Binary GA: A Worked Example

Problem: maximize

$$f(x) = x^2 \qquad \text{subject to } x \in [0, 31].$$

Key idea

- The problem lives in **phenotype space** (x).
- The GA operates on **binary strings** (genotype).

Takeaway

GA does not search the problem space directly.

Representation (Genotype $\leftrightarrow$ Phenotype)

Encode $x \in [0, 31]$ using a 5-bit string

$$\mathbf{b} = b_4 b_3 b_2 b_1 b_0 \in \{0, 1\}^5 \quad \Rightarrow \quad x = \text{integer}(\mathbf{b}).$$

Examples:

- $01001_2 = 9$
- $11111_2 = 31$

Takeaway

Binary encoding discretizes a numeric decision into a finite search space.

Step 1: Initialize Population and Evaluate Fitness

Population size $N = 4$ (kept tiny so every step is visible).

ID	Binary string	x	Fitness $f(x) = x^2$
A	01001	9	81
B	10100	20	400
C	00110	6	36
D	11100	28	784

Takeaway

Fitness evaluation happens in phenotype space (on x), not on bits.

Step 2: Parent Selection (Tournament, $k = 2$)

We use tournament selection with $k = 2$.

Example tournaments:

- (A vs C) $\rightarrow$ A
- (B vs D) $\rightarrow$ D
- (A vs B) $\rightarrow$ B
- (C vs D) $\rightarrow$ D

Selected parents:

$$[A,\ D,\ B,\ D]$$

Takeaway

Selection pressure biases reproduction toward higher fitness, not perfection.

Step 3: One-Point Crossover ($p_c = 0.8$)

Pair parents: (A, D) and (B, D).

Assume crossover occurs (with probability p_c) and choose cut after bit 3:

$$\begin{aligned} A &: 010|01 \\ D &: 111|00 \\ \hline O_1 &: 01000 \\ O_2 &: 11101 \end{aligned}$$

Second pair (same cut point):

$$\begin{aligned} B &: 101|00 \\ D &: 111|00 \\ \hline O_3 &: 10100 \\ O_4 &: 11100 \end{aligned}$$

Takeaway

Step 4: Bit-Flip Mutation ($p_m \approx 1/L$)

Each bit flips independently with probability

$$p_m \approx \frac{1}{L}, \qquad L = 5 \Rightarrow p_m = 0.2.$$

Example mutation events:

- 01000 $\rightarrow$ 01010 (one bit flipped)
- 11101 $\rightarrow$ 11101 (no mutation)

Offspring population O_0 after mutation:

Offspring	Binary	x
O_1	01010	10
O_2	11101	29
O_3	10100	20
O_4	11100	28

Takeaway

Step 5: Replacement and Elitism

We apply **generational replacement** with **elitism**.

- Best individual in P_0: $D = 11100$ ($x = 28$, $f = 784$)
- Elitism: copy D into P_1
- Fill remaining slots with the best offspring from O_0

New population P_1:

$$\{11100,\ 11101,\ 10100,\ 01010\}$$

Takeaway

Elitism guarantees the current best is not lost (but increases pressure).

What Just Happened? (Interpretation + Diagnostics)

Observations:

- Best improved: $28^2 \rightarrow 29^2$
- Population drifted toward larger x
- Diversity decreased slightly (more high-bit patterns)

Diagnostic questions:

- If we increase tournament size k, do we converge faster *and* risk collapse?
- If we reduce p_m, do we lose alleles permanently?
- If we increase elitism, do we freeze the population structure?

Takeaway

GA behavior emerges from interaction of selection, variation, and representation.