INF3490/INF4490 Exercises - Evolutionary Computing

Eivind Samuelsen*

$\mathbb{P}$ marks the programming exercises, we strongly recommend using the python programming language for these. Exercises may be added/changed after publishing.

1 Evolution strategy(ES)

1.a

A common variant of evolution strategies used for (local) search is the $(1+4)$ ES. How would this differ from the $(1+1)$ ES in how the search space is explored? How does this, and $(1+\lambda)$ in general, compared to hill climbing and greedy search?

1.b

What effect does an adaptive search strategy have on optimization performance?

1.c

How would it affect the search if the strategy parameters were mutated after the solution parameters instead of before?

2 ES Implementation

$\mathbb{P}$ 2.a

Ignoring mutation, and starting with the population $\{1, 2, 3, 4\}$, implement and run 3 generations of a $(4+8)$ ES maximizing $g(x) = x$, and observe what the end population looks like (use intermediary recombination).

2.b

If a $(4, 8)$ ES had been used in Problem 2.a, what would the probability of the optimal solution ($x = 4$) surviving the first generation have been?

$\mathbb{P}$ 2.c

Repeat Problem 2.a with an EP with $q = 2$. How do the two algorithms compare?

3 Knapsack problem

In a 0-1 **knapsack problem**, how could you implement a repair mutation to transform infeasible solutions into feasible ones (i.e. make the sum of costs of the selected items go below the budget)?

Contact and Github

Corrections of grammar, language, notation or suggestions for improving this material are appreciated. E-mail me at **tjbersta@uio.no** or use **GitHub** to submit an issue or create a pull request. The **GitHub repository** contains all source code for assignments, exercises, solutions, examples etc. As many people have been involved with writing and updating the course material, they are not all listed as authors here. For a more complete list of authors and contributors see the **README**.

*See **README** for complete list of authors/contributors.