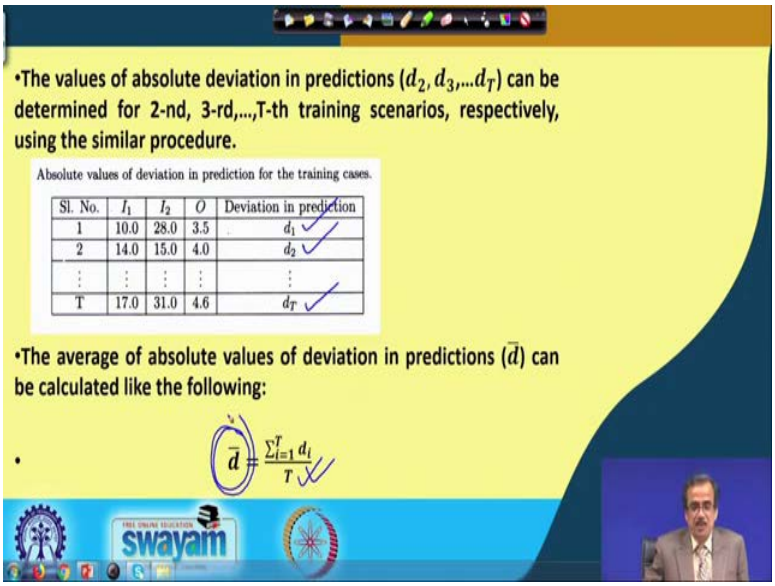
(Refer Slide Time: 18:15)



Now, corresponding to the first fire rule, that is your if I_1 is medium and I_2 is low, the output is medium. So, what you can do is, we can find out what should be the fuzzified output. And, corresponding to the second rule, once again, I can find out what should be the fuzzified output, and then, we combine. Then, we will be getting the fuzzified output for the combined control action considering these two rules. And, once you have got it, now we can use the center of sums method of de-fuzzification. And, if I use it, there is a possibility that you will be able to find out what should be the crisp output and that is coming to be equal to 4.056452.

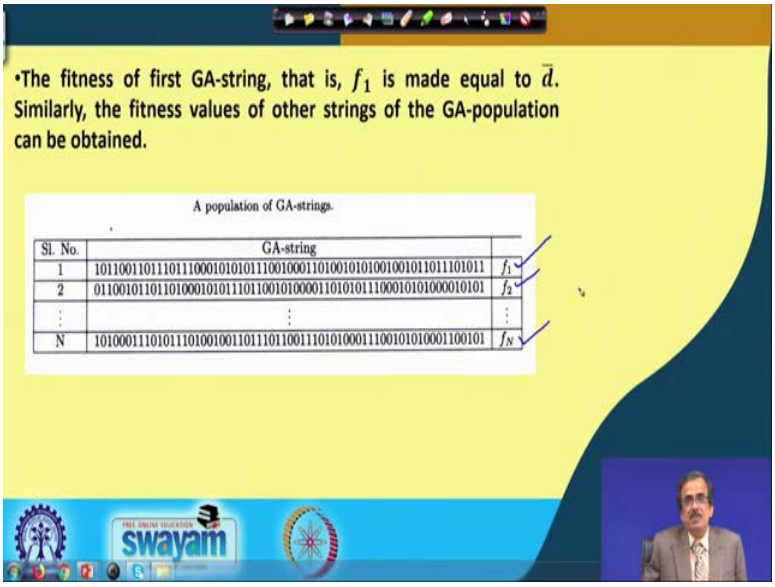
Now, this is the calculated output corresponding to the inputs like I_1 is 10 and I_2 is nothing but 28.0. Now, this is actually the output, but the target output is nothing but is your 3.5. So, there is some deviation and here, this particular deviation, that is, 3.5 minus 4.056452, so this is nothing but a negative value, and that is why we use the mod value just to make it positive. That means out of all the training scenarios, which you have, so if I pass the first training scenario, I am getting this particular deviation. Now, by following the same procedure, I am just going to pass the second training scenario, third training scenario up to the T-th training scenario.

(Refer Slide Time: 20:09)



Now, if I pass all the training scenarios, then I will be getting actually the different deviation values. Now, corresponding to the first training scenario, say the deviation is denoted by say d_1, corresponding to the second training scenario supposing that the deviation is denoted by say d_2. And, corresponding to the T-th training scenario supposing that the deviation is represented by d_T. Now, what we do is, we try to find out the average deviation and that is nothing but the sum of all devalues divided by the number of training scenarios that is nothing but T. So, I can find out what should be this average deviation that is $\bar{d}$.

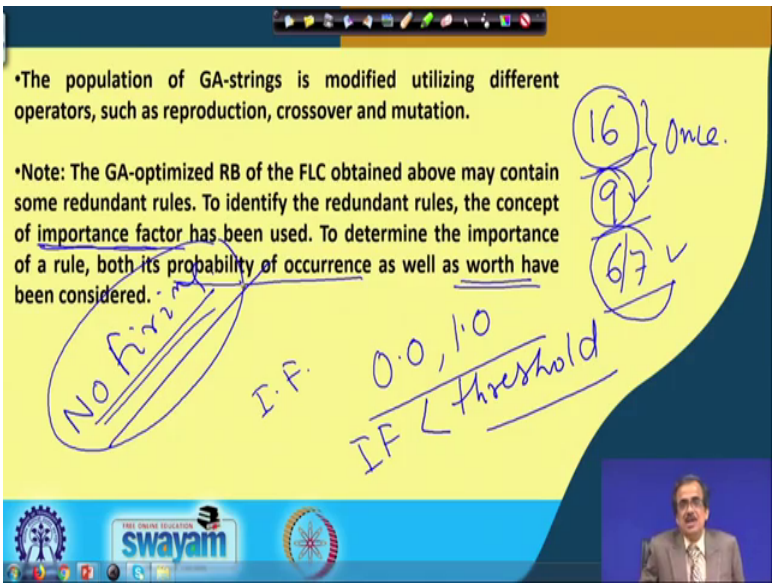
(Refer Slide Time: 20:58)



And, once you have got this average deviation now this average deviation is nothing but the fitness for the first GA string. So, I should be able to find out what should be the fitness for the first GA string that is your f 1. And, by following the same procedure, I can find out the fitness for the second GA string. And we try to find out by following the same procedure the fitness for the other GA string. And for the Nth GA string the fitness is denoted by f_N.

Now, we take the help of the GA-operators like the reproduction, crossover and mutation. And, GA through a large number of iteration we will try to find out what should be the optimal database, and what should be the optimal rule base for this fuzzy reasoning tool. And, once, you have got this optimal database and rule base, so what you can do is, now this optimal fuzzy logic controller, you can use for your per online application. That means, we can pass some test scenarios and we can find out what should be the output for a set of inputs.

(Refer Slide Time: 22:14)



Now, if we optimized or if you try to evolve the knowledge base, that is the rule base and data base, particularly the rule base of a fuzzy logic controller by following this particular the method which I have already discussed, there is a possibility that there will be some redundant rules in the rule base. Now, redundant rule means like, let me try to explain supposing that I have started with says 16 rules. Now, I have started with 16 rules, now if I just do this GA based tuning,

which have already discussed that automatic design of fuzzy logic controller using the genetic algorithm, there is a possibility say I will be getting say 9 good rules out of this particular the 16. Now, this 9, I will be getting if I just do the GA based tuning only once ok.

Now, if I do the same GA based tuning once again, so there is a possibility that from a 9, it may select only 6 rules or the 7th rules out of these 9. So, we will be getting the further tuned rule base. Now, if I follow this particular principle; that means, there are some redundant rules ok, and those redundant rules, in fact, we will have to identify.

Now, to identify the redundant rules, so what we do is, we introduced one technique like we calculated what do you mean by importance factor. Now, this importance factor that is denoted by say I.F. To find out, what we do is, we try to find out, what is the probability of occurrence of the different rules during the training. That means, a particular rule how many times it has been fired during the training scenarios or during the training. And, we try to find out what is the probability of occurrence of all the possible rules during the training. And, moreover, we try to find out what should be the worth of a particular rule.

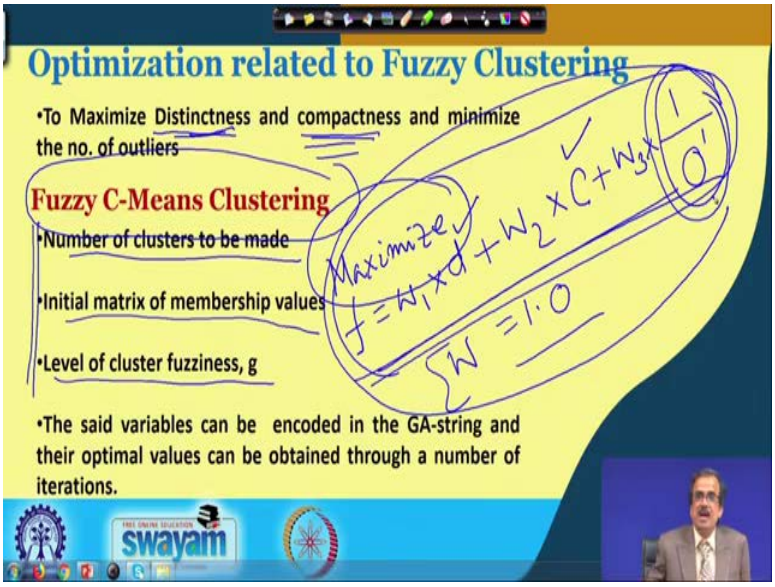
Now, this is what is decided for the set of inputs, what is the output, now out of these 16 rules, the importance of all the 16 rules may not be equally good. Now, what I do is, we try to represent the importance of each of this particular rule in a scale of say 0 to 1. And, if I get the worth of a particular rule and the probability of occurrence, so both the things are going to lie between 0 and 1. So, I will be getting some value lying between 0 and 1. And, if you multiply them, I will be getting another value which is once again will be lying between 0 and 1.

Now, if that particular value that is nothing but the importance factor, now if this importance factor is found to be say either less than some threshold value, that means that particular rule is not very good and that can be declared as a redundant role. So, this is the way actually we declared the redundant rules. But, if it just go on tuning, so this particular the rule base of the fuzzy logic controller, so there is a possibility that it will give raise only a very weak or a very optimized rule based sort of thing.

And, there could be a few test scenarios. Now, if I pass these test scenarios, there is a possibility that not even a single rule is going to be fired and that is actually the problem

of no firing. Now, by no-firing, in fact, we mean a situation, whenever we are passing say one set of training, one set of inputs, but it is not going to actually trigger any of the rules and none of the rules is going to be fired. So, we cannot find out actually the output for these set of inputs, so that particular situation is nothing but actually the no-firing situation. So, our aim is to reduce the redundant rules, but at the same time we should take care that there should not be any such no firing or weak firing. Now, this is the way actually we can optimize the fuzzy reasoning tool.

(Refer Slide Time: 27:09)



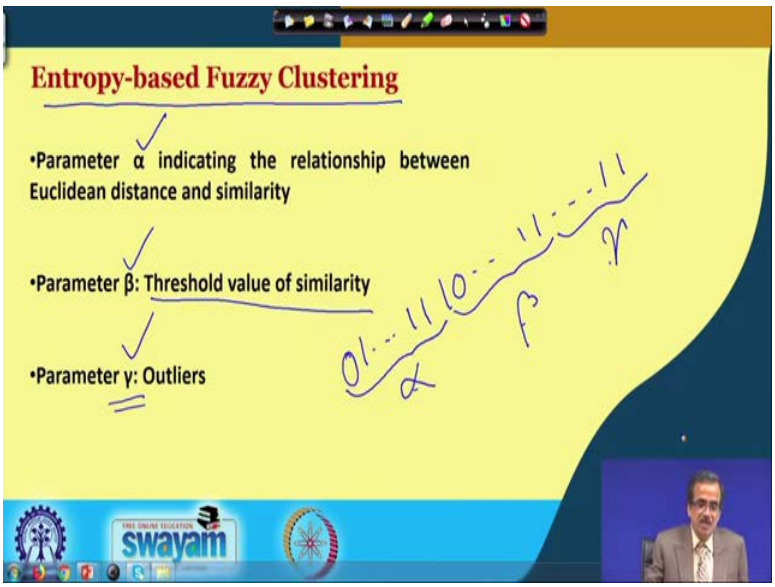
And, now I am just going to discuss, in short, like how to carryout optimization for the fuzzy clustering. Now, we have already discussed that our aim is to determine the clusters, which are very distinct and the clusters should be very compact, and at the same time the number of outlier should be as minimum as possible. And, in fact, ideally we want that there should not be any such outliers.

Now, if I just formulate this particular problem as a maximization problem, so I can formulate like this. So, maximize f, $f = W_1 \times d + W_2 \times C + W_3 \times \frac{1}{O'}$, which indicates your outliers. So, our aim is to maximize this particular objective function. And, once again, we can take the help of your this type of say genetic algorithm. And genetic algorithm is going to encode, the values for this d then comes your c and this say your 1 divided by O prime that is your outliers and the sum of all the W values should be equal to your 1.0.

So, all such values, we can actually find out and we can find out what should be the distinctness, what should be the compactness and what should be the number of outliers. So, these are the things, which will have to find out, but what should be the design variables. The design variables or the performance of a fuzzy clustering tool, so fuzzy C-means clustering depends on the number of clusters to be made. Then comes the initial matrix of your membership values and the level of cluster fuzziness. So, all such things at the design parameters for this FCM that is our fuzzy c means clustering. And we can use some GA strings to represent the design variables and this would be your objective function. And, our aim is to maximize this particular the objective function.

And, GA will try to find out through a large number of iterations like what should be the number of clusters to be made, what should be the the matrix for the C values and what should be the level of cluster fuzziness, so that this particular condition gets fulfilled and it will try to find out; the optimal clustering using the fuzzy C-means clustering.

(Refer Slide Time: 30:24)

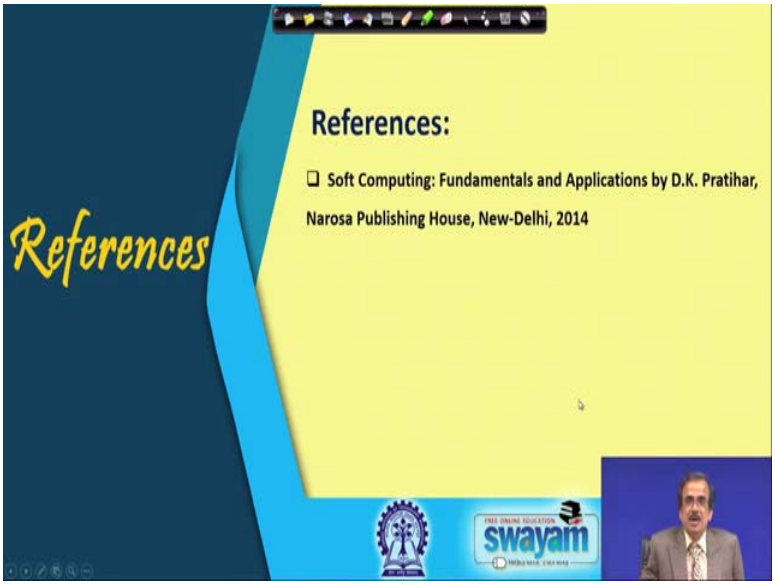


Now, next we try to see another method of clustering, which have been used, that is entropy-based fuzzy clustering. And, if I want to carry out the similar type of optimization, where our aim is to maximize the distinctness to maximize the compactness and to minimize the number of outliers. So, we can use the same principle for this entropy-based fuzzy clustering also. Keeping the same objective function, now here the design variables will be α, β, γ, this have been already discussed. Now, α

actually indicates the relationship between the Euclidean distance and similarity. And, β represents the threshold value of similarity; and γ indicates actually the outliers.

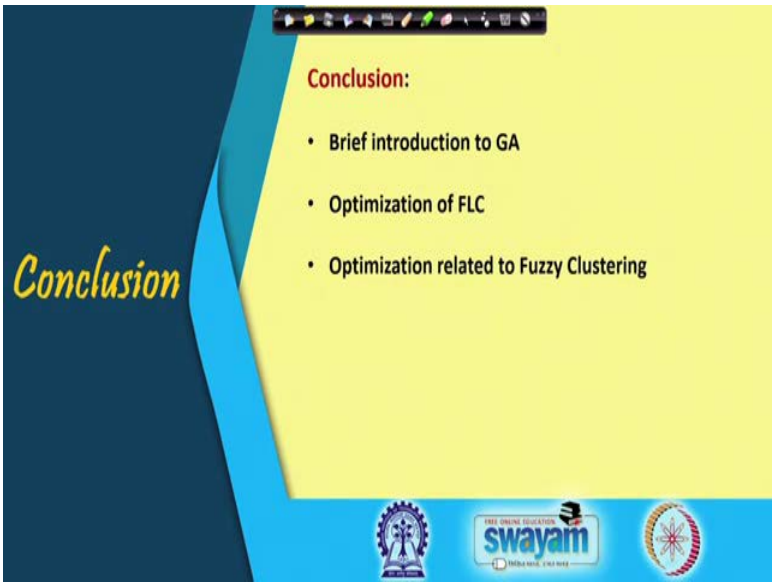
Now, in the GA string, if I consider say a particular GA string something like this, say this is one say GA string. So, if I consider, it can represent the value of α, then comes your β, and then comes your γ, and we can generate a population of solution. Now, corresponding to these α, β, γ, so it will carry out some sort of fuzzy clustering; and we will be getting the quality of clusters in terms of the distinctness, in terms of the compactness and we can also find out, whether there is any such outliers or not. And GA through a large number of iterations, we will try to find out that values of α, β, γ, corresponding to a particular say the datasets, so that it can ensure the optimal clusters.

(Refer Slide Time: 32:22)



Now, this is the way actually, we can carry out some sort of optimization, if I want to optimize the performance of fuzzy logic controller and performance of your fuzzy clustering tools. Now, this is actually the reference based on which so we carried out this discussion, that is, Soft Computing: Fundamentals and Applications by D.K. Pratihar. So, this is the textbook for this course. So, you can refer to this.

(Refer Slide Time: 32:51)



And, here, I just want to summarize, whatever we have discussed in this particular lecture. Now, at the beginning, we gave a brief introduction to the nature-inspired optimization tools particularly the genetic algorithm. We spent some time on optimization of fuzzy reasoning tool or fuzzy logic controller, like how to find out the optimal database, optimal rule base for the fuzzy logic controller, that we discussed in details and we solve some numerical examples also.

Now, next, in fact, we concentrated on how to optimize the clustering algorithms like fuzzy C-means algorithm or entropy-based algorithm, so that it can ensure the optimal clusters in terms of compactness, in terms of distinctness, and there should not be any such outliers.

Thank you.