Exercise sheet 5: Evolution and genetic algorithms

Evolution – Exercise 1: Genetic algorithms I – the knapsack problem

Genetic algorithms (GA) model the process of evolution and use it to solve optimisation problems.

A transport company has been asked to move goods from a distribution centre to a shop. The goods are packed in boxes that differ in weight and in the value of their contents. The van cannot carry all boxes. The client wants the total value of the transported goods to be as high as possible.

Table 1: List of boxes with value in k$ and weight in kg

No.	Value	Weight	No.	Value	Weight
Box 1	9	105	Box 13	28	145
Box 2	3	55	Box 14	26	60
Box 3	4	20	Box 15	17	75
Box 4	8	35	Box 16	15	80
Box 5	2	15	Box 17	2	85
Box 6	11	60	Box 18	12	115
Box 7	30	25	Box 19	12	35
Box 8	7	35	Box 20	10	150
Box 9	13	35	Box 21	23	135
Box 10	23	105	Box 22	4	110
Box 11	13	115	Box 23	34	80
Box 12	8	65	Box 24	3	105

a) Write a GA in a programming language of your choice that maximises the value of the transported goods without exceeding the load capacity of the van.
   - The load capacity of the van is $m_{max} = 920\,\text{kg}$.
   - Use a population A with individuals $a_1 \dots a_N$ and population size $N = 50$.
   - A genome (a combination of boxes) is encoded as a bit string of length L, where L is the number of boxes.
   - Each box k_i has a weight $m(k_i) = m_i$ and a value $w(k_i) = w_i$ (see Table 1).
   - The fitness function $F(a)$ is the total value of the **loaded goods**. Make sure that the total weight of the boxes does not exceed the load capacity of the van.
   - Copy the two best individuals unchanged into the next generation ("elitist selection").
   - Create all other individuals of the next generation in pairs:

- Select two parents by **tournament selection**: draw two individuals at random from the population and take the fitter one as the first parent; repeat for the second parent.
- With probability $p_c = 0.8$, the two parents produce two children by **one-point crossover**; otherwise the children are copies of the parents.
- **Mutation**: in each child, flip every bit independently with probability $p_m = 1/L$.

b) Give the fitness function $F(a)$. How do you treat solutions that exceed the load capacity of the van? Name an alternative and discuss how it affects the optimisation.

c) Plot the course of the optimisation (best and mean fitness of the population per generation).

d) Which combination of boxes is the optimal solution? How can you check that your GA has found the global optimum and not only a local one?

e) Compare your result with three greedy strategies that load boxes until no further box fits: (i) lightest first, (ii) highest value first, (iii) highest value per kg first. Why does none of them find the optimal solution?

f) How do you define the termination condition?

g) Run your GA with (i) crossover only ($p_m = 0$), (ii) mutation only ($p_c = 0$) and (iii) both, as described in a). Use at least 20 runs per variant, each with a different random seed and at most 500 generations. Report the median number of generations until the optimum is found and how often it is found. What is the role of mutation and what is the role of crossover?

h) How and why does the course of the optimisation change when no "elitist selection" is performed?

Evolution – Exercise 2: Genetic algorithms II – traits encoded by several genes

Consider an organism with 16 traits (e.g. an enzyme, an organ, a behaviour). Each trait is encoded by 4 genes, and a trait is only functional if **all four** of its genes carry the beneficial allele. A single beneficial allele on its own has no effect. The genome is a bit string of length $L = 16 \times 4 = 64$, where 1 stands for the beneficial and 0 for the neutral allele. The four genes of trait t ($t = 0, \ldots, 15$) lie next to each other at positions $4t, \ldots, 4t + 3$.

Use the GA from Exercise 1 with the same parameters ($N = 50$, two elite individuals, tournament selection, $p_c = 0.8$, $p_m = 1/L$) and at most 500 generations per run. Fix the random seed of each run so that your results are reproducible.

a) Give the fitness function $F(a)$ as the number of functional traits ($0 \leq F(a) \leq 16$) and implement it.

b) Compare (i) mutation only ($p_c = 0$) with (ii) mutation and one-point crossover. Use at least 20 runs per variant and report the median number of generations until all 16 traits are functional and how often this happens within 500 generations. Plot the course of the optimisation for one typical run of each variant.

c) Explain your observations. Does crossover have the same effect as in Exercise 1g)? Why or why not? Relate your answer to the evolution of sexual reproduction: where do the beneficial mutations come from that end up together in one genome?

d) Implement **uniform crossover**: for each position, choose at random (probability 0.5) from which parent the gene is taken. Compare one-point and uniform crossover.

e) So far, the four genes of a trait lie next to each other. Now scatter them! Shuffle the list of positions $0, \ldots, 63$ once and call the resulting order π. Use the same π for all individuals and all runs. Repeat the comparison of one-point and uniform crossover from d).

f) Interpret the results from d) and e) biologically. What corresponds to one-point crossover and what to uniform crossover during meiosis? Use the terms *linkage* and *recombination frequency*.