1 Pseudo-code

(1+1) EA

1. Choose $s \in \{0,1\}^n$ randomly.
2. Produce s' by flipping each bit of s with probability $1/n$.
3. Replace s by s' if $f(s') \geq f(s)$.
4. Repeat Steps 2 and 3 forever.

RLS

1. Choose $s \in \{0,1\}^n$ randomly.
2. Produce s' from s by flipping **one** randomly chosen bit.
3. Replace s by s' if $f(s') \geq f(s)$.
4. Repeat Steps 2 and 3 forever.

2 Chernoff Bound Initialisation

2.1 Expected Ones Initialisation

Given that both algorithms initialise an n-set of binary string at random specifically $X_i = \text{Unif}\{0,1\}$ for $i = 1, 2, \ldots, n$. Therefore the probability of getting 0 or 1 is the same as

$$P(X_i = 1) = P(X_i = 0) = \frac{1}{2}$$

The expected value of each X_i is

$$\mathbb{E}[X_i] = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} = \frac{1}{2}$$

Suppose that $X = X_1 + X_2 + \ldots + X_n$, then the expected value of X by using the *Linearity of Expectation* is

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \frac{n}{2}$$

If X is the summation of those values inside the binary string, then there will be expected to have $\frac{n}{2}$ ones.

2.2 Chernoff

$$\forall \delta > 0 : P(X > (1+\delta) \cdot \mathbb{E}[X]) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mathbb{E}[X]} \qquad (\bigstar)$$

$$\forall\, 0 < \delta < 1 : P(X < (1-\delta) \cdot \mathbb{E}[X]) < e^{-\mathbb{E}[X]\delta^2/2}$$

From the previous section, it's known that $\mathbb{E}[X] = \frac{n}{2}$, choose inequality $(\bigstar)$ and then substitute.

$$P\left(X > (1+\delta) \cdot \frac{n}{2}\right) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\frac{n}{2}}$$

$$\Rightarrow P\left(X > \left(\frac{1}{2} + \frac{\delta}{2}\right) \cdot n\right) < \ldots$$

Let $2\delta' = \delta$, which is always inside the bound of $\delta > 0$. Now choose the domain of δ as $0 < \delta \leq 1$. Therefore, the new domain will be $0 < 2\delta' \leq 1 \Rightarrow 0 < \delta' \leq \frac{1}{2}$, thus

$$\begin{aligned} P\left(X > \left(\frac{1}{2} + \delta'\right) \cdot n\right) &< \left(\frac{e^{2\delta'}}{(1+2\delta')^{1+2\delta'}}\right)^{\frac{n}{2}} \\ \Rightarrow \ldots &< \rho^{\frac{n}{2}} \\ \Rightarrow \ldots &< e^{\frac{n}{2}\ln\rho} \end{aligned}$$

Evaluate $\rho = \frac{e^{2\delta'}}{(1+2\delta')^{1+2\delta'}}$

$$\ln\rho = 2\delta' - (1+2\delta')\ln(1+2\delta') < 0, \quad \text{Negative constant}$$

Therefore,

$$P\left(X > \left(\frac{1}{2} + \delta'\right) \cdot n\right) < e^{-\Omega(n)}$$

Then,

$$\begin{aligned} P\left(X > \frac{2n}{3}\right) &< e^{-\Omega(n)} \\ \Rightarrow 1 - P\left(X > \frac{2n}{3}\right) &\leq 1 - e^{-\Omega(n)} \\ \Rightarrow P\left(X \leq \frac{2n}{3}\right) &\leq 1 - e^{-\Omega(n)} \end{aligned}$$

This means that both algorithms with uniform random binary string initialisation will have a very high chance of having at most $2n/3$ one-bits inside the initialisation. It's so high that with large n or large binary space it's almost gauranteed.

3 RLS Lower Bound

Given from the initialisation that there are expected to have $2n/3$ one-bits, thus $m = n/3$ zero-bits needed to be flipped. Each zero-bit has a chance of being flipped is $1/n$. We want to know what is the lower bound Ω that will let flipping all distinct 0-bits at least once. This is a Collector's Coupon Theorem.

3.1 Collector's Coupon Theorem

Suppose that we have n distinct cards and we draw each everytime with *replacement*. How many times do I expect to draw to collect all of those cards. This implies a similarity as our prolem with RLS lower bound.

From Wikipedia, let T be the total number of times to finish flipping all the 0-bits, t_i be the number of times when reaching to the i^{th} bit after $i-1$ bits, therefore the probability for reach. Then $T = t_k + t_{k+1} + \ldots + t_m$. The probability of each 0-bit being picked is,

$$p_i = \frac{m - (i-1)}{n}$$

Here comes the *significant* point, each t_i has a *geometric distribution*, recall the geometric distribution from Wikipedia, *The probability distribution of the number X of Bernoulli trials needed to get one success.*

So, each time t_i reflects a geometric distribution since we want to know how many times do we need to be able to pick that 0-bit, by the geometric distribution we can infer the expected value as $\mathbb{E}[t_i] = 1/p_i$, by the *linearity of the Expectation*,

$$\begin{aligned}
\mathbb{E}[T] &= \mathbb{E}[t_k] + \mathbb{E}[t_{k+1}] + \ldots + \mathbb{E}[t_m] \\
&= \frac{1}{p_k} + \frac{1}{p_{k+1}} + \ldots + \frac{1}{p_m} \\
&= \frac{n}{m-(k-1)} + \frac{n}{m-k} + \ldots + \frac{n}{m-(m-1)} \\
&= n\left(\frac{1}{n-(k-1)} + \frac{1}{n-k} + \ldots + \frac{1}{1}\right) \\
&= nH_{n-k+1} \\
&= n\ln(n-k+1) + C \\
\mathbb{E}[T] &= \Omega(n \log n)
\end{aligned}$$

By using the Theorem, we can derive a lower bound for the RLS to be $\Omega(n \log n)$.

4 (1+1)EA Lower Bound

Given the initialisation, as for the RLS the lower bound is $\Omega(n \log n)$, so we are also trying to show that for the EA it's also $\Omega(n \log n)$. We may assume that everytime we flip the 0-bit to 1-bit it's will yield a improvement. We may choose $t = (n-1)\ln n$ iterations until optimum, the reason of choosing such t maybe from more complicated mathmatical proof, since this will give a nice bound for when hitting an optimum.

4.1 Waterfall Probability

$$\begin{aligned}
1 - \frac{1}{n}, &\quad \text{the probability of a specific bit not being flipped.} \\
\left(1 - \frac{1}{n}\right)^t, &\quad \text{the probability of a specific bit not being flipped for all } t \text{ iterations} \\
1 - \left(1 - \frac{1}{n}\right)^t, &\quad \text{the probability of a specific bit being flipped for all } t \text{ iterations} \\
\left[1 - \left(1 - \frac{1}{n}\right)^t\right]^m, &\quad \text{the probability of all } m \text{ bits being flipped for all } t \text{ iterations} \\
1 - \left[1 - \left(1 - \frac{1}{n}\right)^t\right]^m, &\quad \text{the probability of all } m \text{ bits not being flipped for all } t \text{ iterations}
\end{aligned}$$

The last one is called the *failure probability.*

4.2 Some Calculus

For whatever algorithms time complexity analysis, we always let n explodes or in another word let $n \to \infty$. Before that, let,

$$\frac{1}{h(n)} = \left(1 - \frac{1}{n}\right)^t$$

Then,

$$\lim_{n\to\infty} 1 - \left[1 - \frac{1}{h(n)}\right]^{n/3}$$

$$1 - \left[\lim_{n\to\infty} \left(1 - \frac{1}{h(n)}\right)^{n}\right]^{1/3} \qquad (\circ)$$

Now, we need to analyse does $h(n)$ grows therefore $1/h(n)$ will diminish and also will the power of n catch up with the dimishing. Let's analyse the $h(n)$ first,

$$\begin{aligned}
\lim_{n\to\infty} h(n) &= \lim_{n\to\infty} \frac{1}{(1-\frac{1}{n})^{t}} \\
&= \frac{1}{\lim\limits_{n\to\infty} \left(1 - \dfrac{1}{n}\right)^{(n-1)\ln n}}, \quad \text{Using the Quotient Limit Law} \\
&= \frac{1}{\lim\limits_{n\to\infty} \left[\left(1 - \dfrac{1}{n}\right)^{(n-1)}\right]^{\ln n}} \\
&= \frac{1}{\lim\limits_{n\to\infty} \left[e^{-1}\right]^{\ln n}}, \quad \text{Classic Limits Problem check on Math Exchange} \\
&= \frac{1}{n^{-1}} \\
&= n
\end{aligned}$$

After analysis of $h(n)$ it yields n, therefore with $(\circ)$ as $n \to \infty$ it seems to yields back the Original Limits problem. So,

$$\text{From } (\circ) \Rightarrow 1 - e^{-1/3}$$

We have found that the failure probability is constant which is $\Omega(1)$

Observation

~~The limit only approach the value from the side, in this case we only consider approach ∞ from the left to right, therefore as n is slowly increasing but never reach the value thus an observation arises: $\lim_{n\to\infty}\left(1 - \frac{1}{h(n)}\right)^{n} < e^{-1}$. From $(\circ) \Rightarrow \geq 1 - e^{-1/3}$, Minus sign therefore change the equality sign.~~

~~Recall that time complexity for the EA is the number of iterations until reaching the optimum, same notation as T is the total number of iterations until optimum, and we let $t = (n-1)\ln n$ iterations until optimum. Therefore if,~~

- ~~$\boldsymbol{T \leq t}$ meaning sucessfully reaching optimum.~~
- ~~$\boldsymbol{T > t}$ meaning failed to reaching optimum.~~

~~Then the probability of failing to reach optimum is $P(T > t) \geq 1 - e^{-1/3} = \Omega(1)$.~~

Observation

To derive the $\geq$ as the failure probability for $P(T > t)$ can be done by observing the initialisation. As we have known that, there will be $\leq \frac{2n}{3}$ one-bits this mean that there will be $\geq \frac{n}{3}$ zero-bits. We can replace the constant $\frac{1}{3}$ with other scalar in the range of $x \in \left[\frac{1}{3}, 1\right]$, think the failure probability as $1 - e^{-x}$, thus it's increasing therefore we have that

$$P(T > t) \geq 1 - e^{-1/3}$$

Then the probability failing to reach optimum is

$$P(T > t) \geq 1 - e^{-1/3} = \Omega(1)$$

Explanation

The reason of choosing the failure probability instead of successful probability for the lower bound is because we give a candidate of t iterations, the number of iterations we want until it's the iteration of optimum. But then after some maths, it is shown that it's not sufficient enough for t iteration with a probability therefore we need some more iterations, which is the lower bound. It's the buffer of how much more is needed.

4.3 Tying Everything Together

~~After going through the hazard we are blessed with the result of a *survival function* as $1 - e^{-1/3}$, one of the properties we can harness is the monotonic decreasing essence. From Wikipedia~~

> ~~*Every* survival function is monotonically decreasing.~~

Using the Tail sum formula

$$\begin{aligned}
\mathbb{E}[T] &= \sum_{t=1}^{\infty} P(T > t) \\
&= \sum_{t=1}^{(n-1)\ln n} P(T > t) + \sum_{k=(n-1)\ln(n)+1}^{\infty} P(T > k) \\
&\geq \sum_{t=1}^{(n-1)\ln n} P(T > t) \\
&\geq \sum_{t=1}^{(n-1)\ln n} \Omega(1) \\
&\geq (n-1)\ln(n)\Omega(1) \\
\mathbb{E}[T] &\geq \Omega(n \log n)
\end{aligned}$$

Therefore the lower bound for (1+1)EA is $\Omega(n \log n)$