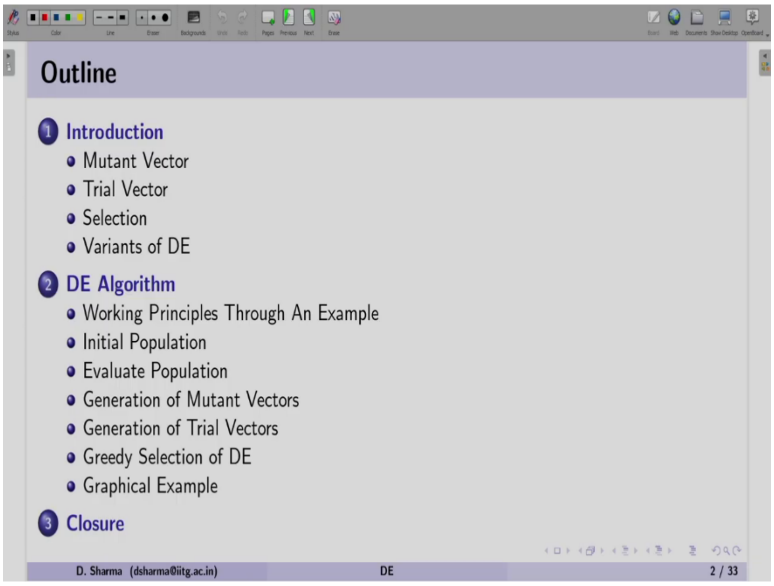
Evolutionary Computation for Single and Multi-Objective Optimization
Dr. Deepak Sharma
Department of Mechanical Engineering
Indian Institute of Technology, Guwahati

Lecture – 10
Differential Evolution

(Refer Slide Time: 00:39)



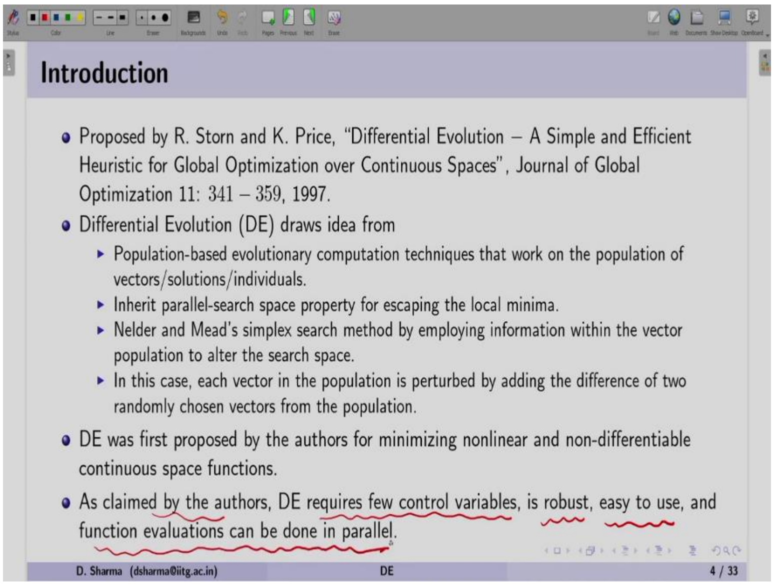
Welcome to this session on Differential Evolution. In this particular session, we will cover the introduction about the differential evolution; in differential evolution rather than we say solutions or individual, we have vectors. So, we are going to use different kind of vectors for perturbing the solution in the feasible search space.

So, as we can see that we will be having mutant vector, trial vector and when we are generating such vectors; we have a selection operator which is going to select the best solution from the population or while comparing one vector to the another vector. After going through the introduction of differential evolution, so we call it as DE.

So, we are going to understand this particular algorithm through an example. So, all working principle of DE; we will understand through the starting with the initial population generation, followed by the followed by the evaluation of the population. Then we will create mutant vector followed by the trial vector and then we will be performing some selection on these two newly generated vectors.

We will go through this hand calculation for one generation and the same set of solution we will be understanding through the graphical illustration and finally, we will conclude this session.

(Refer Slide Time: 02:19)



So, let us begin with the introduction now; in this introduction as we know, as we can see here the differential evolution is proposed by Storn and Price. In this particular; in this particular paper, they have mainly targeted developing differential evolution for continuous spaces.

In differential evolution, this DE draws ideas from; first of all it is from the evolutionary computation technique, it is because it has a population of vectors. So, when we have multiple vectors or multiple solutions; these solutions they interact in some sort of operations and then they generate new solutions; we create while creating the new solution we compare them and we keep the best solutions.

So, similar procedure is followed in differential evolution moreover; as we know that since population based EC computation techniques inherits parallel search space property. As you can see here, this particular property will help us escaping the local minima; it is because when we are performing some operations for creating new vectors, so that operation is performed on the set of solution.

In case, if few solutions are trapped at the local minima the other solutions will help this solution to come out from the local minima and move towards the global optima. Moreover, as it is mentioned in the paper that differential evolution draws an idea from Nelder and Mead's simplex search method.

In this case, it is done by employing the information within the vector population to alter the subspace. So, as we know; in the simplex search method if we consider for example, two variable problem. In this case we consider N plus 1; so basically three solutions, in these three solutions we find what is the best solution based on the fitness value, what is the worst solution and what is the next to the worst solution.

So, the idea of simplex search method is let us move away from the worst solution in such a way that we can improve the solution iteration by iteration. So, the same concept of using those vector operations is borrowed in differential evolution. So, as it can be seen that in differential evolution, each vector in the population is perturbed by adding the difference of two vectors randomly chosen vectors from the population.

So, in the slides which we will be discussing later; you can see that we are going to pick random solutions, making a difference of them, then adding to the third vector. So, the whole idea is to mutate if the given vector in the population so that we can generate a new solution in the population.

Now, yes it is given in the first paper that the DE was proposed by the author for the minimization of non-linear and non differentiable continuous space function. So, that was the main motive when DE was proposed; however, there are different variants of DE are available now that can not only solve such kind of a problem, but that can solve like mixed integer kind of a programming problem as well.

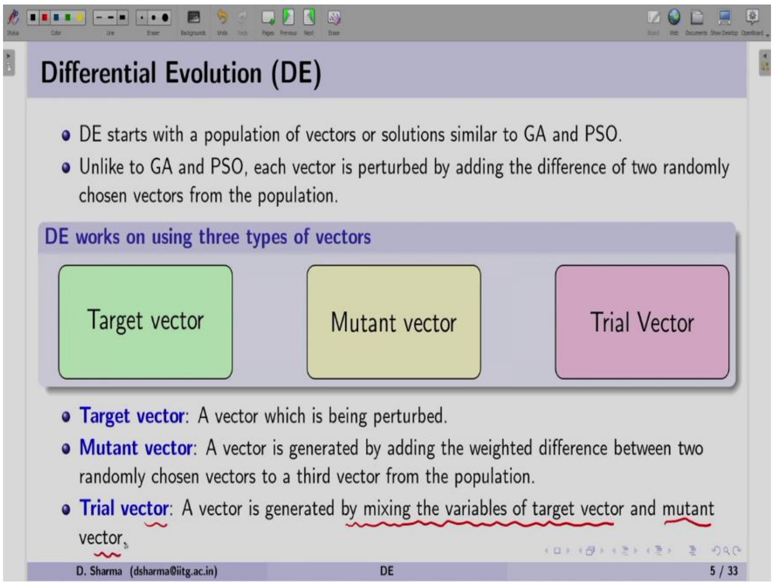
As mentioned by the as mentioned by the authors; DE require few control parameters. So, as we have understood that EC computation techniques need certain inputs to be set before running the algorithm. So, for DE also we need certain input parameters, but that number is quite small.

Second, it is robust; why because DE can be used for various kind of a problem to get an optimum solution. Similarly, it is easy to use because when we are randomly choosing vector and performing some kind of operation on it, you can see that those operations are simple say vector operations.

So, that why; that is why it is easy to understand and we can implement as well, and last but not the least; the function evaluation can be done in parallel. So, that is the advantage as we have discussed with evolutionary computation technique. So, that advantage is also available with DE in which when we have multiple solutions or vectors; so when we are going to calculate the objective function or constrain and assigning a fitness; so these operations are independent.

So, these set of operation on one solution and another solutions, they are independent to each other and therefore, we can perform this function evaluation for solution 1 and solution 2 in parallel. Now, let us understand how what are the features with the differential evolution.

(Refer Slide Time: 08:10)



Now, as we can see that DE starts with a population of a vectors and or we say solutions that is similar to GA and PSO and that is what we understand that evolutionary computation techniques are population based meta heuristic techniques. But unlike to GA and a PSO; each vector is perturbed by adding the difference of the two randomly chosen vector from the population.

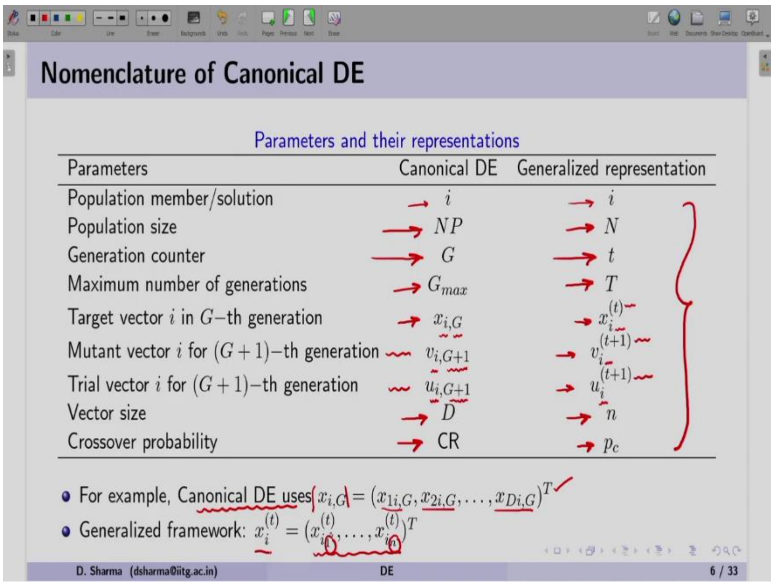
So, that is the new thing the way the new solution or a new vector is created in differential evolution. There are three kinds of vectors in DE; mainly we refer them as a target vector, mutant vector and the trial vector. So, what are those vectors? As we can see, the target vector is the one which is being perturbed.

So, we are going to choose one particular vector of a solution and we have to change the solution using some operation; then comes the mutation mutant vector this vector is generated by adding weighted difference between the two randomly chosen vectors to a third vector from the population.

So, as you can see; the multiple vectors are involved by using vector operations, we are going to generate the mutant vector. And the last is the trial vector, as we can see in the trial vector it is generated by mixing the variables of the target vector and the mutant vector.

So, this vector is generated when we are comparing the fitness and we will be exchanging the variables between the target and a trial vector.So, in this case; the probability will come into the picture by which we will be exchanging these two; we will be exchanging the variable between these two vectors.

(Refer Slide Time: 10:23)



Now, let us come to the nomenclature; it is because in the original paper some different nomenclature is used. As of now we are using certain kind of a nomenclature and we will be understanding a differential evolution using our nomenclature. So, let us see what are the parameters that are being used with DE and those parameters are represented in our generalized representation.

$$x_{i,G} = \left(x_{1i,G}, x_{2i,G}, \dots, x_{Di,G}\right)^T$$

$$x_i^{(t)} = \left(x_{i_1}^{(t)}, \dots, x_{i_n}^{(t)}\right)^T$$

So, let us begin with the population member. So, as you can see that in both the cases; we are representing the members as i. The population size; the second one in canonical DE, it is represented as NP; however, in our representation; it is represented as N; so, N will be represented as a population size.

Similarly, we have the counter; so this is called generation counter, it is represented as a G in DE, but we are representing as a small t. The maximum number of generation; this is represented by G max, in our representation it is represented by capital T. Similarly, a target vector; now, as you can see this target vector in the Gth iteration; the i and G are written in the subscript of x.

However, in our representation we write i in the subscript and the generation number t in the superscript. If and the mutant vector; so the mutant vector for t plus 1 again, in the DE representation; in the subscript you can see i and G plus 1.

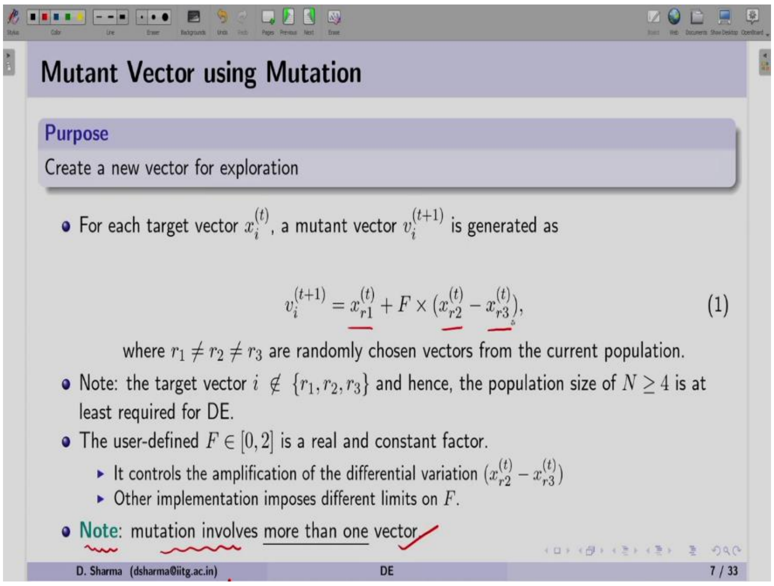
However in our representation; in the subscript we are writing the solution and in the superscript, it is written as the t plus 1th generation. So, v i t plus 1 will become our mutant vector. Thereafter, we have a trial vector and again in DE; we they have used i and a G plus 1 in the subscript; in our representation we are using u, but in the subscript is i and in the superscript, it is t plus 1.

The vector size in DE is represented as capital D, but as we know; we take this as a column vector and the size of the column vector is divided decided by this small n. There is a crossover probability rate. So, in DE; it is represented as a CR, we will be understanding this as a pc which is similar to the probability of crossover.

So, this is; so what we are going to do here is the representation which we are following in our generalized format that we are going to follow to understand how differential evolution works. Now, at the bottom we can see the canonical DE uses; suppose this is the vector here, x i G; this is represented and the component of this vector.

So, basically the values of the decision variables are given and again it is a column vector here. In our format, we are writing x i t and this is x i 1 to x i n; so we are writing these 1 and n in the subscript of i; just to make sure telling that we are talking about the variable of ith vector in tth generation.

(Refer Slide Time: 14:16)



So, let us start with the mutant vector. So, the purpose of the mutant vector is to create a new solution that will be exploring the search space. So, as we can see; we are going to create new vector for the exploration. Now, for each target vector; so we are going to take vector one by one; so let us take ith vector.

$$v_i^{(t+1)} = x_{r1}^{(t)} + F \times \left(x_{r2}^{(t)} - x_{r3}^{(t)}\right)$$

$$r_1 \neq r_2 \neq r_3$$

$$i \notin \{r_1, r_2, r_3\}$$

$$\left(x_{r2}^{(t)} - x_{r3}^{(t)}\right)$$

So, the mutant vector v i t plus 1 is generated as; as you can see equation number 1; that it depends on choosing the vector 1, then r 2 vector, then r 3 vectors. What are these vectors? These are the randomly chosen vectors from the current population, we have to make sure when we are selecting three vectors randomly; they should be different from each other.

Now, if we look at the equation number 1 here; the second part you can see, this second part represents the difference between the two vector that is multiplied by the scaling factor F. And finally, we are adding this component into the first vector; these are the simple vector operations which are performed to generate the mutant vector v i t plus 1.

While using while selecting r 1, r 2 and r 3 basically three random vectors, we have to make sure that these three vectors should not; any of them should not be the same as our target vector. So, therefore if we want to start differential evolution we need at least four members to start with.

So, as we can written; as we have written here, the population size of DE should be greater than or equals to 4 to start with this algorithm. Now, the scaling factor in the formula number 1; the factor F, generally we take it as a scalar factor and we take any value between 0 to 2. Now, what is the effect of this factor F?

As we can see, in the equation number 1 here; when we are finding the difference between the two vectors and when multiplying the scaling factor F; this is controlling the amplification of the difference differential variation. Since, we are subtracting these two vectors; so how much effect we are going to use it. If the value of F is smaller than 1, this means that we are going to take less effect of these two vectors; the difference of these two vectors.

However, when we take F greater than 1; then we are taking more component or the larger component of the difference of these two vectors. Other implementation imposes different limits on F; so this was the original implementation shown in the paper as we have seen in the initial slide.

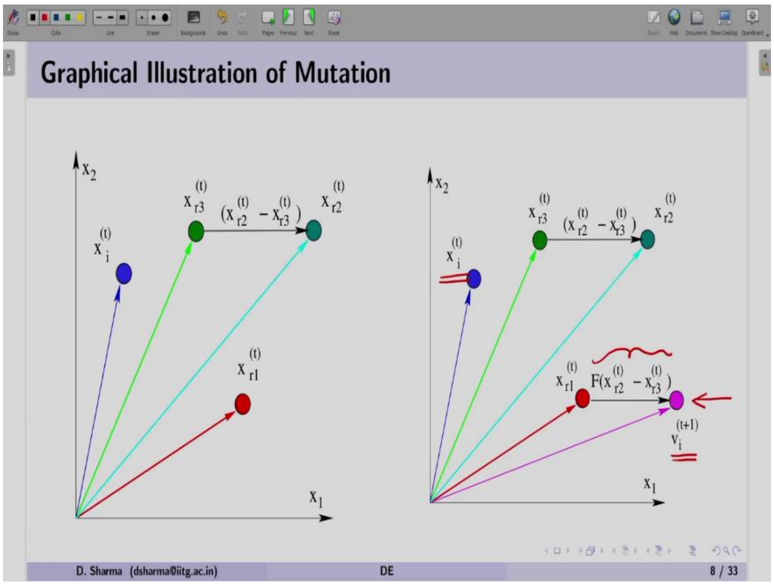
So, that says that if we are going to use a difference of the two vectors; we can use the value of F between 0 and a 2. Now, here there is an important note I have written at the bottom that in this mutation; it involves more than one vector.

Now, let us remember that when we are working with say genetic algorithms; in that case, we understood crossover and mutation operator as; in crossover operator, we need more than one solutions. It is only because these solutions they exchange some information and create new solutions.

So, as and when we need more than one solution that operator was referred to as a crossover operator. Similarly, when we are perturbing a solution; so only one solution is involved so that operator we considered as a mutation operator. However, in differential evolution; the mutation mutant vector which is generated using the mutation operator as referred in the paper that it involves more than one vector.

Now, looking at the equation number 1; we can see that we need r 1, r 2 and r 3; meaning that we need more than one vectors to generate a mutant vector v i t plus 1, let us understand the mutant vector graphically.

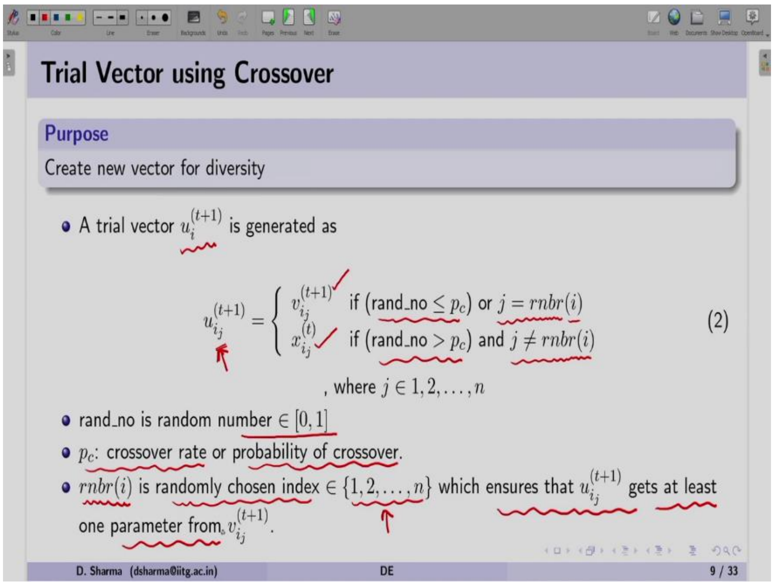
(Refer Slide Time: 19:08)



As we can see here, we have the target vector which is represented in a blue color and randomly chosen three vector; say r 1 and r 2 and r 3. So, the green color and the red color and the cyan color we have used three different colors to use three kinds of randomly chosen vector for a target vector i.

Now, as we know that we are going to use a difference; as you can see this particular difference in the vector, we are going to use it to create a mutant vector. Now, looking at the figure on the right hand side; we have now multiplied with the F and that F with the difference in the vector is creating the mutant vector v i t plus 1. So, this mutant vector and the target vector both of them will be used to create a trial vector.

(Refer Slide Time: 20:09)



Now, let us come to the trial vector; this trial vector is created using similar to the crossover concept. The purpose is; if we are using a trial vector that should help us in maintaining the diversity into the population. Therefore, as we can see the purpose is to create a new vector for diversity in the population.

$$u_{i_j}^{(t+1)} = \begin{cases} v_{i_j}^{(t+1)} & if\ (rand\ no \le p_c)\ or\ j = rnbr(i) \\ x_{i_j}^{(t)} & if\ (rand\ no > p_c)\ and\ j \ne rnbr(i) \end{cases}$$

$$where\ j \in 1, 2, \ldots, n$$

Now, trial vector u i t plus 1 is generated as; now before we explain this particular equation number 2, you can see that we are writing j here. So, the purpose of a j is; this is going to be our jth decision variable. So, variable by variable we are going to take a component either from the mutant vector or from the target vector and in that case while going through one by one, we will be creating the trial vector.

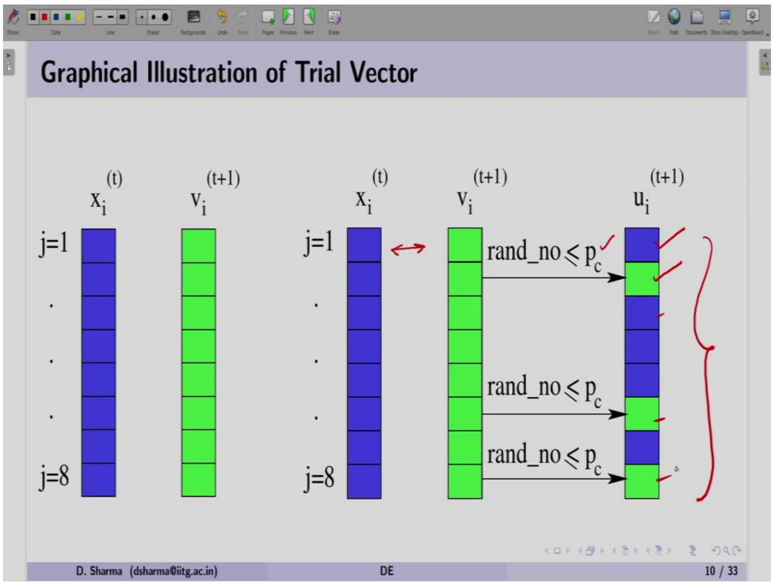
So, let us see the equation number 2 now. It says that say for the jth variable; if the random number is a smaller than pc or j is equals to rnbr i. So, we will see what is that; then we are going to take a jth component from the mutant vector. If the random number is greater than pc and j is not equals to rnbr i, we are going to take the jth variable component from the target vector.

So, here the random number is generated between 0 to 1, p c is the crossover rate or we can say the probability of a crossover. Now, let us come to rnbr i; this is a randomly chosen index between 1 to n. Now, you can see these are the index we are used for the decision variable.

It means that we are creating a random number and as and when this random number is same as the decision variable which we are taking; then we are copying that particular decision variable from the mutant vector to the trial vector.

The only purpose of this particular condition is that; it ensures that the trial vectors; so the jth component or trial vector get at least one parameter from the mutant vector. So, that is the whole purpose that we should get something from the mutant vector while creating a trial vector. So, let us see graphically how this trial vector is generated now.

(Refer Slide Time: 23:02)

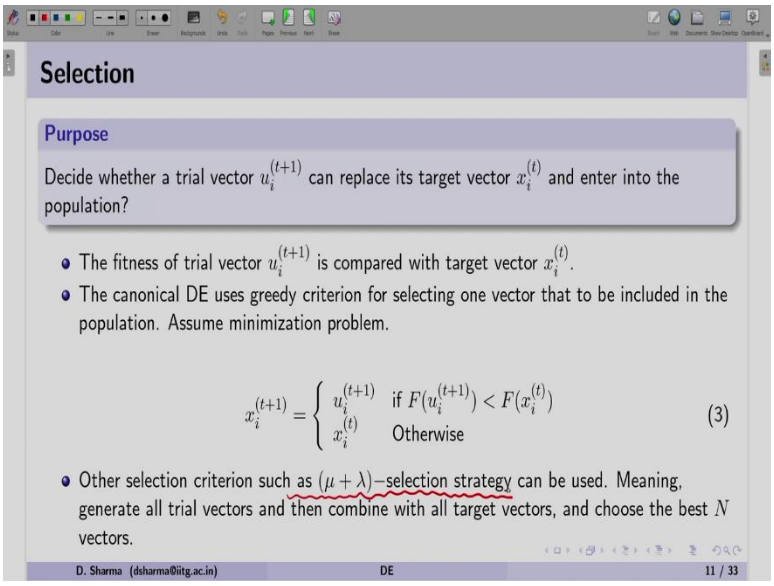


So, in this particular example; you can see on the left hand side, we have the target vector, we have the mutant vector and this vector is made of 8 number of variables. Different color codings are used here to understand how this trial vector is generated; looking at the figure on the right hand side; so suppose for the first variable, so the random number is more than pc.

So, this means that we are going to copy the variable; the first variable from the target variable. Suppose, for the second component, random number is smaller than pc. So, we are going to copy the second variable from the mutant vector. And similarly, if we are going to follow we are we will get a trial vector.

Now, here you can see in the final component of the trial vector; it is made of blue color and the green color. So, the change in the blue and a green color means that this particular vector will be having different set of decision variable values. And when we are changing it; definitely it will be representing a new particular vector or a solution in the population. After understanding, the mutant vector and the trial vector; now we have to select which is better.

(Refer Slide Time: 24:42)



So, in this case; we have this selection. In this particular selection the purpose is we have to decide whether a trial vector can replace its target vector and enter in the population. So, since we have generated the trial vector; we are going to compare with its target vector.

So, as we know when we are going to compare; we generally compare with the help of fitness value. So, the fitness value of both the vectors we have already calculated it.

$$x_i^{(t+1)} = \begin{cases} u_i^{(t+1)} & \text{if } F\left(u_i^{(t+1)}\right) < F\left(x_i^{(t)}\right) \\ x_i^{(t)} & \text{Otherwise} \end{cases}$$

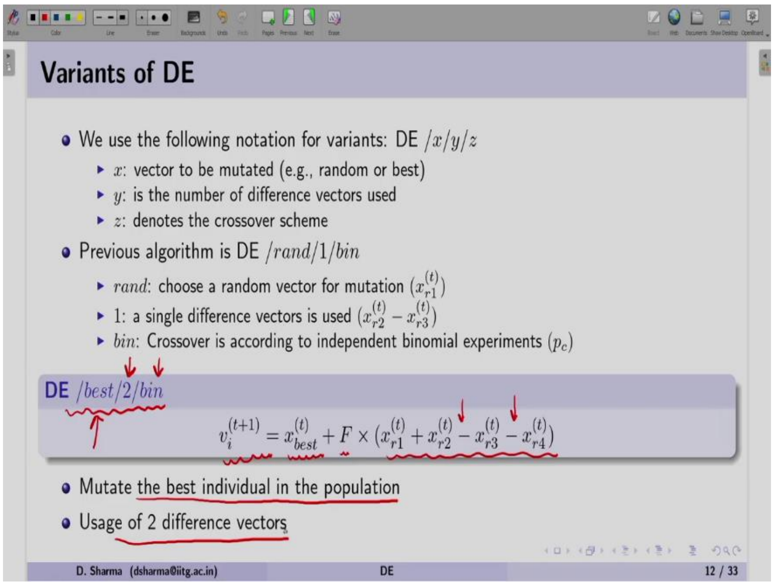
Now, the canonical DE uses some greedy selection criteria for selection of one vector that to be included in the population. So, here we are assuming it is a minimization problem. So, let us look at the formula number 3 here; it says that the target vector for the next generation population which is t plus 1, it will come if the fitness of the trial vector is better than the target vector; so let us copy the trial vector.

Otherwise, we will be copying the target vector as it is into the next generation. Now, what you can find here or we can understand from here? That we are taking the trial vector and the target vector and we are comparing these two solutions and choosing the best.

However, if we want to make a small change into this algorithm; we can use some other selection criterion. Now, as we can see at the bottom as an example, I have written we can use $(\mu + \lambda)$ selection strategy meaning that we can generate we already have a target vector with us, we can generate all the trial vectors using mutant vectors.

So, now we have target plus trial vectors; we can combine both of them and then we can choose the best n solution for the next population and that is what mu plus lambda strategy says that. So, this is the another way of selection we can incorporate with DE to see whether the performance of the DE is improved for certain class of a problem or not.

(Refer Slide Time: 27:10)



There are different variants of DE available in the literature. So, generally those variants are can be differentiated with respect to a form. As we can see in this here that the following notation is used which is represented as DE; x, y and z. So, here x will be representing a vector that to be mutated.

$$v_i^{(t+1)} = x_{best}^{(t)} + F \times \left(x_{r1}^{(t)} + x_{r2}^{(t)} - x_{r3}^{(t)} - x_{r4}^{(t)}\right)$$

So, for example, we can either use random or the best; why is the number of differences vector that can be used with DE and z will be representing the crossover scheme. So, the algorithm which we have gone through; it also has had some version. So, what is that? As we can see, the algorithm which we have just discussed; it has DE; rand random basically, this is 1 means one difference and bin means binomial; so let us see that.

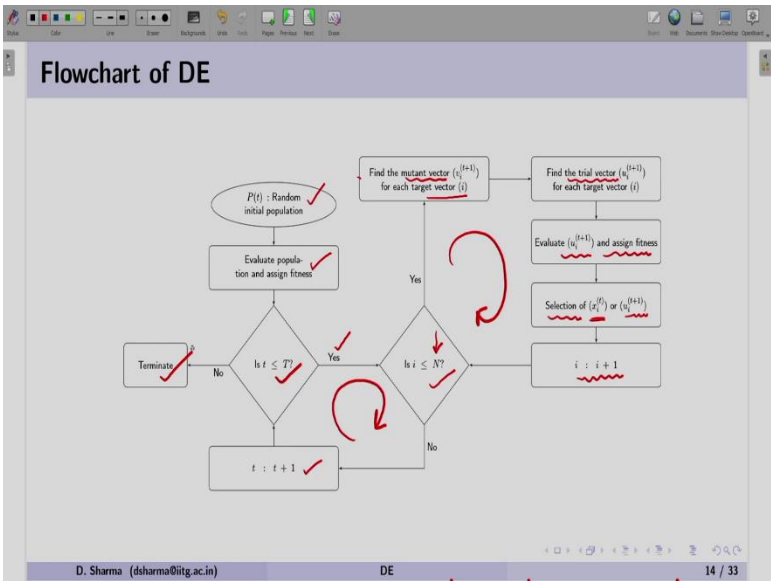
So, random means we are going to choose a vector for a mutation. So, we selected r 1; 1 means a single difference vector is used. So, we have used a difference between r 2; r 2 minus r 3. So, this is the one difference between these two vectors and bin; this is the crossover is according to the independent binomial experiment and that is why we took it as a pc here.

So, what could be the other variant for a DE? In this case, suppose we take DE; best 2 bin. How we can understand what kind of DE it is? So, the first quantity says; so the first term here it is best. So, you can see in the formula here v i t plus 1 is equals to x best. So, meaning that in the given population; we will be finding which vector has the best fitness that vector will be chosen here.

Then we have plus F and then now we are using four vectors in which there are two differences used. So, you can see that the two numbers; so DE best 2 is going to represent, there are two vector differences in the formula and bin is the same as what we have understood earlier.

So, the mutate the best individual in the population and uses of two difference vector that represents DE; best 2 bin. And similarly in the literature, if we change this particular notation; accordingly we are going to get another kind of a DE that is the variant of the original DE which we have understood it. Now, we have come to this stage when we can see how this DE algorithm will work on an example. So, we are going to start this DE algorithm with the help of a Rosenbrock example now.

(Refer Slide Time: 30:34)



Before going that, let us see the flowchart of differential evolution; in this particular flowchart, we can see that we start with the random initial population; so, the vectors that will be generated randomly as we have done it earlier. Thereafter, we are going to evaluate the population and assigning the fitness.

So, here since the problem which we will be solving; we will see that what fitness we are going to assign. Now, come to the first decision box here. So, this decision box says that t is smaller than N equals to capital T meaning that the number of generation, if it is smaller than and equals to capital T; go ahead.

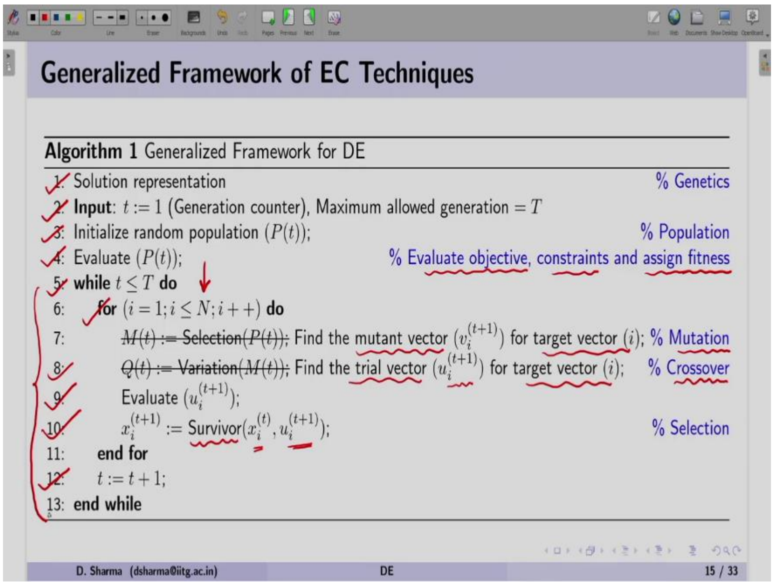
So, yes; so suppose we are at the first iteration here; now we are going into the second decision box. As you can see, this decision box will be running for every vector in the population. So, let us choose; we have the ith decision vector, so for that particular target vector; so the vector i will be referred as a target vector. So, first we will be finding the mutant vector here.

So, the mutant vector formula is known to us; thereafter we are going to find a target trial vector u i t plus 1 using target, as well as mutant vector. Since, this trial vector can be a new one; so we have to evaluate it and assign a fitness to our trial vector. Thereafter, we will be performing a selection between the target vector x i t and the trial vector u i; t plus 1.

Then finally, we are going to increase the counter by 1 and this loop will be working till all the vectors in the populations are undergone through the operations, as creating mutant vector, trial vector and then selection. Thereafter, once the population is done then we will be moving in the upper loop on number of a regeneration; so, we increase the counter by 1.

So, this process will be followed till all the number of generation is more than the maximum number of generation. Once it is done, we terminate the algorithm and report the optimum solution. After understanding the flowchart of differential evolution; now, let us fit DE with our generalized framework of EC computing technique.

(Refer Slide Time: 33:31)



Now, in this case we can see that we have this step 1 which is the solution representation. So, as of now we are assuming that we are using DE for continuous function; for real value and for the real values of x 1, x 2, x 3 and x 4; thereafter we have to give certain input to the DE.

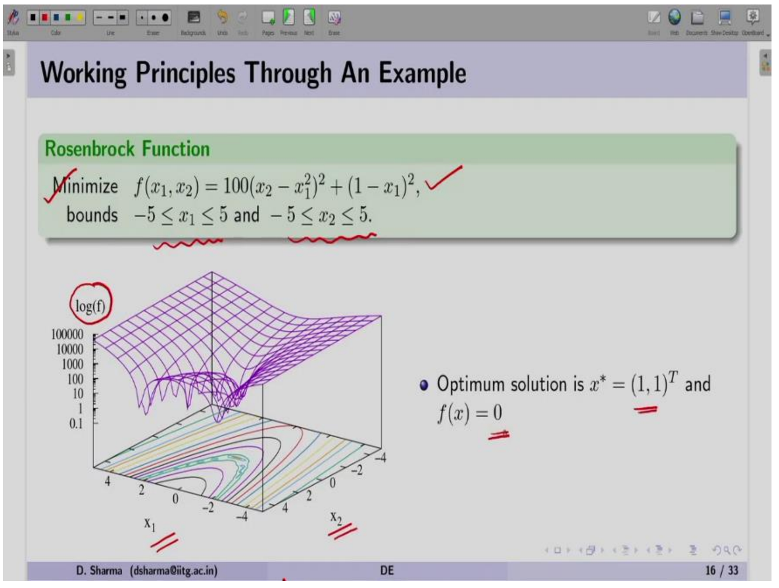
So, as we know that for evolutionary computation techniques; we have to set certain input that will be given to the differential evolution. Thereafter, in a step number 3; we initialize the random population and in step number 4, we evaluate the population. As can be seen; evaluation of the population means we will be evaluating the objective function, we will be evaluating the constraint and finally, we will be assigning the fitness to each vector.

So, as we remember; in DE the solution is referred to as a vector; as a vector. Now, in a step number 5; we are in the standard loop of generation, inside this particular loop; we have one for loop, this for loop works for the number of vectors that is represented by capital N. So, we are going to take vector one by one; so those vectors will be considered as a target vector and once.

So, for that one first we will be creating a mutant vector; for the target vector i, the operation is referred as a mutation; then in step number 8, we will be finding a trial vector; u i t plus 1 for target vector i, this operation is referred as crossover operator. Since, new vector is generated; so in step number 9, we will evaluate this new vector and in step number 10, we have the survival stage.

This survival stage will be working on two solutions only that is the target vector and the trial vector u i t plus 1; whosoever is having a better fitness is selected for the next generation and in the step number 12, we increase the counter by 1 and the loop over the number of generation from 5 to 13 will terminate, once the small t is greater than the capital T.

(Refer Slide Time: 36:21)



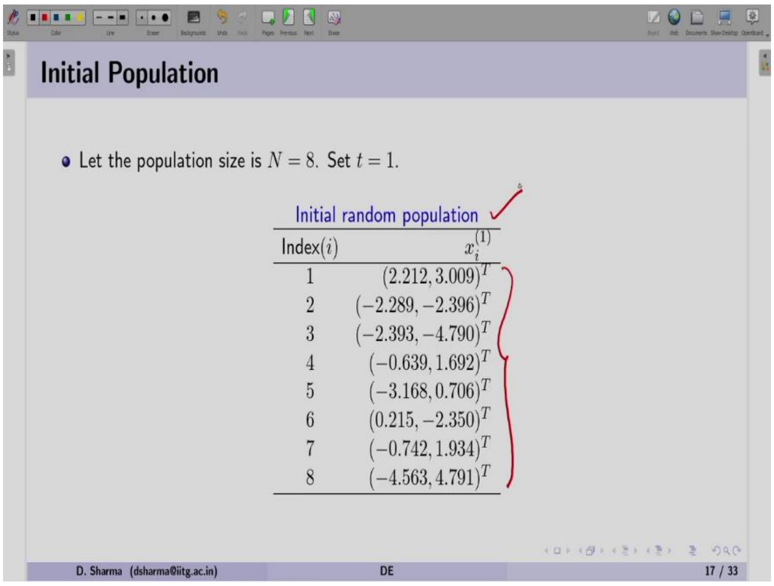
Now, let us understand the working principle of differential evolution. So, we have taken an example of Rosenbrock function. We can see on the top that we want to minimize the function and the two variable Rosenbrock function is given here and the variable bound for the Rosenbrock function is considered as between minus 5 to plus 5 for both the variables.

$$Minimize \quad f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

$$\text{bounds} \quad -5 \le x_1 \le 5 \quad and \quad -5 \le x_2 \le 5$$

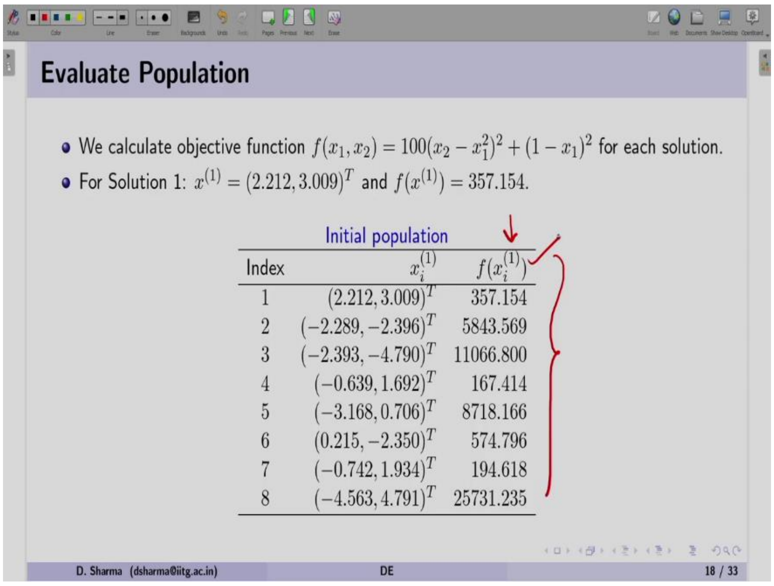
Now, the figure on the left hand side; we have drawn the third axis as a logarithmic of the function, you can see x 1 and x 2 plane at the bottom. Now, the surface you can see that this particular surface has generate the contours which shows that we have many local optimum solution for the given problem; However, the global optimum solution is at 1 and a 1 and the function value at this global optimum solution is 0.

(Refer Slide Time: 37:20)



So, let us solve this problem using differential evolution; so we have to generate the initial population first. So, let us assume that we have a population size of 8. So, as you remember; DE need at least four number to start with. So, we are taking 8 as a number and the generation counter is considered as first. So, what we did is; we created our random population as you can see in the table that x 1 and x 2 values for each index i is generated between minus 5 to plus 5.

(Refer Slide Time: 38:01)

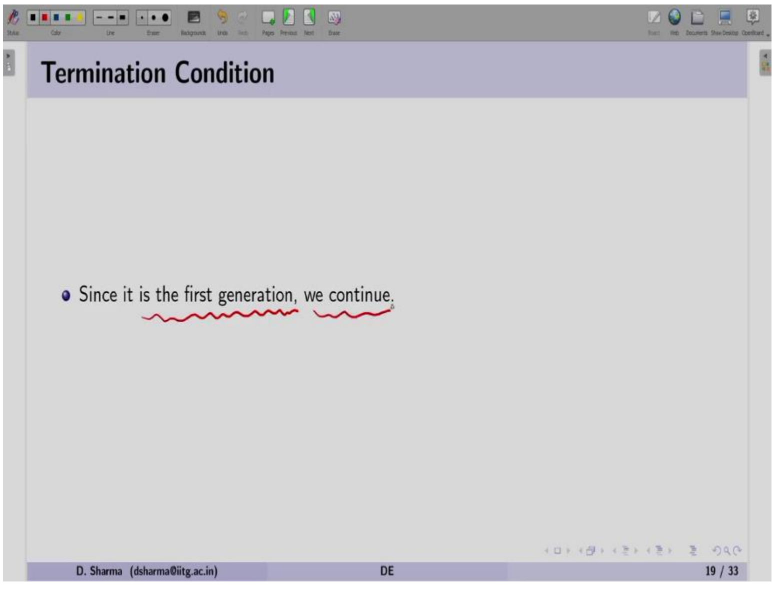


Once this initial random population is generated; now we have to evaluate the population; for evaluating the population the objective function is given to us. Now, let us take a solution number 1; for this particular solution, the column vector or the variable values are given here. By including these values we will get the function value is 357.154. If we continue to fitting the values of the variable for vector 1, vector 2, vector 3 till vector 8, we are going to get the fitness of all the vectors.

$$x^{(1)} = (2.212, \&3.009)^T \ and \ f\left(x^{(1)}\right) = 357.154$$

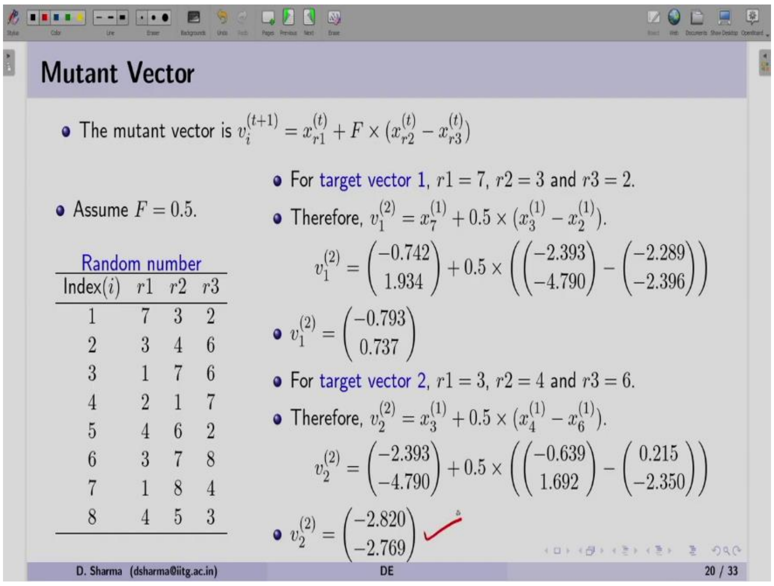
So, as can be seen in this particular table here; the third column represents the fitness or the function value of all the vectors. So, here we are assuming that the fitness function is the same as the objective function value. So, we are considering the function value as our fitness in this particular example.

(Refer Slide Time: 39:07)



Now, let now we are in the standard loop of the number of a generation; since this is the first generation. So, we continue and we move inside the loop of generation.

(Refer Slide Time: 39:23)



So, we are going to do the operations now; vector y vector. So, let us take; so first operation is the mutant vector. So, the formula is given to us which is represented as the vector operation; so, the difference between the vector and adding to the third vector. Now by for performing the mutant vector we are assuming that F is 0.5.

$$r1 = 7, r2 = 3 \; and \; r3 = 2.$$

$$v_1^{(2)} = x_7^{(1)} + 0.5 \times \left(x_3^{(1)} - x_2^{(1)} \right)$$

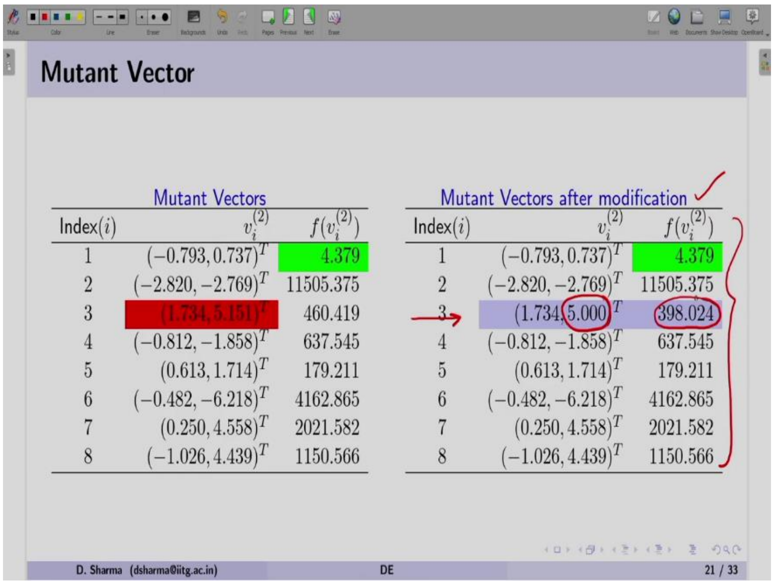
Now, let us see the table now; for the target vector 1, we have chosen three vector randomly; so these are 7, 3, 2. So, what we did? Inside the population, we chose three vectors randomly those are 7, 3 and a 2. Similarly, as you can see in the table for target vector 2, the other three vectors are chosen. So, we are following the route that i should not be equals to r 1, r 2 and r 3; as well as r 1, r 2 and r 3 should be different from each other.

So, if we follow the same thing; then we are going to get a table of random numbers for each target vector. So, let us go one by one; let us choose the target vector 1, for the target vector one r 1 is 7, r 2 is 3 and r 3 is 2; so that you can make it out from this particular table.

Now, we are going to calculate the mutant vector for this target vector 1; so, it is simply the vector operations as you can see. So, we are including the value here in the column vector form and then finally, we will get the value or the mutant vector 1, for the target vector 1. Just to explain one more time, let us do the same calculation on target vector 2.

So, for this target vector 2; we have r 1, r 2 and r 3; that is we have selected from this particular table as 3, 4 and 6. By including their vector values or the column decision variable values here, in the column vector form; I can find, what is the value of the mutant vector for target vector 2 and finally, we will get the value here. So, in this particular process we are going to take one vector; one target vector at a time and then creating the mutant vector for the same target vector.

(Refer Slide Time: 42:14)



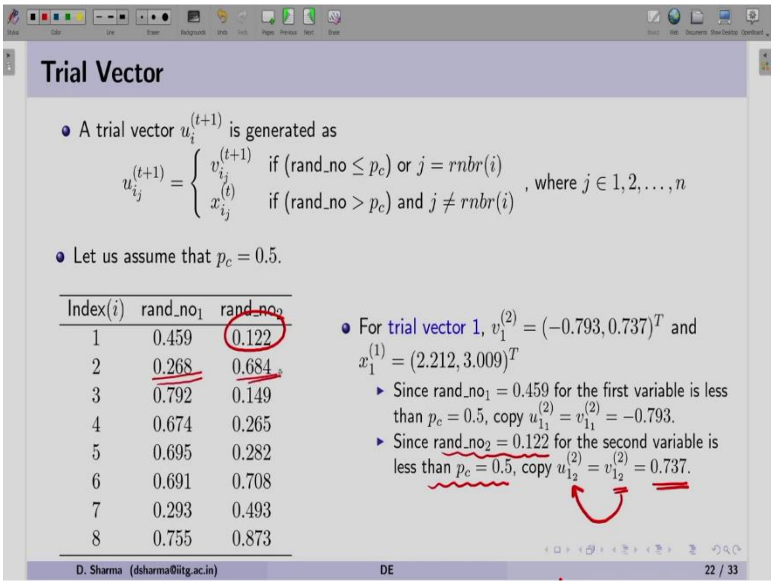
Now, here when we will be performing it; we are going to get a table as mutant vector. So, there are two observation here; so let us see the column number 2 here. Now, the column number 2 is represented in a red color is shown in the red color, it is because look at the x 2 value. This x 2 value is actually greater than 5 and in our problem; we consider x 1 and x 2 should lie between minus 5 to plus 5. So, in this case when the variable is going out of the bounds; we will keep this particular variable on the bound.

So, that is the first observation that the operations which we did earlier that can make our vector out of the bounds of the variable. Second observation is shown in the third column that is represented in the green color. Now, in this green color what you can see that we started with a fitness with initial population which have certain fitness.

After performing the mutant; after calculating the mutant vector, you can see the fitness has already improved for vector 1. And that is what we expect is that when we are performing the operation, our solution should improve and should move to the optimum solution. So, the mutant vector can create a better solution for us and it can create bad solution as well.

So, as a modified table; as we can see on the right hand side, the values corresponding to the vector 3; you can see that the x 2 component is kept on 5 and accordingly, the fitness value is also changed. With this modified mutant vectors, now let us create the trial vectors.

(Refer Slide Time: 44:13)



Now, as you can see here; the target trial vectors will be made with the help of every variable called j. So, using this particular formula as given in the top; we will be generating the trial vectors. For creating the trial vectors, we are assuming the probability of crossover or a crossover rate is 0.5.

$$u_{i_j}^{(t+1)} = \begin{cases} v_{i_j}^{(t+1)} if \quad (rand\ no\} \le p_c) \ or \ j = rnbr(i) \\ x_{i_j}^{(t)} \ if \quad (rand\ no \ > \ p_c\ ) \ and \ j \ne rnbr(i) \end{cases}$$

$$v_1^{(2)} = (\& - 0.793\&, \&0.737\&)^T \text{ and } x_1^{(1)} = (2.212, 3.009)^T$$

So, that says that 50 percent should come from the mutant vector and 50 percent should come from the target vector. So, let us see how; now in this case, as we know we are solving two variable problem that is x 1 and x 2. So, for both the variables; we are creating random numbers as you can see in the table.

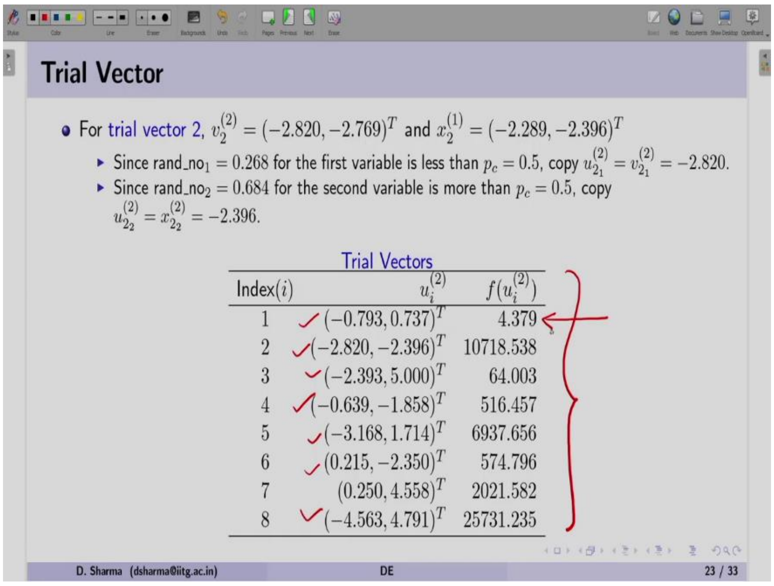
So, for every vector 1 to 8, we are creating random numbers to find out which particular component of the decision variable will be copied into the trial vector. So, let us take the trial vector 1, in this trial vector; we have already computed the mutant vector and we had the target vector. Now, let us look at the random numbers.

So, random number 1 is 0.459; as you can see from the table as well. So, the first random number is 0.459. Now, since this random number is smaller than pc which is 0.5 and looking at the formula on the top; we are going to get a component. So, the first variable value, we are copying from the mutant vector to the our trial vector.

So, the first variable will be minus 0.793. Now, let us do the same calculation for the variable number 2. So, the random number is now 0. 12; 0.122 and from the table also, you can see the same random number is taken here.

Now, since this random number is smaller than probability of a crossover or pc; we can take the second component of the mutant vector to the second component of the trial vector; so, this is 0.737. This is we did it for the; for creating a trial vector 1. Let us perform the same calculation for the trial vector 2.

(Refer Slide Time: 46:56)



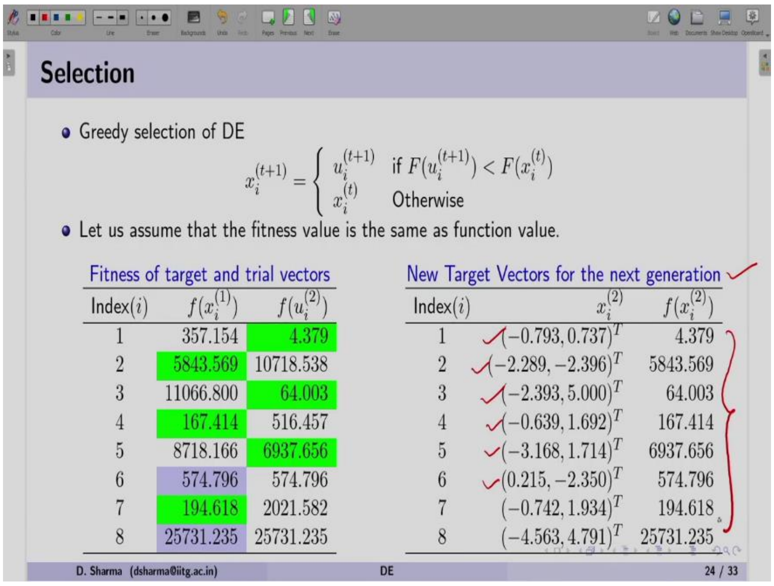
As you can see in the table, the random numbers are given here; so, these random numbers we are going to use them. For trial vector 2; we have the mutant vector v 2 and we have a target vector x 2. In this case, as we have seen the first random number for the first variable is 0.268 which is less than the pc meaning that; the first variable from the mutant vector will be copied to the trial vector.

Similarly, if we are going to do the same thing for the second variable; so here random number is 0.684. Now, this random number is more than pc; it suggests that we have to take the second component from the target vector into the trial vector; so, the component as you can see here.

So, what we can see? That this target trial vector can take a component either from the mutant vector or from the target vector; by following those set of random numbers we can thus generate the trial vector. Now, looking at the table here; this particular table says that based on the random number, these trial vectors for every solution we have created.

So, the fitness value is shown for our reference and we can see that the best solution was preserved, but it is random. In this case, we have preserved it, but it can be distorted or it can be changed to some other value.

(Refer Slide Time: 48:40)



Now, we have created a trial vector; so in the at this particular stage; we have the target vector, as well as the trial vector. Now, we have to choose the best and therefore, we need a selection. So, in this selection operator; we are using the greedy selection of a DE, as we have understood from the earlier slide which says that the target vector for the t plus 1 generation that is the next generation is equals to the trial vector F, the fitness of the trial vector is better than the fitness of target vector.

$$x_i^{(t+1)} = \begin{cases} u_i^{(t+1)} & \text{if } F\left(u_i^{(t+1)}\right) < F\left(x_i^{(t)}\right) \\ x_i^{(t)} & \text{Otherwise} \end{cases}$$

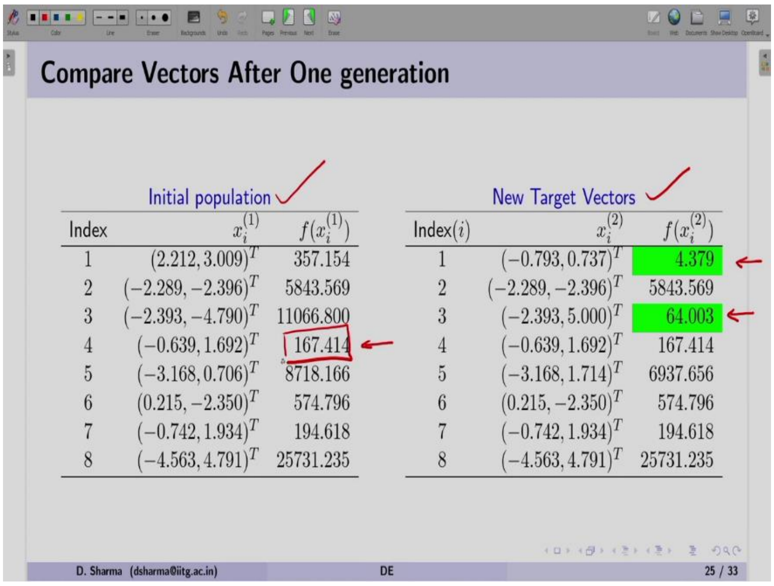
Otherwise, we are going to copy the same target vector. Here, we are assuming that the fitness value is the same as the function value. So, this assumption we are using it and we will be choosing the solution one by one; let us see how. So, for the first solution; we will be comparing the target and the trial vector; looking at the fitness, we can see that we have to select the; we have to select the trial vector for solution number 1.

Now, look at the solution number 2 here; in this case the fitness value of the target vector is better than the trial vector; so this is going to be selected. This process we will continue and we will get this kind of a table here. Here, you can see all the green particular cells says that which particular vector we are going to select.

Here, some different color such as in 6 and 8 are shown; it is because our target vector as well as our trial vector both of them are same; since both of them are same, we can select any one. So, that is why I am just showing as a representation purpose that anyone can be selected because it is not going to change our member.

So, the new target vector for the next generation you can see on the right hand side and the vector components are shown in the column number 2 and their fitness values are shown in the third column of the table.

(Refer Slide Time: 51:16)



Index	$x_i^{(1)}$	$f(x_i^{(1)})$
1	$(2.212, 3.009)^T$	357.154
2	$(-2.289, -2.396)^T$	5843.569
3	$(-2.393, -4.790)^T$	11066.800
4	$(-0.639, 1.692)^T$	167.414
5	$(-3.168, 0.706)^T$	8718.166
6	$(0.215, -2.350)^T$	574.796
7	$(-0.742, 1.934)^T$	194.618
8	$(-4.563, 4.791)^T$	25731.235

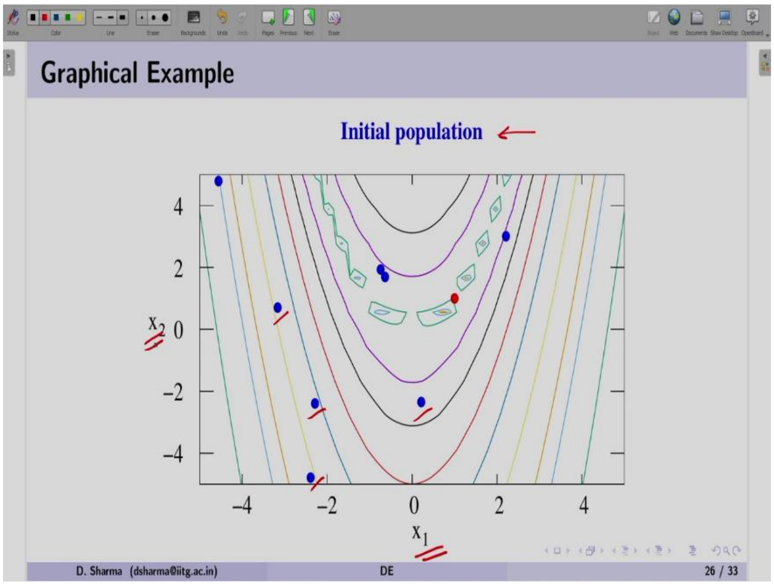
Index(i)	$x_i^{(2)}$	$f(x_i^{(2)})$
1	$(-0.793, 0.737)^T$	4.379
2	$(-2.289, -2.396)^T$	5843.569
3	$(-2.393, 5.000)^T$	64.003
4	$(-0.639, 1.692)^T$	167.414
5	$(-3.168, 1.714)^T$	6937.656
6	$(0.215, -2.350)^T$	574.796
7	$(-0.742, 1.934)^T$	194.618
8	$(-4.563, 4.791)^T$	25731.235

Just at the end let us compare the initial population versus the new target vectors after one generation. Now, as you can see the left hand table represents the initial population and the new target vector after one generation is shown on the right hand side. The observation is that we are we have now two solutions which is better than the best solution in the initial population.

And that is what we expect from differential evolution that the way we perform the mutation in the form of mutant vector, as well as crossover in the form of trial vector; these operation should support DE to improve the solution generation by generation.

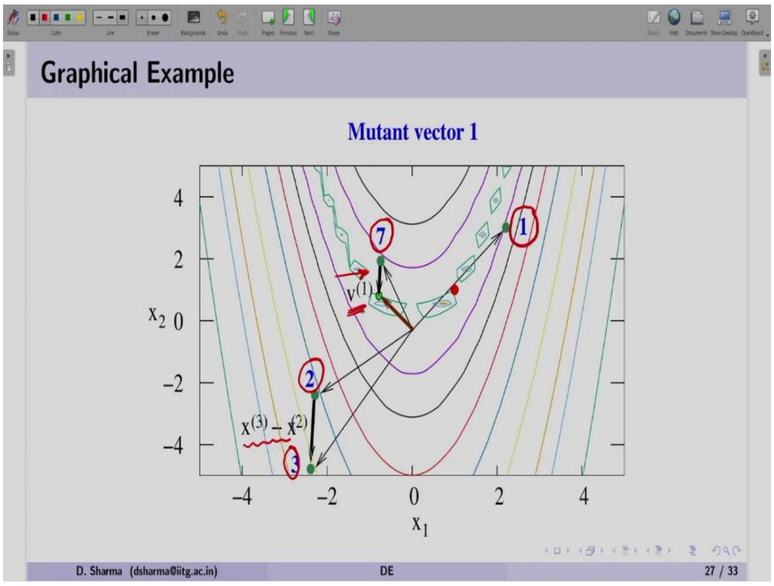
And after few generation, we should expect that these target vector will be close to the optimum solution. Now, the process which we have gone through using our hand calculation; now let us see how the solution graphically are moving in a one generation.

(Refer Slide Time: 52:37)



So, we are we are starting with the graphical example; now let us look at this figure, this is the initial population and this initial population all these vectors are generated randomly in the plane of X 1 and X 2.

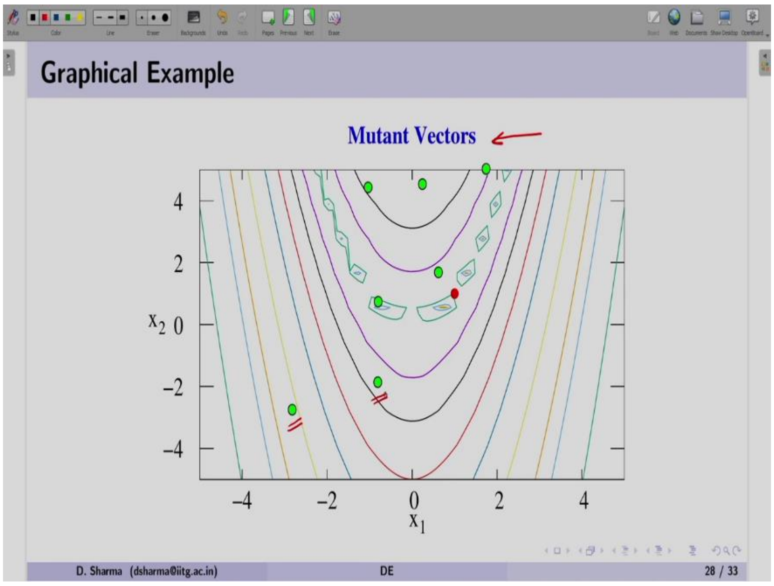
(Refer Slide Time: 52:56)



Once we evaluated that and then we are under the; we are inside the generation loop, the first operation which we do is the mutation in the form of mutant vector. So, let us take an example of first target vector 1. Now, look at the figure here; the target vector 1 is here; for this particular vector we selected r 1 as 7, r 2 as 3 and r 3 as 2.

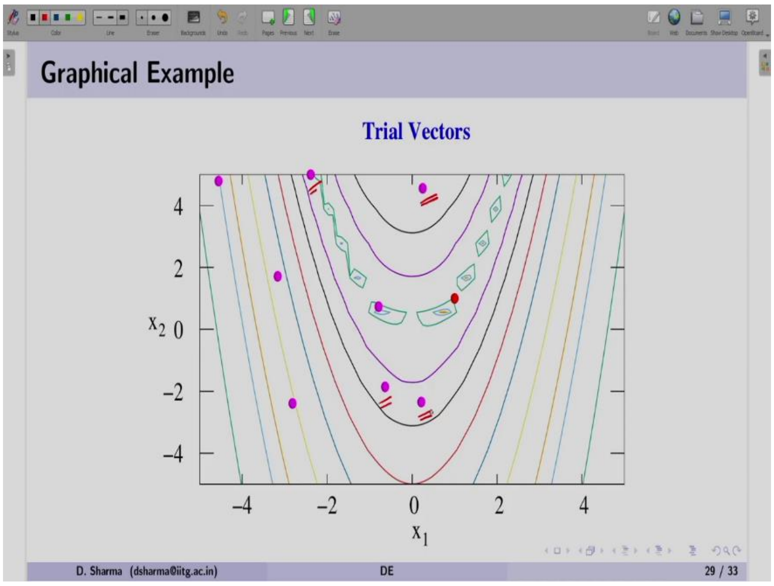
So, what we are going to do is; we are first finding the difference between the two vector which is X 3 minus X 2 and then and then multiplying with the F, we are adding this vector to the vector 7; that is going to give me my first mutant vector v 1 and that is that process will be happening for all the target vector in the population.

(Refer Slide Time: 53:56)



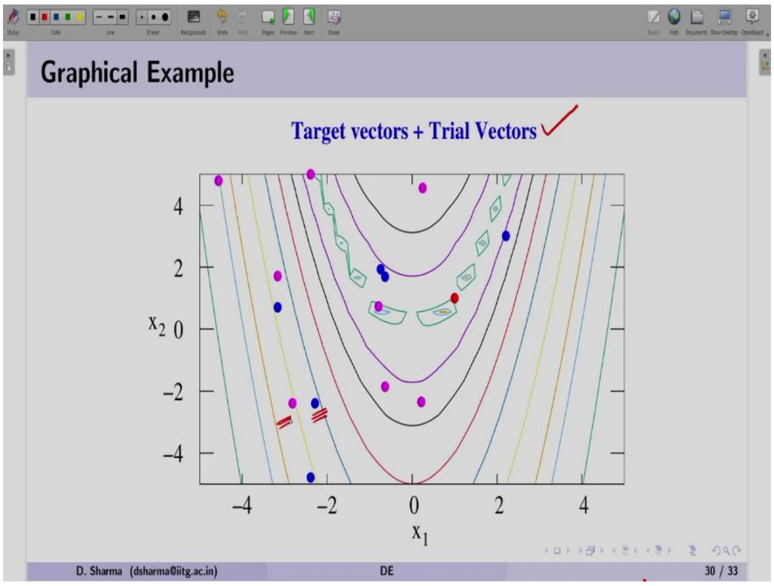
Now, in the figure; you can see that all the mutant vectors are shown in these green colors. So, that are generated using the formula that we know for mutant, for mutation.

(Refer Slide Time: 54:15)



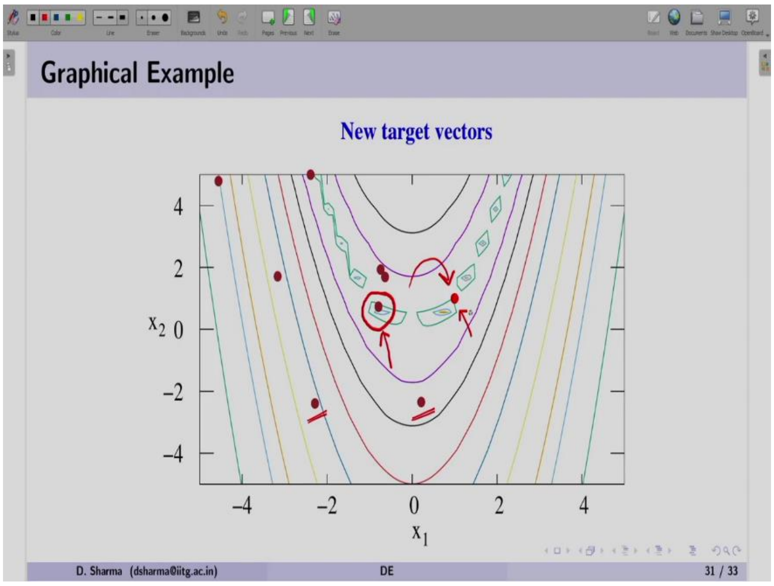
After performing this mutation and creating the mutant vectors, now let us create the trial vectors. So, following the hand calculations; we use the formula and the tables which is shown in the trial vector, the solutions are shown in the pink color; as you can see here. So, these are the solutions that are created after trial vectors. Now, the process which is left is the selection; so we have to either select our target vector or a mutant vector, just for our reference I am showing you both the solutions now.

(Refer Slide Time: 54:42)



So, the target and trial vectors are shown. So, the blue color solutions are the target vectors and the pink color solutions our trial vectors. So, here the solutions will be compared and we have to select the best solution.

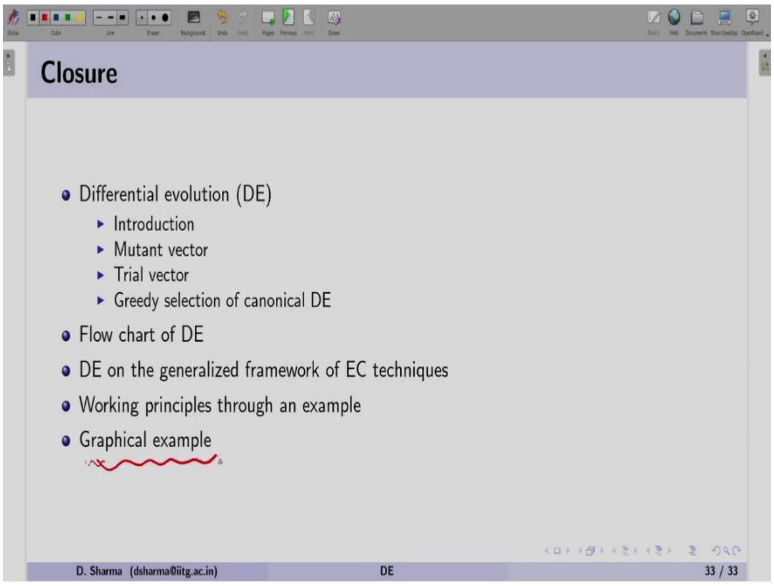
(Refer Slide Time: 55:06)



Now, after selecting using greedy selection method; as you can see these brown points these points are the new target vectors. So, what we can see? That we started randomly in the X 1 and X 2 plane and after one generation, the solutions are start moving towards the optimum solution; let us look at this particular solution. So, you remember that one solution was having a fitness close to 4, it is because this particular solution as you can see in the figure; it is close to one of the local minima in this problem.

So, this solution will be trying to converge to this local minima, but using the parallel search space property of evolutionary computation; the other solution will help this solution to come out and then move towards the optimum solution, as it is shown in the red color. So, this way generation by generations these vectors will move towards the optimum solution for the problem.

(Refer Slide Time: 56:26)



So, here we have come to the conclusion of this particular session. So, what we have learned in this session about differential evolution is first we introduce the differential evolution. So, what we understood that DE is borrowed the idea from evolutionary computation. Why? Because it uses multiple solutions or vectors and then the operations were performed to create new solutions, those new solutions were compared and the best one was kept.

Second idea was borrowed which is similar to the vector operations in the simplex search method. So, borrowing these two idea differential evolution came with the concept of mutant vector. So, as we understand that this particular mutant vector was generated by taking a difference of randomly chosen two vectors, multiplied by the scaling factor F and this particular component is now multiplied with the third component for mutating target vector.

Once the mutant vector was generated, then we had a formula for a trial vector where with the help of probabilities or the random numbers; we are taking the variable either from the mutant vector or from the target vector, to create a trial vector. And we also showed the greedy selection of the canonical DE and the; this says that the best of target or trial vector should be chosen based on the fitness value.

Thereafter, we understood the flowchart of a DE; in this case we had two loops, one is the decision space decision box with respect to the number of regeneration, another decision box with respect to the number of solutions in the population. So, after following that; we have fit the flowchart of a DE or we have explained the flowchart of a DE.

Thereafter, we fit DE on the generalized framework; so in that generalized framework we understood the operators and the processes what we have changed and then accordingly we created the mutant vector, the trial vector for each target vector i.

Then, we understood differential evolution using a working example; so we took the Rosenbrock function which has local; many local minimum and it has one global minimum solution. And we have showed the hand calculation for one generation and the same set of calculations which we did it; so, the same calculations we have showed in terms of graphical example.

So, from starting from the initial population, to the mutant vector to the trial vector and finally, the new target vector for the next generation; we understood how these vectors are improving using mutant and the trial vector operations. And finally, we can expect that after few generation, DE will converge to the optimum solution. So, with this, these understanding and the concepts of differential evolution, I conclude this session.

Thank you.