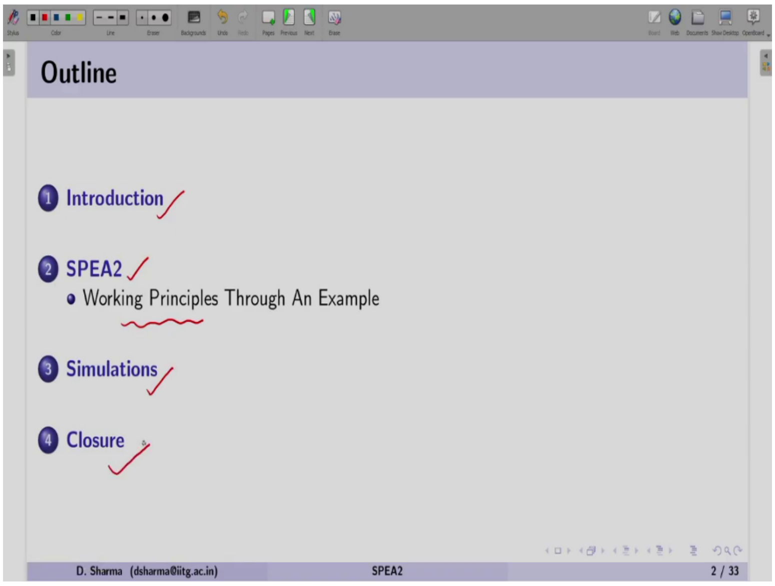
Evolutionary Computation for Single and Multi-Objective Optimization

Dr. Deepak Sharma
Department of Mechanical Engineering
Indian Institute of Technology, Guwahati

Lecture - 23
Strength Pareto Evolutionary Algorithm: SPEA2: Introduction

Welcome to the session on EC Computing Techniques for Multi Objective Optimization. In this particular session, we will be discussing another benchmark algorithm, which is known as SPEA2. So, in the previous session, we focused our discussion on NSGA2; we understood the hand calculations followed by the simulation results. In this particular session, our main focus will be on SPEA2.

(Refer Slide Time: 01:13)



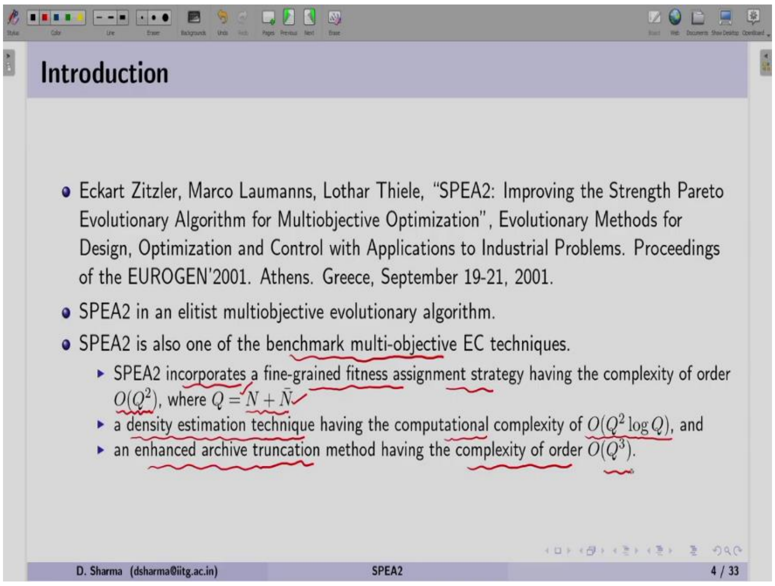
Outline

1. Introduction
2. SPEA2
   - Working Principles Through An Example
3. Simulations
4. Closure

D. Sharma (dsharma@iitg.ac.in) SPEA2 2 / 33

So, we will start with the introduction and thereafter we will discuss SPEA2. We will again understand this algorithm using the working principle through an example, some simulations will be shown of this algorithm and finally, we will conclude the session on SPEA2.

(Refer Slide Time: 01:45)



Introduction

- Eckart Zitzler, Marco Laumanns, Lothar Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization", Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN'2001. Athens. Greece, September 19-21, 2001.
- SPEA2 in an elitist multiobjective evolutionary algorithm.
- SPEA2 is also one of the benchmark multi-objective EC techniques.
  - SPEA2 incorporates a fine-grained fitness assignment strategy having the complexity of order $O(Q^2)$, where $Q = N + \bar{N}$
  - a density estimation technique having the computational complexity of $O(Q^2 \log Q)$, and
  - an enhanced archive truncation method having the complexity of order $O(Q^3)$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 4 / 33

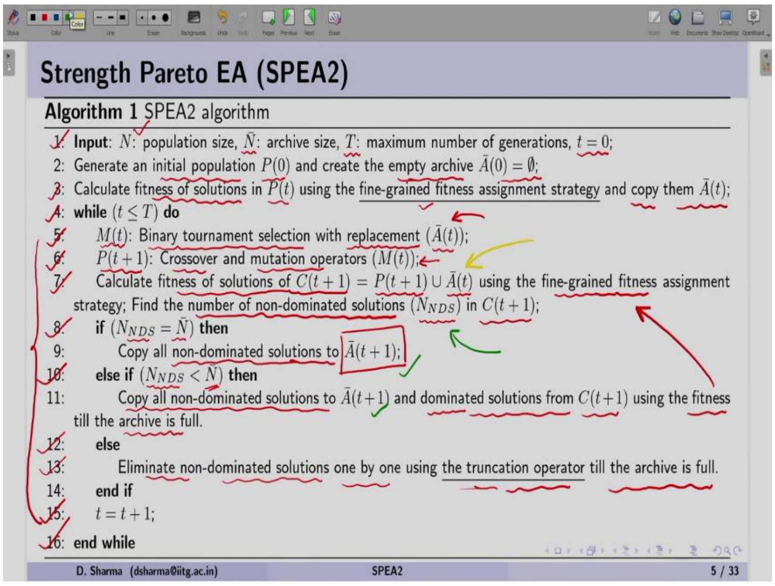
So, let us begin with the introduction now. This SPEA2, this is the second version of algorithm called SPEA, which we understand as Strength Pareto Evolutionary Algorithm. SPEA2 is proposed by Zitzler, Laumanns and Thiele and this is a conference paper which was published in 2001. So, it is an improvement, improving the earlier algorithm which we know that as SPEA.

So, that is why they have included the improved version as SPEA2. SPEA2 is also an latest multi objective evolutionary algorithm, in which we preserve good solutions while going from one generation to another generation. Since it is one of the benchmark, benchmark multi objective EC techniques as NSGA 2; it has certain features, such as SPEA2 incorporates a fine grained fitness assignment strategy. The computational complexity of that strategy is of order of Q square, where Q is N plus N bar.

So, as per our notation, N is the population size and N bar is going to be the size of archive. So, we will see later that the archive is maintained in this algorithm. There is a density estimation technique, which is used for preserving the diversity in the population.

This technique has a complexity of order Q square log Q and Q is the summation of N plus N bar. Moreover there is a enhanced archive truncation method, which is also included in SPEA2, which is having the computational complexity of order Q to the power 3.

(Refer Slide Time: 04:13)



Strength Pareto EA (SPEA2)

Algorithm 1 SPEA2 algorithm

```
1:  Input: $N$: population size, $\bar{N}$: archive size, $T$: maximum number of generations, $t = 0$;
2:  Generate an initial population $P(0)$ and create the empty archive $\bar{A}(0) = \emptyset$;
3:  Calculate fitness of solutions in $P(t)$ using the fine-grained fitness assignment strategy and copy them $\bar{A}(t)$;
4:  while $(t \leq T)$ do
5:      $M(t)$: Binary tournament selection with replacement $(\bar{A}(t))$;
6:      $P(t+1)$: Crossover and mutation operators $(M(t))$;
7:      Calculate fitness of solutions of $C(t+1) = P(t+1) \cup \bar{A}(t)$ using the fine-grained fitness assignment
        strategy; Find the number of non-dominated solutions $(N_{NDS})$ in $C(t+1)$;
8:      if $(N_{NDS} = \bar{N})$ then
9:          Copy all non-dominated solutions to $\boxed{\bar{A}(t+1)}$;
10:     else if $(N_{NDS} < \bar{N})$ then
11:         Copy all non-dominated solutions to $\bar{A}(t+1)$ and dominated solutions from $C(t+1)$ using the fitness
        till the archive is full.
12:     else
13:         Eliminate non-dominated solutions one by one using the truncation operator till the archive is full.
14:     end if
15:     $t = t + 1$;
16: end while
```

D. Sharma (dsharma@iitg.ac.in) SPEA2 5 / 33

Now, let us understand SPEA2 in our generalized framework. In the original paper, it is represented in a different way; but we can easily write the algorithmic steps in our generalized format. So, let us go one by one and understand how this SPEA2 works. The SPEA2 as we have earlier, there are certain inputs at step number 1; such as the population size, we have archive size of N bar and we have the maximum number of generation.

So, we are saying that currently t is equal to 0, means the counter for number of generation is 0 as of now. Now, at the beginning, we generate initial population; that is a random initial population and we are saving all the member into P 0. And at the same time, we also create an archive which is currently this, which is currently empty. Once it is done at the step number 3, we calculate fitness of solutions in P t using the fine grained fitness assignment strategy.

We will be understanding this strategy using an example and then we copy all the solution in P t into the archive A bar t. Now, in step 4, we are at the standard loop of our generation.

So, the first step inside this while loop is the binary tournament selection. So, as of now we know how the binary tournament selection is performed; only small change was done that, the members which we selected at a random to perform random to perform binary tournament selection operator, now those solutions will be replaced.

So, we are going to perform this binary tournament selection on the archive. So, this is important step to remember. This is going to create a mating pool called M t. In step 6, we perform crossover and mutation operator on the mating pool. Now, in this in the original paper of SPEA2, the crossover and mutation is done using SBX cross over operator and polynomial mutation.

So, here we will be restricting our discussion for solving real parameter optimization problem; because we can use these two operators. These two operators we discussed when we were discussing the real coded genetic algorithm. After crossover and mutation, the population which is created is P t plus 1; so this is the next generation population.

Now, at step number 7, we calculate fitness of a solution of C t. Now, C t is a combined population, that is we have the new population after crossover and mutation P t plus 1 and then we are also adding our archive population, which is A bar t. Now, this fitness we are going to assign using fine grained fitness assignment strategy. Once it is done, then we find the number of non dominated solution in step 7 itself.

We are representing the number of non dominated solution as N and in subscript N D S is written, so that we can see it is non-dominated solution. And we are finding it from the combined population C t plus 1. Now, when we check the non-dominated solution, we have different conditions. So, the first condition as written in step number 8 that, the number of non-dominated solution is equal to the number of archive that is the size of archive.

If it is so, then we are going to copy all non dominated solution to the archive. Now, what you can see, it is written as A bar t plus 1; meaning we are updating our archive using the current non-dominated solutions.

Thereafter in step 12, at this particular stage, suppose the number of non-dominated solution is less than the size of archive. In this particular case, since we have to fill the archive completely; first step is we are going to copy all non-dominated solution to the archive, thereafter some space will be left out.

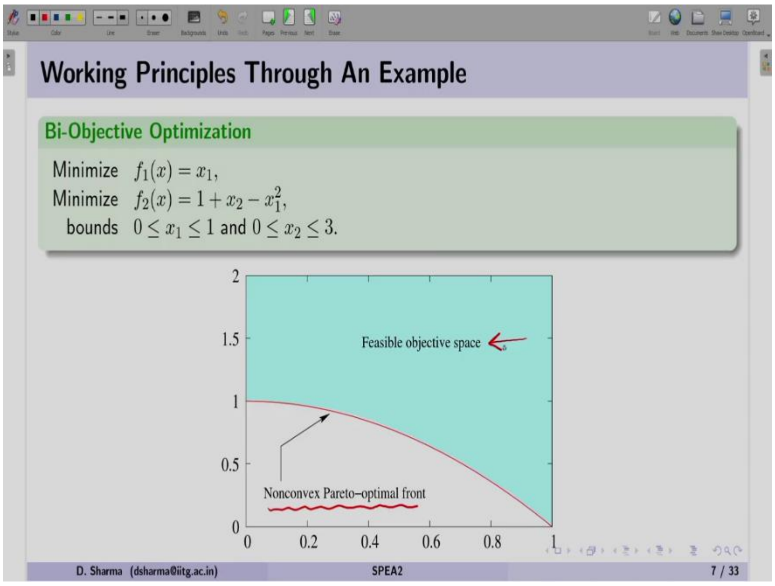
So, the remaining space we are going to copy the dominated solution from C t plus 1 using the fitness, which we have calculated using fine grained fitness assignment strategy. And we are going to copy this solution one by one till our archive is full. At step number 13, so

12 means there is a third case; in this particular case, the non-dominated solution is greater than the size of the archive. In this case, we are going to eliminate non-dominated solutions one by one using the truncation operator till our archive is full.

So, as we said, the number of non-dominated solution is more than the archive size. So, we are going to delete the solutions one by one. And once these non-dominated solution can fit in into the archive of size N bar, we will stop this truncation operator. Once it is done, in step number 15, we will increase the generation counter by one and then we will approach at the last step of the while loop. All these steps which are mentioned from step number 5 to 15, all these steps will be followed till the termination condition gets satisfied.

So, here the important point which we have to remember that, we are going to combine the new population and the archive here and then we are going to find the non-dominated solution. Now, these non-dominated solution will be used to update the archive; these small changes we are going to remember, because these changes will help us to understand SPEA2 algorithm. Now, let us move to understand this algorithm through an example.

(Refer Slide Time: 12:27)



Working Principles Through An Example

Bi-Objective Optimization

Minimize $f_1(x) = x_1$,
Minimize $f_2(x) = 1 + x_2 - x_1^2$,
bounds $0 \le x_1 \le 1$ and $0 \le x_2 \le 3$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 7 / 33

$$Minimize \quad f_1(x) = x_1$$

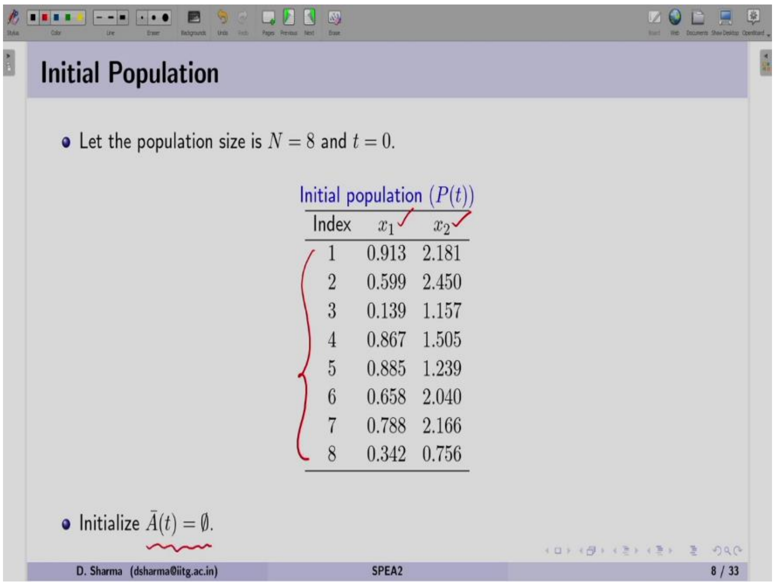
$$Minimize \quad f_2(x) = 1 + x_2 - x_1^2$$

$$bounds \ \ 0 \le x_1 \le 1 \ and \ 0 \le x_2 \le 3.$$

So, here we are taking the bi objective optimization problem, in which we want to minimize f 1 which is x 1 and we want to minimize f 2 which is given as 1 plus x 2 minus x 1 square. For the given problem, the bounds are given as x 1 is lying between 0 to 1 and x 2 is lying between 0 to 3. For the given problems since it is a mathematical problem, we can find where is the Pareto optimal front.

So, as can be seen from the figure; for the given problem, we have a non-convex Pareto optimal front which is shown in the red curve and the above this particular line, we have the feasible objective space. So, this particular figure is drawn between f 1 and f 2.

(Refer Slide Time: 13:27)



Initial Population

- Let the population size is $N = 8$ and $t = 0$.

Initial population ($P(t)$)

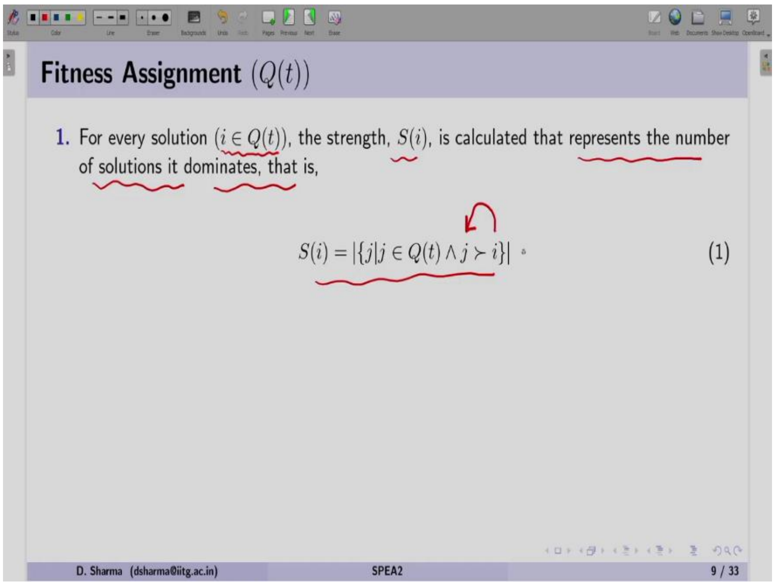
Index	x_1 ✓	x_2 ✓
1	0.913	2.181
2	0.599	2.450
3	0.139	1.157
4	0.867	1.505
5	0.885	1.239
6	0.658	2.040
7	0.788	2.166
8	0.342	0.756

- Initialize $\bar{A}(t) = \emptyset$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 8 / 33

Now, as we understand that, to start with SPEA2; we have to decide say what is the population size. So, let us assume we are, we are having N equals to 8 and we are saying t equals to 0. This initial population meaning, the value of x 1 and x 2 that are created randomly within the bound for the 8 solution as given in the table. Since it is the starting, so the archive size is kept empty currently.

(Refer Slide Time: 14:13)



Fitness Assignment $(Q(t))$

1. For every solution $(i \in Q(t))$, the strength, $S(i)$, is calculated that represents the number of solutions it dominates, that is,

$$S(i) = |\{j | j \in Q(t) \wedge j \succ i\}| \qquad (1)$$

D. Sharma (dsharma@iitg.ac.in) SPEA2 9 / 33

Thereafter we have to assign a fitness. As we can see that, Q t is a population which is input to the fitness assignment. Now, let us see how we are going to calculate the fitness. Now, in SPEA2, the fitness to an to each solution; we are calculating using the concept like strength and the raw fitness.

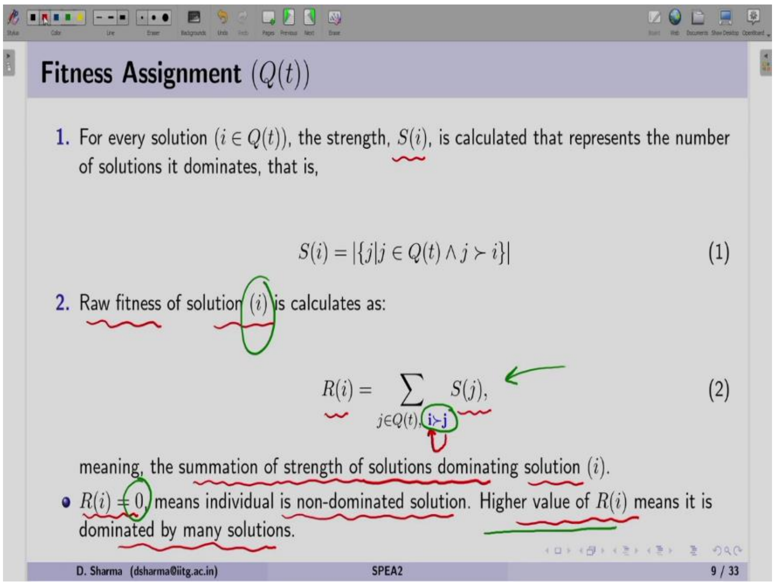
$$S(i) = |\{j | j \in Q(t) \wedge j \succ i\}|$$

We will be going through it one by one. Let us understand how we can calculate the strength of a solution here. As we can see, for every solution say i which belong to the population say Q t, the strength S i is calculated as given in equation number 1.

What it represents that, this S i represents the number of solution it dominates; meaning that a solution i is dominating the solution j. So, we will count the number of solution i is dominating in the given population. So, as we remember, we have gone through the three kinds of rank that was dominance rank, dominance count and dominance depth. So, the dominance count says that, a solution is dominating how many solutions.

So, the strength in SPEA2 is based on dominance count concept, in which we will be finding how many solutions a solution i is dominating.

(Refer Slide Time: 16:13)



Fitness Assignment $(Q(t))$

1. For every solution $(i \in Q(t))$, the strength, $S(i)$, is calculated that represents the number of solutions it dominates, that is,

$$S(i) = |\{j | j \in Q(t) \wedge j \succ i\}| \tag{1}$$

2. Raw fitness of solution (i) is calculates as:

$$R(i) = \sum_{j \in Q(t), i \succ j} S(j), \tag{2}$$

meaning, the summation of strength of solutions dominating solution (i).

- $R(i) = 0$ means individual is non-dominated solution. Higher value of $R(i)$ means it is dominated by many solutions.

D. Sharma (dsharma@iitg.ac.in) SPEA2 9 / 33

Once we calculated the strength for every solution in the population, now we are going to calculate the raw fitness. So, what is raw fitness? As can be seen in equation number 2 that, the raw fitness is the summation of strengths.

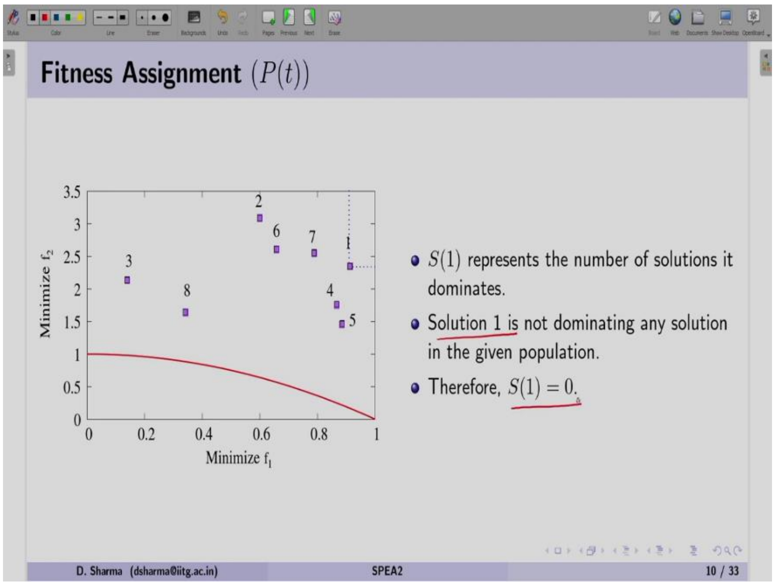
$$R(i) = \sum_{j \in Q(t), i \succ j} S(j),$$

So, the strength means, the solution j which are dominating I, so those strengths will be added. So, as we have written here, the summation of strength of solutions dominating solution i. When we are going to find the raw fitness here, when raw fitness is going to be 0; this means that, the individual or a solution is a non-dominated. And higher value of R i means, it is dominated by many solution.

So, if we look into the equation number 2 here. So, this equation suggest that, if any solution j which is dominating the solution i as can be seen here, so the strength of those solutions j will be added. Say suppose if any of the solution in the population is not dominating solution i currently; meaning that, there is the strength, there will be no strength.

So, this means the raw fitness of a solution i will become 0; but as and when there is a solution dominating i, meaning that we are going to add this strength. So, that suggest that, if we are going to get higher value of R i; means the solution is dominated by many members in the population.

(Refer Slide Time: 18:13)



Fitness Assignment $(P(t))$

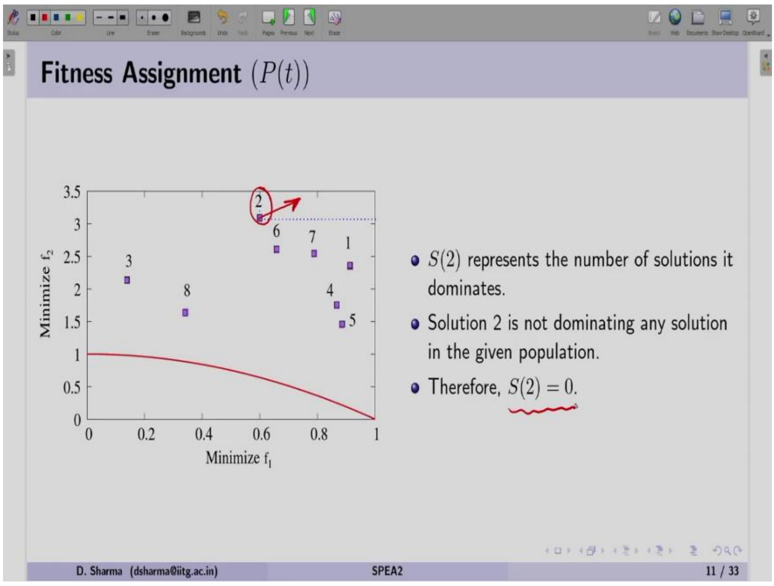
- $S(1)$ represents the number of solutions it dominates.
- Solution 1 is not dominating any solution in the given population.
- Therefore, $S(1) = 0$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 10 / 33

So, let us understand how we can apply the strength as well as the raw fitness to raw fitness for each solution. We started with one example. So, we are going to assign the strength in the raw fitness using the figure here. As can be seen here that, our problem was minimizing f 1, minimizing f 2; so all eight solutions are given. Let us take solution number 1. So, as the strength says, the number of solution it dominate; means how many solution are dominated by solution 1.

So, as per our understanding, since it is a minimization problem and also it is two objective problem; so we know that, from the solution 1. So, with respect to solution 1, if any solution which is coming into the first quadrant; so those solutions will be dominated by 1. In the current situation, there is no solution which is dominated by solution 1 in the current population; so the strength of this solution is 0.

(Refer Slide Time: 19:53)



Fitness Assignment $(P(t))$

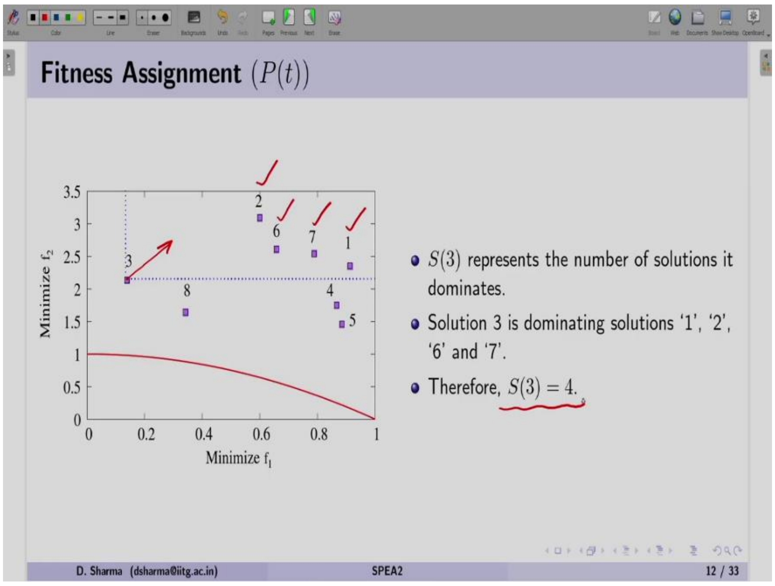
- $S(2)$ represents the number of solutions it dominates.
- Solution 2 is not dominating any solution in the given population.
- Therefore, $S(2) = 0$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 11 / 33

.

Let us take another example now. Here we are taking solution number 2. In this particular solution, if we see with respect to this solution, if we look into the first quadrant; we can see there is no solution which is dominated by 2. Therefore, the strength of solution 2 is also 0. Since solution 1 and 2 are having strength 0.

(Refer Slide Time: 20:21)



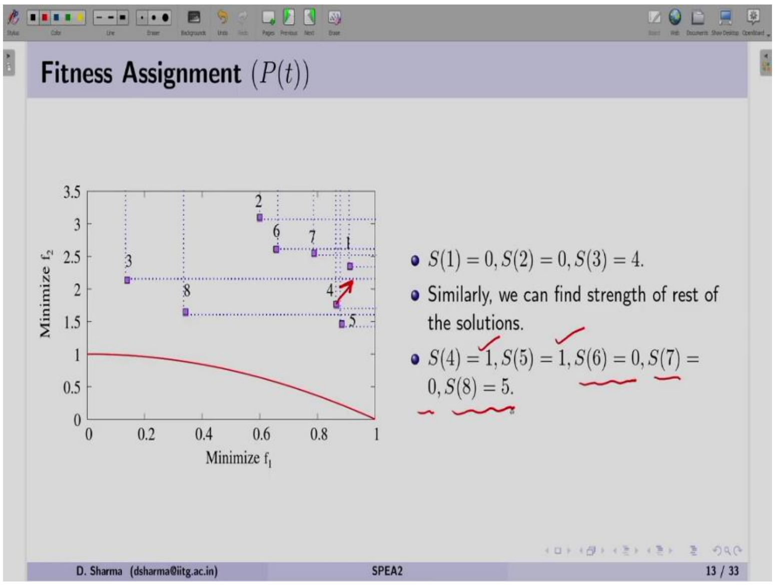
Fitness Assignment $(P(t))$

- $S(3)$ represents the number of solutions it dominates.
- Solution 3 is dominating solutions '1', '2', '6' and '7'.
- Therefore, $S(3) = 4$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 12 / 33

Let us take the another example of say, solution number 3. Now, here we know that, with respect to solution number 3; if I look into the first quadrant, there are four solutions that are 1, 2, 6 and 7. These are the solutions which are dominated by solution number 3. So, therefore, strength of solution 3 is 4.

(Refer Slide Time: 20:57)



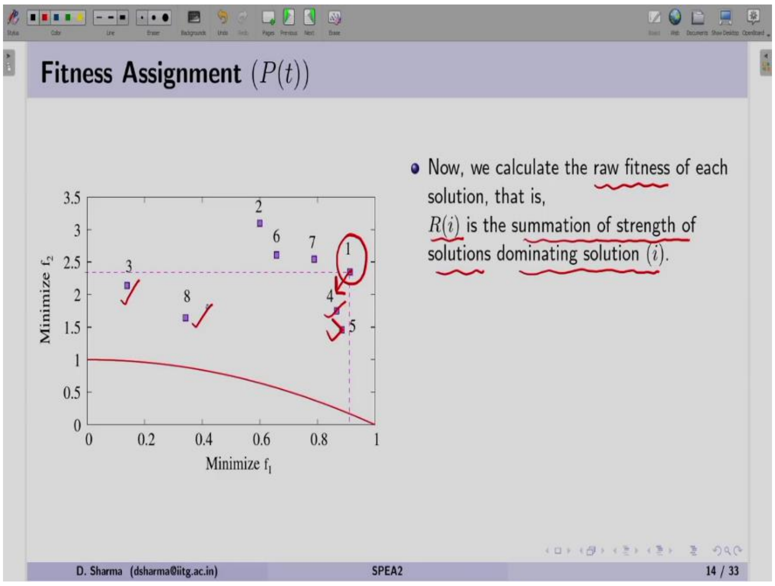
Fitness Assignment $(P(t))$

- $S(1) = 0, S(2) = 0, S(3) = 4.$
- Similarly, we can find strength of rest of the solutions.
- $S(4) = 1, S(5) = 1, S(6) = 0, S(7) = 0, S(8) = 5.$

D. Sharma (dsharma@iitg.ac.in) SPEA2 13 / 33

Now, by taking the three examples, now we can find strength of all the solution. So, in this case, if you look at solution 4, looking into the first quadrant. So, it is only 1 solution which is dominated by 4, similarly by 5, 6 is not dominating anyone; 7 is not dominating any solution in the current population and then finally, we have the solution 8, which is dominating multiple solution that are 5. So, in this particular case, we have calculated the strength of all the solutions.

(Refer Slide Time: 21:43)



Fitness Assignment $(P(t))$

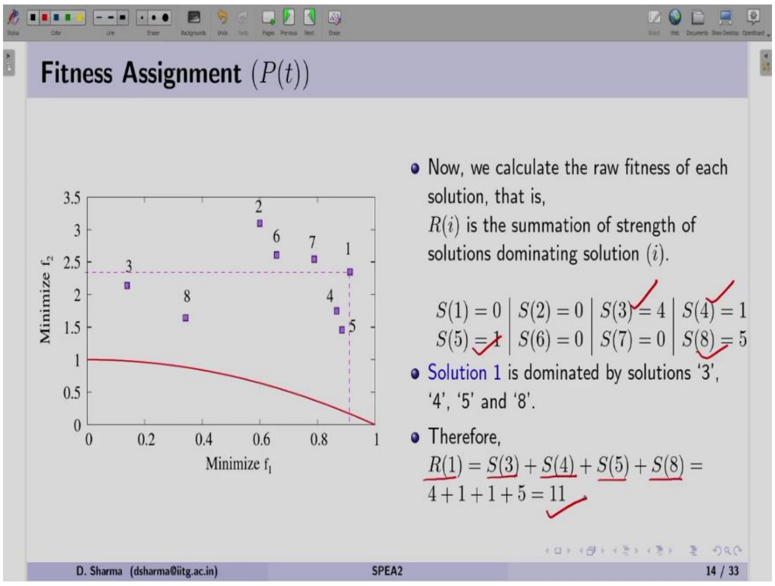
- Now, we calculate the raw fitness of each solution, that is, $R(i)$ is the summation of strength of solutions dominating solution (i).

D. Sharma (dsharma@iitg.ac.in) SPEA2 14 / 33

Now, now we have to calculate the raw fitness of each solution. As per equation number 2, R i is. So, this is R i the raw fitness is the summation of strength of solutions dominating solution i. So, let us take solution number 1. Since it is a two objective problem, we know that if we take one as a reference and look into the third quadrant with respect to the solution number 1; we can find what are the solutions that are dominating 1.

In the current scenario, we can see solution number 3, 4, 5 and 8; these are the solutions which are dominating solution 1.

(Refer Slide Time: 22:43)



Fitness Assignment $(P(t))$

- Now, we calculate the raw fitness of each solution, that is, $R(i)$ is the summation of strength of solutions dominating solution (i).

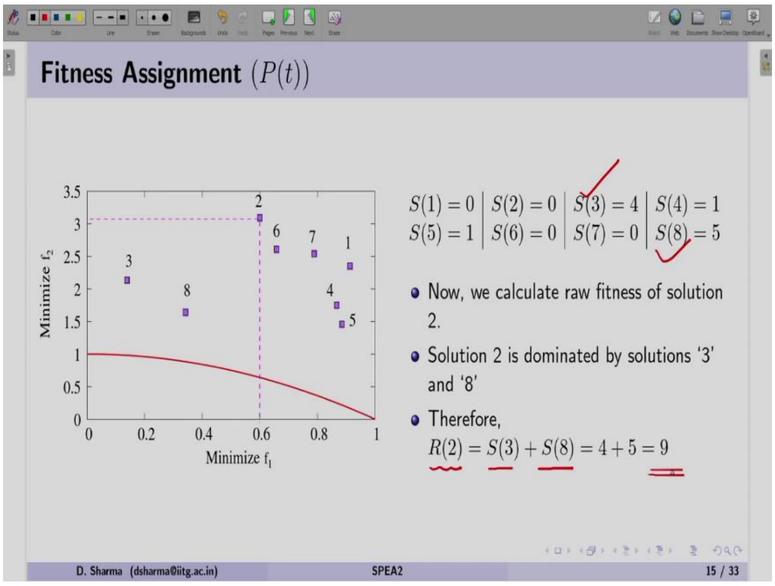
$$\begin{array}{c|c|c|c} S(1)=0 & S(2)=0 & S(3)=4 & S(4)=1 \\ S(5)=1 & S(6)=0 & S(7)=0 & S(8)=5 \end{array}$$

- Solution 1 is dominated by solutions ‘3’, ‘4’, ‘5’ and ‘8’.
- Therefore, $R(1)=S(3)+S(4)+S(5)+S(8)=4+1+1+5=11$

D. Sharma (dsharma@iitg.ac.in) SPEA2 14 / 33

So, we have written the strength of all the solution as of now. Now, since solution 1 is dominated by 3, 4, 5 and 8. So, the raw fitness of 1 will become strength of 3 plus strength of 4 plus strength of 5 plus strength of 8. If we take the value of 3, 4, 5 and 8; so the raw fitness summation is going to be 11. So, the, so this way, we can calculate its raw fitness.

(Refer Slide Time: 23:25)



Fitness Assignment $(P(t))$

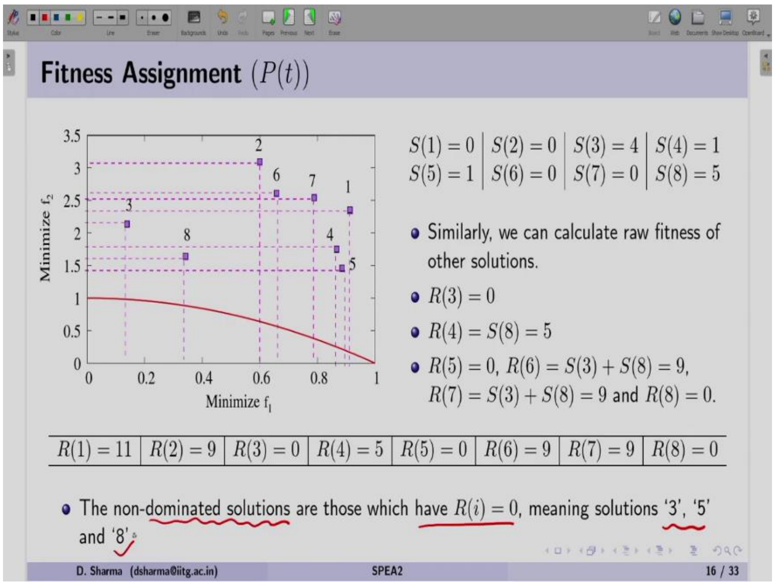
$S(1) = 0$	$S(2) = 0$	$S(3) = 4$	$S(4) = 1$
$S(5) = 1$	$S(6) = 0$	$S(7) = 0$	$S(8) = 5$

- Now, we calculate raw fitness of solution 2.
- Solution 2 is dominated by solutions '3' and '8'
- Therefore, $R(2) = S(3) + S(8) = 4 + 5 = 9$

D. Sharma (dsharma@iitg.ac.in) SPEA2 15 / 33

Let us take another example, here we are taking solution number 2 and since it is minimization two objective problem, so we are looking into the third quadrant. In this third quadrant we can see that, the solution number 2 is dominated by solution 3 and 8; meaning that, the raw fitness of solution 2 is equals to the strength of 3, 8. So, looking at the table on the top, we have the strength. So, currently the raw fitness of solution 2 is 9.

(Refer Slide Time: 24:13)



Fitness Assignment $(P(t))$

$S(1) = 0$	$S(2) = 0$	$S(3) = 4$	$S(4) = 1$
$S(5) = 1$	$S(6) = 0$	$S(7) = 0$	$S(8) = 5$

- Similarly, we can calculate raw fitness of other solutions.
- $R(3) = 0$
- $R(4) = S(8) = 5$
- $R(5) = 0$, $R(6) = S(3) + S(8) = 9$, $R(7) = S(3) + S(8) = 9$ and $R(8) = 0$.

$R(1) = 11$	$R(2) = 9$	$R(3) = 0$	$R(4) = 5$	$R(5) = 0$	$R(6) = 9$	$R(7) = 9$	$R(8) = 0$

- The non-dominated solutions are those which have $R(i) = 0$, meaning solutions '3', '5' and '8'.

D. Sharma (dsharma@iitg.ac.in) SPEA2 16 / 33

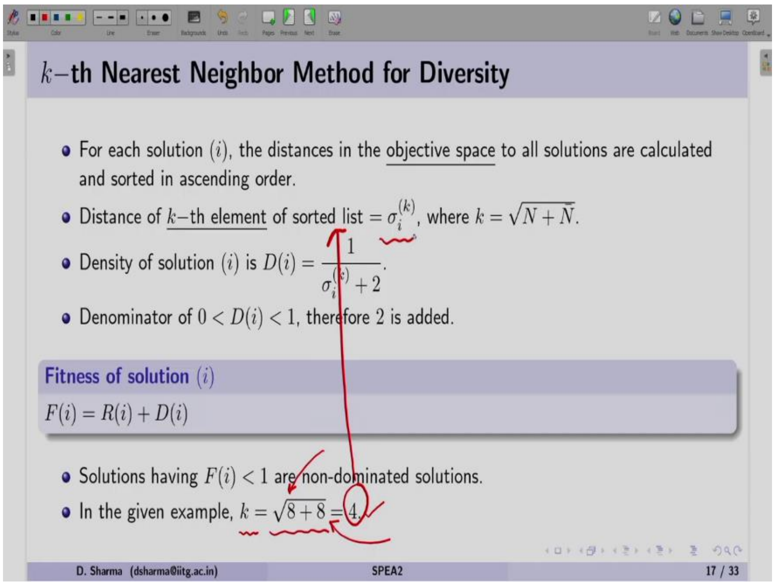
In a similar way, we can calculate raw fitness of other solutions; that that are say for example, solution number 3. Now, as we know, if we look at the third quadrant with respect to solution number 3.

So, there is no solution which is dominating 3; say they similarly you can look into solution number 8 as well as 5. So, since these solutions are not dominated, we will see the strength of solution 3 is 0. Now, similarly for 8 and 5 we will see afterwards. If we take solution number 4 and if we look into the third quadrant, only solution 8 is dominating; so therefore, the raw fitness is fine 5.

In a similar way as we have discussed, raw fitness of a 5 is 0; because it is not dominated by anyone, raw fitness of a 6 is S 3 plus S 8, because 3 and 8 are dominating 5. So, the fitness is 9. For solution 7, it is dominated by 3 and 8 again, the fitness is 9. And solution 8 as we discussed earlier, since it is not dominated by anyone; so the raw fitness is 0.

If we go if we tabulate all the raw fitness of these solutions, we can see here from R 1 to R 8. As per our earlier discussion, any solution which is having raw fitness equals to 0; so we know that, those solutions are become non non-dominated solution. Therefore, solution 3, 4 and 8 are the non-dominated solutions in the given population.

(Refer Slide Time: 26:19)



$k-$th Nearest Neighbor Method for Diversity

- For each solution (i), the distances in the objective space to all solutions are calculated and sorted in ascending order.
- Distance of $k-$th element of sorted list $= \sigma_i^{(k)}$, where $k = \sqrt{N + \bar{N}}$.
- Density of solution (i) is $D(i) = \dfrac{1}{\sigma_i^{(k)} + 2}$.
- Denominator of $0 < D(i) < 1$, therefore 2 is added.

Fitness of solution (i)

$F(i) = R(i) + D(i)$

- Solutions having $F(i) < 1$ are non-dominated solutions.
- In the given example, $k = \sqrt{8+8} = \boxed{4}$.

D. Sharma (dsharma@iitg.ac.in) SPEA2 17 / 33

$$\sigma_i^{(k)}, \qquad \text{where } k = \sqrt{N + \bar{N}}$$

$$D(i) = \frac{1}{\sigma_i^{(k)} + 2}$$

Now, as of now we have calculated the fitness of each solution; we started with their strength to calculate a strength for every solution and thereafter we use those strength to calculate the raw fitness of each solution. Now, since this fitness involve concept of a dominance concept, so this is going to help us in the convergence.

But as we know we have two objectives to solve a multi objective optimization problem; then we have to understand how SPEA2 is keeping the diversity among the solution. SPEA2 uses k kth nearest neighbour method for the diversity. How it is calculated? For example, a solution i is given.

So, we are going to find the Euclidean distance in the objective space with respect to all the solutions. And these distances we are going to sort in an ascending order. Once it is done, we are trying to find out the kth element in the sorted list and that is represented by say sigma i k. So, sigma i k is going to be the distance of the kth element. And what is kth element? As a thumb rule it is given as k equals to under root of N plus N bar.

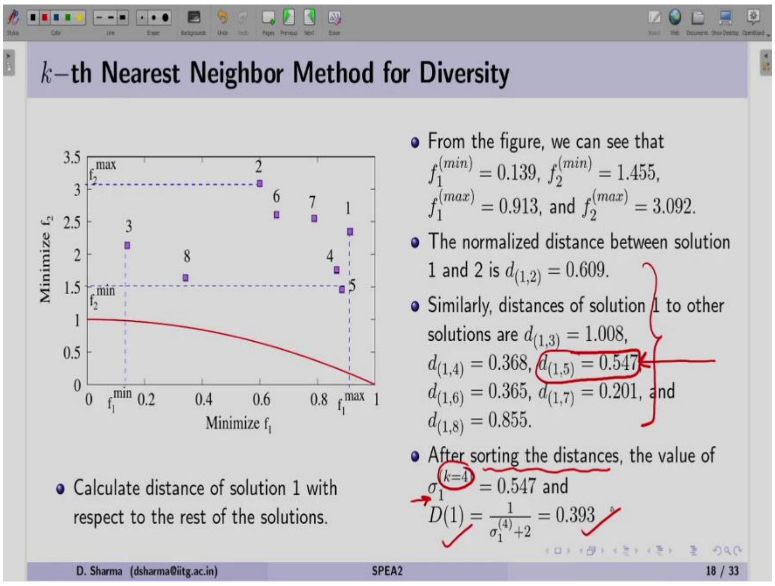
So, we remember that, N is the population size and N bar is the archive size. Now, the density of solution i which is represent as, which is represented as a capital D i; we are calculating as 1 divided by the distance of the kth element in the sorted list plus 2. Why we are keeping 2? Because we want our diversity should lie between 0 to 1 and therefore, we added 2 in the denominator. So, the overall fitness of a solution now will become the raw fitness plus the diversity D i.

Now, here the important point we can understand that, if any solution is having fitness less than 1; then the all those solutions become the non-dominated solution. How do we know that? So, as per our discussion, if the raw fitness is going to be zero; then that particular solution is non-dominated. And now we have discussed the diversity D i and we have made the D i in such a way that, it always lies between 0 to 1. So, meaning that the diversity factor will be less than 1.

So, since the raw fitness was 0 and the diversity will always be lying between 0 to 1. So, the non-dominated solutions are going to have the composite fitness or a combined fitness less than 1. This particular observation will be useful when we will be performing the other operators with SPEA2. Now, in the given example, we have k which is 8 plus 8. Why? Because the population size we have kept 8 similarly, just for the simplicity we have taken the archive size of 8 as well. So, we know 8 plus 8, 16 and under root of 16 becomes 4.

So, this means that, we will be looking for the 4th solution in the sorted list, so that we can calculate what is the sigma i k value.

(Refer Slide Time: 30:51)



$k-$th Nearest Neighbor Method for Diversity

- Calculate distance of solution 1 with respect to the rest of the solutions.

- From the figure, we can see that $f_1^{(min)} = 0.139$, $f_2^{(min)} = 1.455$, $f_1^{(max)} = 0.913$, and $f_2^{(max)} = 3.092$.
- The normalized distance between solution 1 and 2 is $d_{(1,2)} = 0.609$.
- Similarly, distances of solution 1 to other solutions are $d_{(1,3)} = 1.008$, $d_{(1,4)} = 0.368$, $\boxed{d_{(1,5)} = 0.547}$ $d_{(1,6)} = 0.365$, $d_{(1,7)} = 0.201$, and $d_{(1,8)} = 0.855$.
- After sorting the distances, the value of $\sigma_1^{(k=4)} = 0.547$ and $D(1) = \frac{1}{\sigma_1^{(4)}+2} = 0.393$

D. Sharma (dsharma@iitg.ac.in) SPEA2 18 / 33

Let us perform some hand calculation now. Here first we will take solution number 1 and we will try to find out what is the kth nearest neighbour distance. In order to do that, we can see in the figure; we have to first normalize f 1 as well as f 2. So, we have to find out what is f 1 min, f 2 max; similarly f 1 min f 1 max, similarly f 2 min f 2 max. Now, this is the current population which is given in the figure.

So, as a for the hand calculation, we have taken the extreme in f 1 which is corresponding to the solution 1. So, f 1 max we can calculate using solution one; similarly f 1 min we can calculate with respect to the solution 3. Since the objective function values are known; so we can calculate as f 1 min and f 2 min f 1 max and f 2 max as given on the top. Thereafter, let us calculate the normalized distance.

So, as our discussion says that, this minimum and a maximum in both the objective will help us to calculate the normalized distance. This is important, it is because our objective function in solving practical optimization problem; they can vary in different ranges, for example, f 1 can vary in a range of say thousands and f 2 can vary say in the range of 0.1 and 0.2.

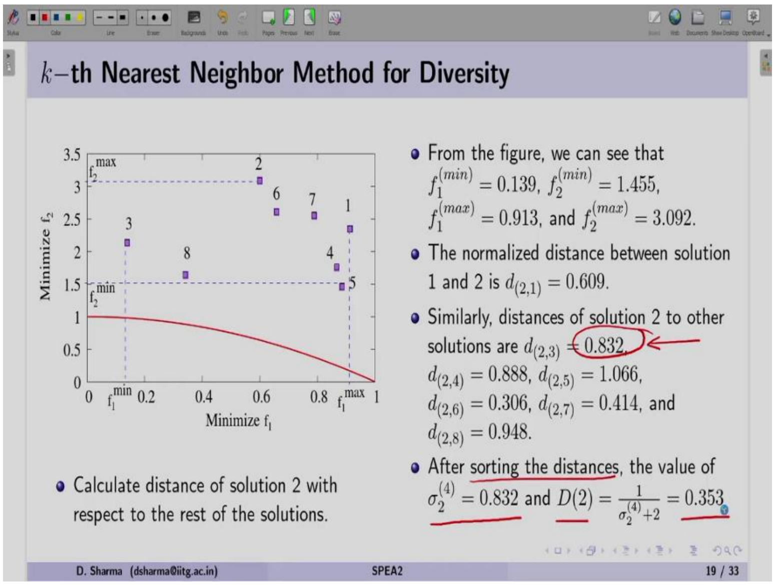
So, since these two objectives are varying in a different ranges, when we are going to normalize; so the value will always be lying between 0 to 1 and that is the whole motive of normalization here. So, let us see here that the normalization distance between solution 1 and solution 2.

So, basically we are finding the Euclidean distance between solution 1 and A2 and that distance is normalized using f 1 and f 2 min and a max. So, since it is a straight forward calculation; we are showing that the distance between 1 and A2 is 0.609. Similarly, we can find solution; the we can find the distance between solution 1 and a 3; 1, 4; 1, 5; 1, 6; 1, 7 and 1, 8.

Now, all these distances are given here. Now, as our understanding says that, we have to first sort these distances. So, the seven distances are given to us with respect to the solution number 1. When we are going to sort, we have to find which is the fourth solution, which is the element which is coming at the fourth place, since k is equals to 4. So, we are going to find sigma 1 k 4.

So, sigma 1 represent, this is the solution one and k 4 is the fourth which element. So, k equals to 4 means, it is the fourth element in the sorted list. So, when you are going to sort, you will see that distance 1, 5 will come at the fourth place in the sorted list of distance. We will include this value in D 1 and we can get as 0.393.

(Refer Slide Time: 34:55)



$k-$th Nearest Neighbor Method for Diversity

- Calculate distance of solution 2 with respect to the rest of the solutions.

- From the figure, we can see that $f_1^{(min)} = 0.139$, $f_2^{(min)} = 1.455$, $f_1^{(max)} = 0.913$, and $f_2^{(max)} = 3.092$.
- The normalized distance between solution 1 and 2 is $d_{(2,1)} = 0.609$.
- Similarly, distances of solution 2 to other solutions are $d_{(2,3)} = \boxed{0.832}$, $d_{(2,4)} = 0.888$, $d_{(2,5)} = 1.066$, $d_{(2,6)} = 0.306$, $d_{(2,7)} = 0.414$, and $d_{(2,8)} = 0.948$.
- After sorting the distances, the value of $\sigma_2^{(4)} = 0.832$ and $D(2) = \frac{1}{\sigma_2^{(4)}+2} = 0.353$

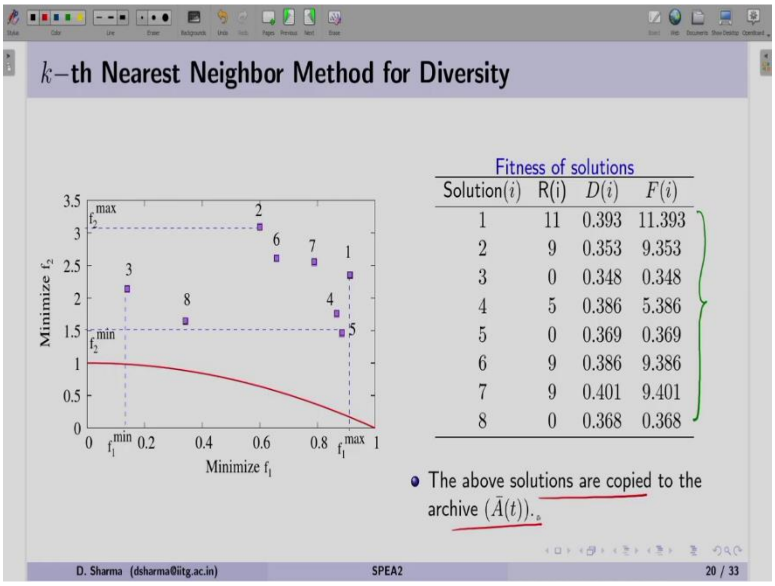
D. Sharma (dsharma@iitg.ac.in) SPEA2 19 / 33

Let us take another example of say solution number 2 and in this case we our f 1 min and a max, f 2 min and a max both will remain the same. Now, we are going to calculate the normalized distance.

So, this normalized distance from 2 to 1 that remains the same as we have found earlier. But we have to find the distance of solution 2 to 3, 2 to 4 and the other solution, as we can see all the distances are given here. Our practice says that, we are going to sort these solutions based on the distances and we will be looking which is the solution coming at the fourth place.

So, solution number 3 is going to come at the fourth place in the sorted list of distance and therefore, sigma 2, 4 will become 0.832. And when we are putting into our formula, we will get the diversity D 2 as 0.353.

(Refer Slide Time: 36:15)



k–th Nearest Neighbor Method for Diversity

Fitness of solutions

Solution(i)	R(i)	$D(i)$	$F(i)$
1	11	0.393	11.393
2	9	0.353	9.353
3	0	0.348	0.348
4	5	0.386	5.386
5	0	0.369	0.369
6	9	0.386	9.386
7	9	0.401	9.401
8	0	0.368	0.368

- The above solutions are copied to the archive ($\bar{A}(t)$).

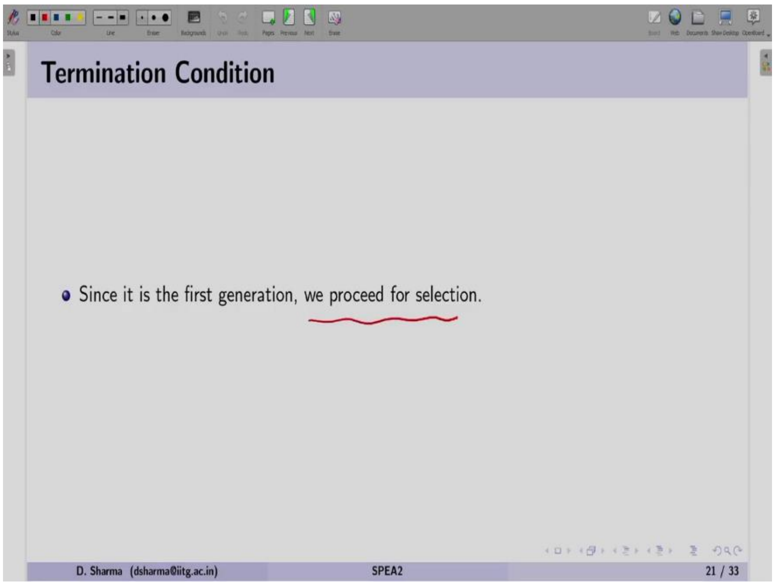
D. Sharma (dsharma@iitg.ac.in) SPEA2 20 / 33

Now, if we are going to follow the same procedure for rest of the solution; we can find the diversity measure, which is D i. And since we have already calculated the raw fitness and the diversity D i, then we can find the fitness.

So, in this here, we can see that all the solutions in a given table on the right hand side, we have the raw fitness given; the distance as we have discussed earlier all the distances are given. And the fitness is nothing, but the summation of R i plus D i and that is why we get these distances. Now, as per our earlier discussion we remember that, the solution 3 was a non-dominated solution.

And if we look at the fitness of solution number 3, this is smaller than 1. So, we can identify the solution is non-dominated. Similarly, if we look at solution number 5; so the fitness is less than 1 and similarly for solution 8. Now, here since we started with the random initial population and we calculated the fitness of all the solutions. So, we are going to copy all these solution into the archive population as per the algorithm.

(Refer Slide Time: 38:01)



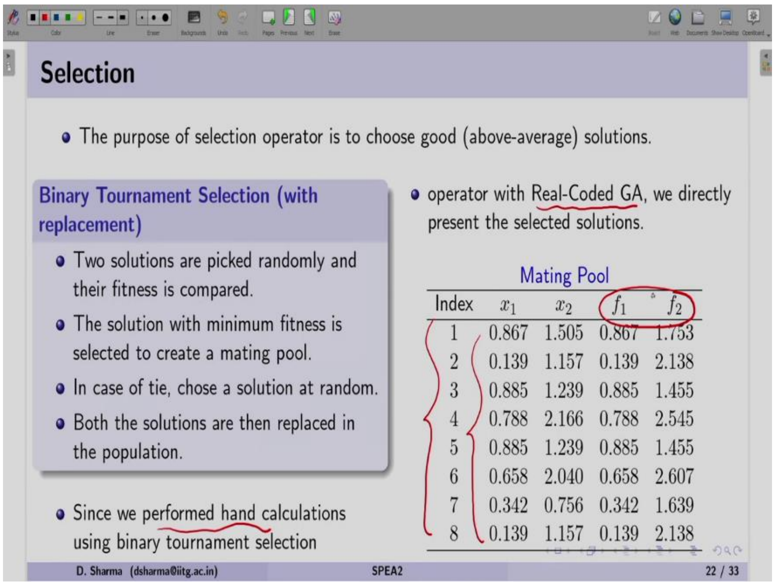
Termination Condition

- Since it is the first generation, we proceed for selection.

D. Sharma (dsharma@iitg.ac.in) SPEA2 21 / 33

So, by this step, we have already calculated the fitness of the population. So, currently it is the initial population and thereafter we have copied into the archive. Now, we are at the decision box, basically the while loop. Since it is the first generation, we will proceed further to apply other operators. So, as mentioned here, we will proceed for the selection.

(Refer Slide Time: 38:31)



Selection

- The purpose of selection operator is to choose good (above-average) solutions.

Binary Tournament Selection (with replacement)

- Two solutions are picked randomly and their fitness is compared.
- The solution with minimum fitness is selected to create a mating pool.
- In case of tie, chose a solution at random.
- Both the solutions are then replaced in the population.

- Since we performed hand calculations using binary tournament selection
- operator with Real-Coded GA, we directly present the selected solutions.

Mating Pool

Index	x_1	x_2	f_1	f_2
1	0.867	1.505	0.867	1.753
2	0.139	1.157	0.139	2.138
3	0.885	1.239	0.885	1.455
4	0.788	2.166	0.788	2.545
5	0.885	1.239	0.885	1.455
6	0.658	2.040	0.658	2.607
7	0.342	0.756	0.342	1.639
8	0.139	1.157	0.139	2.138

D. Sharma (dsharma@iitg.ac.in) SPEA2 22 / 33

As a recap we know, the purpose of the selection operator is to choose good and above average solution. In SPEA2 the selection is perform using binary tournament selection operator and that is with replacement; so we will understand what is this small change with the binary tournament selection, which we have gone through earlier.

So, we know that for performing binary tournament selection operator, we pick two solutions randomly and we compare their fitness. As per our earlier discussion fitness, we have already calculated for all the solutions. Thereafter, the solution having minimum fitness is selected to create a mating pool.

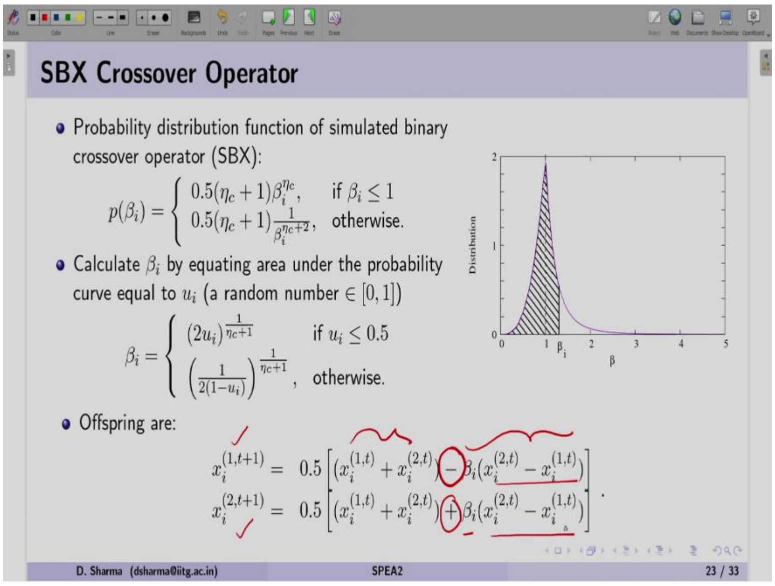
Now, in case if there is a tie. So, basically the fitness of both the solution is same; then we are going to choose a solution at random. Finally, the solutions which we have selected for performing the tournament selection; they are now replaced to the population. When we are saying replaced, so that is called with replacement. The binary tournament selection operator which we have discussed earlier with real coded GA and with NSGA 2, in which we pick two solutions and once these two solutions are picked, they are not considered in the same tournament.

For example, if we take a population size of 8 and if we pick two solutions randomly; in that case, these two solutions will be removed and we are left with six solutions. Thereafter we pick another two random solutions and the size will reduce from 6 to 4. This strategy we call it as binary tournament selection without replacement. But in SPEA2, the strategy adapted is binary tournament selection with replacement.

Meaning that, when we have eight solutions, when we are going to take two solutions for tournament; we select we will copy one solution. And these two solutions were will again replaced into the population; meaning that the population size will remain the same. Now, since we have performed the binary tournament selection earlier; so we are going to show you the solutions which are selected in the mating pool.

And since we did it with the real coded GA; so we are showing the result. So, these are the index and the solutions which are selected after binary tournament selection operator are given here. Only for the representation purpose, we are showing f 1 and f 2 values of these two these solutions which are selected.

(Refer Slide Time: 41:57)



SBX Crossover Operator

- Probability distribution function of simulated binary crossover operator (SBX):

$$p(\beta_i) = \begin{cases} 0.5(\eta_c+1)\beta_i^{\eta_c}, & \text{if } \beta_i \le 1 \\ 0.5(\eta_c+1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise.} \end{cases}$$

- Calculate β_i by equating area under the probability curve equal to u_i (a random number $\in [0,1]$)

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & \text{if } u_i \le 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise.} \end{cases}$$

- Offspring are:

$$x_i^{(1,t+1)} = 0.5\left[(x_i^{(1,t)} + x_i^{(2,t)}) - \beta_i(x_i^{(2,t)} - x_i^{(1,t)})\right]$$
$$x_i^{(2,t+1)} = 0.5\left[(x_i^{(1,t)} + x_i^{(2,t)}) + \beta_i(x_i^{(2,t)} - x_i^{(1,t)})\right].$$

D. Sharma (dsharma@iitg.ac.in) SPEA2 23 / 33

$$p(\beta_i\ ) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & if\ \beta_i \le\ 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & otherwise \end{cases}$$

$$\beta_i\ = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & if\ u_i\ \le\ \ 0.5 \\ \left(\frac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}} & otherwise \end{cases}$$

$$x_i^{(1,t+1)}\ \ =\ \ 0.5\left[\left(x_i^{(1,t)} + x_i^{(2,t)}\right) - \beta_i\left(\ x_i^{(2,t)} - x_i^{(1,t)}\ \right)\right]$$

$$x_i^{(2,t+1)}\ \ =\ \ 0.5\left[\left(x_i^{(1,t)} + x_i^{(2,t)}\right) + \beta_i\left(\ x_i^{(2,t)} - x_i^{(1,t)}\ \right)\right]$$

Now, thereafter once the mating pool is created, we have to create new solutions. These new solutions are created using crossover and mutation. SPEA2 in its original paper uses SBX operator for crossover, and polynomial mutation for mutation operator for mutating the solutions.

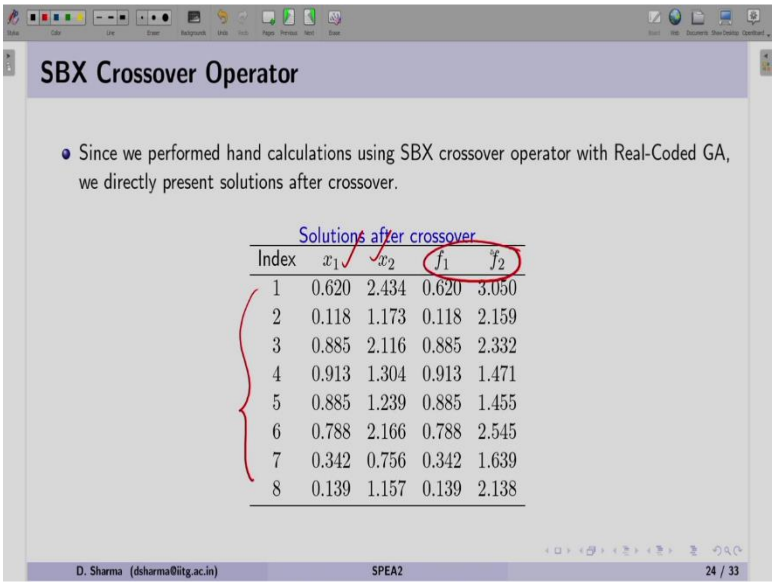
So, as a recap, let us see what was SBX crossover operator? So, the probability distribution of SBX cross over operator is given here; as we can see there are two parameters. So, beta i is the x axis for us and eta c is the user defined parameter. This probability distribution can be seen on the right hand side of the figure.

It says that, we have two nature of the curve along the beta i; because here you can see when beta i is smaller than and equals to 1, we have one equation and otherwise we are going to use another equation. How this probability distribution is going to work? We will be first generating a random number u i and that random number we will be including with the area under the curve.

So, when we are going to integrate this probability distribution and we equate this integral to the random number u i, we can calculate the value of a beta i. So, again the beta i value depends on the random number u i; it says that if it is smaller than and equals to 0.5, we are going to use this formula, otherwise we are going to use the another formula. Since beta i we have calculated.

So, the new solutions, so x i 1, t plus 1; x i 2, t plus 1 these solutions we are we can calculate as given in the formula. So, the difference which you can see that at one part, we are subtracting minus beta i and the and the factor in the bracket; in another part, we are adding beta i with the factor given in the bracket.

(Refer Slide Time: 44:31)



SBX Crossover Operator

- Since we performed hand calculations using SBX crossover operator with Real-Coded GA, we directly present solutions after crossover.

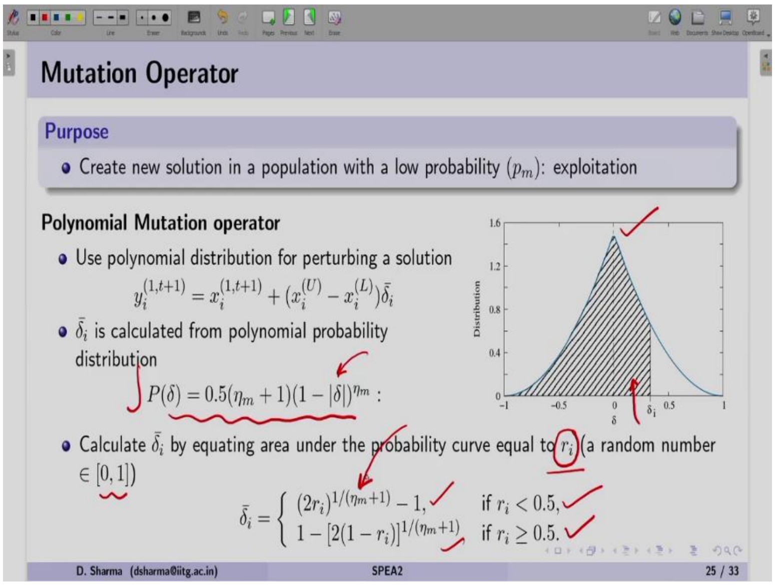
Solutions after crossover

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.118	1.173	0.118	2.159
3	0.885	2.116	0.885	2.332
4	0.913	1.304	0.913	1.471
5	0.885	1.239	0.885	1.455
6	0.788	2.166	0.788	2.545
7	0.342	0.756	0.342	1.639
8	0.139	1.157	0.139	2.138

D. Sharma (dsharma@iitg.ac.in) SPEA2 24 / 33

So, here since this SBX operator, which we have performed earlier with real coded genetic algorithm; we will be presenting the solutions that are created after crossover operator. So, here we can see that, the solution are given in the table; there x 1 and x 2 values are given. Now, here f 1 and f 2 values are given for the representation only.

(Refer Slide Time: 45:05)



Mutation Operator

Purpose

- Create new solution in a population with a low probability (p_m): exploitation

Polynomial Mutation operator

- Use polynomial distribution for perturbing a solution
  $$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(U)} - x_i^{(L)})\bar{\delta}_i$$
- $\bar{\delta}_i$ is calculated from polynomial probability distribution
  $$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}:$$
- Calculate $\bar{\delta}_i$ by equating area under the probability curve equal to r_i (a random number $\in [0,1]$)
  $$\bar{\delta}_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{if } r_i < 0.5, \\ 1 - [2(1-r_i)]^{1/(\eta_m+1)}, & \text{if } r_i \geq 0.5. \end{cases}$$

D. Sharma (dsharma@iitg.ac.in) SPEA2 25 / 33

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(U)} - x_i^{(L)})\bar{\delta}_\iota$$

$$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}:$$

$$\bar{\delta}_\iota = \begin{cases} (2r_i)^{\frac{1}{\eta_m+1}} - 1, & if\ r_i < 0.5 \\ 1 - [2(1-r_i)]^{\frac{1}{\eta_m+1}}, & if\ r_i \geq 0.5 \end{cases}$$

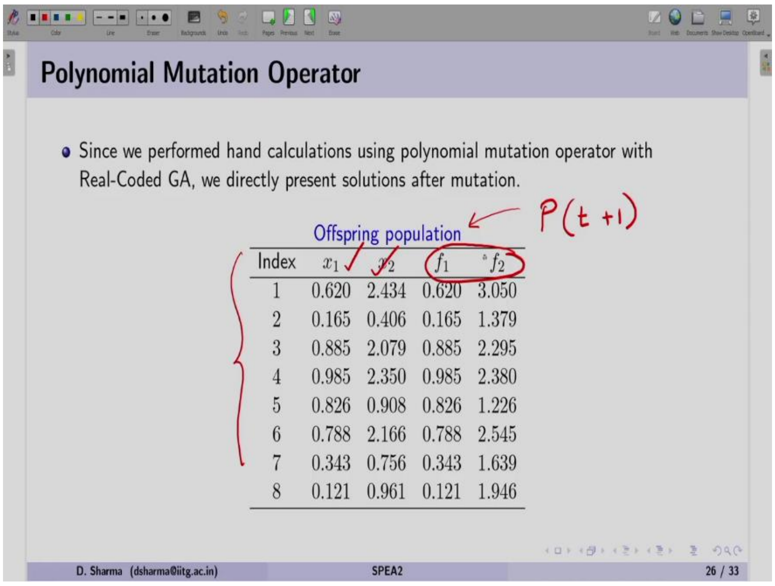
Once it is done, we have to perform the mutation. As we remember, the purpose of the mutation is to perturb the solution in the vicinity with the help of some probability distribution function for the real number. So, this mutation again creates new solutions, but with a low probability. So, as crossover is used for exploration, mutation is used for exploitation.

As the polynomial mutation was used in the original paper of SPEA 2; so let us see how the solution is mutated. So, we remember that, this is the solution which is created by the crossover operator, which is SBX. And then we have a difference between the lower and the upper bound and we are multiplying by the delta i. How we are calculating this delta i?

So, we have a probability distribution here and this probability distribution basically tells us this the figure, which is shown with respect to this probability distribution. When we are integrating this probability distribution and equating it to the random number R i, which is 0 to 1 in this case; so we will calculate the value of delta with respect to the random number R i.

So, we are going to integrate the probability distribution and equating into the R i as given in the figure. So, the area under this particular curve will be equating to the random number. This will arise two situation here; when random number is smaller than 0.5, we are going to use the formula number one; when it is greater than and equals to 0.5, we are going to use the another formula. Similar to the SBX crossover operator, we have the user defined parameter eta m that we have to fix it before working on the mutation operator.

(Refer Slide Time: 47:35)



Polynomial Mutation Operator

- Since we performed hand calculations using polynomial mutation operator with Real-Coded GA, we directly present solutions after mutation.

P(t +1)

Offspring population

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.165	0.406	0.165	1.379
3	0.885	2.079	0.885	2.295
4	0.985	2.350	0.985	2.380
5	0.826	0.908	0.826	1.226
6	0.788	2.166	0.788	2.545
7	0.343	0.756	0.343	1.639
8	0.121	0.961	0.121	1.946

D. Sharma (dsharma@iitg.ac.in) SPEA2 26 / 33

Now, here since it is a polynomial mutation, we have already performed the hand calculation with the real coded GA. So, we are presenting the offspring population or the new solutions directly after mutation. As can be seen in the in the in the table here, these solutions; the solutions after mutation are shown here and this population is called at say offspring population. In the current SPEA2, it is represented as P t plus 1. All these solution x 1 and x 2 values are given and f 1 and f 2 values are given for the representation purpose.

So, in this particular session, we have gone through strength how to calculate the fitness of each solution using SPEA2, where we calculated the strength followed by the raw fitness. Then we also calculated the distance D i and this distance D i at the raw fitness, club together will make a fitness of a solution.

And thereafter binary tournament selection operator with replacement was discussed and that will help us to select the solution that on which the crossover will be applied. Since SBX operator and polynomial mutations which we discussed, so we presented these solutions directly.

So, in this session the part one of SPEA2, we have discussed these things; in the following session, we will be understanding the survival stage of SPEA2 that will be useful for updating the archive. So, with this understanding on the first part of SPEA2, I conclude this session.

Thank you.