PROOF: RANDOM SEARCH LOWER BOUND ON F1 (ONEMAX)

Theorem

Random Search needs with probability **1 - e^(-Ω(n))** at least a budget of **2^(n/2)** fitness evaluations to reach an optimal search point for the function F1 (OneMax).

Proof

Problem Setup

Function F1 (OneMax):

$$OneMax(x) = \Sigma(i = 1 \; to \; n) \; x\_i, where \; x \in \{0,1\}^\wedge n$$

Properties:

- Search space size: $|\{0,1\}^\wedge n| = 2^\wedge n$
- Unique global optimum: $x* = 1^n \; with \; OneMax(1^n) = n$
- Number of optima: 1

Random Search Algorithm: In each iteration, samples a solution uniformly at random from {0,1}^n, independent of previous samples.

Step 1: Probability of Success in a Single Evaluation

The probability that Random Search finds the optimal solution in one random sample is:

$$p = P(sample \; x* = 1^n) = \frac{1}{2^n}$$

This follows from uniform sampling over the search space.

Step 2: Probability of Success After t Evaluations

Let t be the number of fitness evaluations (budget). Since Random Search samples independently, the probability of **not finding** the optimum after t evaluations is:

$$P(failure \; after \; t \; evaluations) = \left(1 - \frac{1}{2^n}\right)^t$$

Equivalently, the probability of success is:

$$P(success\ after\ t\ evaluations) = 1 - \left(1 - \frac{1}{2^n}\right)^t$$

Step 3: Setting Budget $\boldsymbol{t = 2^{\frac{n}{2}}}$

We now evaluate the probability of success with budget $\boldsymbol{t = 2^{\frac{n}{2}}}$:

$$P(success) = 1 - \left(1 - \frac{1}{2^n}\right)^{2^{\frac{n}{2}}}$$

To bound this probability, we use the inequality $(\mathbf{1} - \boldsymbol{x}) \le \boldsymbol{e^{-x}}\ for\ x \ge 0$:

$$\left(1 - \frac{1}{2^n}\right)^{2^{\frac{n}{2}}} \le e^{-\frac{2^{\frac{n}{2}}}{2^n}} = e^{-2^{-\frac{n}{2}}}$$

Step 4: Evaluating the Exponential

For large n, note that:

$$2^{-\frac{n}{2}} \to 0\ as\ n \to \infty$$

Therefore:

$$e^{-2^{-\frac{n}{2}}} \to e^0 = 1\ as\ n \to \infty$$

This means:

$$P(failure\ after\ 2^{\frac{n}{2}}\ evaluations) \to 1$$

Step 5: Converting to Big-O Notation

More precisely, using $\boldsymbol{ln(2) = \Theta(1)}$:

$$2^{-\frac{n}{2}} = e^{-\frac{n}{2} \cdot \ln(2)} = e^{-\Theta(n)}$$

Therefore:

$$P(success\ after\ 2^{\frac{n}{2}}\ evaluations) = 1 - e^{-2^{-\frac{n}{2}}}$$

$$= 2^{-\frac{n}{2}} \cdot (1 + o(1))$$

$$= e^{-\Omega(n)}$$

And consequently:

$$P(failure\ after\ 2^{\frac{n}{2}}\ evaluations) = 1 - e^{-\Omega(n)}$$

Conclusion

With probability $1 - e^{-\Omega(n)}$, Random Search requires more than $2^{\frac{n}{2}}$ fitness evaluations to find the optimal solution for F1 (OneMax).

Interpretation: Since $e^{-\Omega(n)} \rightarrow 0$ exponentially fast as n increases, Random Search fails with overwhelming probability even after $2^{\frac{n}{2}}$ evaluations. This demonstrates that Random Search has exponential complexity on F1.

Comparison to (1+1) EA: While Random Search requires exponential time, the (1+1) EA solves OneMax in expected time O(n log n) using fitness-based selection, demonstrating the power of evolutionary mechanisms over pure random sampling.