RLS algorithm and (1+1) EA algorithm Proof

We will define:

$k =$ number of 1-bits in current solution x
$d =$ number of 0-bits remaining in current solution x

And we will note that:

The RLS algorithm flips exactly 1 bit per iteration.
The (1+1) EA algorithm flips each bit independently with probability $1/n$.

RLS
We can find the probability of RLS decreasing d by 1 to be:
$P(flip\ a\ zero\ bit) = d/n$

Since RLS only flips one bit, d is either decreased by 1 or stays the same.
Therefore, we can find the expected number of steps for RLS to decrease d to be:

$$N[steps\ for\ d \to d-1] = \frac{1}{d/n} = n/d$$

To find the optimal search point, $x^* = (1,1,\ldots,1,1)$,

$$N[RLS\ steps] = \sum_{d=1}^{n} \frac{n}{d} = n \sum_{d=1}^{n} \frac{1}{d}$$

This format now resembles the harmonic numbers, and we can find that:

$$n \sum_{d=1}^{n} \frac{1}{d} = nH_n$$

Where H_n is the nth harmonic number. Since the harmonic series demonstrates a logarithmic growth, we know that the big O notation form of H_n would be $O(\log n)$.

Combining this with n, we can figure out that the runtime for RLS would be:

$$N[RLS\ steps] = O(n \log n)$$

(1+1) EA
In (1+1) EA, each bit flips independently with a probability $P(flip) = 1/n$.

For a solution, x, with d zero bits:

$$P(flip\ exactly\ one\ zero\ bit, and\ no\ one\ bits) = d * \frac{1}{n} * \left(1 - \frac{1}{n}\right)^{n-1}$$

$$\approx \frac{d}{n} * \frac{1}{e} = \Theta\left(\frac{d}{n}\right)$$

Therefore, the expected time it would take to find one improvement is the reciprocal:

$$\Theta\left(\frac{n}{d}\right)$$

To find the expected time it would take to reach the optimal solution, x^*, where $d = 1$, we will need to sum over the expected waiting times for all d values:

$$N[(1+1)EA\ steps] = \sum_{d=1}^{n} \Theta\left(\frac{n}{d}\right)$$

Since n can be taken out of the summation, we arrive at another harmonic series equation:

$$N[(1+1)EA\ steps] = \Theta\left(n \sum_{d=1}^{n} \frac{1}{d}\right)$$

Therefore, this follows a similar route to RLS, where the harmonic series can be denoted as $O(\log n)$ when referring to its big O notation. As such, we would find that:

$$N[(1+1)EA\ steps] = \Theta(n \log n)$$

And we can therefore say that the runtime of the (1+1) EA algorithm is also $O(n \log n)$.