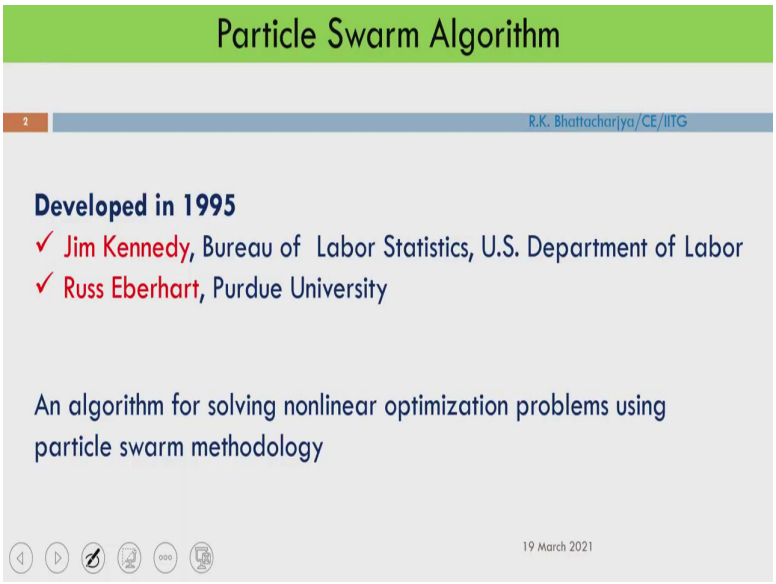
Optimization Methods for Civil Engineering
Dr. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 27
Particle Swarm Algorithm

Hello student. Welcome back to the course on Optimization Methods for Civil Engineering. So, today, we will discuss a different metaheuristic optimization method and that is your Particle Swarm Optimization.

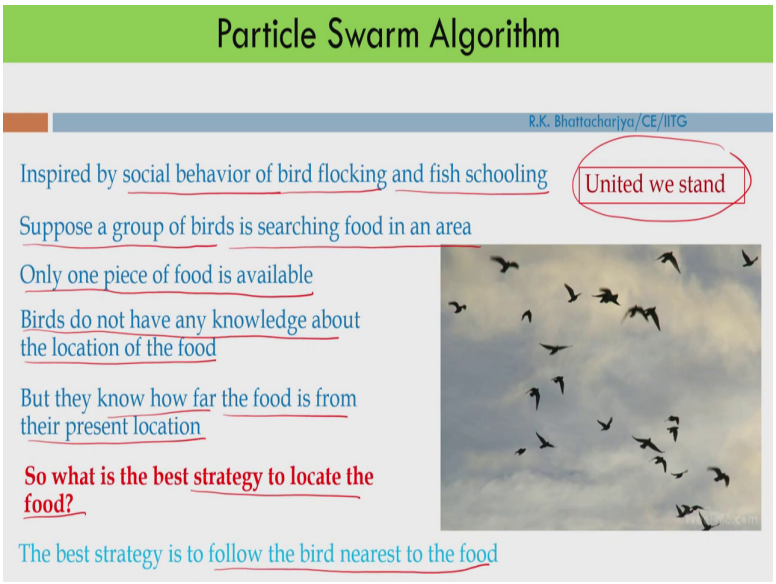
(Refer Slide Time: 00:46)



This algorithm was developed in 1995 by Jim Kennedy, so he was in Bureau of Labor Statistics, U.S. Department of Labor and Russ Eberhard, so he was in Purdue University. So, they have proposed particles from algorithm in 1995. So, this is an algorithm for solving non-linear optimization problem using particle swarm methodology. So, we will discuss this

particular algorithm. This is also one of the very popular optimization algorithm, metaheuristic optimization algorithm and basically, it can be used for solving a non-linear non convex optimization problem.

(Refer Slide Time: 01:31)



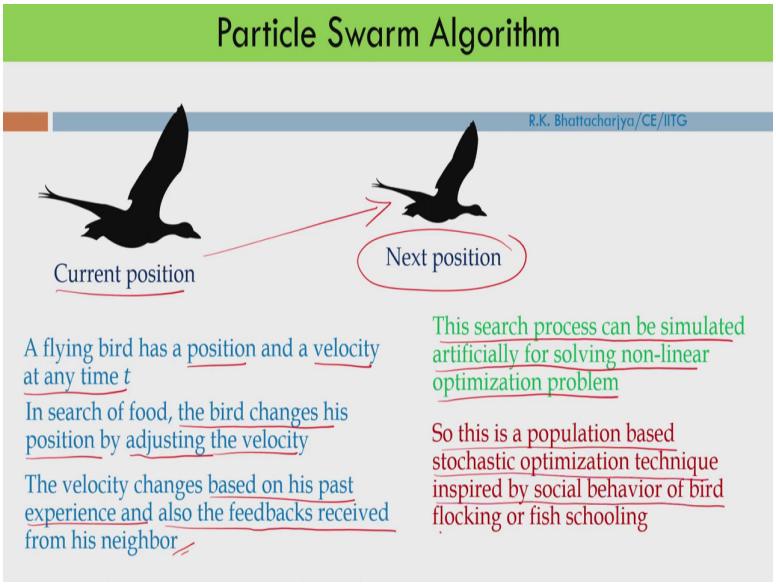
This algorithm is inspired by social behavior of bird flocking and fish schooling. United we stand. So, basically this is a group activity and they basically they are when the birds are trying to find food. So, whatever process they are following, so that is artificially simulated for finding an optimal solution of an optimization problem.

Suppose, a group of bird is searching for food in an area and only one piece of food is available. So, that is the solution of that particular search process. The birds do not have any knowledge about the location of food ok. So, they do not have any knowledge about the

location of the food and however, they know how far they are from the food from their present location.

So, what is the best strategy to locate the food? So, best strategy is that you follow the bird nearest to the food. So, what they are doing? They are trying to or they are communicate each other. So, the bird has the information about his position how far he is from the food. Similarly, they also have the information about the location of other bird and then, the best strategy and therefore, the best strategy is to follow the bird nearest to the food.

(Refer Slide Time: 03:01)



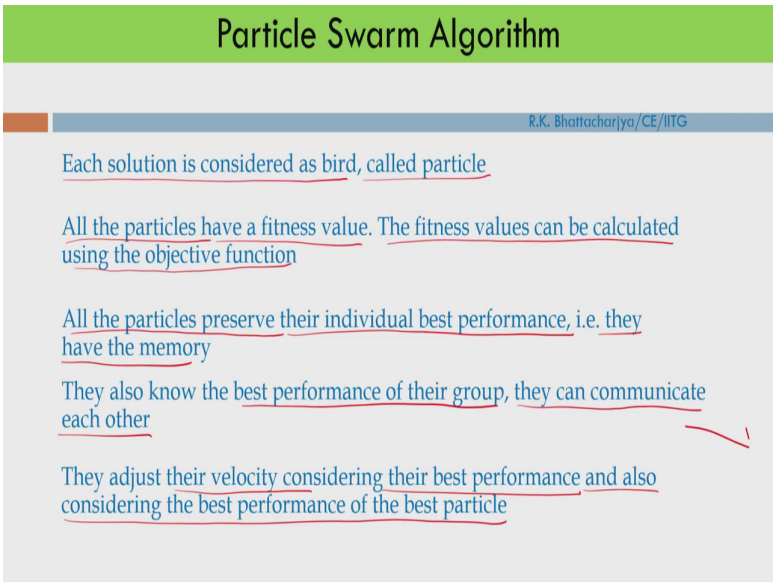
Suppose, this is the current position of a flying bird. A flying bird has a position and a velocity at time t ok. So, this is the current position. So, he has a velocity and he has also a position. In search of food, the bird changes his position by adjusting the velocity. So, velocity means he will change his speed as well as his direction.

The velocity changes based on his past experience and also the feedback received from his neighbor. So, what he is doing; a bird is doing? So, he is changing his velocity based on his past experience. So, he actually he has the memory. So, he is storing all his past your location; that means, best past based location that is his personal experience, he is storing as well as he is also getting the information from his neighbor ok.

So, therefore, he is changing his location or he is changing his position based on his past experience and also the feedback received from his neighbor. So, based on that, he will change his position and this will be the new position of this bird. So, basically from here to he is moving up to this location. So, this search process can be simulated artificially for solving non-linear optimization problem.

So, this is a population based stochastic optimization technique inspired by the social behavior of bird flocking and fish schooling. So, this is a population based algorithm. So, just like classical optimization algorithm, so we are not going from one solution to another solution; but we are actually taking a population and we are trying to search in multiple direction. So, this is again stochastic method, so this is not deterministic and then, as I said that this is inspired by the social behavior of birds flocking and or and fish schooling.

(Refer Slide Time: 05:14)



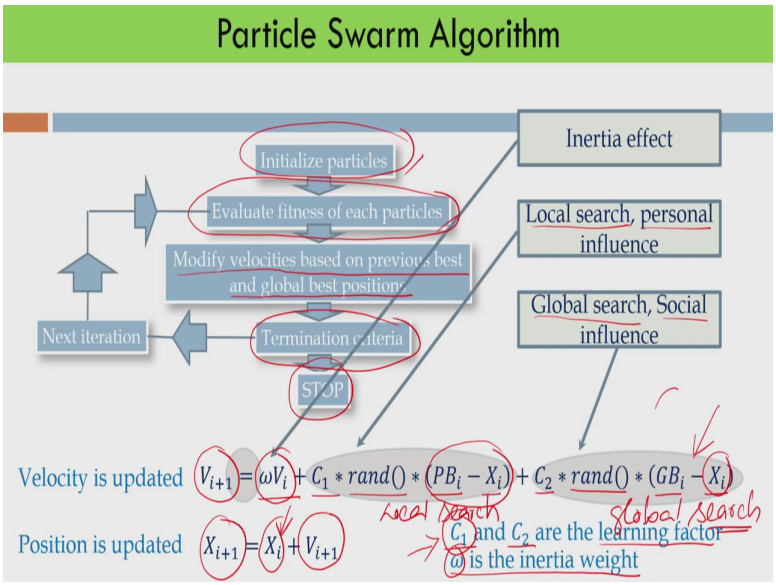
So, here, each solution is considered as bird. We call it particle. So, this is the solution. All these particles have fitness value. The fitness value can be calculated using objective function. So, what we are doing here? So, we call the solution as bird or we call it particle and for each particle, I can evaluate his position by using the objective function; that means, how far or what is the quality of that particular position or particular solution.

So, that can be calculated using objective function. All the particle preserve their individual best performance. So, that means, they have the memory. So, what they are doing? They are preserving their individual best performance; that means, best location, they are preserving.

They also know the best performance of their group; that means, they can communicate each other. So, these are two qualities. So, one is that they have memory. So, they are preserving their best performance and second one is they are also aware about the best performance of

their group; that means, they can communicate each other and they are aware about the group performance. They adjust their velocity considering their best performance that is his personal best and also, considering the best performance of the best particle.

(Refer Slide Time: 06:42)



This is the flowchart of particle swarm optimization algorithm. So, we have initialized the population and then, next step is that we will evaluate the fitness of each particle. So, we are initializing the particle, then we are calculating the fitness value of each particle and then, we are modifying the velocity based on previous best and global best position.

Then, we will check the termination criteria. If termination criteria is satisfied, then it will terminate; otherwise, it will go to the next iteration so, this process will continue and till we are not satisfying the termination criteria. So, once it is satisfied, then you will basically

declare the optimal solution. So, here basically we are using two equations; one is to update the velocity. So, velocity is updated something like that.

So, this is the current velocity of the previous location and this is the updated velocity. So, which is equal to omega V i, then plus C, a random number and then, this is PB is the personal best and this is the X i that is the current location and similarly, C 2 multiplied by a random number, then a global best minus X i; X i is the current location.

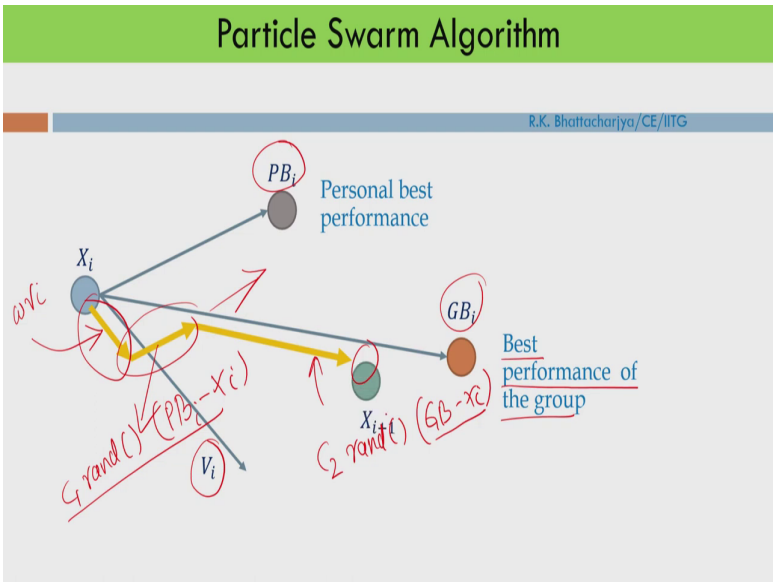
And once you are getting the new velocity and then, we can calculate the new position and which is equal to X i plus V i plus 1. So, this is the previous location. So, this is the new location and new location is X i plus V i plus 1. So, here this is the inertia effect; that means, before taking a decision, suppose the particle or the bird has taken a decision to sense his velocity and but due to inertia effects, so he will moves some distance.

So, this is the inertia effect and this is basically give you the direction towards his personal best. So, therefore, you can say this is local search or personal influence, this part and the other one is the global search or basically the social influence. So, here GB minus X i is giving you the direction towards the best particle, towards the best particle and P B minus X i is giving the direction towards his personal best.

So, therefore, this is a combination of local search. So, this is you can say local search and this is your global search ok and anyway, so this is the inertia effect. So, as you have seen this is a very simple algorithm. So, it can be implemented easily. So, with few lines of code, you can implement this particular algorithm and as I said this is one of the most popular and powerful meta heuristic optimization algorithm for solving non-linear optimization problem.

So, here C 1 and C 2 are the learning factor and omega is the inertia factor. So, therefore, suppose you want to so or you can say that C 1 and C 2 are basically step length. If you are using large value of C 1, so you will give a big jump; but if you are using smaller value of C 1, that means, you are basically searching near that particular solution. So, I can actually gives suitable value of C 1, C 2 and omega and can control the search process.

(Refer Slide Time: 10:42)



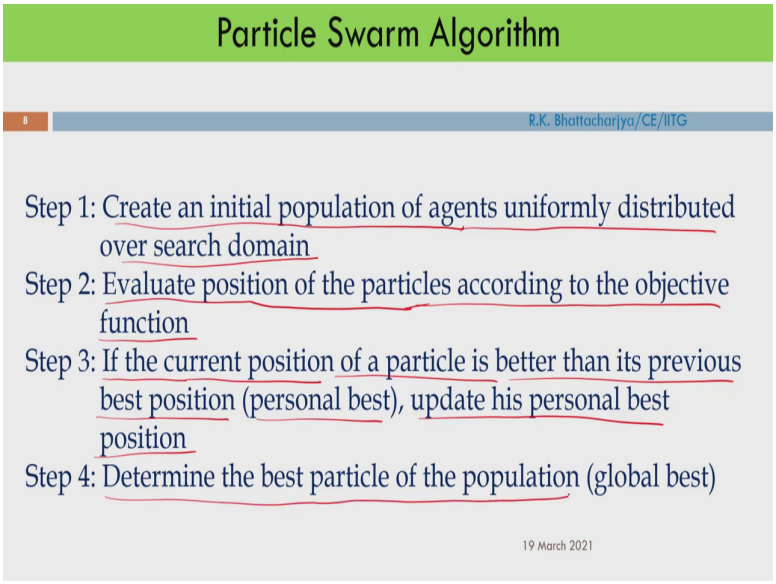
Suppose, this is the current location of a particle and this is his personal best. So, that he has preserved his location, personal best performance and this is the global best performance or you can say the best performance of the group so far. And now, suppose your velocity at this location is V i.

So, therefore, how he will adjust the velocity or how he will go to the next location? So, this is the effect of inertia. So, because of inertia, so he will initially move in this direction and then, he is going towards his personal best direction and this part is this part is coming C 1 rand. Then, personal best minus X i ok.

So, this is his personal best and this is basically omega V i ok. So, this is the inertia effect and then, he will move towards the global best. So, this is basically C 2 rand is GB minus X i ok.

So, this is your GB. So, that is GB minus X i. So, and that will be his new location that is X i plus 1.

(Refer Slide Time: 12:18)

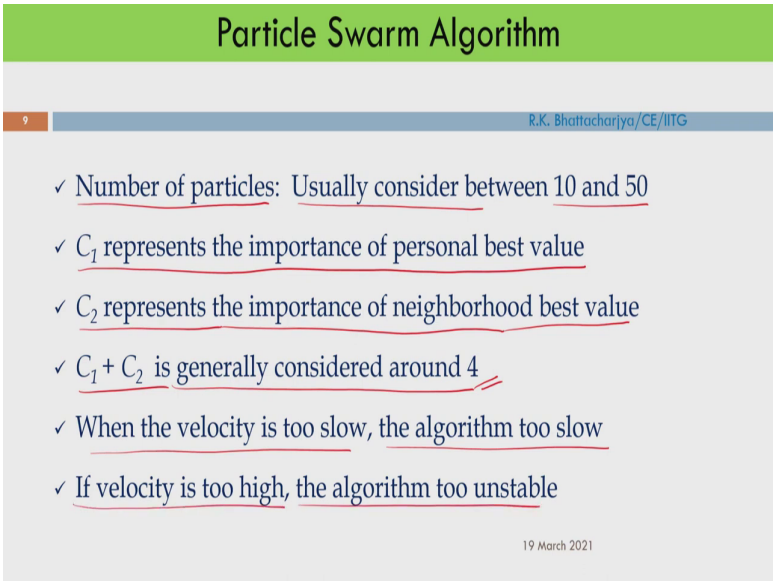


So, what are the steps of this algorithm? So, step 1, so we will create initial population of agent uniformly distributed over the search domain. So, I can use any uniform distribution to generate this population and once, I am getting this population, we will evaluate the position of the particles according to the objective function.

So, I can use any fitness function and fitness function is basically you are getting from the objective function and you can evaluate the particles and so, whatever new position you are getting, if the current position of a particle is better than its previous best position that is the personal best. So, we will update his personal best position.

So, whatever new position you are getting that is X i plus 1, if this position is better than his previous personal best position, so we will update that one. So, we will update his personal best position and then, we will determine the best particle of the population and we call it global best ok. So, this process we will continue unless and until we are not reaching the termination criteria. Once you are reaching the termination criteria, so we will stop the iteration and we will the final solution will be displayed.

(Refer Slide Time: 13:40)

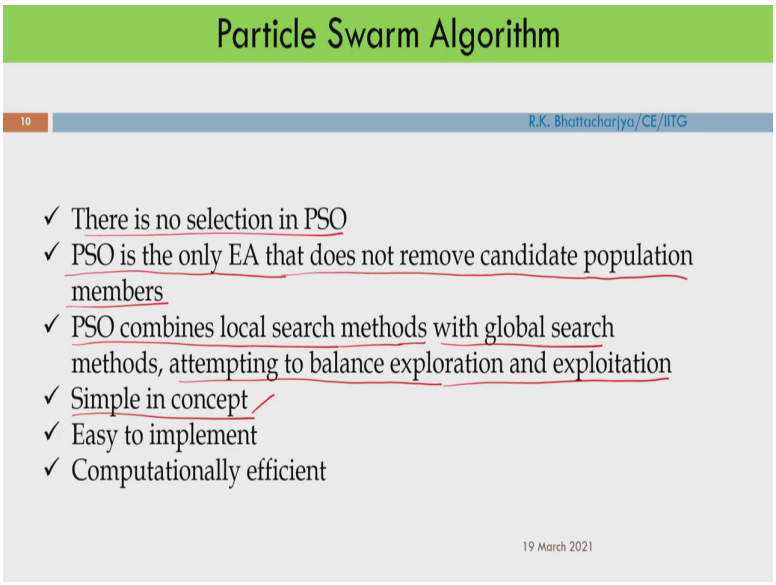


Now, how many numbers of particle you should consider? So, usually, consider between 10 to 50; so, number of particles, so you can minimum 10, you can consider and I think 50 is sufficient in this case. And C 1 represent the importance of personal best value. So, as I said that by C 1, I can control the step length.

So, that will represent the importance of personal best value and C 2 represents the importance of neighborhood best value; that means, the importance of the global solution and generally, C 1 plus C 2 is around 4. So, you can consider that C 1 equal to 2 or C 2 equal to 2 or C 1 equal to 1.5, C 2 equal to 2.5 something like that. So, summation should be around 4.

So, when the velocity is too low, the algorithm is too slow ok. So, velocity is less means the algorithm is too slow and you may need a lot of iteration to reach the optimal solution. On the other hand, if velocity is too high, the algorithm will be unstable. So, in that case, it will be unstable. So, therefore, you have to choose this particle in such a way that velocity is neither too slow in that case, it will take lot of time or neither it is too fast, in that case the search process will be unstable.

(Refer Slide Time: 15:11)



So, unlike genetic algorithm, then other metaheuristic algorithm we have discussed suppose ES. So, PSO does not have selection operators. So, as you have seen, so we do not have in PSO does not have any selection operator; that means, you are not deleting any particle. So, what you are doing basically? You are trying to get you are trying to get his new position.

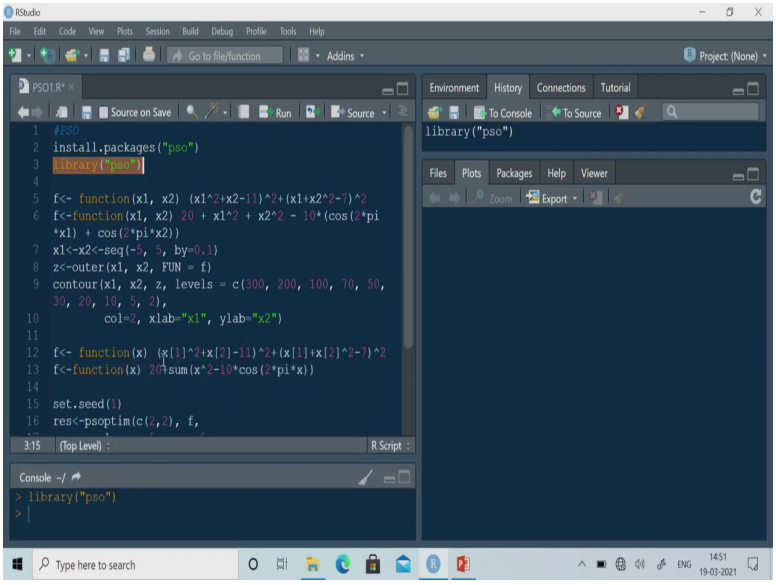
So, you are not deleting any particles and therefore, we do not have any selection in PSO. PSO is the only ES that does not remove candidate population member. So, as you have seen, so we are not removing any solution; so, only we are updating his position. PSO combines local search.

So, as we have seen. So, we are also giving importance to his earlier best performance that is the personal best as well as global search. So, PSO combines local search method with global search method, attempting to balance exploration and exploitation. So, the algorithm is very simple in concept as you have seen, the algorithm is very simple.

So, what we what we need basically? So, we need only two equation to update the velocity and then, we are updating the position. So, it is a very simple algorithm. I can show you that this is a very powerful algorithm in finding global optimal solution of a function and it is easy to implement.

So, the code is very simple and computationally efficient. So, this is all about particle swarm optimization. So, as you have seen, the algorithm is very simple in concept; we can implement this algorithm with 10 to 15 lines code. So, you can write a code in either MATLAB or your R program or any other language ok. So, you can use any other language. But here in this particular course, I will solve some problem using the PSO package available in R. So, here I am not writing the code. So, directly, I am using the PSO function available in R to solve the problem.

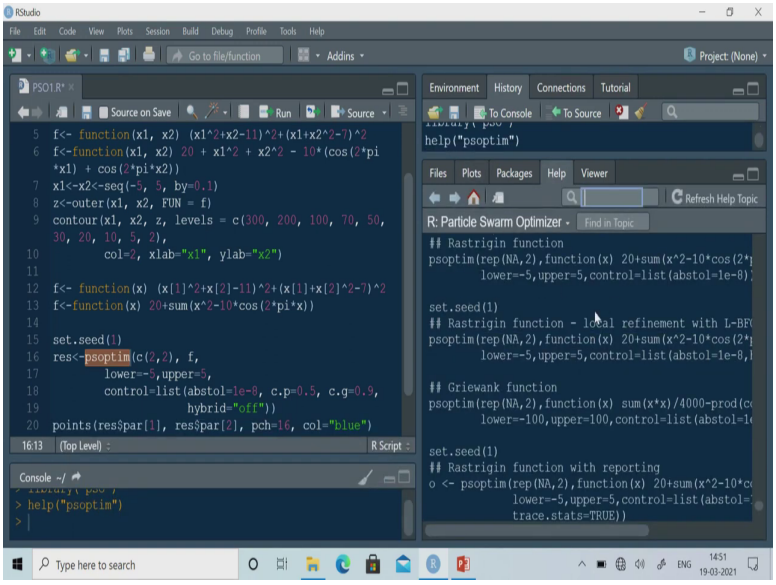
(Refer Slide Time: 17:31)



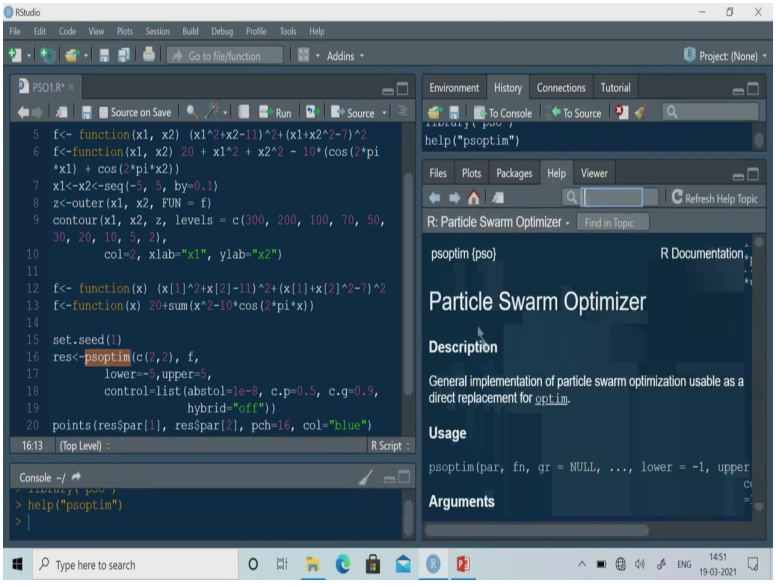
So, let us solve some problem using R. So, I will use PSO algorithm here. So, for using PSO, you have to install the PSO package. So, as you have seen. So, this is you have to you have to do only for the first time. So, you have to install this one. So, I have already installed in my computer. So, I will not install it again; however, if you have not installed these thing.

So, you please install fast. So, you can write install dot packages and within codes you can write “pso”. So, this will install the PSO package; but you need internet connection to install that one. So, I have already installed. So, therefore, I will not run this particular line. Then, next step is to include the library. So, let us include this library that is your “pso” library.

(Refer Slide Time: 18:32)

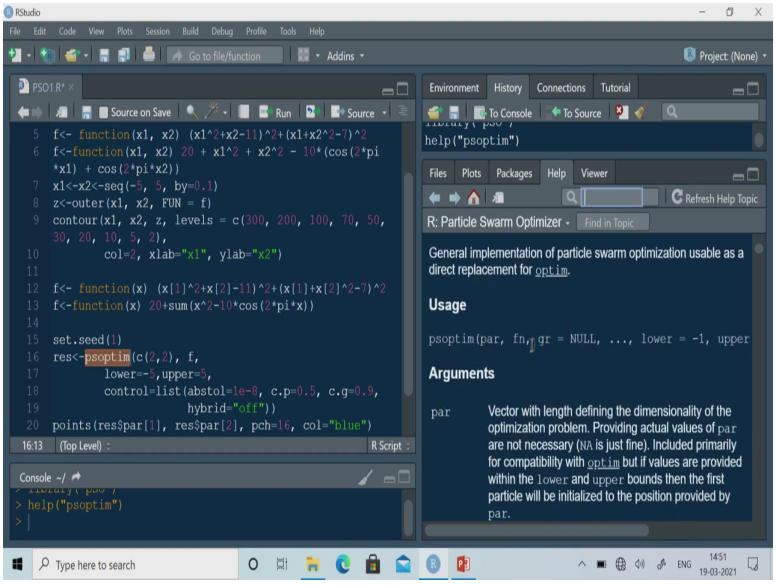


(Refer Slide Time: 18:55)



So, now it has been included. So, I can see actually, so here the function is psoptim. So, let us see this function first. So, I can go to help and then, I am writing. So, dso; this is pso ptim and then, you can see this function here, you can see this. So, this is particles from optimizer. So, this is the function; PSO function.

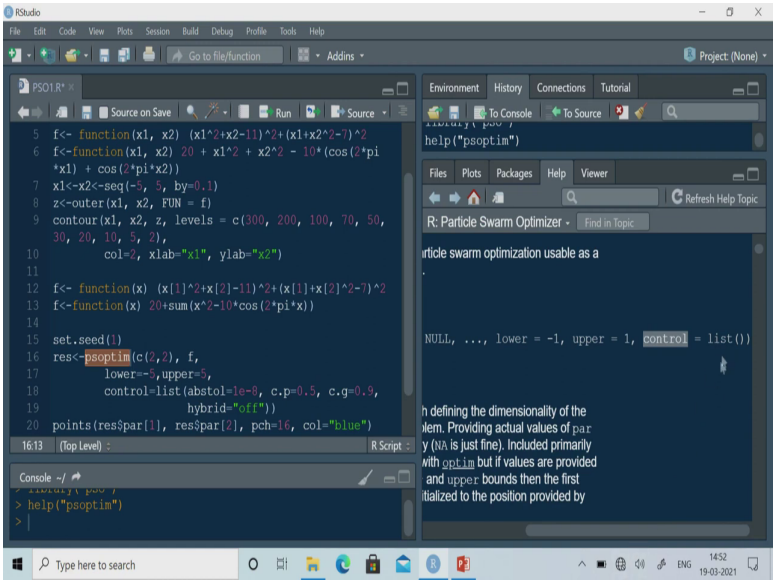
(Refer Slide Time: 19:00)



So, what you have to do here. So, the function name is p s optim and here, you have to define the parameter. That means, just like the classical method, so you can define the initial solution because this is a part of optim package. So, in case of classical method, so we generally define an initial solution. So, in order to make its similar, so in PSO algorithm, so the provision is there, you can define an initial solution.

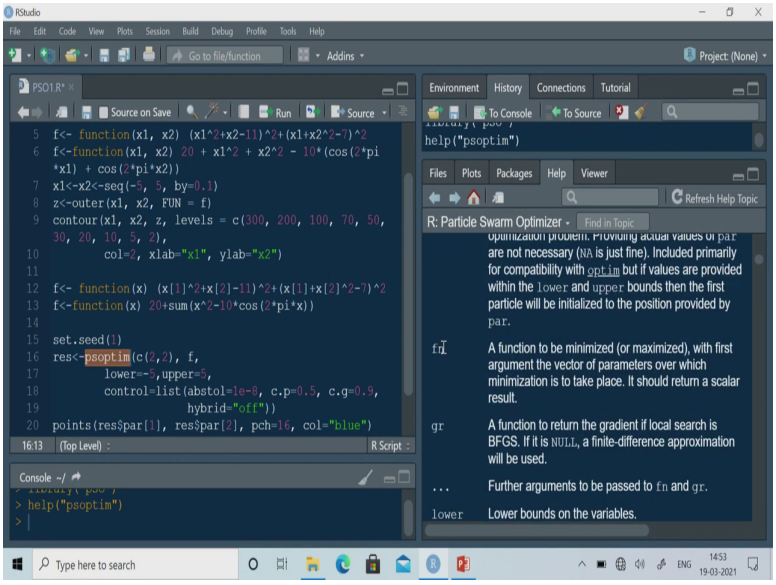
So, in that case, what will happen? The particle will be generated near to that solution; but that is not mandatory. So, you can also define the number of variable only; that means, I can define NA; that means, I will not define the initial solution. But I will define the number of variable and next is you have to give the function name and then, some gradient information all those thing you can put; but if you are using the hybrid method.

(Refer Slide Time: 20:01)

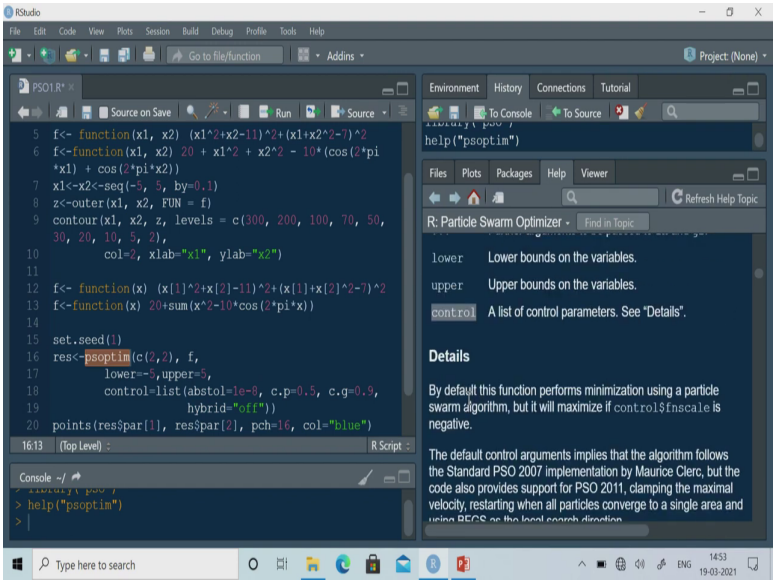


And basically, you have to define lower bound and upper bound and finally, there is some control and you can list it the control parameters here ok. So, what I need basically? So, I have to define this function fn and I have to define lower bound and upper bound and I have to define the parameter.

(Refer Slide Time: 20:21)

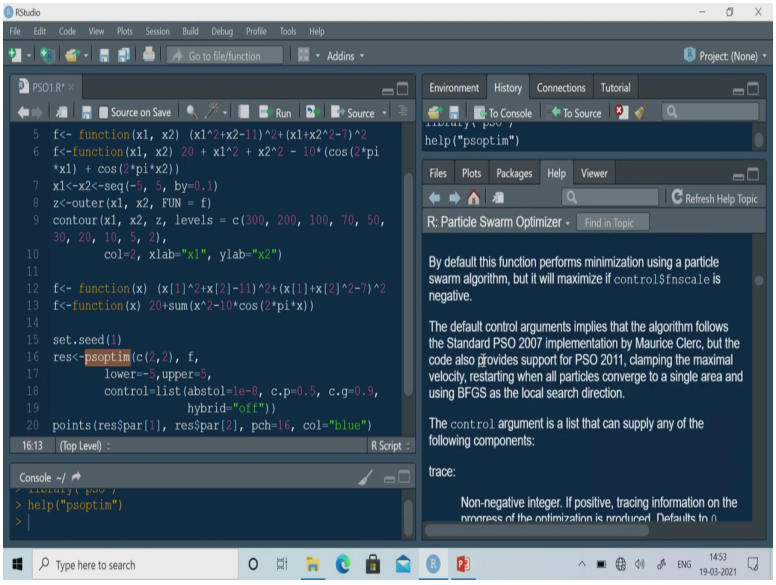


(Refer Slide Time: 20:37)

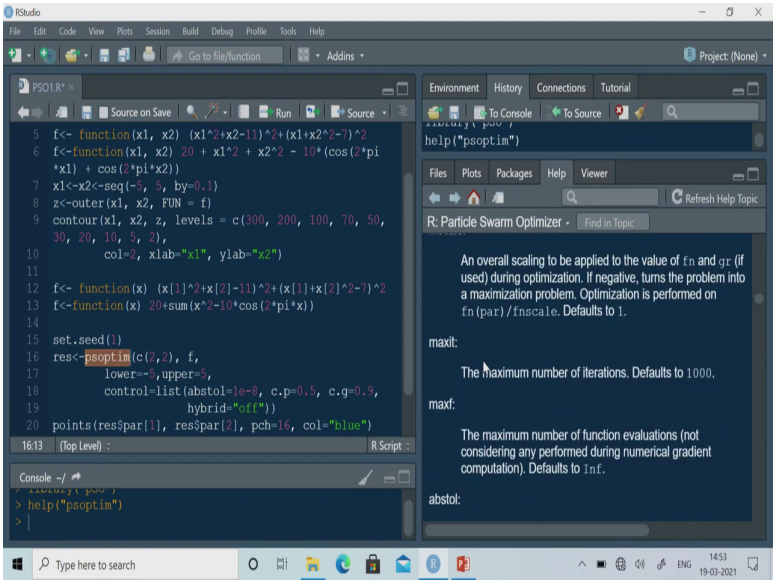


So, it has been defined here; what is parameter what is fn, what is gr. So, if you are using hybrid method, a function to return the gradient if local search BFGS is used basically. So, by default, it is null and then, you can define what is lower bound; what is upper bound and you can define the control parameter. So, you can see the details, by default this parameter has been defined, but you can also define it to if you want to sense those parameters.

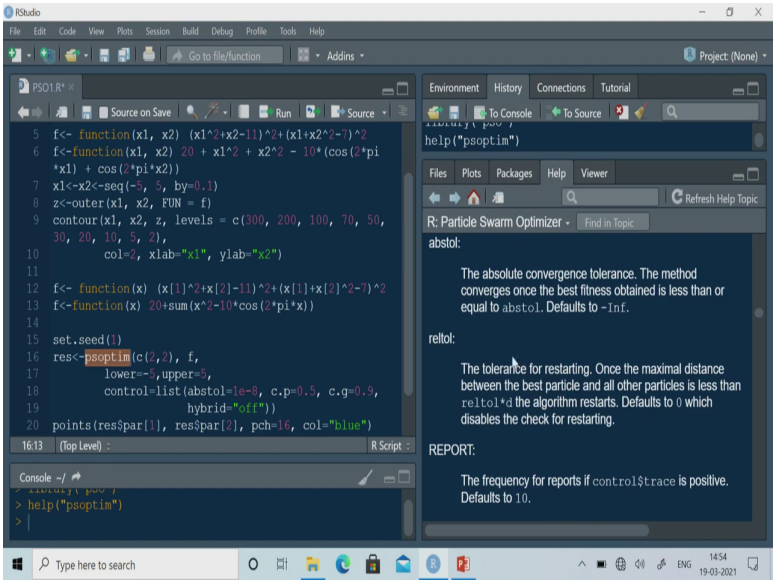
(Refer Slide Time: 20:53)



(Refer Slide Time: 20:57)

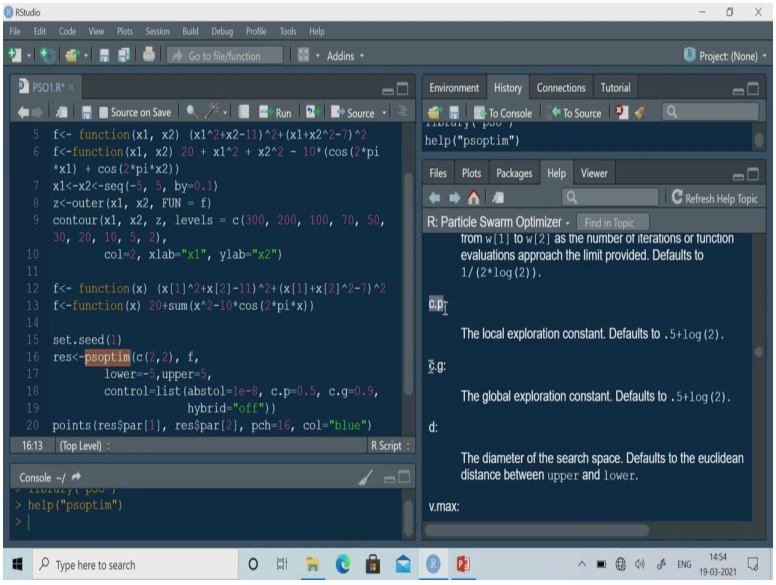


(Refer Slide Time: 21:03)



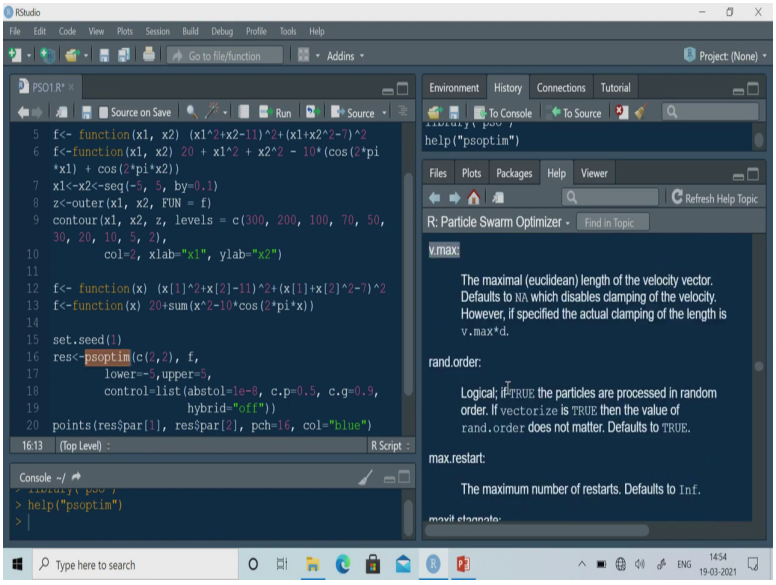
So, I will show you how you can change these parameter and here, you can define maximum number of iteration; then maximum number of function evaluation, you can put actually that maximum iteration I can put by default it is your 1000 and by default it is infinity. Maximum function evaluation is infinity; that means, it can go up to any value and similarly, other parameters you can define ok.

(Refer Slide Time: 21:25)



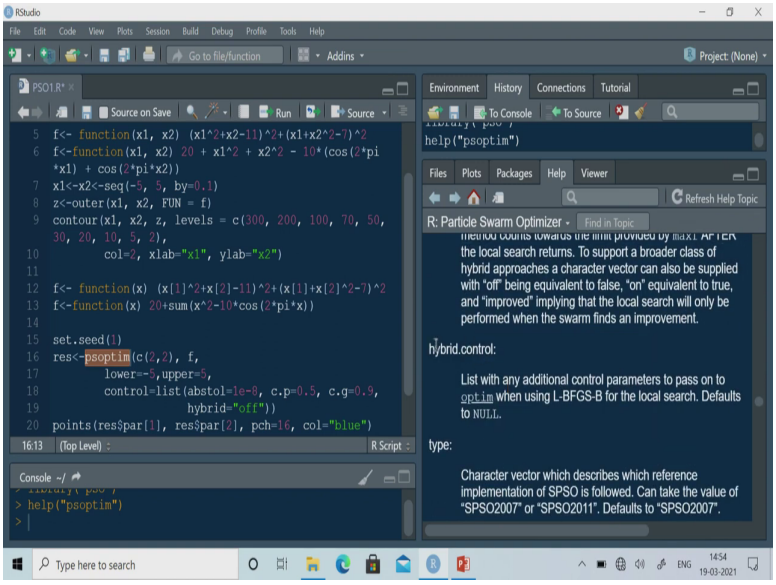
So, generally, I would be I would like to define what is c.p that is basically C 1 and c.g that is the your C 2. So, this is local exploration constant and this is global exploration constant. So, in my presentation, so I have written this is C 1 and this is C 2.

(Refer Slide Time: 21:45)



And similarly, you can also define what is v.max ok. So, these are some of the things you can and you can also use hybrid ok. So, you can also use hybrid method.

(Refer Slide Time: 21:51)



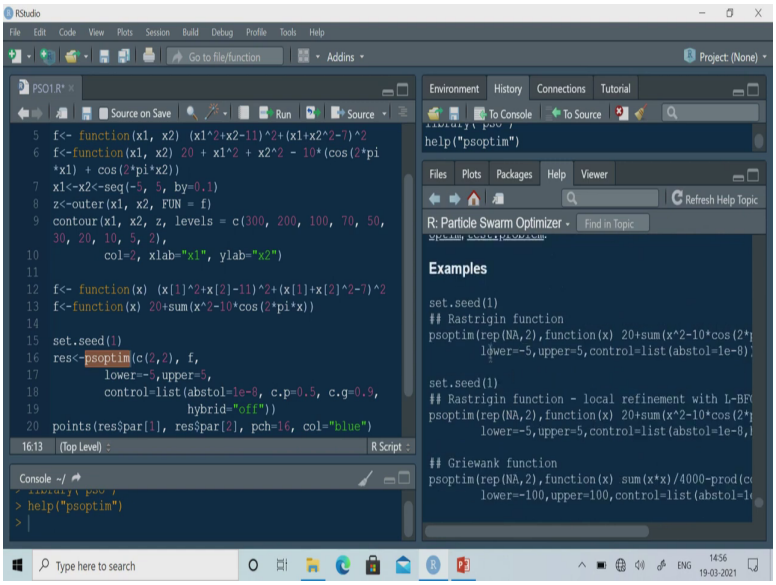
So, hybrid method is something like that. So, as you have seen, so whenever you are applying the genetic algorithm or any metaheuristic optimization method, so we are not calculating the gradient of the objective function as well as constraint function. So, therefore, what may happen?

So, this algorithm may not give you the exact optimal solution; but it will give the near optimal solution. So, you may not get the exact optimal solution, where gradient is 0; but you may get a solution which is near to the global optimal solution. So, in that case, what you can do basically? So, you can apply the classical method.

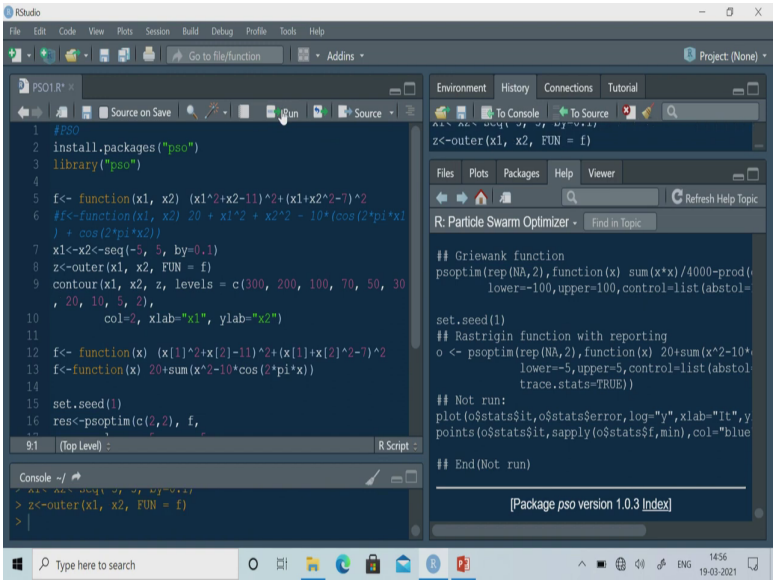
So, classical method is using gradient information and you may you will get the exact optimal solution. So, we call it hybrid optimization method; that means, initially, we will apply PSO

and then, we are applying classical optimization method. So, in this case, we are applying BFGS algorithm that is for local search ok. So, I can also define that one.

(Refer Slide Time: 23:04)



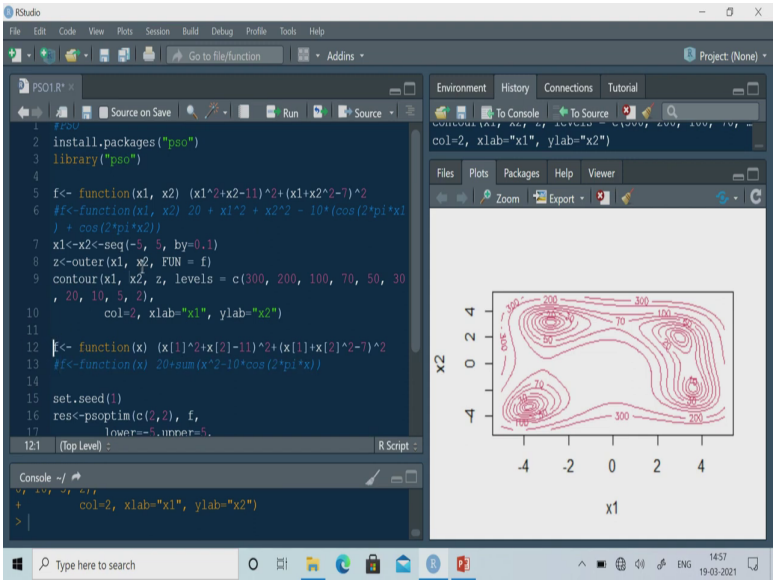
(Refer Slide Time: 23:05)



So, and these are some of the control parameter you can see and they have also given some example. So, for particular this function, they have shown you the example. So, anyway, so let us discuss this thing this particular problem. So, first, I will solve the function 1. So, what is function 1?

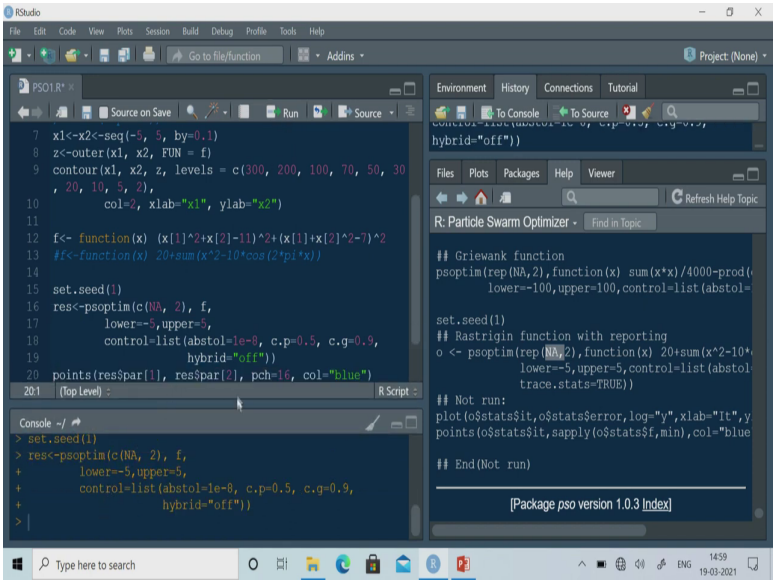
So, maybe I will mark comment this one. So, this function 1 is x 1 square plus x 2 minus 11 whole square plus x 1 plus x 2 square minus 7 whole square and this is between minus 5 and plus 5. So, let us execute this particular line and then, let us execute x 1, x 2 values; then I am calculating the z values, then I am plotting the contour ok.

(Refer Slide Time: 23:46)



So, I am getting this contour, then this part is just to plot the function ok. So, this is not part of PSO algorithm. So, I am just plotting this particular function as you know. Now I have to write this function in this order in order to apply PSO. So, I can actually write an x and within x basically I have two variable that is your x 1 and x 2. So, this function I am not using; second function. So, first function I am using.

(Refer Slide Time: 24:16)



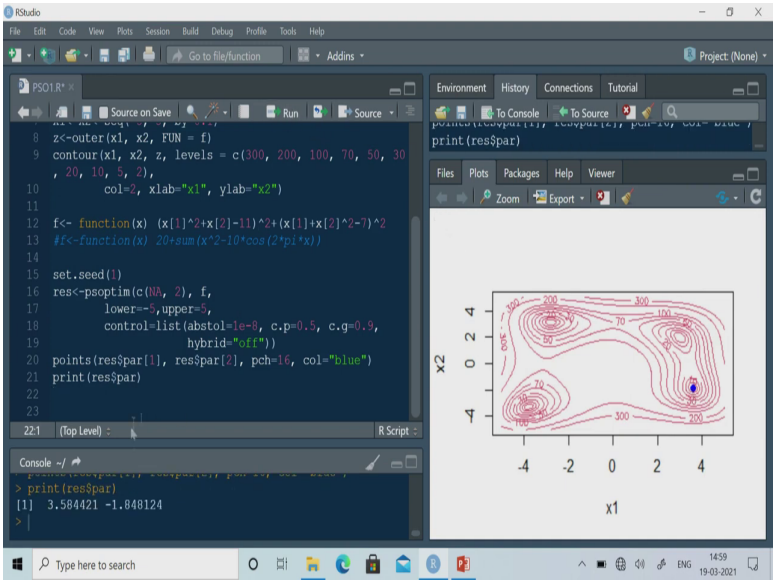
So, let us execute this particular function and then, here I am using seed value 1. So, this set seed value we are doing just to get the same result ok. So, this is you can use or if you are not using, you are not getting similar same result. So, let us execute this thing and then, I am using psoptim. So, what we are doing here? I am defining the initial point that is 2 and 2, but as I said that you may not define.

So, just like here suppose we can also write that NA 2. So, if we are writing, we are we may not define; this is NA and this is 2 that because we have two variable here and lower bound is minus 5, upper bound is plus 5 and the function is f. So, this function f function I am using. So, function is f and in control, this is the absolute tolerance I am putting that is your 1 e to the power minus 8.

This is the tolerance I am putting and then, c.p I am putting 0.5. So, I may define or I may not define. If I not define this value, so c.p and c.g I may define or I may not do that. So, if you are if we are not giving that one the default value will be taken and hybrid off ok. So, I am not using any hybrid.

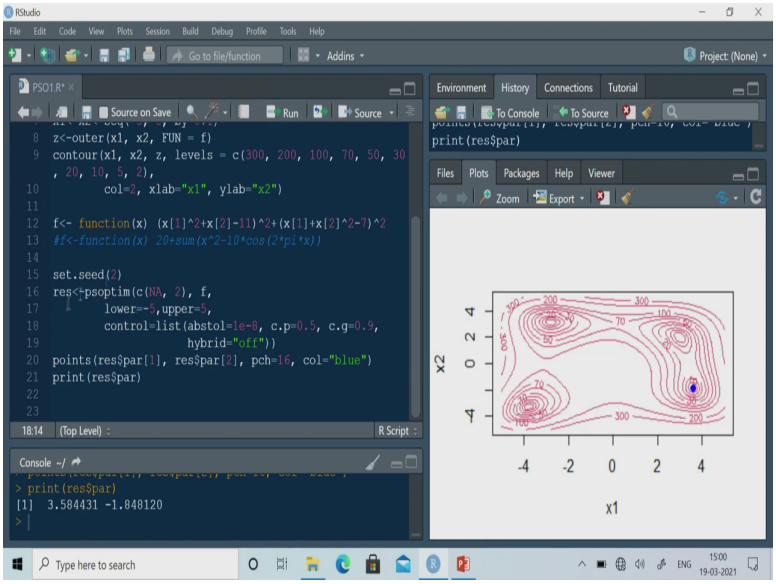
So, you can. So, it is off basically or you may not write this thing. So, if we execute this particular line, so it will execute the PSO algorithm. So, we are getting the solution, you can see that one. So, we have executed this one and then, so let us plot this particular points over this contour map.

(Refer Slide Time: 26:06)



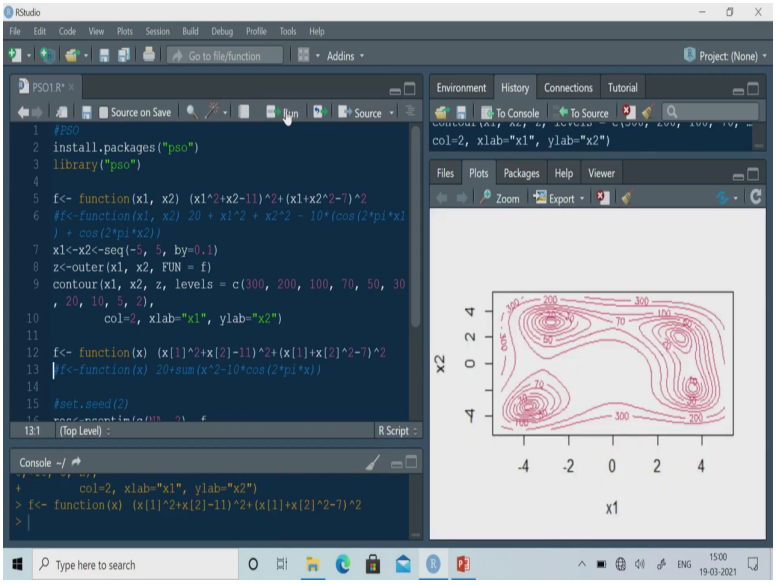
So, let us execute this one. So, I am getting this solution ok and the point is this one 3.584421 and this is minus 1.848124 ok. So, what I can do basically? So, maybe I can execute this one. So, every time, I should get the similar result.

(Refer Slide Time: 26:34)

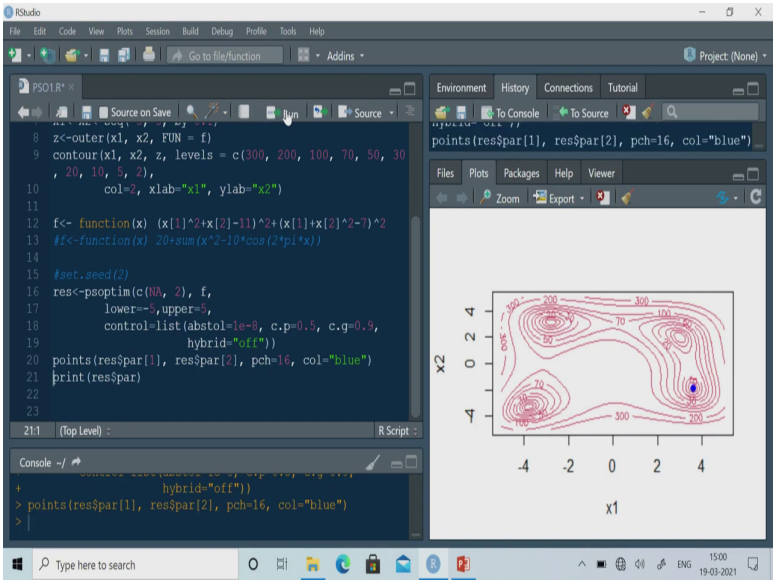


So, let us see yeah. So, you just see I am getting the similar result. If I am not executing this one or if I put a different number, let us see I am getting the same result. So, let me start it again.

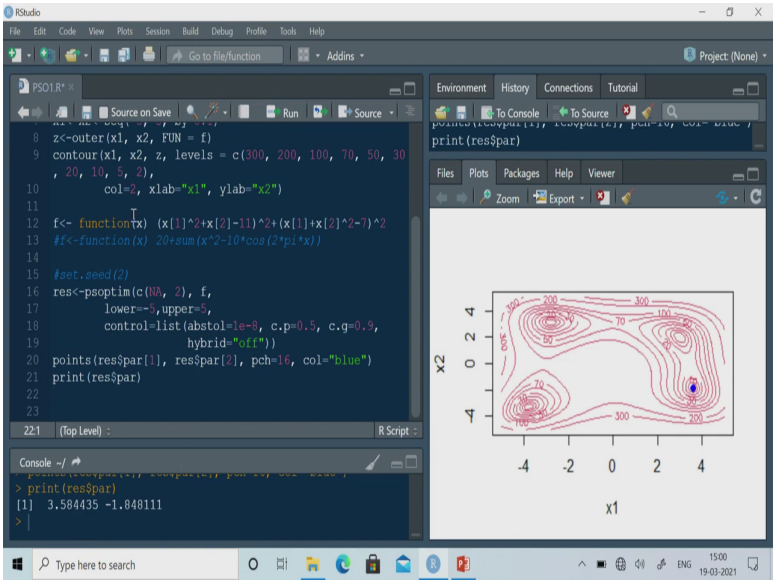
(Refer Slide Time: 26:50)



(Refer Slide Time: 27:05)

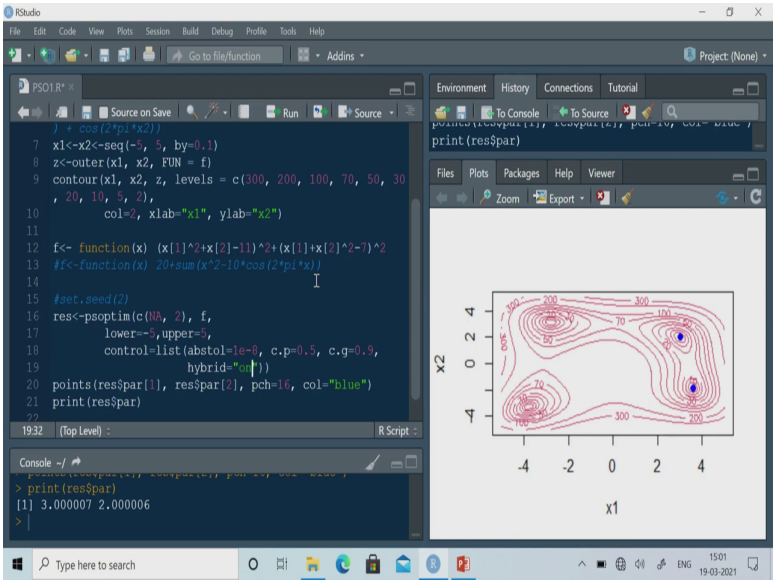


(Refer Slide Time: 27:12)



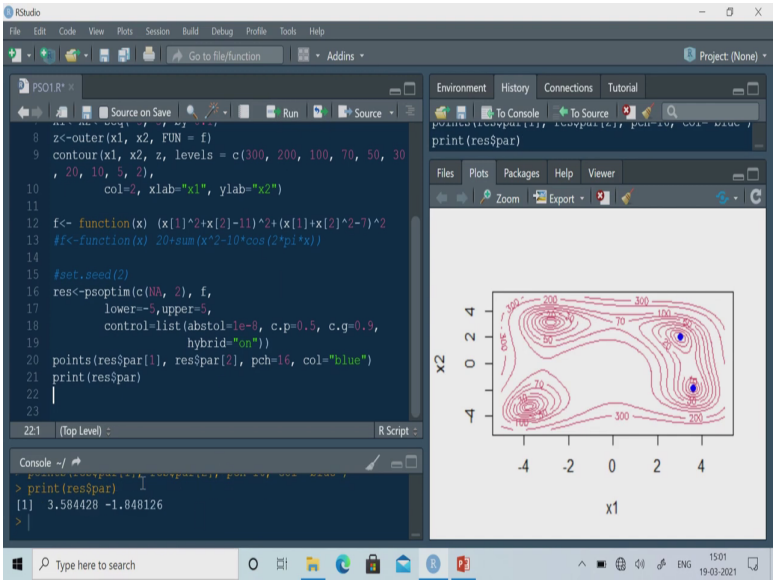
So, this time I am not defining this one ok. So, let me execute from here. So, this is the function and so, I am getting this solution again. So, I am getting this particular solution yeah. So, this time, you just see I am getting this solution.

(Refer Slide Time: 27:15)



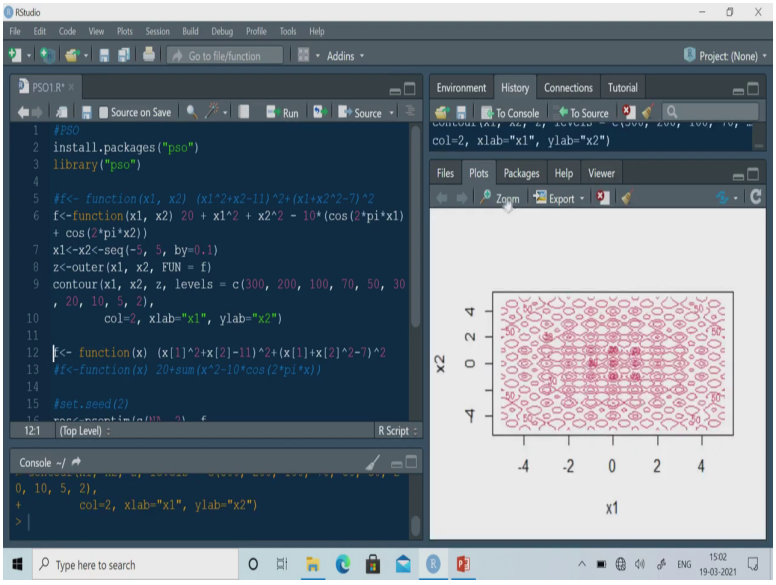
And if you so this solution is 3, 2, but I am getting three and 3.007 and 2.00 something like that. So, what I can do basically? So, I can also use hybrid ok. So, if I use hybrid, then suppose this is hybrid on, then so, I am getting this solution.

(Refer Slide Time: 27:43)



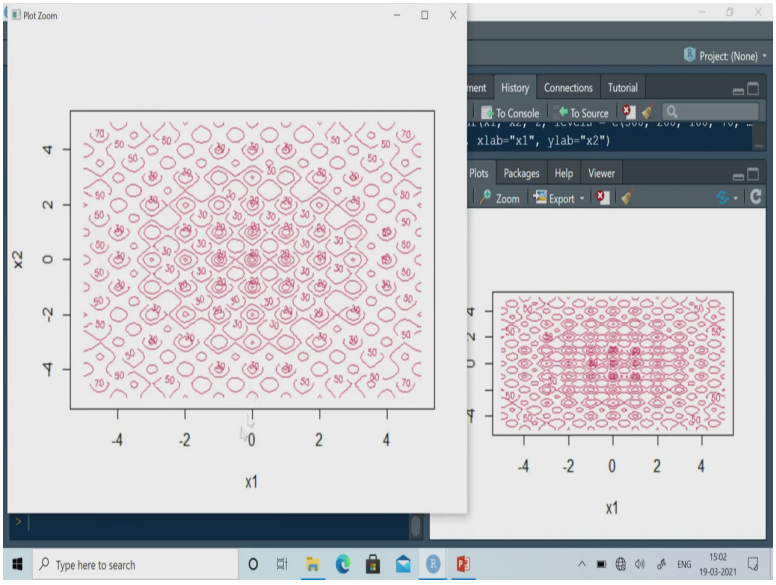
So, in that case, what will happen? There will be some refinement. So, what you will do basically? If you are getting near optimal solution, then after that if you are applying classical methods, so you will get the exact optimal solution of the problem, where gradient is 0. So, I hope this is fine. So, let us try with the next function. So, I would like to clean this part. So, I will delete the history. Similarly, I will also clean the console.

(Refer Slide Time: 28:11)



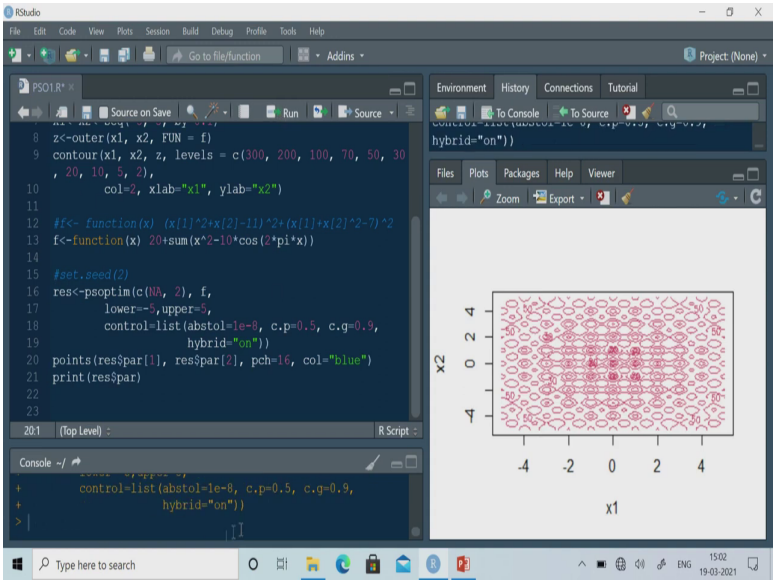
So, now let us execute the second function. The second function if you plot it, so this is a very complicated function as you have seen earlier also.

(Refer Slide Time: 28:33)

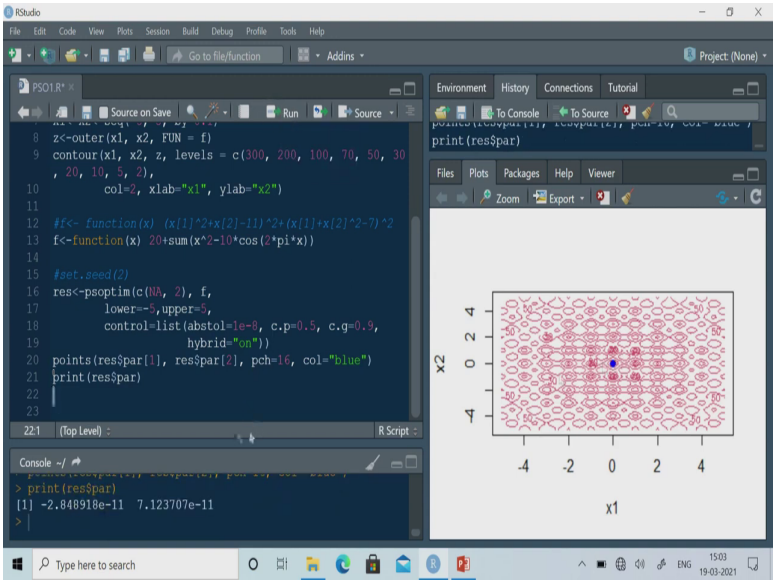


So, here, the solution is somewhere here 0, 0. This is the global minima of this particular. There are several local optimal solution. So, and global solution is somewhere here at 0, 0 and function value is 0. So, we will try to find out the solution using PSO algorithm.

(Refer Slide Time: 28:52)

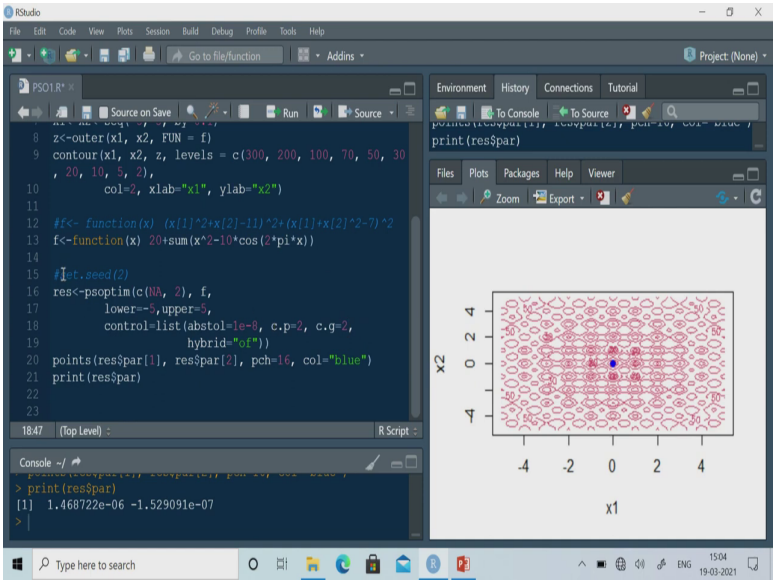


(Refer Slide Time: 29:07)



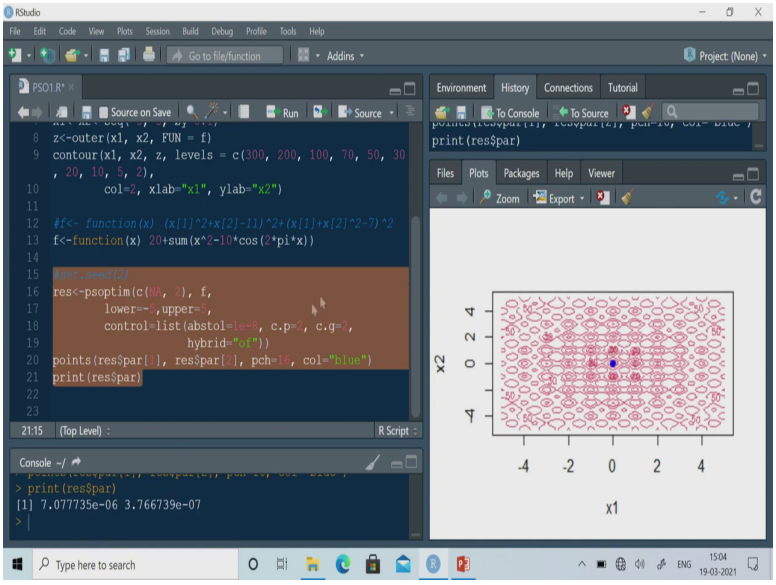
So, I am using PSO now and so, let us execute this particular line and then, apply PSO ok. So, you are getting this global optimal solution. Let us see what is this solution and solution is 0, 0; but you are getting 10 to the power minus 11 or 10 to the power minus 11 is exactly it exactly not 0, but it is near 0; basically both x1 and x2, x1 and x2 are 0 and function value is also near 0.

(Refer Slide Time: 29:30)

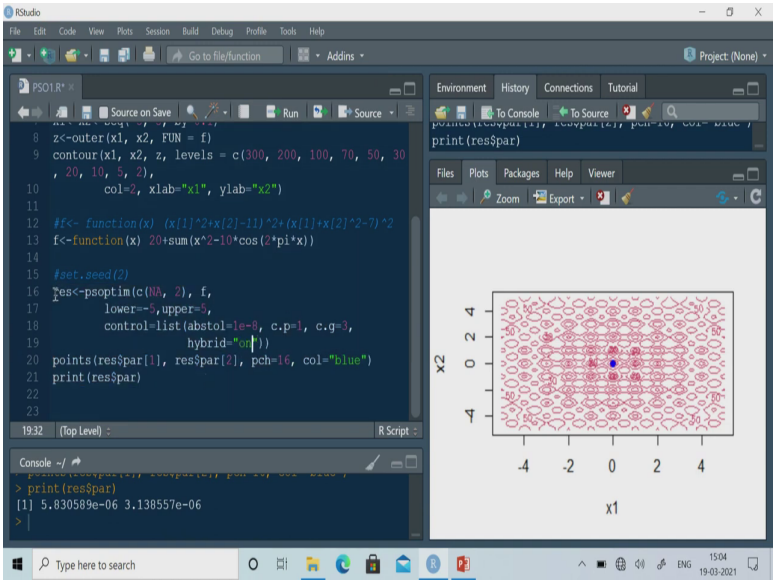


So, what I can do basically? Suppose, if I write off, then what will happen just see. So, this time, I am getting this is 10 to the power minus 6 and 10 to the power minus 7. But if you are using hybrid, so you are getting little bit better solution. So, as I said that I can sense this thing, so c.p right now I am putting 0.5 and 0.9; but I can use as per that rule. So, I can use 2 and 2. So, it should be around 4 ok. So, 2 and 2, I am using. So, in that case just see, so if you are executing this, yeah.

(Refer Slide Time: 30:12)

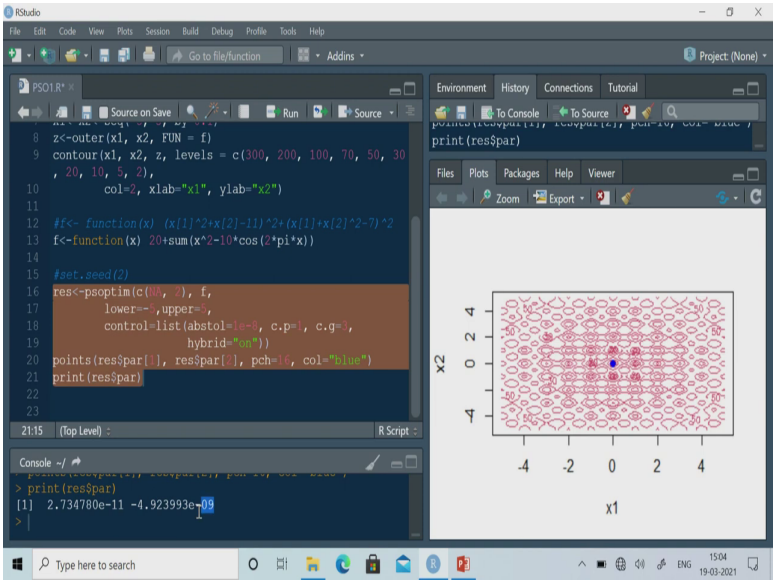


(Refer Slide Time: 30:15)



So, I am getting this value and I can use suppose 1 and 3 also I can use; just see, yeah. So, I am getting this thing and if you are putting this is hybrid on and then, you will get little bit better solution yeah.

(Refer Slide Time: 30:36)



So, it is coming 10 to the power minus 11 and 10 to the power 9. So, you can improve your solution using hybrid optimization technique. The idea is that for this particular problem, so if you are using classical method, so you will not get the global optimal solution. So, you will only get a local optimal solution.

So, therefore, idea is that initially you apply a metaheuristic optimization algorithm, just like genetic algorithm, PSO then we have also discussed ES; evolutionary strategy. So, and you can apply any metaheuristic optimization method. So, in that case, you will get a near global optimal solution that whatever solution you are getting that may not be exactly optimal solution global optimal solution, but it will be near the global optimal solution.

So, after that you apply any classical method or any local search technique to find out the exact optimal solution of the problem and this is known as hybrid optimization technique.

That means, we are using two algorithms. So, one is that metaheuristic optimization algorithm and then, we are applying the classical method. So, as you know that once you are applying classical methods, so classical methods are using gradient information. So, therefore, you will get a solution where gradient is near to 0 basically. So, it is it will never be 0, but it is very small gradient ok.

So, this is the way I can solve a non-linear nonconvex problems, problem with several local optimal solution. So, we can find out the global optimal solution of this problem. The problem you have seen, this problem has several local optimal solutions and one of them is the global optimal solution and basically if you apply classical methods, so you will never get this one.

So, you may get it, but by changing the initial solution; but once you are taking initial solution, near the global optimal solution, then only you will get the global optimal solution. However, if you are applying PSO first and then, classical method, so you will get the global optimal solution of the problem.