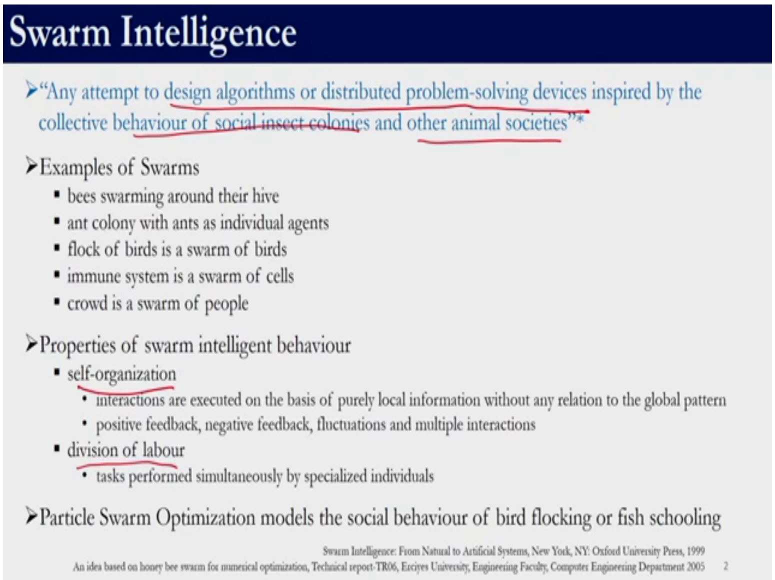
Computer Aided Applied Single Objective Optimization
Dr. Prakash kotecha
Department of Chemical Engineering
Indian Institute of Technology, Guwahati

Lecture – 17
Artificial Bee Colony Algorithm

Welcome to the session. In this session we will be looking at Artificial Bee Colony Algorithm. Artificial bee colony algorithm is a metaheuristic algorithm, which was proposed by Karaboga in 2005. It is a swarm intelligence algorithm, swarm intelligence has been formally defined as any attempt to design algorithms or distributed problem solving devices which are inspired by the behavior of social insect colonies or other animal societies.

(Refer Slide Time: 00:45)

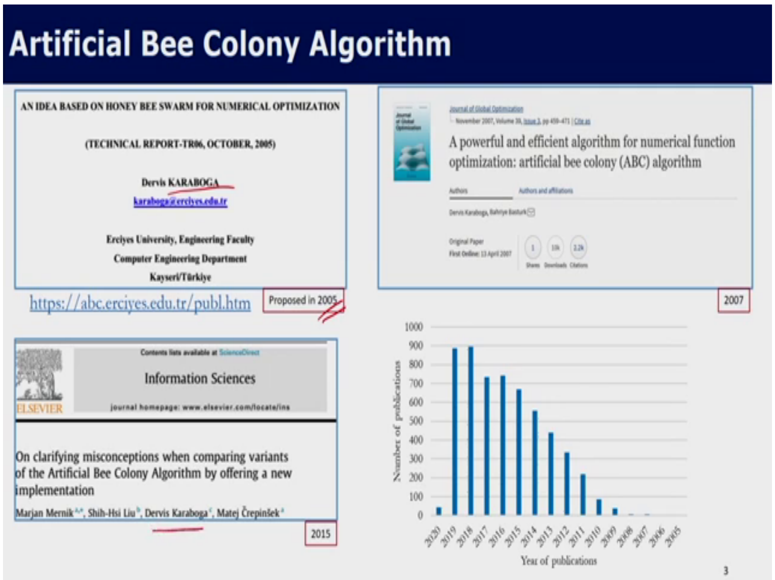


So, one other swarm intelligence algorithm which we have so far seen is particles swarm optimization right. So, this is also a swami intelligence algorithm, examples of swarms are bees swarming around their hive, ant colony with ants as individual agents, flock of birds it is a swarm of birds, you mean system is a swarm of cells and crowd is swarm of people. So, these are examples of swarms. So, any algorithm which is designed with this as its inspiration is a swarm intelligence algorithm. Again these are only examples there can be many other examples.

So, the two properties of swarm intelligent behavior are self organization and division of labor by self organization, we mean that the interactions are executed on the basis of purely local information without any relation to the global pattern right. So, the four components of self organization are positive feedback, negative feedback, fluctuations and multiple interactions. By division of labor we mean that the task that need to be performed are simultaneously performed by specialized individuals.

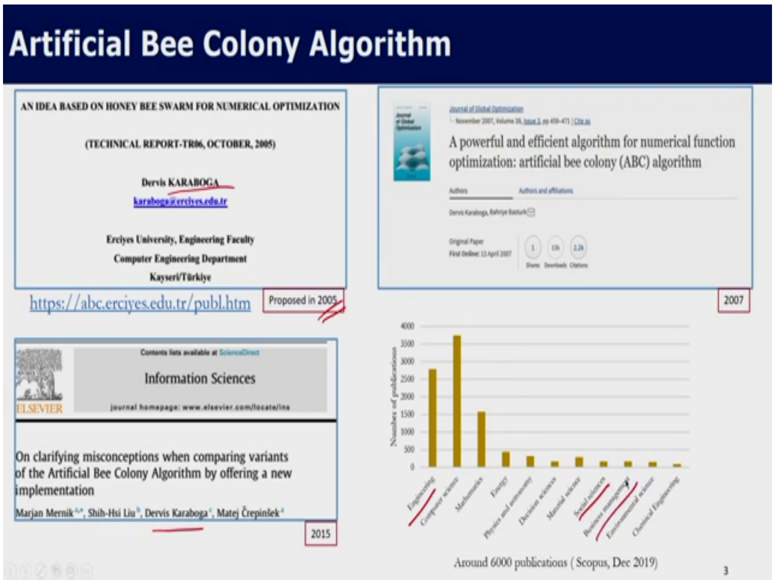
So, particles form optimization which we have previously studied was modeled based on the social behavior of bird flocking or fish cooling.

(Refer Slide Time: 02:11)



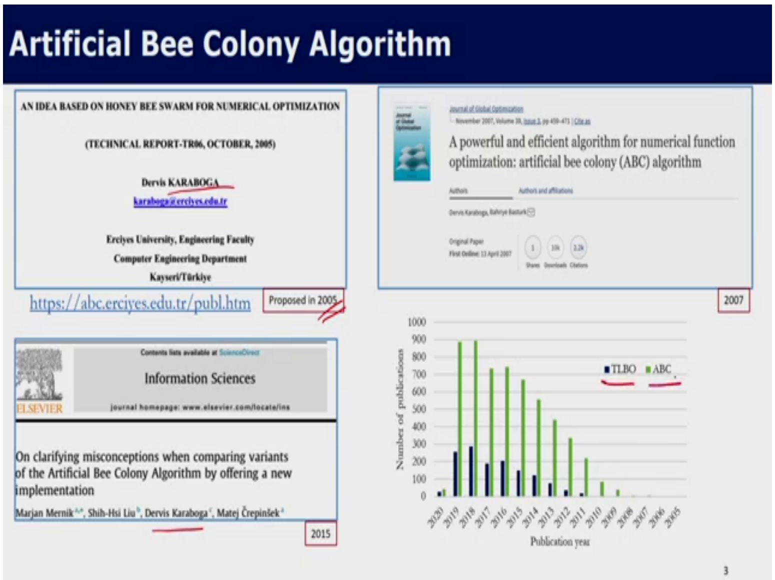
So, artificial bee colony algorithm was first proposed in 2005 by Dervis Karaboga right. Subsequently, it was published in 2007 in Journal of Global Optimization. So, ever since its publication, it has been receiving increased interest from researchers both in terms of its applications as well as variants of ABC algorithm that have been proposed. In 2015, Dervis Karaboga co authored a publication in which they clarified some of the misconceptions with regard to artificial bee colony algorithm right.

(Refer Slide Time: 02:44)



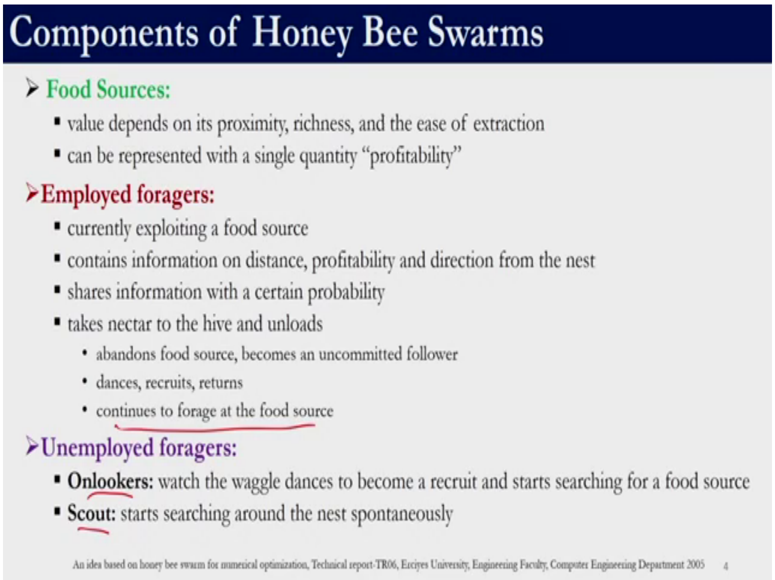
Like all other algorithms; ABC has found applications in engineering social sciences and also in business management. So, this shows the comparison of TLBO and ABC.

(Refer Slide Time: 02:52)



Before we move on to the algorithm, we need to first understand what are the components of honey bees swarms.

(Refer Slide Time: 03:00)



So, there are three major components; one is the food source, the employed bees and the unemployed bees. So, food sources value depends on its proximity, richness and ease of extraction and they can be represented with a single quantity known as profitability. So, this is similar to our solutions. So, each solution has an objective function value associated with it right. So, the objective function value can be considered as profitability.

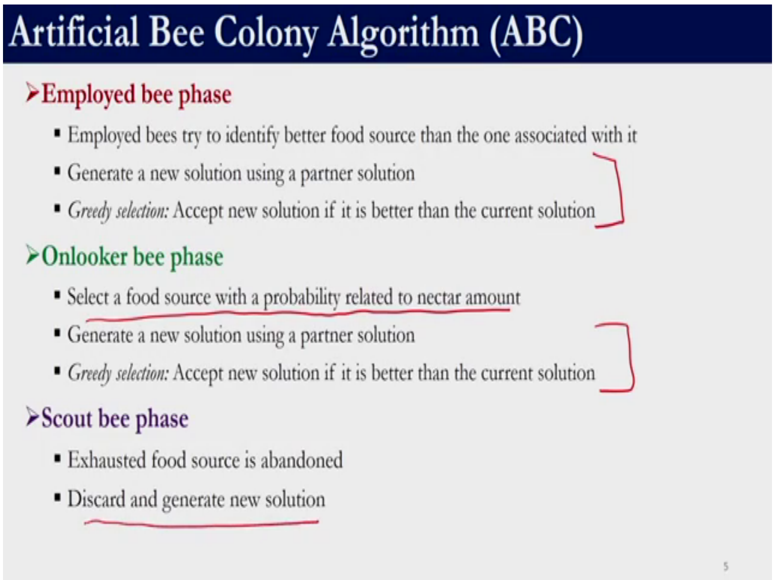
So, what we consider as food sources will be the solutions in optimization. Employed bees are currently associated with food source, they are exploiting a food source, they contain information on distance profitability and direction from the nest. These employed bees share the information with a certain probability to other bee's right. So, when a bee collects nectar and comes back to the hive to unload it has three options; one it can either abandon the food source and thus becomes an uncommitted follower.

The second is to dance in the dance area right these bees perform something called as waggle dance. So, the other bees which are unemployed as of now look at this dance and they may choose to follow this particular employed bee to go to that particular food source to collect more nectar right. So, the second option for the bees is to dance recruit other bees and return back to the food source and keep collecting the nectar from the food source.

The third option is not to dance, but to continue to forager at the food source. In this case it does not recruit any other bees, but it continues to forager at the food source and collect the nectar and come back to the hive right. So, that is the employed bee. So, there are two type of unemployed foragers; one is onlookers and the other one is scout.

So, these onlookers watch the waggle dance, they may choose to follow that particular bee and they start searching for a food source; whereas, scouts do not necessarily watch the waggle dance, but starts exploring around the nest spontaneously right. So, we have basically three components; one is food source, employed foragers and unemployed foragers.

(Refer Slide Time: 05:04)

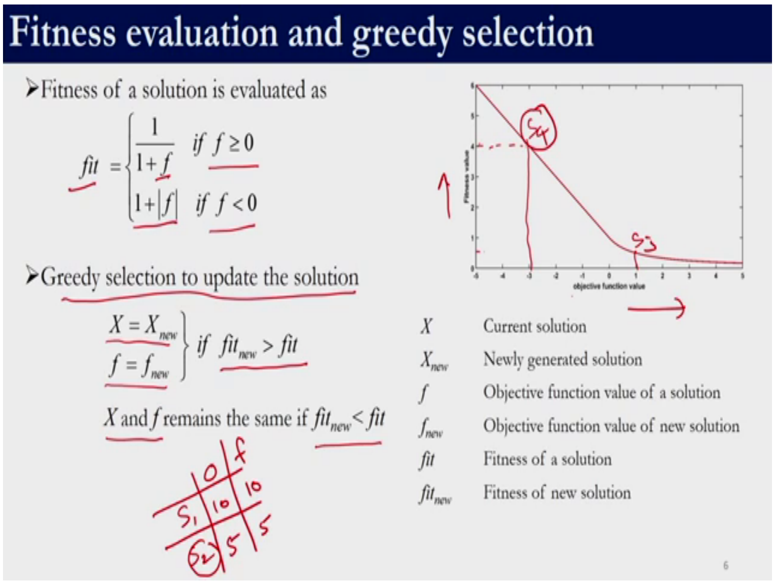


Artificial bee colony algorithm consists of three phases; one is the employed bee phase, onlooker bee phase and the scout bee phase. In the employed bee phase the employed bees try to identify a better food source than the one they are currently associated with right. In terms of optimization, we will be generating a new solution using a partner or solution in the employ bee phase and then we will be performing a greedy selection. So, as you know greedy selection; in greedy selection, we will be accepting a new solution if it is better than the current solution.

In onlooker bee phase; we will be generating a new solution and performing a greedy search, so this is similar to employed bee phase. Only thing is that in employed bee phase every bee associated with a particular food source was generating a new solution, here we will select a food source based on a probability related to the nectar amount right.

So, it is not like every food source will be used to generate a new solution. In the scout bee phase the exhausted food source is abandoned. So, in terms of optimization; we will be discarding a particular solution and we will be generating a new solution. A major difference between many of the other algorithms which we have covered in this course and ABC is the use of the term fitness. In most of the algorithms which we have discussed so far, the term fitness directly corresponded to the objective function value.

(Refer Slide Time: 06:24)



So, if there are two solutions say S1 and S2 right and if the objective function value of S 1 was 10 and S 2 was 5 right. So, then the fitness of solution 1 was 10 and the fitness of solution 2 was 5. So, the objective function directly correspondent to the fitness function value right. And if we are solving a minimization problem, then solution S 2 is better than solution S1 right.

So, objective function directly corresponded to our fitness function value. That is not the case in artificial bee colony optimization right. In artificial bee colony algorithm; the fitness is related to the objective function using the solution right. So, if the objective function value is greater than or equal to 0, then the fitness is 1 by 1 plus f and if the objective function value is less than 0, then the fitness is given by 1 plus absolute of f right.

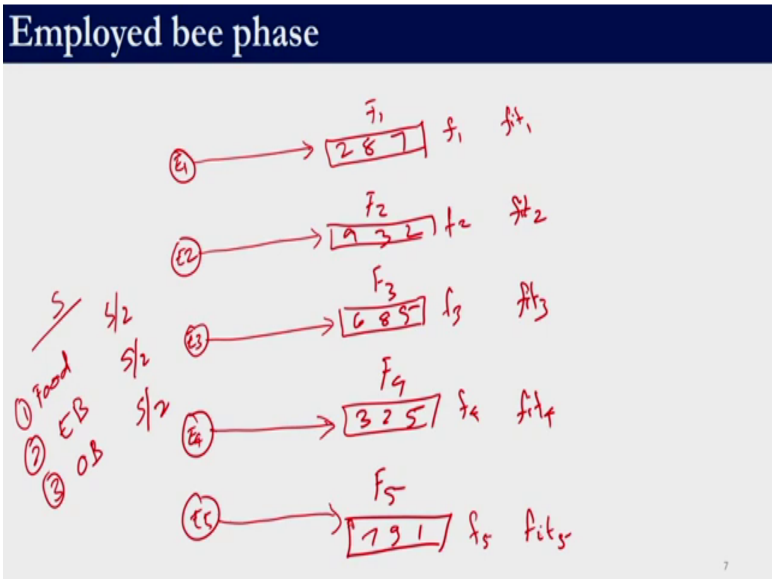
So, this graph shows the variation of fitness function value on the Y axis and the objective function value on the X axis. We can see as the objective function value increases; the fitness value actually decreases right. So, if we consider two solutions let us say this is S 3 and this is S 4 right. So, if you are solving a minimization problem; we would have taken S 4 as a better solution because it has lower objective function value. Now, since we are working with the fitness value which is inversely related, we are supposed to take a solution which has maximum fitness.

So, given two solutions if I need to perform a greedy selection, I will have to select a solution which has higher fitness, not lower fitness, a higher fitness. In the other algorithm, it was lower fitness because objective function value directly corresponded to the fitness function value whereas, in ABC, the objective function value and the fitness function value are inversely related right.

So, if I am solving a minimization problem I will have to select a solution which is actually having a higher fitness right. So, that is what is shown over here, greedy selection to update the search. So, if the fitness function of the new solution is greater than the fitness of a particular solution then we will replace.

So, X is equal to X new, f is equal to f new. So, this is the selection criteria that the fitness of the new solution has to be better than the fitness of the solution which is either undergoing the employed bee phase or the onlooker bee phase right. If this condition is not satisfied like if the fitness of the new solution is not better than the fitness of the existing solution that is, if the fitness of new solution is less than the fitness of the old solution then, we retain the same values for the decision variable and the objective function value. So, this is a major difference between many of the algorithms which we have studied and ABC algorithm right.

(Refer Slide Time: 09:09)



So, let us actually see what is employed bee phase right. So, let us say I have 5 foot source F 1, F 2, F 3, F 4 and F 5 right. These are my decision variables or population numbers this these are the decision variables that we have. Let us say, if I have a three variable problem then these are the values let us say 2 8 7, 9 3 2, 6 8 5, 3 2 5, 7 9 1. So, these are our solution vectors which we have been calling population members so far right. So, corresponding to each of this solution we will have a objective function value right.

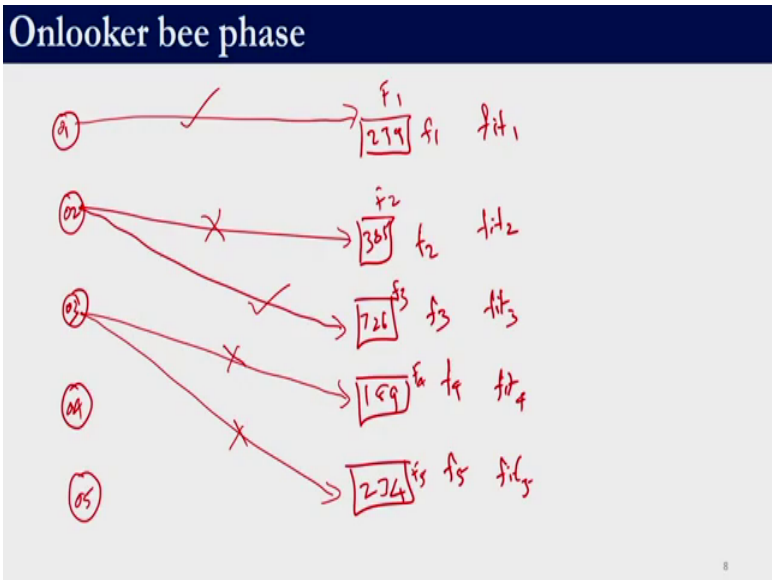
So, objective function value, I am indicating by f 1, f 2, f 3, f 4 and f 5, these are the objective function values. Based on objective function value, I can also calculate the fitness value right based on the formula which was shown in the previous slide right. So, these are food sources 5 food sources; food sources are nothing but, the decision variables right.

So, here we have shown it for 3 variable, for each food source; we have a objective function and the fitness function value. Corresponding to each food source in the employed bee phase there are bees right. So, this is let me say E 1, employed bee 2, employed bee 2, employed bee 3, employed bee 4 and employed bee 5. So, in ABC algorithm, we need to fix something called as swan size right. So, the number of food sources, the number of employed bees and the number of onlooker bees is S by 2 right. So, this is S by 2, S by 2, S by 2.

So, if I fix the swamp size as 10, then the number of food sources is 5. The number of employed bee is 5 and the number of onlooker bee is 5 right. So, it is S by 2. So, in employed bee phase what will happen is this bee is exploiting this particular food source. The first bee is exploiting the first food source, the second bee is exploiting the second food source, the third bee is exploiting the third food source, the fourth bee is exploiting the fourth food source and the fifth bee is exploiting the fifth food source right. What is the meaning of exploiting food source?.

We will come to it a little bit later, but this is what is happening in the employed bee phase that I have 5 food source, I have 5 bees. So, each bee is going to exploit a particular food source right. So, that is our onlooker bee phase.

(Refer Slide Time: 11:54)



So, every bee is going to exploit a particular food source, that is one thing that you need to keep in mind for the employed bee phase. In onlooker bee phase; again I have the same 5 food source; F 1, F 2, F 3, F 4 and F 5. Again these are decision variables I have let us say some decision variable, let us say 2 3 4, 1 8 9, 7 2 6, 3 8 5, 2 7 9 right. So, these are food source. So, my food source in the employed bee phase and the onlooker bee phase may or may not be the same.

So, as we do an example you will realize it that why I am saying that the food sources may be same or not. So, for each food source which corresponds to the solution we also have the objective function value denoted by f 1 to f 5 and we also have the fitness value; fitness 1, fitness 2, fitness 3, fitness 4 and fitness 5. And I also have 5 onlooker bee, so let me call it as O 1, O 2, O 3, O 4 and O 5. So, remember in employed bee phase every bee did exploit the food source. In onlooker bee phase that is not necessary right. So, first we will start with onlooker bee 1.

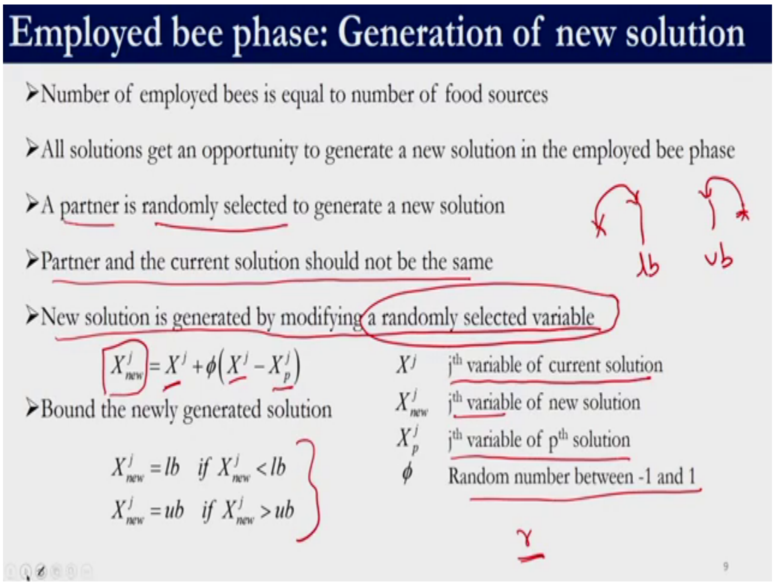
So, onlooker bee based on a particular condition may or may not exploit food source. Let us say it exploits the food source, we will exactly see what is the condition a little bit later right, but as of now let us superficially look into what is onlooker bee phase right. So, onlooker bee will exploit food source 1 based on some condition, if the condition is met it will exploit. If the condition is not met it will not exploit food source 1, but it will exploit food source 2. So, let us to begin with let us assume that this condition is satisfied.

So, onlooker bee 1 is exploiting food source 1, for the second onlooker bee; let us assume that this whatever that condition is not satisfied right. So, since if this condition is not satisfied onlooker bee 2 will exploit the food source 3. Remember this did not happened in employed bee phase. In the employed bee phase; each employed bee was exploiting a particular food source. There was a 1 to 1 correspondence. Here there is not that correspondence right.

So, if this condition happens to be true then this will exploit it otherwise this will not exploit it. Let us say O 3 is not meeting the condition of food source 4 as well as it is not meeting the condition of food source 5 right. So, now if we see we have exploited all the 5 food sources, 3 has not completed the onlooker bee phase because it did not find a food source right. So, in that case we will again start from f 1, f 2, f 3, f 4, f 5. So, we will formally see it in a detailed way subsequently right. So, this is essentially the onlooker bee phase and the employed bee phase.

So, in ABC algorithm we have a unique feature which keeps track of the number of failures. So, whenever we use a solution to generate a new solution. If we are generating a solution which is better right, then the new solution is; obviously taken inside the population because of the greedy selection, but if a solution is not able to generate a better solution, then we keep a track of the number of failures right.

(Refer Slide Time: 15:04)



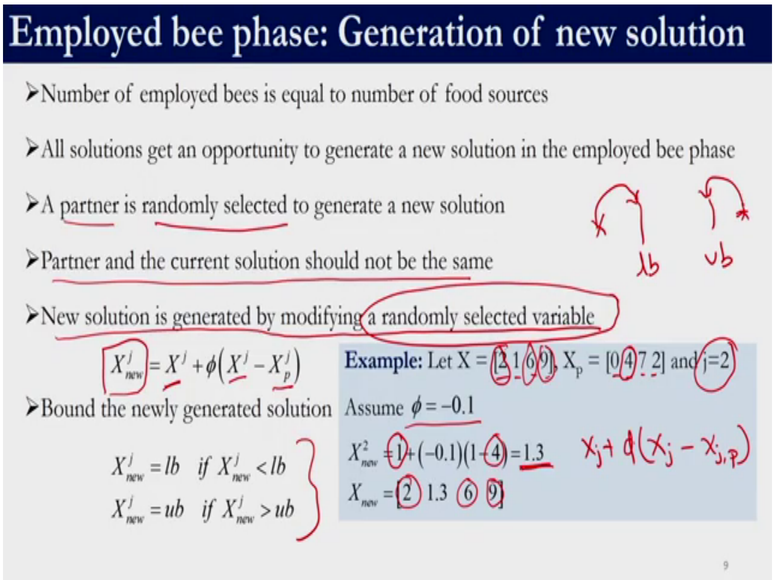
Let us now formally see what is employed bee phase. So, the number of employed bee is equal to the number of food sources. In this, all the solutions get an opportunity to generate a new solution right. So, how do we generate a new solution? It is based on a partner right; obviously, the partner and the current solution should not be the same right and this is the relation that is used to generate.

So, the new solution is generated by modifying a randomly selected variable. So, remember, it is a randomly selected variable. Not all the variables are to be changed, but only one variable has to be changed right. So, this equation if you see X j corresponds to the j th decision variable which was randomly selected right, j th variable this was this j th variable was randomly selected.

X j is the j th variable of the current solution which is undergoing the employed bee phase and X p j is the j th variable of the p th solution, where p is the partner. Again partner has to be randomly selected right. In this relation phi is a random number between minus 1 and 1, remember it is not r which we were using to denote a random number generated between 0 and 1, but here the random number has to be between minus 1 and 1.

So, now this variable may or may not be in bounds. So, if it is in bounds well and good, if it is not in bounds we will again employ the corner bounding strategy. So, the corner bounding strategy is I have a lower bound, I have a upper bound if a solution is anywhere between the lower and upper bound, then we do not need to employ a bounding strategy. But, if it violates the lower bound we bring it back to the lower bound and if a solution violates the upper bound we bring it back to the upper bound right. So, this is the bounding strategy which we have been using in all techniques.

(Refer Slide Time: 16:45)



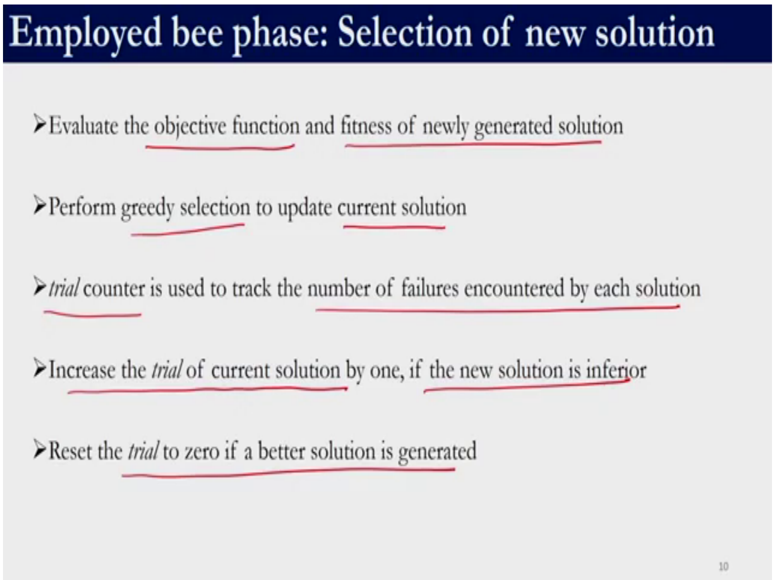
So, let us quickly see an example to generate a solution. Let the solution which is undergoing the employed bee phase bee 2 1 6 9, it is a 4 variable right. So, 2 1 6 9 and the partner which is selected from the population or the food source let it be 0 4 7 2 right. So, the partner has been randomly selected, X is the solution which is undergoing the employed bee phase and let me randomly select the variable 2. So, it means that we are going to change the variable 2 right. And if you remember that equation it required a phi value which was supposed to be between minus and 1 and 1.

So, let us assume that the phi value is minus 0.1. There would not be any change in this, so that is why we have retained 2 6 and 9. Remember the value of only one decision variable is change for generating a new solution right. So, here the equation was X j right. So, X j is 1 plus phi; phi was minus 1 X j minus X j p right, so p denoted the partner. So, the value of partner is 4 over here right. So, the value 1 and the value 4 are used to generate a new solution 1.3.

If this 1.3 is within bounds, we directly take it into the solution, but if this 1.3 is not within the bounds we bound it using the corner bounding strategy. So, this is how a solution is generated. The procedure to generate a new solution either in the employed bee phase or in the onlooker bee phase is the same. The way we generate a solution in the scout phase is different, we will to that a little bit later, but for employed bee phase and the onlooker bee phase the procedure to generate a particular new solution is the same right.

We select a random partner, we select a random variable and then modify just one variable to generate the new solution. So, once we have generated this new solution within the bounds; we need to evaluate the objective function value. So, that is done using the objective function right.

(Refer Slide Time: 18:38)



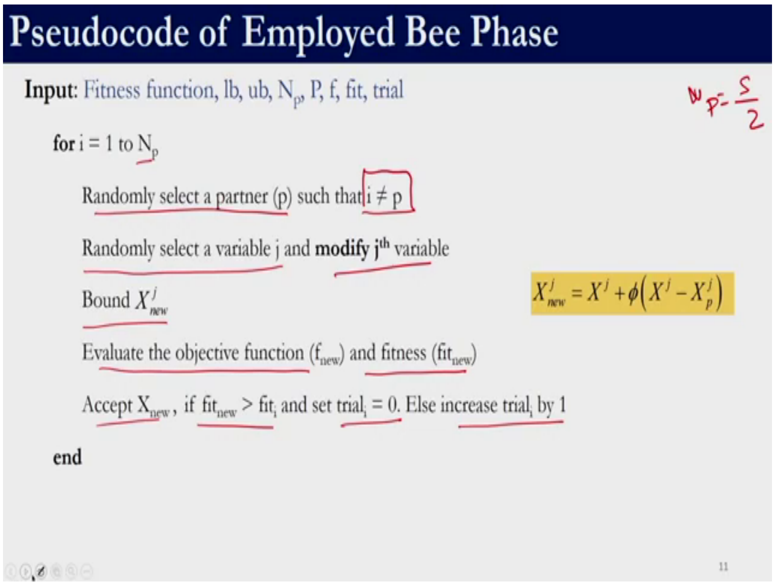
Once we have determined the objective function value; we can use the relation to find out the fitness of the newly generated solution. So, we will perform a greedy selection over here between the current solution and the newly generated solution, whichever solution is better survives the greedy selection procedure right. As we had mentioned a little bit earlier, we will be using a counter known as trial. The name of the counter is trial, this will be used to track the number of failures encountered by each solution. So, each solution has a trial associated with right.

If you have 5 solutions you will have 5 values of trial. So, every time a solution encounters a failure its trial value is increased by 1. At any phase by looking at the trial value, we will be able to say that for how many times that particular solution did generate a new solution, but that new solution was not better. So, it gives a measure of how many times the solution has failed to generate a better solution.

So, we will be having this trial vector. If the new solution is inferior we will increase the trial counter by 1 or if the new solution is better right. So, the new solution will come into the population and since this is a new solution which has come into the population the trial is set to 0. This trial vector will be handled in the same way in the onlooker bee phase also.

So, we will generate a solution, if the solution is good we will take it inside the population and reset the trial, if the newly generated solution is bad then we will update the counter by increasing its value by 1. So, it will keep track of the number of failures. The trial vector keeps a count of the total number of failures, irrespective of whether the failure happened in the employed bee phase or the onlooker bee phase.

(Refer Slide Time: 20:26)



So, now let us look at the Pseudocode of the employed bee phase right. So, in employed bee phase we know that every solution has to undergo this employed bee phase right. So, this for i equal to 1 to N p right. N p is number of food source or the number of employed bees or the number of onlookers bees which is S by 2 right; S is the swarm size.

So, we have this external loop right. So, we are going to repeat whatever is inside this for loop for N p times right. So, what we are going to do is, we are randomly going to select a partner p such that i is not equal to p right. So, because, if i is equal to p this equation this value will become 0. So, the effort that we are putting in to generate a new solution would be meaningless. So, we need to ensure that the partner is not the same solution and then we need to randomly select a variable j.

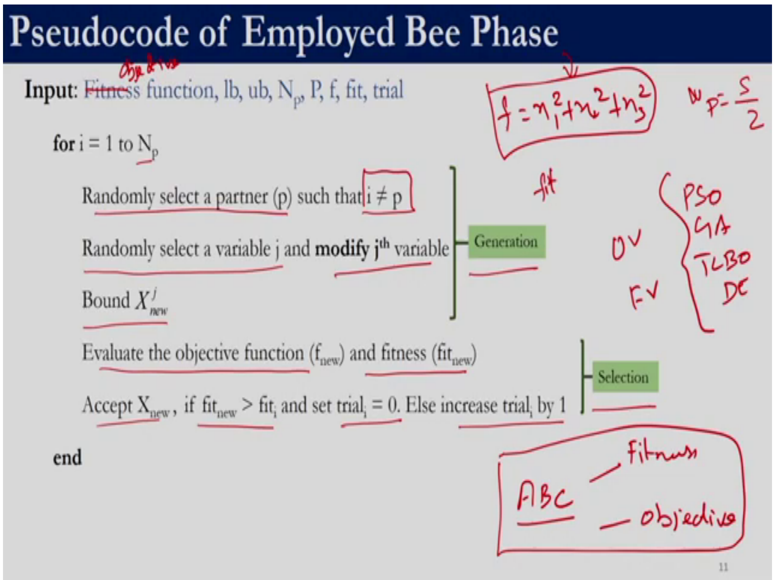
So, it does not matter if your problem is having 4 decision variable or 100 decision variable or 10000 decision variable, we will be modifying only one variable to generate a new solution right. This equation which we are using is more of exploitation rather than exploration right. This is because we are modifying only one variable and very likely the new solution is in the vicinity of the current solution right.

So, once we modify this, we need to apply this equation and generate a new value for the j th variable and then we need to check for the bounds. So, if it is within bounds; well and good, if it is not in the bounds will have to bound the solution. Once we bound the solution we are ready to evaluate the objective function and also fitness because, ABC algorithm works on the basis of fitness and not directly on the objective function value.

So, now we have two solutions. So, the solution which we generated and the solution that is undergoing the employed bee phase right. So, we will do a greedy search. So, if the fitness of the new solution is greater than the fitness of the i th solution, then we will accept the new solution. Since we are accepting the new solution, the trial counter has to be set to 0 right. If this condition is not satisfied right, if the fitness of the new solution is less than the fitness of i th solution, then we will increase the trial counter and discard the new solution.

So, here what we are saying is the old solution is retained, but we also have a track of how many times that solution has failed right. So, this is the pseudocode for the employed bee phase as you can see it is fairly simple to implement right.

(Refer Slide Time: 22:50)



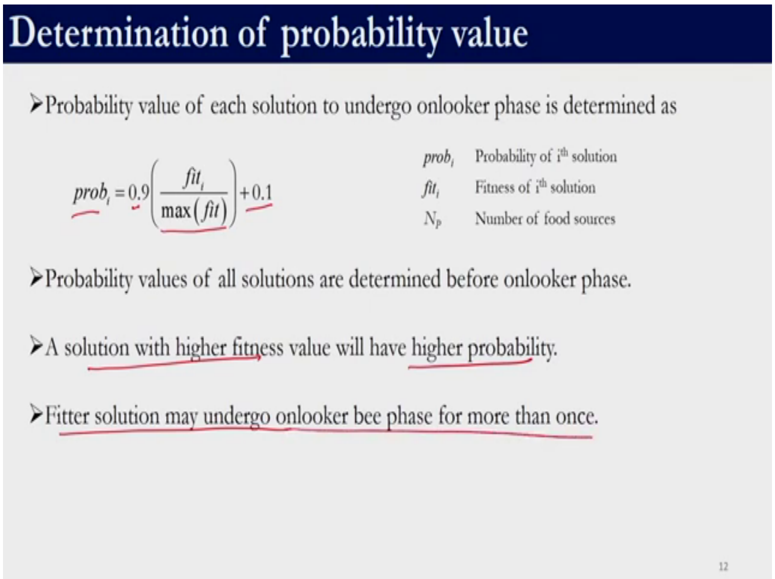
So, this is the generation phase wherein we are generating a new solution and this is the selection phase wherein we are selecting a better solution. So, just to avoid any confusion, this is not actually fitness function this is objective function. Remember particle swarm optimization, genetic algorithm, teaching-learning based optimization, differential evolution. In all four of this objective function value is the same as fitness function value right.

So, that is why we worried using it interchangeably right. We sometimes called it objective function and we sometimes call this as fitness function; particularly since we were dealing with unconstrained optimization problem. ABC explicitly uses a fitness function which is not the same as objective function right. So, both of them are different right.

So, what we are supposed to provide here is what is the objective function. So, once we know the objective function right. So, when we say objective function it is like this function f is equal to let us say x 1 square plus x 2 square plus x 3 square right. So, this is what we mean by the objective function right. So, that has to be passed. So, as in when a solution is received the objective function value is calculated and ABC specifies explicit mechanism to calculate the fitness right.

So, in ABC alone, we have this thing that objective function is not the same as fitness function right. So, just to avoid any kind of confusion; we want to just make sure that you understand that it is objective function over here and not fitness function.

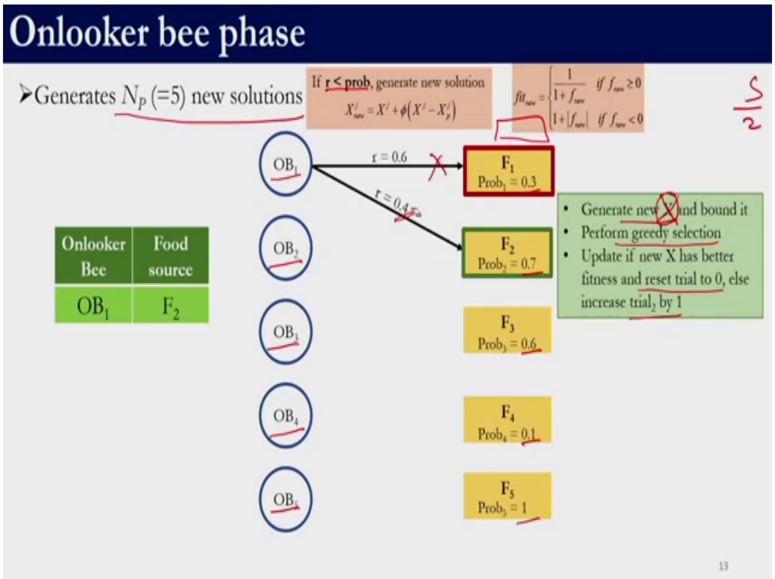
(Refer Slide Time: 24:19)



So, remember in the onlooker bee phase, we said there is a condition which has to be met for a bee to exploit a particular food source right. We know the fitness of the each food source. So, for each food source, we calculate something called as a probability.

So, the probability is given by this formula. Probability of the i th solution is 0.9 times the fitness of the i th solution divided by the maximum fitness in the food source plus 0.1 right. So, the probability of all the solutions are to be determined before we undertake the onlooker bee phase right. So, here we can interpret that a solution which has a higher fitness value will have higher probability right.

(Refer Slide Time: 25:06)



And as you will realize that a fitter solution has a greater probability of undergoing onlooker bee phase for more than once. The onlooker bee phase is supposed to generate N p new solutions where N p is again S by 2, where S is the swarm size user defined parameter, if the swarm size is 10, we need to generate 5 new solutions. These solutions may or may not be better than the solution used to generate the new solutions, but we need to definitely generate 5 solutions, 5 atoms need to be made. The atoms may be successful or may fail, but 5 attempts have to be made.

So, this is the figure similar to what we have seen. So, in this case we have 5 onlooker bees right onlooker bee 1, onlooker bee 2, onlooker bee 3, 4 and 5 and similarly we have 5 food source F 1 to F 5. Since, we know the food source which is nothing but the solution we can calculate its objective function, since we know the objective function we can calculate the fitness.

Now, that we know fitness, we can calculate this probability. Let the probability calculated for food source be 0.3, for food source 2 0.7, 0.6 for food source 3, 0.1 for food source 4 and 1for food source 5. Since the denominator here is maximum fitness and since we are calculating this probability for every solution. At least one of these food source will have a probability value of 1 right.

So, now we have 5 onlooker bees, we have 5 food source, we have the probability associated with it. So, the condition that is to be used to check if a particular onlooker bee will exploit a particular food source is given over here. That for every onlooker bee will generate a random number. If this random number happens to be less than the probability of the food source, then that particular onlooker bee will exploit that particular food source. If that condition is not satisfied; we will move on to the next food source for the same onlooker bee right.

So, let us see a small example, let us say for onlooker bee 1, the random number which was generated is 0.6. So, in this case 0.6 is not less than 0.3 right. So, this condition is not satisfied. So, onlooker bee 1 is now exploring the second food source. Remember from when we moved from one food source to the other food source, this 0.4 changed the random number is to be changed.

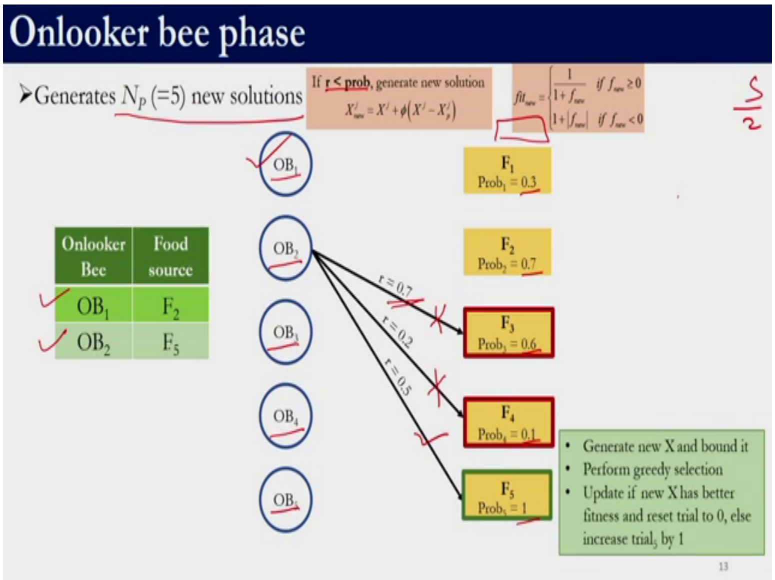
So, here if we see 0.4 is less than 0.7 right. So, it will use this particular food source to generate a new solution right. So, this condition is satisfied, so we need to generate a new solution. Generating new solution is similar to what we discussed in employed bee phase is that for F 2, we will have to randomly select a partner from 1 to 5, we will have to randomly select a variable and then apply that equation which we have seen right. So, that we get a new solution.

So, once we get a new solution, we need to check the bounds for it if it is within the bounds well and good otherwise we need to bound it. Once we have the bounded solution, we need to perform a greedy selection right; greedy selection between the newly generated solution X and the solution F 2.

So, if X happens to be better than F 2, then X will be taken inside the population right and the trial will be set to 0 because, trial is a counter which keeps track of the number of failures right. And since this solution is entering the population for the first time and has not been used to generate any new solution the trial value has to be set to 0, else if this new solution was not better than F 2 right in that case, we discard this new solution and increase the trial by 1 right. Remember there are 5 trial values.

So, we need to the increase the trial value corresponding to food source 2, because that is the food source which failed to generate a new solution right. So, this is for onlooker bee 1. So, now that we have completed for onlooker bee 1, we need to go for the next bee. So, here we are keeping track of which onlooker bee exploited which food source. So, in this case onlooker bee 1 has exploited food source 2 right.

(Refer Slide Time: 29:09)

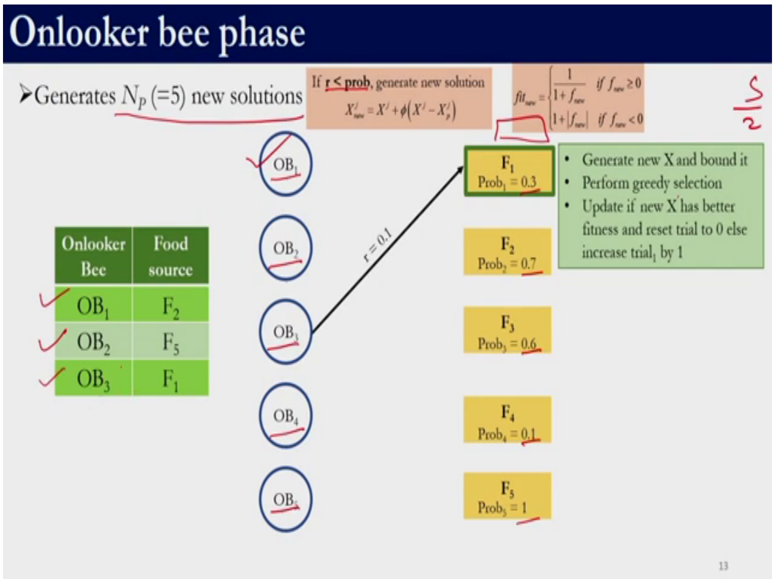


So, now, onlooker bee 1 is over. So, now for onlooker bee 2, we again generate a random number. So, let the random number be 0.7. So, this onlooker bee 2 will not exploit the food source F 3 right, then again we generate a random number, let us say we generate 0.2 and the probability of this solution is 0.1 right. So, again that is not satisfied right.

So, we go on to generating another random number, let us say r is equal to 0.5 in this case we get the relation is satisfied right. So, onlooker bee 2 exploits food source 5. So, what do we mean by exploits food source 5? We will randomly select one partner, we will randomly decide on a particular variable, we will generate a new solution right. If the new solution is within the bounds well and good, otherwise will bound it and then we will perform a greedy selection right.

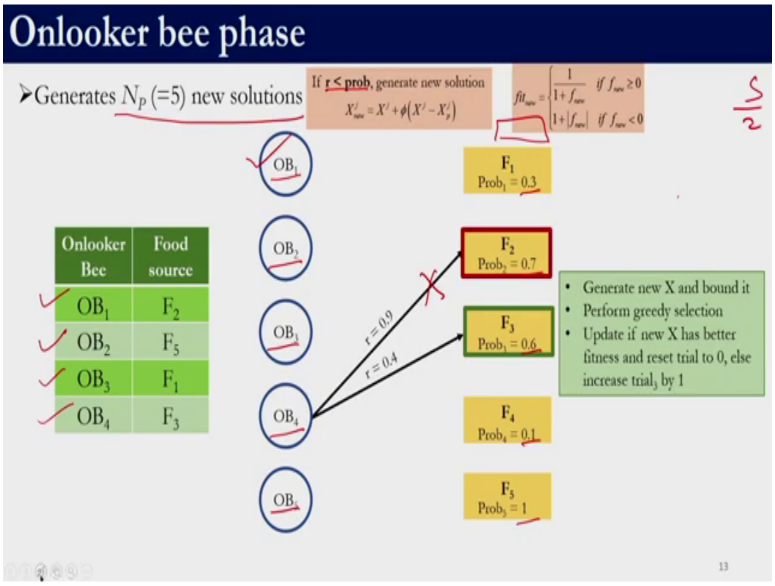
Once we do the greedy selection, we will be able to know whether the new solution is better or not. If it is better, we will take it inside the population and reset the trial to 0, else will increase the trial by 1. So, now, we have completed onlooker bee phase for the second onlooker right. So, onlooker bee 2 exploited F 5.

(Refer Slide Time: 30:20)



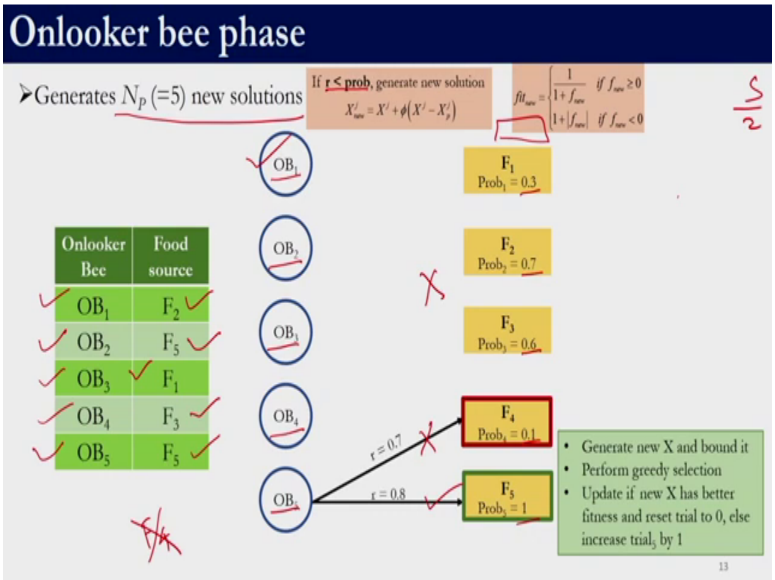
So, now let us move on to onlooker bee 3. Again, we need to generate a random number 0.1. So, we will generate a new solution using F 1 and one of the randomly selected population member. So, OB 3 completes the onlooker bee phase.

(Refer Slide Time: 30:34)



Similarly, we try it for OB 4. So, here the random number we generate is 0.9. So, this does not satisfy the condition. So, we move on to the next food source right. So, here second time we generate a random number which happens to be less than 0.6. So, again we need to generate a new solution, we need to bound the solution, we will perform a greedy selection. If the solution is better, we will take it inside the population reach at the trial else will increase the trial counter by 1 because this particular solution failed to generate a new solution right.

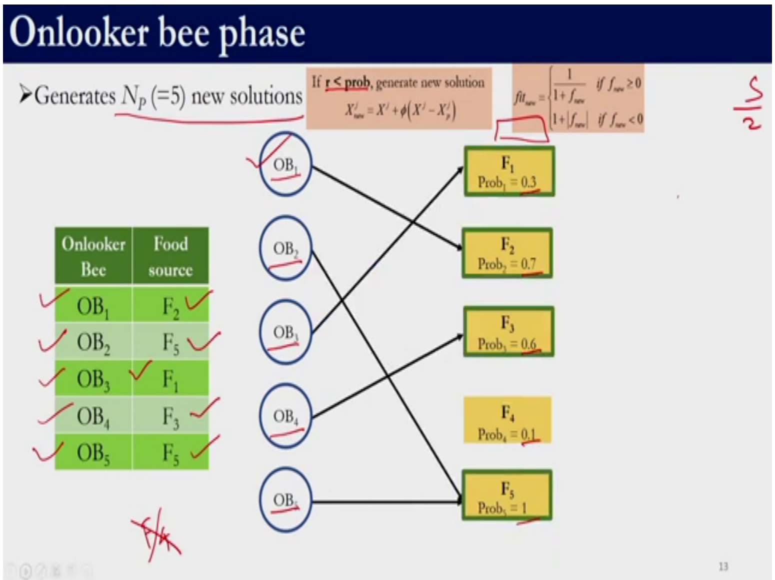
(Refer Slide Time: 31:07)



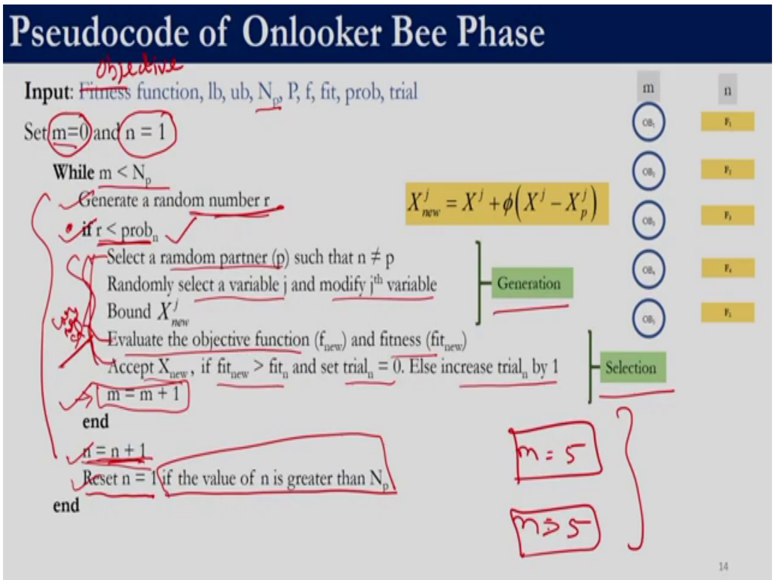
So, OB 4 undergoes F 3. So, then onlooker bee 5, again we generate a random number let us say the random number is 0.7, this condition is not met. So, we generate another random number, so here the relation will be satisfied and again we will generate a new solution. So, this way all the five bees have completed the onlooker bee phase right. So, here if we see F 5 has come twice, F 1 has come once and F 2 and F 3 has come once, F 4 has never been used right.

This is unlike the employed bee phase wherein every solution was used to generate a new solution right. So, that is the major difference between employed bee phase and onlooker bee phase. So, this shows the solutions which were used for every onlooker bee.

(Refer Slide Time: 31:48)



(Refer Slide Time: 31:52)



So, now let us quickly consolidate whatever we discussed in onlooker bee phase using a pseudocode right. So, initially we set m is equal to 0 and n is equal to 1. So, n is curve going to correspond to our food source, m is going to correspond to our onlooker bee right. You will see why we are initializing m to 0 in a little while. So, we are supposed to generate N p new solutions in this example we are taking 5. So, we need to generate 5 new solutions.

So, we use a while condition over here. Initially we have not generated any new solution, so that is why we had initialized m to be 0 right. So, this condition will be satisfied till we do not get five solutions. So, that way we will ensure that we are doing this loop with whatever is there within this while loop for N p number of times. So, first what we need to do is, we need to generate a random number r right. If the random number r is less than probability of n. So, initially n is 1. So, first we are checking the random number which we generated with the probability of F 1.

So, if that condition is satisfied, we will select a random partner we will select a random variable and modify the j th variable using this equation which we have been using. And then since we have generated a new value for the decision variable we need to check whether it is in bounds. If it is in bounds it is well and good else we will use the corner bounding strategy to bound it. After bounding it we need to evaluate the objective function since ABC works with the fitness, we need to calculate the fitness of the new solution.

Once again objective function in ABC is not the same as fitness right it is inversely related. So, we need to calculate the fitness. If the solution which we generated is better that is if the fitness of new solution is better than the fitness of the n th solution right. Then we need to accept the solution into the population and set trial to 0 else will increase the trial counter by 1 right.

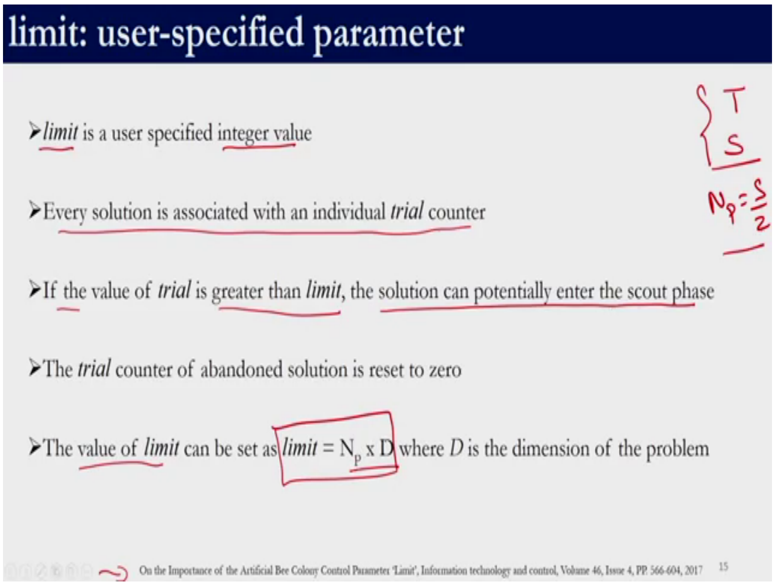
So, this way we have made a first attempt right. So, we increase the counter of m by 1, but if this condition was not satisfied, let us say if this condition is not satisfied. Then if you remember the figure which we saw a little while earlier, we need to move on to the next food source. So, n is equal to n plus 1.

So, the second time we are again generating a random number and checking the probability this condition for the second food source. So, if you carefully analyze it might happen, as it happened in the previous example 4 m becomes 5 because, we require five new solutions right before m becomes 5, this n can go above 5 right. If this n goes above 5 then we do not have any food source. So, in that case what we did was we initialize it right. So, that is what we say, reset n equal to one if the value of N p is greater than N p.

So, you can refer to the example we discussed to further understand why we are resetting 1. Since, this can happen we reset the value of n to 1. So, now, we have completed the employed bee phase as well as the onlooker bee phase. So, if you think about it we started with a swarm size of let us say 10, then the number of food source is 5, the number of employed bees are 5 and the number of onlooker bees are 5 right. So, in employed bee phase we generated 5 new solutions.

In onlooker bee phase we generated 5 new solutions again as discussed previously, this is not fitness function, this is objective function. Because, ABC has different meaning for objective function and fitness function whereas, in all other algorithms for unconstrained optimization problem objective function was the same as a fitness function. Again if you compare this with our metal heuristic techniques we have a generation phase over here and a selection phase over here right.

(Refer Slide Time: 35:36)



So, like other algorithms, we need to specify that number of cycles or the termination criteria for ABC algorithm and we also need to specify the swarm size right. So, if we know the swarm size we can calculate the number of food sources S by 2, the number of food source is equal to the number of onlooker bees and the number of onlooker bee is the same as the number of employed bee phase.

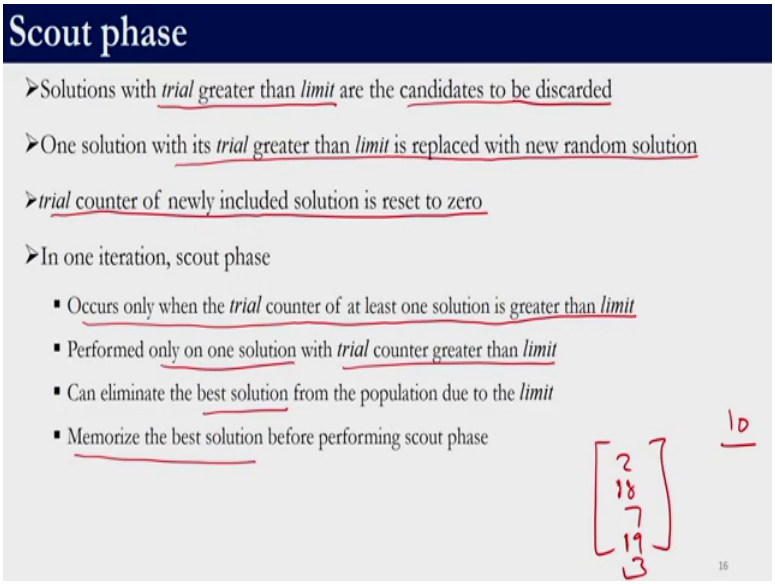
In addition to these two parameters we also need to specify a parameter called as limit right. So, it is a integer value. So, as we have seen that every solution is associated with an individual trial counter. So, for every solution we had a trial counter that is not set by user, but that is just keeping a track of how many times a solution is failing to generate a better solution right.

So, this limit is an integer value. So, if the value of trial for a particular solution is greater than the limit which has been set by the user, then the solution can potentially enter this scout phase. So, not all the solutions will undergo scout phase right. Only those solutions which have failed more than the specified number of times let us assume that I have set a limit value of 5.

So, in that case the solutions which have failed more than 5 times can potentially enter scout phase not all of them will be entering, but only one of them would be entering right. So, we look at it in a little bit later. Usually, the value of limit is given by N p into D, where D is the dimension of the problem and N p is swarm size by 2 or the number of food source right.

So, this is the most commonly used value for limit further research has been done on how to optimally set this limit if you are interested you can look into this, we will not go into detail of that. But, usually the limit value is said to be N p into number of decision variables.

(Refer Slide Time: 37:26)



Now, that we have the limit parameter, we can look into scout phase. So, solutions which have their trial values greater than limit are the candidates to be discarded right. So, only those solutions whose trial is greater than the limit can undergo scout phase right. So, one solution with its trial greater than limit is replaced with new random solution.

So, you may have many solutions, but only one of them is replaced. So, that solution is totally discarded and a newly generated random solution is included in the population right. Since a new solution is generated the trial counter for the new solution will be reset to 0. So, in every iteration or cycle, scout phase may be encountered or may not be encountered right. So, it occurs only when the trial counter of at least one solution is greater than limit.

So, if your limit is let us say 10 and you have 5 solutions whose trial counter is 2, 8, 7, 9, 3 right. So, it indicates that the first solution failed for 2 times to generate a new solution right.

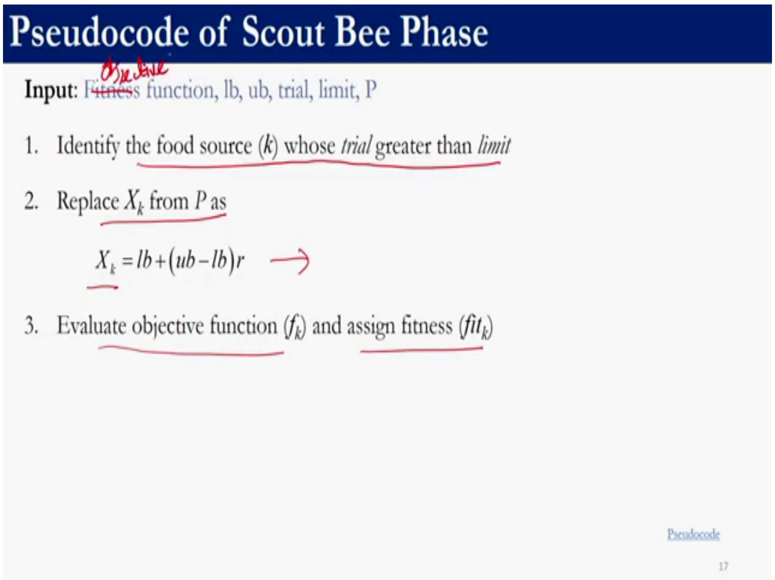
Similarly, the other values can be interpreted right. In this case the scout phase will not be encountered because all of this are less than 10 right. The scout phase may or may not be encountered in an iteration that depends upon the trial vector and the limit value right.

So, this scout phase is performed on only one solution. Let us say instead of this I had trial vector as this right. So, now, when I complete the onlooker bee phase, two solutions are exceeding the limit right. So, even then only one of them would be replaced. So, that is why it is performed only on one solution with trial count greater than limit. What can happen in scout phase that the best solution right can have a very high limit right.

Let us say we encountered a best solution and let us say that best solution is not able to generate better solutions right either in the employed bee phase or in the onlooker bee phase. It is continuously encountering failures right and the number of failures is higher than the limit right. So, it can potentially get eliminated. The best solution can potentially get eliminated from the population right.

So, that is why before we enter the scout phase we will memorize the best solution. So, even if it enters scout phase and it is discarded we will not lose that solution right. Because, we have stored that best solution in a particular variable.

(Refer Slide Time: 39:47)



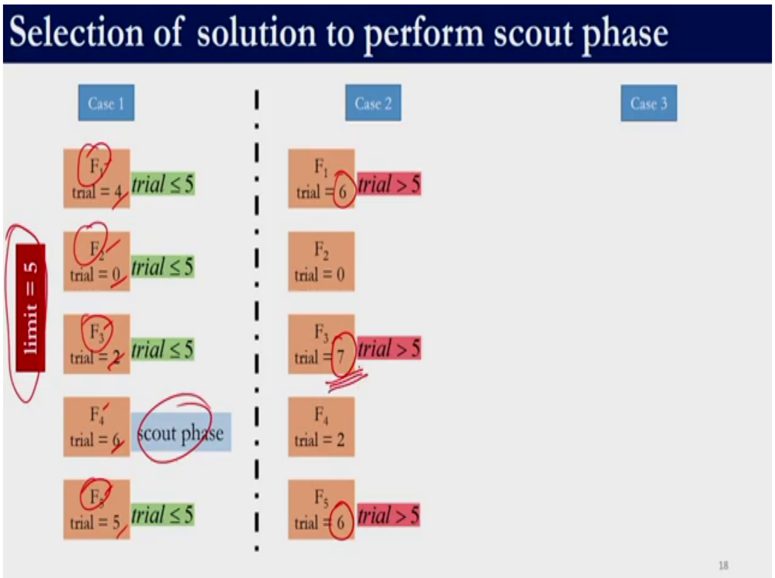
Now, let us look at the pseudocode of scout bee phase right. So, we need to identify the food source whose trial is greater than the limit right. We need to replace that the entire solution. To once we replace the entire solution, we need to evaluates its objective function and assign the appropriate fitness function, remember there is no greedy selection over here right. Because, that solution which has been selected to be discarded from the population has not been able to generate new solutions. So, it may happen that the new solution which we are generating over here can be actually bad than the solution that is being discarded right. So, we are not employing a greedy selection strategy over here.

So, that is how scout bee phase is different from onlooker bee phase and employed bee phase. One is that we change all the decision variable values whereas, in employed bee phase and onlooker bee phase; we were changing only one decision variable value right. And secondly, both in employed bee phase and onlooker bee phase we employed a greedy selection strategy whereas, in scout bee phase we will directly take the newly generated solution into the population we will not employ a greedy selection strategy. Similar to our previous discussion right, this is not fitness function, but this is objective function.

(Refer Slide Time: 41:05)

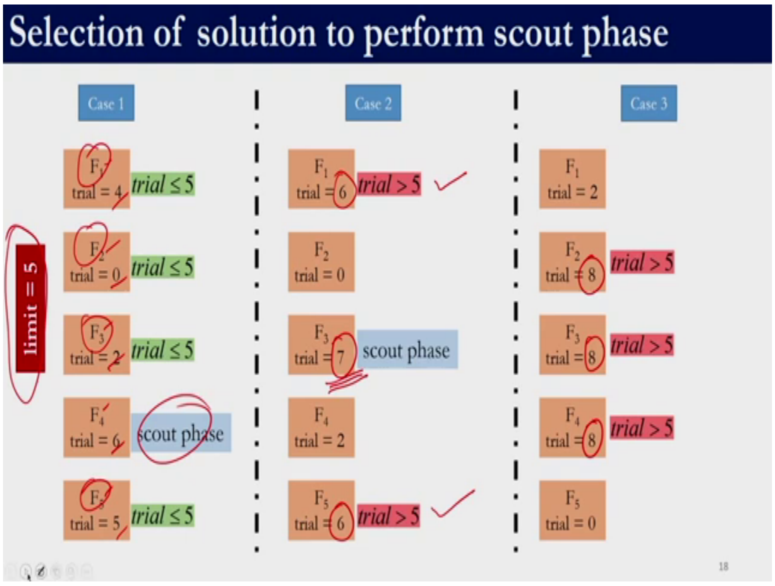
Let us look at three cases that can be encountered in the scout phase right. Let us consider these are our 5 solutions F 1, F 2, F 3, F 4, F 5 and these are their respective failures right. So, solution 4 has encountered 6 failures so far and let our limit value be 5 right. So, in this case these four solutions, solution F 1, F 2, F 3 and F 5 have a trial value which is less than or equal to 5. So, they will not definitely undergoes scout bee phase right. So, the only solution which can undergoes scout bee phase is this one because it has a trial value greater than 5.

(Refer Slide Time: 41:41)



So, in case two, if we see there are three solutions F 1, F 3 and F 5 whose trial is greater than 5. So, one of these three solutions will undergo the scout phase, the one that I will undergo is F 3, because it has the maximum number of failures.

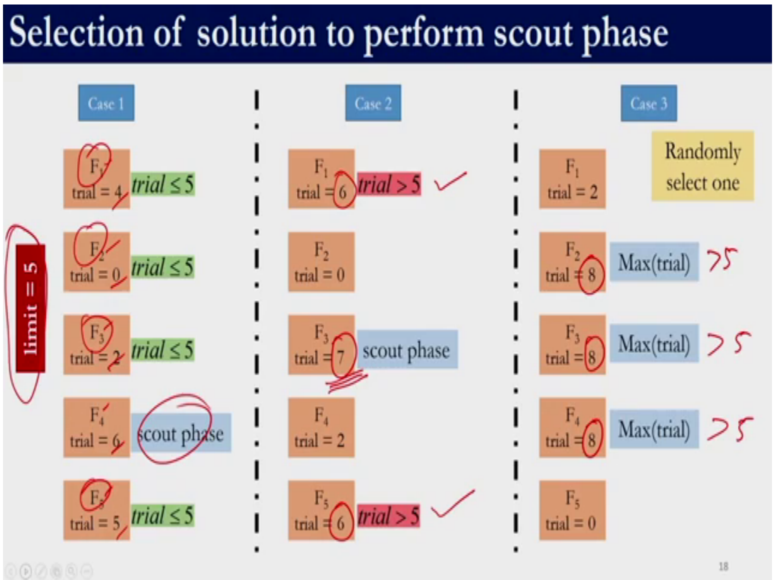
(Refer Slide Time: 42:02)



So, in case 1 there was only one solution. So, there was no problem. In case 2, there were multiple solutions which violated this limit parameter right. So, in that case we will select the one which has failed the maximum number of times. So, this is also violating this condition, this is also violating the condition, but these two food sources will not undergo the scout phase because in the scout phase only one solution has to be discarded right.

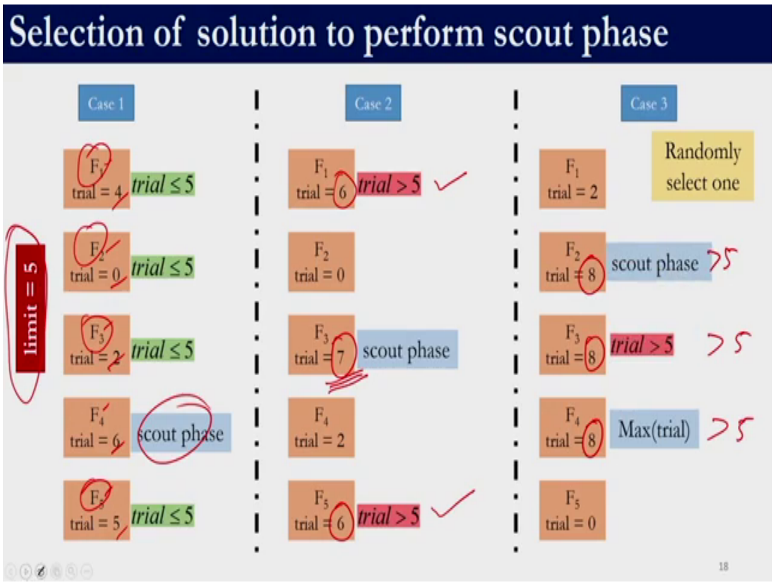
So, in case 3, these three solutions have identical trial values 8 8 8 right. So, all of them are greater than 5 which is our trial limit. So, in this case we need to randomly select one of these solutions right.

(Refer Slide Time: 42:40)

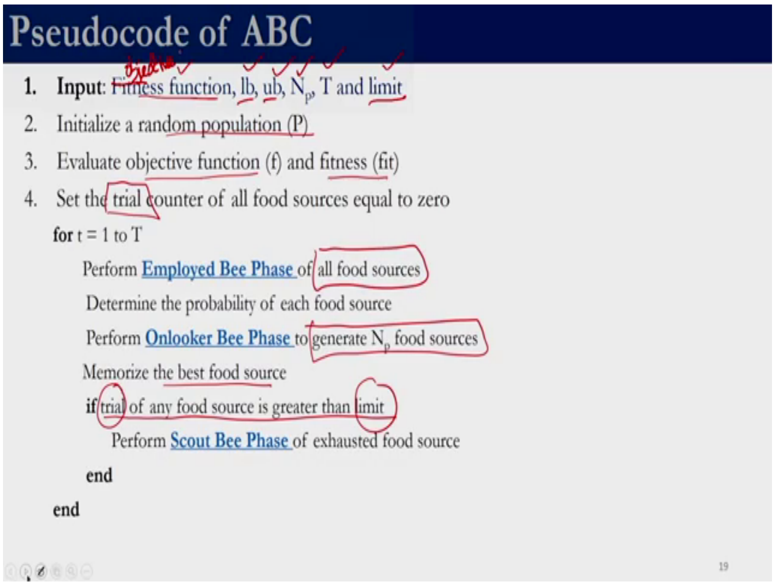


So, all of these have the maximum number of trails. So, we will select randomly one solution because they have the same number of maximum trials which is also greater than 5. So, this is all of these are greater than 5. So, any one of this can be selected. These are the three cases in which we will have to decide which particular solution undergoes the scout phase.

(Refer Slide Time: 43:05)



(Refer Slide Time: 43:06)



So, the entire pseudocode of ABC can be given as this thing that we need to first initialize a random population within the bounds of the decision variable. So, the bounds should be known right and we need to have a fitness function. So, that we can calculate the objective function right fitness function as in like objective function is required and then we will use the relationship between objective function and the fitness function to calculate the fitness of every solution right.

And then we need to set the trial counter of all food sources equal to 0. So, this trial counter will keep track of the number of failures. So, if the number of food sources is 5, trial is a vector of 5 elements. ABC is an iterative technique right. So, we will have to perform the steps of ABC multiple times, it is either known as the cycles or we can call it as iterations.

So, we will have to perform it for t iterations will have to first perform the employed bee phase on all the food sources right and then we will have to determine the probability of each food source and then we perform onlooker bee phase to generate N p food source right. In employed bee phase all the food sources are going to definitely be used to generate a new solution that is not the case in onlooker bee phase. But still we need to generate N p food sources. Once we are done with that we need to memorize the best food source. This is an important step because the best food source can be lost in scout phase right.

So, we need to memorize the best food source and we need to check if trial of any food source is greater than limit. We need to first determine if the scout phase is to be performed using the trial vector and the limit value which has been set by the user right. If the condition is met then we perform the scout bee phase of the exhausted food source over here it has to be objective function right and not the fitness function. So, this completes the pseudocode of ABC. So, with that we conclude this session.

Thank you.