Hans Lienhop, 114926
Lucas Hübner, 116232

Exercise 2

Search Based Software Engineering
May 20, 2019

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.

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

c) Implement Intermediate Line Recombination as a python function.

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.