

Task 1

a) What are $(1 + 1)$, $(1 + \lambda)$, $(1, \lambda)$ and (μ, λ) ?

They describe instances of evolutionary strategies.

b) Explain differences between them.

$(1 + 1)$: One parent produces one candidate solution and parent and child compete based on objective fitness for a position in the next generation.

$(1 + \lambda)$: One parent produces λ candidate solutions and parent and children compete based on objective fitness for a position in the next generation.

$(1, \lambda)$: One parent produces λ candidate solutions and only the children compete based on objective fitness for a position in the next generation while the parent is disregarded.

(μ, λ) : The μ fittest parents from randomly selected λ individuals from the population produce $\frac{\lambda}{\mu}$ candidate solutions each and these children form the new population.

Task 2

a) How does Line Recombination work? Explain in detail.

In genetic algorithms, children are bred by performing crossover operations using the copies of two parents and possibly mutating the results. The problem is that the children are limited by the hyperspace spanned by the parents. The best solution may be outside this hyperspace. Line recombination offers a method to explore solution outside this hyperspace. Line recombination can generate any point on the line defined by the two parents. First, a positive value p is set that defines, how far the hyperspace can be outreached. Next, two random values α and β between $-p$ and $1 + p$ are generated. These values are used to create new values based on a combination of the parents. With two parent vectors \vec{x} and \vec{v} containing float values, all of their individual values x_i and v_i will be replaced with weighted recombinations. The new x_i value will be $\alpha x_i + (1 - \alpha)v_i$ and the new v_i will be $\beta v_i + (1 - \beta)x_i$. This results in two new points which are either on the line spanned by the two original points or exceeding it by at most p . This way, values outside of the given hyperspace can be reached.

b) How can Line Recombination be extended to get Intermediate Line Recombination?

It can be extended by choosing new α and β values for each new set of x_i and v_i . This way, children are not limited to the line vector, but can take values from the entire hypercube.

c) Implement Intermediate Line Recombination as a python function.

Check tasks/linearRecomb.py

Task 3

- a) What is Fitness-Proportionate Selection (FPS) and how does it work?
- b) What is Stochastic Universal Sampling (SUS) and how does it work?
- c) Implement SUS as a python function.