# Optimization Methods
# Lab 6 Session

By Shi Jimao

# Task1(1)

We want the worst solution is final solution. However, here K=2, so if we let the worst solution is final solution, we must make sure the worst solution is in what we randomly choose 2 solutions from all solutions.

The total number of choice is 10*10=100. While only one situation is correct which is selected the worst solution twice.

Therefore, the probability is $\frac{1}{100}$

# Task1(2)

Since we want to calculate the possibility of i-th worst solution as our final solution, first we need to make sure that i-th worst solution is in K solutions that we random select. In order to make sure i-th worst solution is the best solution in K solutions, we can only choose from the worst solution to i-th worst solution as the others in K solutions. We can calculate it as $(\frac{\mathrm{i}}{10})^2-(\frac{\mathrm{i}-1}{10})^2$

So the probability is:

$$P = \frac{2i-1}{100}$$

# Task1(3)

Similar to Task1(2), only K varies from 2 to 3.

$$P = (\frac{\mathrm{i}}{10})^3 - \left(\frac{\mathrm{i} - 1}{10}\right)^3 = \frac{3i^2 - 3i + 1}{1000}$$

# Task2

(μ+μ)ES: The standard comma strategy is best in my point of view.

1. **Balanced Exploration and Exploitation**: (μ+μ)ES generates twice the number of offspring as the parent population in each generation and then selectively replaces individuals based on their performance. This approach maintains diversity within the population while promoting the evolution towards better solutions, effectively balancing exploration and exploitation.

2. **Avoidance of Premature Convergence**: By retaining diversity and promoting selective replacement, (μ+μ)ES helps avoid premature convergence to local optima, thus enhancing the algorithm's ability to explore the entire search space and potentially find global optima.

3. **Simplicity and Effectiveness**: Compared to more complex generation update mechanisms like (μ,λ)ES or (μ,μ)ES, implementing (μ+μ)ES is relatively straightforward and computationally efficient. Its simplicity makes it more accessible and easier to fine-tune in practice.

4. **Good Convergence Performance**: By preserving promising individuals in each generation, (μ+μ)ES facilitates rapid convergence towards better solutions. This strategy can accelerate the algorithm's ability to find solutions, particularly in optimization problems.