Mathematical Proof 2: RLS & (1+1) EA

Prove that the number of fitness evaluations of RLS and (1+1) EA to reach an optimal search point for the function F3 is $O(n \log n)$ with probability $\Omega(1)$.

F3: A Linear Function with Harmonic Weights

$$f : \{0,1\}^n \to \mathbb{R}, x \mapsto \sum_i ix_i$$

The problem is a linear function with harmonic weights.

Randomised Local Search:

1) Let the bitstring length be n, and let $w_i > 0$ denote the weight of the i-th bit. RLS flips exactly one randomly chosen bit per iteration. Since all weights are positive:

   - Flipping a 0-bit to 1 strictly increases the fitness, and the offspring is accepted.
   - Flipping a 1-bit to 0 decreases the fitness, and the offspring is rejected.

   Thus, the algorithm only accepts a mutation when it flips a 0-bit, reducing the zero count by 1. Flipping a 1-bit leaves the state unchanged.

2) Let the algorithm be in a state with i zeros, where the RLS algorithm selects a uniform index from $\{1, \dots, n\}$ at random. The probability that RLS selects one of the i zero bits is,

   $$\boldsymbol{p_i = \frac{i}{n}}, \textit{ where n is the bitstring length}$$

3) Let T_i denote the waiting time to reduce the zero count from i to $i-1$. Then T_i is a geometric random variable with success probability p_i,

   $$\therefore \ \boldsymbol{E[T_i]} = \frac{1}{p_i} = \frac{1}{\frac{i}{n}} = \boldsymbol{\frac{n}{i}}$$

   The total expected number of iterations to reach the optimum, starting from an initial zero count of n is,

   $$\boldsymbol{E[T]} = \sum_{i=1}^{n} E[T_i] = \sum_{i=1}^{n} \frac{n}{i} = n \sum_{i=1}^{n} \frac{1}{i} = \boldsymbol{n \cdot H_n},$$

   where H_n is the n-th harmonic number. Since $H_n = \Theta(\log n)$, we have,

   $$\boldsymbol{E[T] = \Theta(n \log n)}$$

4) Since we know the expected runtime, we can use Markov's inequality to bound the probability that the runtime is at most a multiple of this expectation,

$$\Pr(T \geq 2E[T]) \leq \frac{E[T]}{2E[T]}$$

$$\therefore \boldsymbol{\Pr(T \geq 2E[T]) \leq \frac{1}{2}}$$

Hence, with constant probability, the algorithm reaches the optimum in at most $2E[T]$ iterations,

$$\Pr(T \leq 2E[T]) \geq \frac{1}{2}$$

Now, we can substitute the expectation bound such that,

$$2E[T] = 2 \cdot \Theta(n\, log\, n) = O(n \log n)$$

$$\therefore \boldsymbol{\Pr\big(T \leq O(n \log n)\big) \geq \Omega(1)}$$

Thus, with constant probability $\Omega(1)$, the runtime of the RLS algorithm is $O(n \log n)$.

So, RLS requires $\boldsymbol{O(n \log n)}$ evaluations with constant probability $\boldsymbol{\Omega(1)}$ to reach an optimal search point for the function F3.

(1+1) Evolutionary Algorithm:

1) Let the bitstring length be n, and let $w_i > 0$ denote the weight of the i-th bit. The (1+1) EA flips each bit independently with probability $\frac{\mathbf{1}}{\boldsymbol{n}}$.

2) If the current solution contains i zero-bits, then the algorithm improves by flipping exactly one of these zeros to one while leaving all other bits unchanged. The probability of flipping a specific bit and no other bits is,

$$p = \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{e \cdot n}, \qquad \left(since\left(1 - \frac{1}{n}\right)^{n-1} \approx \frac{1}{e}\right)$$

Hence, the probability of flipping exactly one of the i zero-bits to one is,

$$\boldsymbol{p_i \geq \frac{i}{e \cdot n}}$$

3) Let T_i denote the waiting time to reduce the zero count from i to $i-1$. Then T_i is a geometric random variable with success probability at least p_i , and therefore

$$\boldsymbol{E[T_i]} \leq \frac{1}{p_i} \leq \boldsymbol{\frac{e \cdot n}{i}}$$

The total expected number of iterations to reach the optimum, starting from an initial zero count of n is,

$$\boldsymbol{E[T]} \leq \sum_{i=1}^{n} E[T_i] \leq \sum_{i=1}^{n} \frac{e \cdot n}{i} = e \cdot n \cdot \sum_{i=1}^{n} \frac{1}{i} = \boldsymbol{e \cdot n \cdot H_n},$$

where H_n is the n-th harmonic number. Since $H_n = \Theta(\log n)$, we have,

$$\boldsymbol{E[T]} = e \cdot n \cdot \Theta(\log n) = \mathbf{O}(\boldsymbol{n} \log \boldsymbol{n})$$

4) Since we know the expected runtime, we can use Markov's inequality to bound the probability that the runtime is at most a multiple of this expectation,

$$\Pr(T \geq 2E[T]) \leq \frac{E[T]}{2E[T]}$$

$$\therefore \mathbf{Pr}(\boldsymbol{T} \geq \mathbf{2}\boldsymbol{E}[\boldsymbol{T}]) \leq \frac{\mathbf{1}}{\mathbf{2}}$$

Hence, with constant probability, the algorithm reaches the optimum in at most $2E[T]$ iterations,

$$\Pr(T \leq 2E[T]) \geq \frac{1}{2}$$

Now, we can substitute the expectation bound such that,

$$2E[T] = 2 \cdot O(n\, log\, n) = O(n \log n)$$

$$\therefore \mathbf{Pr}(\boldsymbol{T} \leq \mathbf{O}(\boldsymbol{n} \log \boldsymbol{n})) \geq \boldsymbol{\Omega}(\mathbf{1})$$

Thus, with constant probability $\Omega(1)$, the runtime of the (1+1) EA is $O(n \log n)$.

So, the (1+1) EA requires $\mathbf{O}(\boldsymbol{n} \log \boldsymbol{n})$ evaluations with constant probability $\boldsymbol{\Omega}(\mathbf{1})$ to reach an optimal search point for the function F3