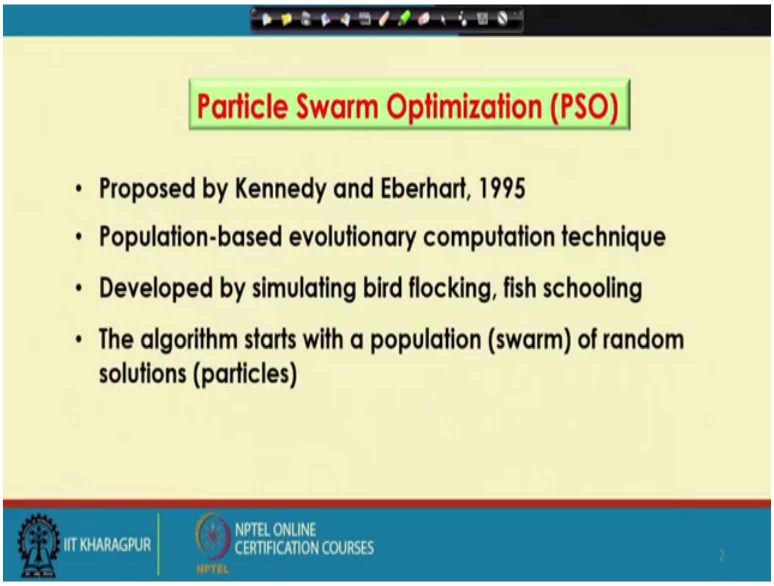
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 21
Particle Swarm Optimization

Now, I am going to start with the discussion of another very popular non-traditional tool for optimization and which is popularly known as vertical swarm optimization in short PSO.

(Refer Slide Time: 00:40)

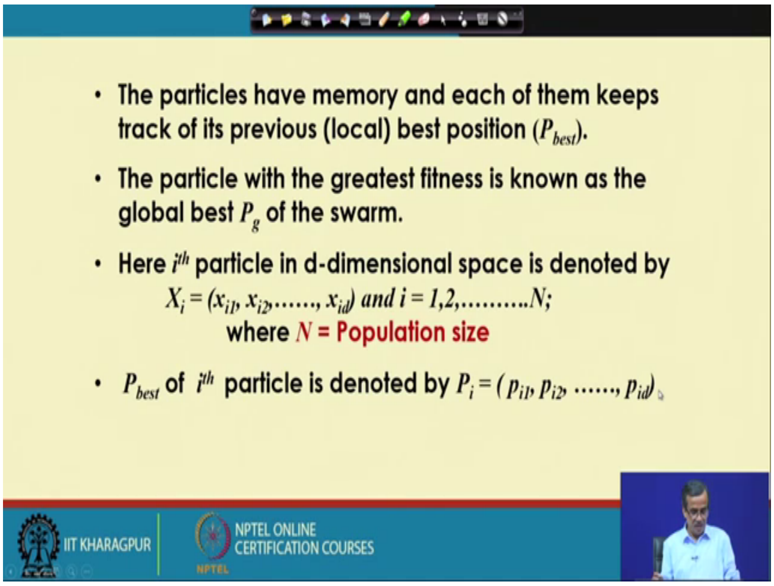


Now, particle swarm optimization; the concept came in the year 1995 and it was introduced by Kennedy and Eberhart.

Now, this is just like the genetic algorithm is a population based approach. Now to develop this particular algorithm, actually, we try to copy some physical phenomena like bond flocking fish cooling and so on. Now, let me take a very simple example. We have seen the way pigeons fly in the sky. They do not collide with each other. They use the principle of optimization.

Now, that particular principle has been copied here to develop this algorithm that is vertical swarm optimization. Now here the name indicates that this is particle swarm that is swarm of particles that is a population of solutions. So, we are using the term swarm in place of population and particle in place of that solution. So, we used to use population of solution and in PSO it is actually the swarm of particles.

(Refer Slide Time: 02:08)



Now, here these particles have memory and they try to update their position, their information, they compare with their previous situations, previous positions and try to find out; what should be its action in future. So, this is actually a method where the particles can keep their memory and this memory is going to help during its search. So, we try to find out actually the P best that is the population based and we also find out the g best.

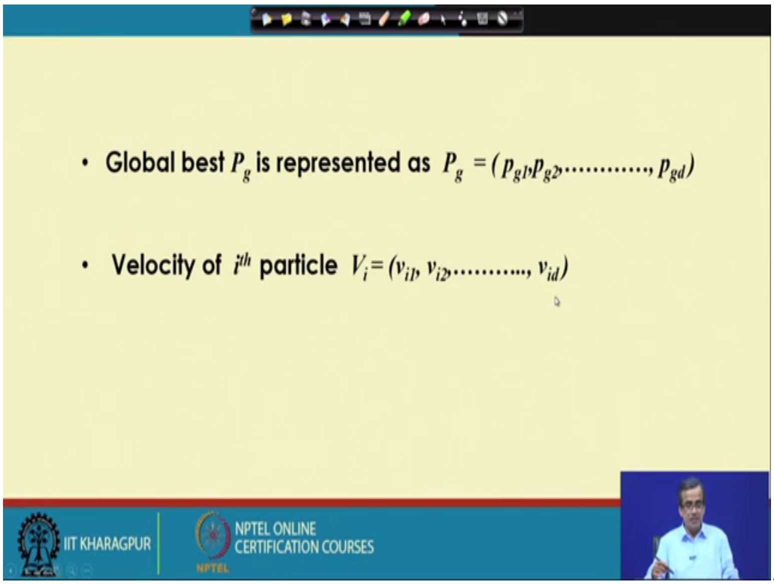
Now, how to find out this previous and g best, we will be discussing in much more details with the help of one numerical example, but before that let me tell you, suppose, I did that in one swarm a few particles are there; now what we do is we consider their positions initially, those are selected at random and we compare their objective function values and if it is solving the maximization problem.

Now, corresponding to that particular set of design variables which gives the maximum value of the objective function that will be considered as the globally best; so, I am just going to take some numerical example to make it much more clear, but before that let me tell that we try to find out the best position that is nothing, but the P best and we also trying to find out the globally best position that is the P g.

Now, here the i th particle in d dimensional space is denoted by capital X i and capital X i is nothing, but the connection of small x i 1 comma x i 2 up to x i d because it is having d dimension and i varies from 1, 2, up to n that is actually the n is nothing, but the swarm size and here, although I have written here population size, but truly speaking, it should be this one size in PSO.

Now, P best of each particle; now each particle that is each solution is having d dimension. So, P i is nothing, but the collection of P i 1 comma pi 2 up to pi d.

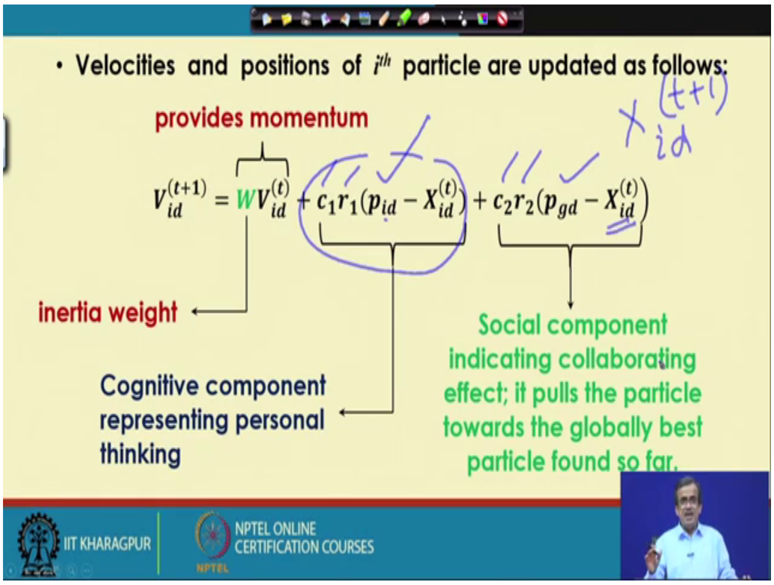
(Refer Slide Time: 04:48)



Now, the globally best solution that is denoted by P g in d dimension that is denoted by P g 1 comma P g 2 up to P gd. Now here actually what we do? We initially assigned some velocity and position to the particle and then we try to update their velocity and position through a large number of iterations.

The velocity of i th particle once again that is denoted by capital V i and that is a collection of small vi 1 small vi 2 up to v id because this is in d dimension.

(Refer Slide Time: 05:37)



Then once they have got this particular the velocity updating, now I can update the position also. Now I want to discuss like how to update this particular velocity in much more details. Now this is actually the updating rule for the velocity. Now this corresponds to the i th particle and its d th dimension. So, V i d represents the velocity of each particle corresponding to the d th dimension at t plus 1 th iteration that is nothing but w multiplied by V i dt. Now this W is nothing, but the inertia weight and we will have to assign some numerical value to this particular the W and V id is nothing, but the velocity of i th particle corresponding to d th dimension at t h iteration. So, W rei d t plus c 1 r 1 multiplied by P i d minus x id corresponding to the t th iteration.

Now, here actually what you do is; so, this c 1 is nothing, but a constant that is called the cognitive constant and this r 1 is nothing, but a random number lying in the range of 0 to 1 and P id is nothing, but your that population best or the present best I should say minus x id t. So, this is nothing, but this constitutes actually the cognitive component representing the personal thinking like how to update myself.

So, this is my own thinking. So, this component is going to represent the cognitive component. Now the last component that is c 2 r 2 multiplied by P g d minus x i dt. So, here c 2 is nothing, but the social constant r 2 is nothing, but the random number lying between 0 and 1 and P gd P g indicates the globally best solution and corresponding to the d th iteration minus x i dt. Now what I am going to do is I am trying to find out what will be my x id t plus 1.

So, my aim is to determine x id corresponding to t plus 1 and supposing that this is known and P gd t and P id t, I have already determined and I know the values of these W c 1 c 2 and r 1 r 2 are generated, then I can find out, this v id t plus 1. Now here the significance of the last part of this particular expression that is c 2 r 2 multiplied by Pg d minus x i dt, it represents actually the social component; that means, if I want to improve the solution.

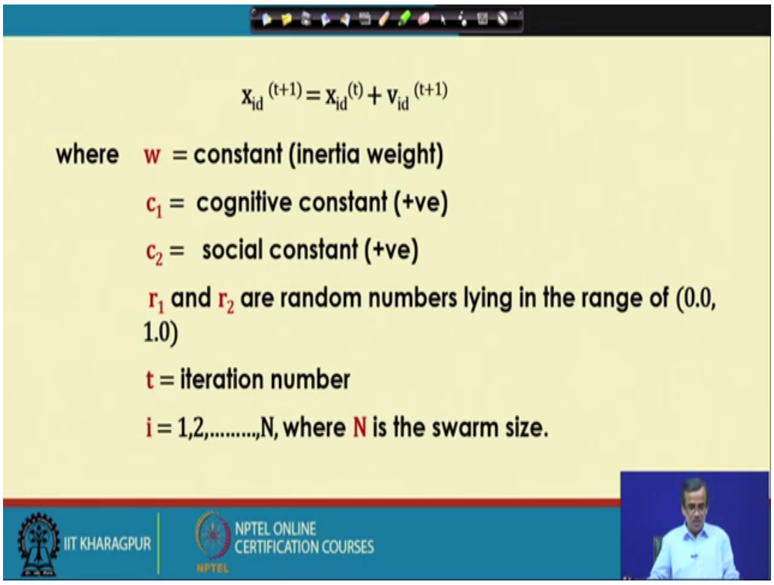
So, I will have to see; what is the global best what is happening outside and the globally best solution is going to attract that particular the solutions towards it. So, there will be some improvement. So, the social component indicates collaborating effect it pulls the particles towards the globally best globally best solution. So, this W multiplied by Vi dt that will provide the initial momentum and the second component is nothing, but the cognitive component and the third component is the social component.

Now, let me take a very simple example very practical example. Now supposing that mister point x has got some training he has got some degree a PhD degree and now he has joined some profession, might be say research, he wants to improve. Now if he wants to improve; what he will have to do is he will have to depend on the initial momentum, there is no doubt in it; at the same time, he will have to improve himself. So, and that indicates actually the second component and another thing, he will have to do? what is happening outside the world and what others are doing.

What is the globally best solution that also will have to keep in touch or keep track of that then only mister x or the dr point x will be able to make significant improvement in his professional life. So, the same principle of this particular the particle swarm optimization is applicable to our life also now as I told that it has got 3 components like initial momentum cognitive component and social component and using these 3 components. So, the velocity for the i th particle corresponding to the real dimension will be updated.

And once we are able to update this.

(Refer Slide Time: 11:35)



Now, I will be able to find out what should be the updated information in terms of the position. So, the velocity updating is done then position updating that is X id t plus 1 is nothing, but X i dt plus V id multiplied with v id t plus 1; so, using this particular expression. So, we can find out what should be this particular updated position for the i th particle corresponding to the d th dimension at k plus 1 th iteration.

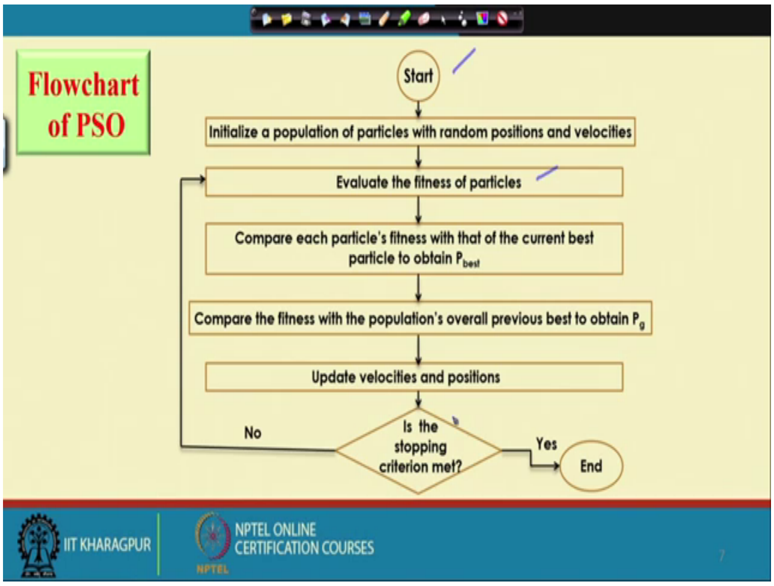
So, this particular updating of the velocity and updating of the position are to be done through a large number of iterations and then this particular algorithm will be able to find out that particular the optimal solution now this optimization algorithm that is PSO principle wise is very simple much simpler compared to the genetic algorithm and here actually, there is no such complete complex operators like cross over mutation and all such things it is very straight forward.

But here, there is one thing it has to be kept in mind that is the memory. So, here the solutions that is the particles are having their own memory and they update their position they update their velocity based on their past experience and based on actually the globally best solution and due to this particular updating this particular algorithm can perform both the global search as well as the locals are very efficiently.

Now, here all such things whatever I discussed. So, I have mentioned here w is a constant inertia weight cognitive constant is c 1 social constant c 2 r 1 r 2 are nothing, but the random number now here I just want to tell that the performance of this particular algorithm is dependent on the number of parameters. For example, say we have got the parameters like the w then comes your this c 1, then c 2 the performance will depend on this particular the values of the parameters.

Now, what you will have to do is you have to find out the values of these parameters w c 1 and c 2 in a very efficient way in an adaptive way so that this particular algorithm can become more efficient. Now, what I am going to do is; so, I am just going to solve one numerical example; just to explain the working principle of this particular the algorithm, but before that let me try to concentrate on the flowchart of this particular algorithm which I have already discussed, but less let me try to summarize with the help of this particular the flowchart.

(Refer Slide Time: 14:22)



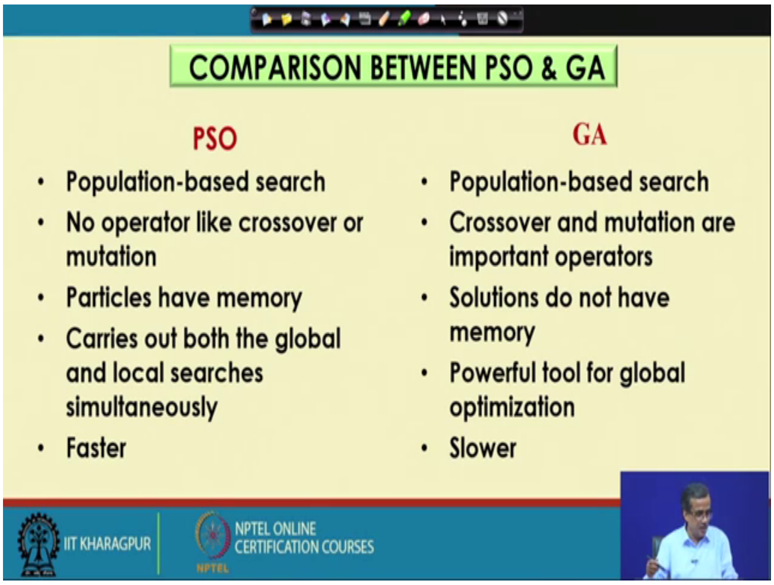
So, we start with the algorithm, we initialize a population of particles at random; that means, the random position and the random velocities will have to assign and we evaluate the fitness value by using the expression of objective function we try to find out the fitness value for each of the particles lying in that particular the swarm.

We compare each particles fitness and with that of the current base to obtain the p best. So, now, we try to find out what should be the P best. Now we compare the P best values just to find out the globally best that is nothing, but the P g, then we update the velocity and the position and this completes actually one iteration of this particular algorithm.

Now here we have got the checking that is the termination criteria if the termination criteria is met then it is the end of the algorithm.

Otherwise we are going to repeat. So, once again we are going to start with the evaluation of this particular the quality of the solution in terms of the fitness and then we try to find out previous G best and we update this particular velocity position and this process will go on and go on. So, this is the way actually this particular algorithm works and as I told that it is very simple and much simpler than the genetic algorithm.

(Refer Slide Time: 16:14)

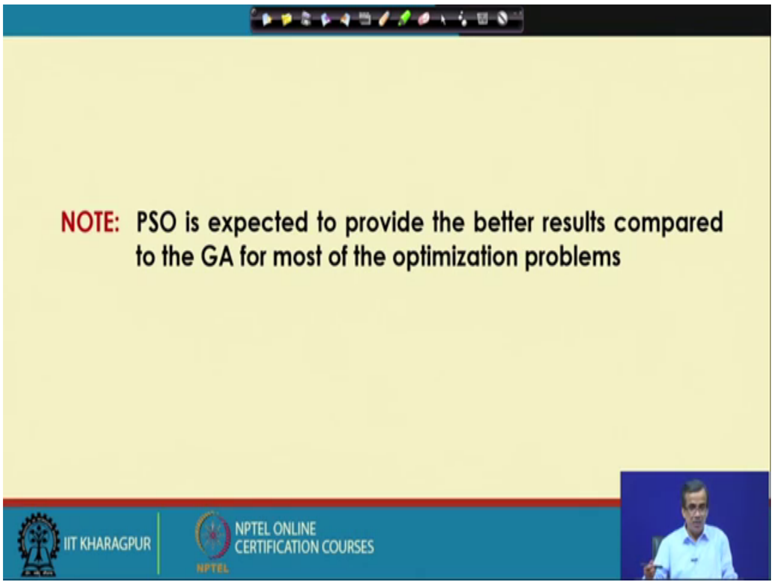


Now, if I compare this particular PSO with the genetic algorithm, we can see that both PSO as well as genetic algorithm are population based approach. Now in PSO, there is no operator like crossover mutation. On the other hand, in GA, we use crossover mutation all such operators in PSO actually the particles have memory and it has got some; I should say, some; this is the plus point of this particular algorithm. On the other end, in genetic algorithm solutions do not have memory and that is why GA is suitable only for the global search.

On the other hand, the PSO is suitable for both local as well as the global search. Now the PSO carries out both the global and the local search, simultaneously on the other hand, g is a powerful tool for global optimization, but its local search capability is poor and PSO is faster because here there is no such operator like reproduction crossover and mutation it is very straight forward on the other hand the g is lower compared to the PSO.

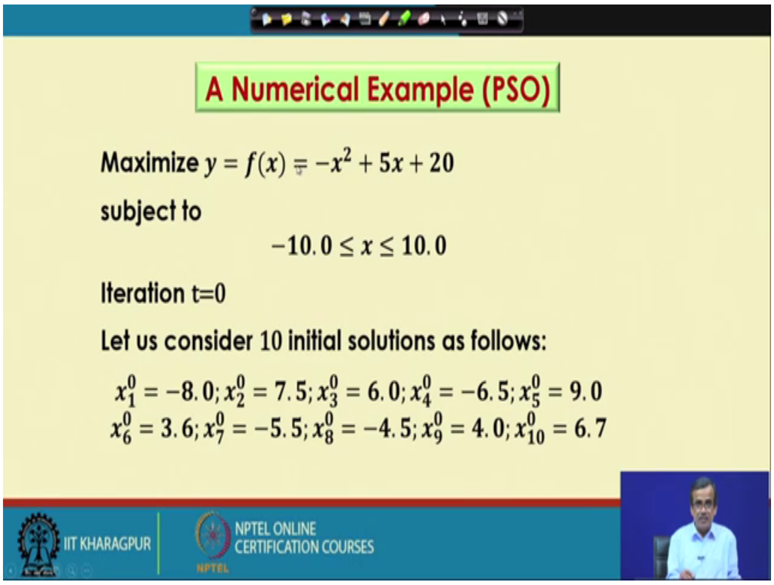
Now, we have solved actually a number of optimization problems using both PSO and genetic algorithm.

(Refer Slide Time: 17:47)



And based on that particular experience, we can say that PSO is expected to provide better results compared to the GA in most of the optimization problem and that is why the PSO is becoming. In fact, more popular and now it is even compared to the genetic algorithm, but of course, as I mentioned there are some problems there are some effort has to be made to find out what should be the values for the parameter. So, that PSO can perform in the optimal sense.

(Refer Slide Time: 18:26)



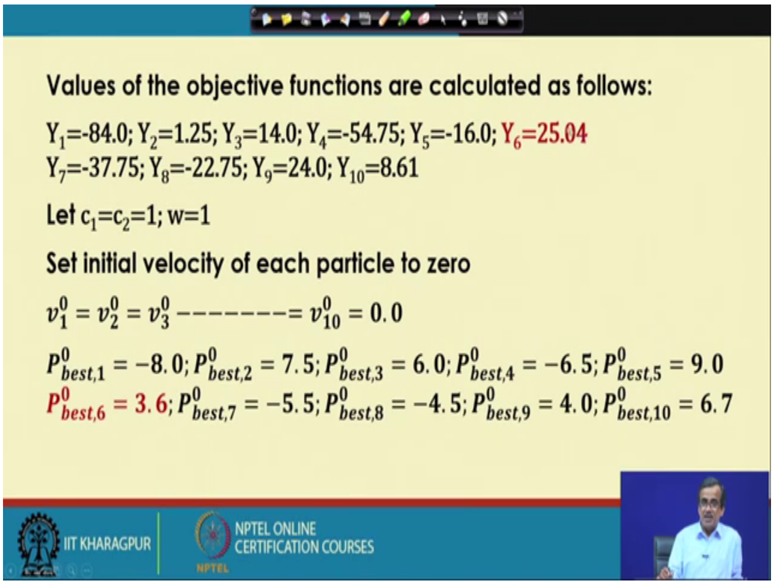
Now, to explain the working principle of this particular PSO I am just going to take the help of one numerical example.

Now, this is a very simple example involving only one variable. Now what I do is we try to maximize y is a function of only 1 variable. So, fx equals to minus x square plus 5 x plus 20 and for simplicity, I am just going to consider that this is a function of only 1 variable subject 2 the condition that x is lying between minus 10 and plus 10. So, this is the range for the variable.

So, what do you do? We start with iteration we set iteration t equals to 0. Let us consider that there are only 10 initial solutions for simplicity; that means, the swarm size is nothing, but 10 and lying within the range of this particular the design variables; let us generate some number; for example, say x 1 0 is minus 8.0, x 2 0 is 7.5, x 3 0; 6.0, x 4 0; minus 6.5, x 5 0; 9.0, x 6 0 is 3.6, x 7 0 is minus 5.5, x 8 0; it is minus 4.5, x 9 0 is 4.0, x 10 0 is 6.7.

Now, all such numerical values for the design variables are generated at random and within this particular the range for the variable and for simplicity, I have considered the swarm size is equal to 10.

(Refer Slide Time: 20:27)



Now, based on this particular your; the values of the design variables, now I can find out the function value the objective function value using this particular expression. So, if I just substitute the values for the x. So, I will be getting the Y values and those Y values are nothing, but this say Y 1 will come as minus 84.0, Y 2 is 1.25, Y 3 is 14.0, Y 4 is minus 54.75, Y 5 is minus 16.0.

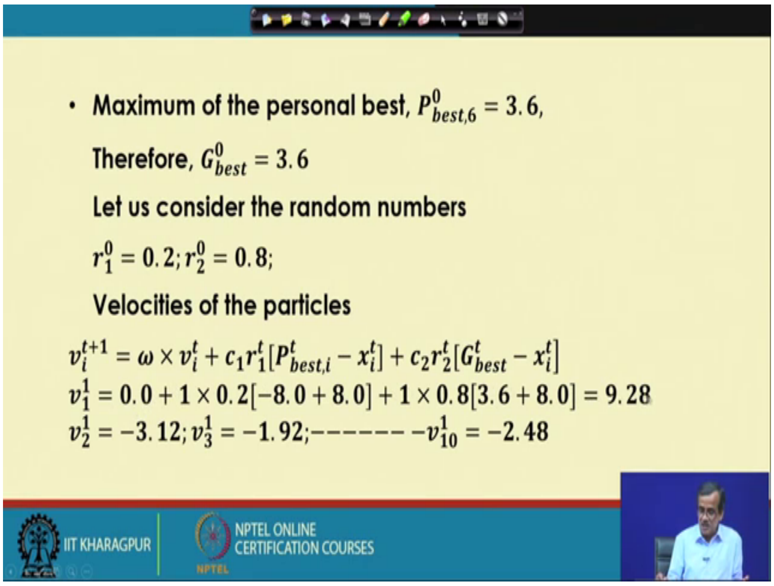
Y 6 is 25.04, Y 7 is minus 37.75, Y 8 is minus 22.75, Y 9; 24.0, Y 10 is 8.61. Now remember; we are solving one maximization problem. Now if I compare the values of this particular Y corresponding to this Y 6, I am getting the maximum value of the objective function. So, this is the best solution and corresponding to that actually what I do is. So, this particular Y 6 is the maximum that is the best solution and this solution is coming due to; what it is coming corresponding to this when we consider like your if you just see the last slide corresponding the x 6 0 that is equal to 3.6.

So, I am getting the best solution; that means, you are in this particular swarm P best 6 0. So, this will be the locally best. Now if I compare actually corresponding to the location one the variable one there is P best one corresponding to iteration one that is actually, we assign that in the best because there is starting 8.0, then P best comma to 0 is 7.5, P best comma 3 is 6.0, P best comma 4 pole 0 is minus 6.5, P best comma 5 0 is 9.0, P best 6 0 is 3.6, P best 7 0 is minus 5.5, P best 8 0 is minus 4.5, P best 9 0 is 4.0, P best 10 0 is 6.7.

Now, if you remember initially, we assign or we assume these are the values of the design variable, but this is the starting. So, what we assign is we assign these are the P best solution. Now we compare this P best in terms of their objective function value. Now why 6 is found to be the maximum as I told and that is coming due to this when x is equals to 3.6. So, this particular P best 6 0 will be declared as the globally best.

So, this is the way we will have to identify the P best as well as the g best. Now let us assume that c 1 equals to c 2 equals to 1, there is a cognitive constant and the social constant and w is equals to 1 and initially we said the velocities are equal to 0. So, all the velocities for the 10 the particles are set equal to 0. Now we generally we just go for. So, we have got this particular P best, we have got the G best also, this will be the G best, we will declare that this is the G best.

(Refer Slide Time: 24:27)



And so, the maximum of the personal best that is P best 6 0 is 3.6 and that is nothing, but the globally best that is G best 0, 0 means corresponding the 0th iteration that is equal 3.6. So, we have seen how to get the P best solution how to get the G best solution and we have assumed the values for c 1, c 2 and w. Now what we do is we assign some random value say r 1 corresponding to the 0th iteration since 0.2 r 2 corresponding to 0th iteration it says 0.8.

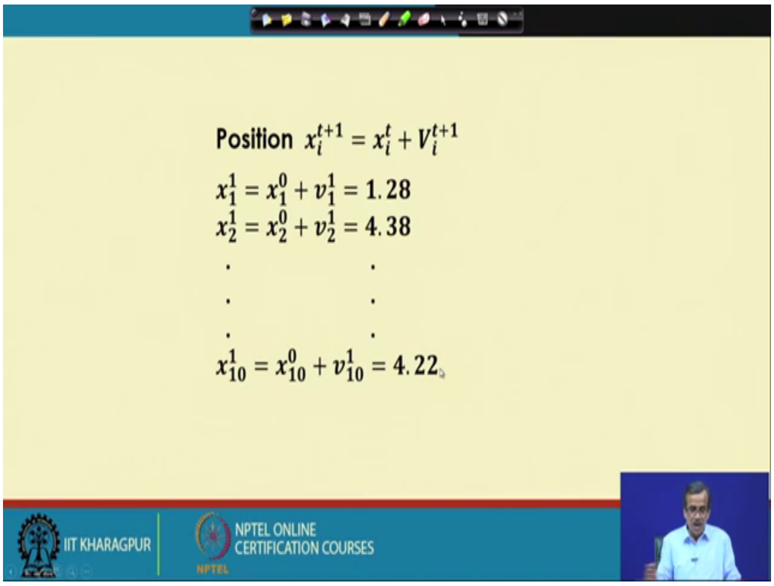
So, using this, now I am in a position to find out the numerical value for this particular the updated velocity that is v i corresponding to t plus 1 th iteration is nothing, but w multiplied by vi t plus c 1 r 1 t P best i t minus x i t plus c 2 r 2 t G best t minus xit, t indicates actually your; the iteration number. It is not the power, it is the superscript. Now if I assign the numerical value, for example, say v 1 corresponding to 1.

Now, the w is 0, sorry, w is not 0, w was equal to 1.0, but the initial velocity initial velocity for all the particles are assumed to be equal to 0. So, this particular contribution is nothing, but is equals to 0. Now c 1 was 1 r 1 is 0.2, I am putting r 1 x 0.2, then P best i corresponding to location one if you remember. So, what was we are your P best 1 corresponding to 0 that was minus 8.0. So, minus 8.0 minus x i t once again x it if we remember x 1 0 was minus 8.0.

So, minus of minus; so, this will become plus plus C 2. C 2 is nothing, but 1 r 2 t is nothing, but 0.8, then globally best is 3.6. So, 3.6 minus this x x i t that is your x 1 0 that was minus 8.0; so, this will become plus 8.0 and if you calculate, this will become 9.28. So, this is what this is nothing, but the velocity of particle one at the next iteration. Similarly, I can find out the velocity of the second particle third particle up to the tenth particle. So, I can find out the velocity of the second particle that is v 2 corresponding to 1 and if I follow this principle, I will be getting minus 3.12. Similarly v 3 1 is minus 1.92 and we can find out; as other velocities and the last one that is v 10 1 will be minus 2.48.

And once we have got all the velocities for 10 particles in the swarm.

(Refer Slide Time: 28:13)



Now, I can find out the position. So, position x i corresponding to t plus 1 is nothing, but x i t plus vi t plus 1 and I can find out x 1 1 is nothing, but x 1 0 plus v 1 1. So, v 1 1; I have already calculated. So, I can find out what is x 1 1 and that is equals to 1.28. Similarly, I can find out x 2 14.38 and all the positions I can find out and the last is x 10 1 that is nothing, but 4.22.

Now, if I see the initial position let me go back if I see the initial position. So, this was my initial position. So, this was my initial position the starting position for all 10 particles now at the end of this particular iteration. So, I am able to find out what should be my the position of the 10 particles at the end of this particular the iteration this completes one iteration of this particular the algorithm and this iteration will proceed and through a large number of iteration the algorithm will try to find out the optimal solution and that could be the globally optimal solution.

So, this is the way actually the particle swarm optimization can find out the optimal solution, but as I told that the performance of this particular algorithm depends on the number of parameters and we will have to set the parameters, accordingly, if we want to get a very good performance of this particular the algorithm now performance in the sense performance in terms of the quality of the optimal solution and in terms of the computational complexity an ideal optimization algorithm should be able to hit the globally optimal solution in less number of iterations and as I told there is a chance of further improvement and many people worked on this particular algorithm to further modify to make it more efficient and faster.

Thank you.