Now I will introduce the last algorithm.

NSGA-II is the basic adaption of single-objective Genetic algorithm using non-dominated sorting.

It tries to find pareto fronts by comparing every pair of individuals’ objective function.

We first record the pareto front 1, and generate next pareto fronts by decreasing the dominated number n\_q.

For each generation of NSGA-II, we made a child offspring from the current population with the same size. Then we sort the entire individuals by non-dominated sorting, and cut at the half to maintain the population size.

We first initialized the employee allocation matrix for each individual gene as the random number selected from 0 to 0.9. For each individual, we followed the same fitness function.

To generate the offspring, we used simulated binary crossover as crossover operator and polynomial-probability based mutation.

For simulated binary crossover, we first selected 2 pairs of 2 individuals and pick 1 from each pair by crowding distance order, and crossovered that 2 individuals.

For polynomial mutation, we selected each element in the employee allocation matrix by the probability of the proposed function. Then we allocated new random value for them.

We set the population size as 32, crossover rate as 0.9. And we made 1 thousand generation.

Now we will move to the results and discussion.

Here is the comparison between NSGA2 and MOCell. In the big sight, you can see that the fitness value of both cannot exceed 0.01. It is because that these two GAs failed to get the valid instance from the initial invalid instances only with SBX and the mutation. In detail, You can see the main difference of the increasing phenomena between two algorithms. For each generation, MOCell archives the fitnesses by the comparison of two randomly selected individuals. Because of that, it starts from not the best value among the entire population, but gradually increases. NSGA2 has quite different phenomena. Since the next generation of NSGA2 is made from the sorted mixture of the current population and generated offspring, while the highest fitness value barely changes, the lowest fitness value changes more often.