Linear Function with Harmonic Weights Problem $\mathsf{F3}$

The $\mathsf{F3}$ problem is intened to maximise value of the linear function, but this can deduce to the OneMax problem, as for the definition of the linear function of $i \cdot x_i + \ldots + n \cdot x_n$, the value of the function only incease when there is increasing in the number of ones . Given $\mathbf{x} \in \{0,1\}^n$ thus we have that,

$$\arg\max_{\mathbf{x}\in\{0,1\}^n} \mathsf{F}(\mathbf{x}) \text{ where } \mathsf{F}(\mathbf{x}) = \sum_{i=1}^{n} i x_i$$

RLS Setup

RLS

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

Problem

Prove that Random Local Search with fitness evaluations to reach an optimal search point for the function $\mathsf{F3}$ is

$$\mathcal{O}(n \log n) \text{ with probability of } \Omega(1)$$

Proof. From the *Put the link here: Chernoff Bound Initialisation*, we know that the tuple $\mathbf{x}$ will expect to have $2n/3$ ones after initialisation with a high probability. Thus now we care for $m = n/3$ 0-bits to be flipped. Let t_i be the number of times to flip each 0-bit where $i = k, k+1, \ldots, m$ $t_i \sim \text{Geo}(p_i)$, where p_i is defined as

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

And the expected number of times to flipped the i^{th} to 1 is

$$\mathbb{E}[t_i] = \frac{n}{m-(i-1)}$$

Let $T = \sum_{i=k}^{m-1} t_i$ be the total time to get to optimum. Then, *from the note, put link here using the Coupon Theorem*,

$$\mathbb{E}[T] = n\ln(n-k+1) + \mathcal{C}, \qquad \text{where } \mathcal{C} > 0$$

Recall that for upper bound,

$$\mathcal{O}(n \log n) = \left\{ g(n) : \exists c > 0 : \exists n_0 \in \mathbb{N}^+ : \forall n \geq n_0 : g(n) \leq c \cdot n \log n \right\}$$

Given that $g(n) = n\ln(n-k+1) + \mathcal{C}$, thus:

$$\begin{aligned} n\ln(n-k+1) + \mathcal{C} &\leq cn \\ &\Rightarrow c \geq \ln(n-k+1) + \mathcal{C} \end{aligned}$$

Therefore there exists such $c > 0$, hence

$$\mathbb{E}[T] = \mathcal{O}(n \log n)$$

Recall the Markov's inequality

$$P(X \geq s \cdot \mathbb{E}[X]) \leq \frac{1}{s}, \quad \text{where } s > 0$$

Then,

$$P(T \geq \mathbb{E}[T]) \leq 1$$
$$\Rightarrow P(T < \mathbb{E}[T]) \geq 0 = \Omega(1)$$

Thus the probability of having at least one successful flip to 1-bit in less than the expected time, = is neglectible for large time is $P(T < \mathbb{E}[T]) = \Omega(1)$ ■

(1+1) EA Setup

(1+1) EA

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

Problem

Prove that (1+1) EA with fitness evaluations to reach an optimal search point for the function F3 is

$$\mathcal{O}(n \log n) \text{ with probability of } \Omega(1)$$

Proof. Given that the $(1+1)$ EA and $(1+1)$ EA_b is similar from the pseudo-code of [1, p.42]. From [1, p.44]

> Assume that we are working in a search space S and consider w.l.o.g. a function $f : S \to \mathbb{R}$ that should be maximized. S is partitioned into disjoint sets $A_1, \ldots, A_m$ such that $A_1 <_f A_2 <_f \cdots <_f A_m$ holds, where $A_i <_f A_j$ means that $f(a) < f(b)$ holds for all $a \in A_i$ and all $b \in A_j$. In addition, A_m contains only optimal search points. An illustration is given in Figure 4.1. We denote for a search point $x \in A_i$ by $p(x)$ the probability that in the next step a solution $x' \in A_{i+1} \cup \cdots \cup A_m$ is produced. Let $p_i = \min_{a \in A_i} p(a)$ be the smallest probability of producing a solution with a higher partition number. [1, p. 44]

There are $n = m$, as similar problem to OneMax, levels. Each level are defined as $A_i = \{1\}^i$ for $i = 1, 2, \ldots, m$. Suppose that the algorithm is currently in A_{m-k} level. There are k 0-bits left.

$$\frac{1}{n}, \quad \text{A specific bit is flipped to 1}$$
$$1 - \frac{1}{n}, \quad \text{A specific bit is not flipped to 1}$$
$$\left(1 - \frac{1}{n}\right)^{(n-1)}, \quad \text{All } n-1 \text{ bits are not flipped to 1}$$
$$\frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{(n-1)}, \quad \text{All } n-1 \text{ bits are not flipped to 1, except for a bit to be flipped to 1}$$
$$\frac{k}{n} \cdot \left(1 - \frac{1}{n}\right)^{(n-1)}, \quad \text{All } n-1 \text{ bits are not flipped to 1, except for the } k \text{ bits to be flipped to 1}$$

We have that,

$$\left(1-\frac{1}{n}\right)^{(n-1)} \geq \frac{1}{e}$$

Thus,

$$\frac{k}{n} \cdot \left(1-\frac{1}{n}\right)^{(n-1)} \geq \frac{k}{ne}$$

Then the transition probability to move to the next level for a better result is.

$$p_{n-k} \geq \frac{k}{en}$$

From Lemma 4.1[1, p.45] we have that $A_{n-k} \leq \frac{1}{p_{n-k}} \leq \frac{en}{k}$. The algorithm must move up all the levels until to reach level m for the optimum. Thus we have that,

$$\mathbb{E}[T] \leq \sum_{k=1}^{m-1} \frac{en}{k}$$
$$\mathbb{E}[T] \leq en \sum_{k=1}^{m-1} \frac{1}{k}$$
$$\mathbb{E}[T] \leq en \cdot H_{m-1}$$

Using the proof from above note we have that

$$en \cdot H_{m-1} = \mathcal{O}(n \log n)$$

Thus

$$\mathbb{E}[T] = \mathcal{O}(n \log n)$$

■

References

[1] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Natural Computing Series. Springer, 2010. DOI: 10.1007/978-3-642-16544-3.