

# Nature-inspired Algorithms

## Lecture 3

Algorithm Engineering Group  
Hasso Plattner Institute, University of Potsdam

22 May 2017



## Why is theory important?

We want to understand how an algorithm behaves over certain inputs.

**Idea:** run the algorithm over a large set of instances and observe its behavior.

**Problem:** sometimes evidence can be deceiving! Even when we think a process is well-behaved, it may not behave as we expect for all inputs.

## Why is theory important?

*At least half of the natural numbers less than any given number have an odd number of prime factors.*  
— George Pólya (1919)

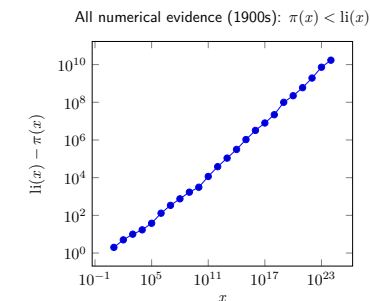
factor parity $m < n = 20$	
odd	even
19	$16 = 2^4$
$18 = 2 \cdot 3^2$	$15 = 3 \cdot 5$
17	$14 = 2 \cdot 7$
13	$10 = 2 \cdot 5$
$12 = 2^2 \cdot 3$	$9 = 3^2$
11	$6 = 2 \cdot 3$
$8 = 2^3$	$4 = 2^2$
7	
5	
3	
2	

Resolved (**false**) by C. Brian Haselgrove (1958).

Smallest  $n$  for which the conjecture fails:  
 $n = 906\,150\,257$  found by Minura Tanaka (1980).

## Why is theory important?

Let  $\pi(x)$  be the prime counting function and  $\text{li}(x) = \int_0^x \frac{dt}{\ln t}$ .



Skewes (1955): there must exist a value of  $x$  below

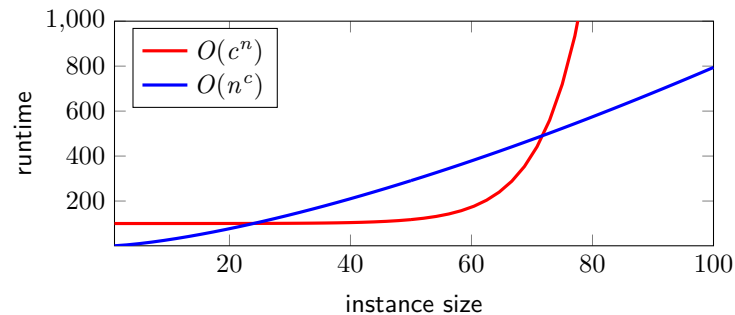
$$e^{e^{e^{7.705}}} < 10^{10^{10^{963}}}$$

**for which**  $\pi(x) > \text{li}(x)$ . Currently, explicit  $x$  is unknown, but the bounds are

$$10^{14} < x < e^{727.951346801}.$$

Furthermore, this occurs **infinitely often!**

## Why is theory important?



We want to make **rigorous, indisputable** arguments about the behavior of algorithms.

We want to understand how the behavior **generalizes** to any problem size.

## Design and analysis of algorithms

### Correctness

“does the algorithm always output the correct solution?”

### Complexity

“how many computational resources are required?”

## Design and analysis of algorithms

### Nature-inspired algorithms

- Random local search
- Metropolis algorithm, simulated annealing
- Evolutionary algorithms, genetic algorithms
- Ant colony optimization

**General-purpose** (randomized search heuristics) can be applied to any optimization problem

### Challenges:

- Unlike classical algorithms, they are not designed with their analysis in mind
- Behavior depends on a random number generator

## Convergence

**First question:** does the algorithm even find the solution?

### Definition.

Let  $f: S \rightarrow \mathbb{R}$  for a finite set  $S$ . Let  $S \supseteq S^* := \{x \in S : f(x) \text{ is optimal}\}$ . We say an algorithm *converges* if it finds an element of  $S^*$  with probability 1 and holds it forever after.

### Two conditions for convergence (Rudolph, 1998)

1. There is a **positive probability** to reach any point in the search space from any other point
2. The best solution is never lost (elitism)

**Does the (1+1) EA converge on every function  $f: \{0, 1\}^n \rightarrow \mathbb{R}$ ?**

**Does RLS converge on every function  $f: \{0, 1\}^n \rightarrow \mathbb{R}$ ?**

Can you think of how to modify RLS so that it converges?

## Runtime analysis

In most cases, randomized search heuristics visit the global optimum in finite time (or can be easily modified to do so)

**A far more important question:** how long does it take?

To characterize this unambiguously: count the number of “primitive steps” until a solution is visited for the first time (typically a function growing with the input size)

We typically use asymptotic notation to classify the growth of such functions.

## Runtime analysis

### Nature-inspired algorithms

- time to evaluate fitness function evaluation is much higher than the rest
- do not perform the same operations even if the input is the same
- do not output the same result if run twice

Given a function  $f: S \rightarrow \mathbb{R}$ , the runtime of some algorithm  $A$  applied to  $f$  is a random variable  $T_f$  that counts the number of calls  $A$  makes to  $f$  until an optimal solution is first generated.

We are interested in

- Estimating  $E(T_f)$ , the **expected runtime** of  $A$  on  $f$
- Estimating  $\Pr(T_f \leq t)$ , the **success probability** of  $A$  after  $t$  steps on  $f$

How to do this rigorously?

## Basic Definitions

- $\mathbb{N} = \{0, 1, 2, \dots\}$ .
- $(\Omega, P)$ : *discrete probability space*;
  - $\Omega$  is a countable set;
  - $P: \Omega \rightarrow [0, 1]$  a function such that  $\sum_{\omega \in \Omega} P(\omega) = 1$ .
- $\omega \in \Omega$ : *elementary events*.
- For  $\omega \in \Omega$ :  $P(\omega)$  is the *probability of  $\omega$* .
- $A \subseteq \Omega$ : *event*.
- For  $A \subseteq \Omega$ :  $P(A) = \sum_{a \in A} P(a)$ .

Example (rolling a die; generalizes to *uniform distribution*):

- $\Omega = \{1, 2, 3, 4, 5, 6\}$ .
- For all  $\omega \in \Omega$ ,  $P(\omega) = 1/6$ .

Another example (flipping a coin until we see tails):

- $\Omega = \mathbb{N}$ .
- For all  $n \in \mathbb{N}$ ,  $P(n) = 2^{-n-1}$ .

Note that  $\sum_{n \in \mathbb{N}} 2^{-n-1} = 1$  (*geometric sum*).

## Some Laws

For all events  $A, B \subseteq \Omega$ :

1. Recall:  $P(A) = \sum_{a \in A} P(a)$ .
2. If  $A \subseteq B$ , then  $P(A) \leq P(B)$ .
3.  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ .
4.  $P(\Omega \setminus A) = 1 - P(A)$ .
5. For all sequences  $(A_i)_i$  of events,  $P(\bigcup_i A_i) \leq \sum_i P(A_i)$ .

The last item is called the *union bound*.

## Random Variables

- Mapping  $X : \Omega \rightarrow \mathbb{R}$ : *random variable*.

Example (rolling two dice):

- $\Omega = \{1, 2, 3, 4, 5, 6\}^2$ ,  $P$  uniform.
- For  $a, b \in \{1, 2, 3, 4, 5, 6\}$ ,  $X(a, b) = a + b$ .
- “ $X = 12$ ” and similar things are now events ( $= \{\omega \in \Omega \mid X(\omega) = 12\}$ ).

Exercise: What are the following probabilities?

$$P(X = 12) \quad P(X = 11) \quad P(X = 0) \quad P(X \geq 7)$$

Another example (rolling two dice):

- Same  $(\Omega, P)$ .
- For all  $a, b \in \{1, 2, 3, 4, 5, 6\}$ ,  $X(a, b) = a$  and  $Y(a, b) = b$ .
- “ $X = Y$ ” and similar things are now events ( $= \{\omega \in \Omega \mid X(\omega) = Y(\omega)\}$ ).

Exercise: What are the following probabilities?

$$P(X = 3) \quad P(X = Y) \quad P(X = Y + 2) \quad P(X \geq Y)$$

## Properties of Random Variables

- $X, Y$  *identically distributed* ( $X \sim Y$ ): for all  $r \in \mathbb{R}$ ,  $P(X = r) = P(Y = r)$ .
- $X, Y$  *identical* ( $X = Y$ ): implies  $P(X = Y) = 1$ .
- $X, Y$  *independent*: for all sets  $A, B \subseteq \mathbb{R}$  we have

$$P(X \in A \text{ and } Y \in B) = P(X \in A) \cdot P(Y \in B).$$

- $(X_i)_i$  *independent*: for all sequences  $(A_i)_i$  of subsets of real numbers

$$P\left(\bigwedge_i X_i \in A_i\right) = \prod_i P(X_i \in A_i).$$

- $X, Y$  *i.i.d.*: identically distributed and independent.
- Extends to sequences of random variables.
- When  $X, Y$  are r.v.,  $X + Y$  is r.v.: for  $\omega \in \Omega$ ,  $(X + Y)(\omega) = X(\omega) + Y(\omega)$ .
- When  $X$  is r.v. and  $r \in \mathbb{R}$ ,  $rX$  is r.v.:  $(rX)(\omega) = rX(\omega)$ .

## Expected Value

The *expected value* of a random variable  $X : \Omega \rightarrow \mathbb{R}$  is defined as

$$E(X) = \sum_{\omega \in \Omega} P(\omega) \cdot X(\omega).$$

$$\begin{aligned} E(X) &= \sum_{\omega \in \Omega} P(\omega) \cdot X(\omega) \\ &= \sum_{r \in \mathbb{R}} \sum_{\omega: X(\omega)=r} P(\omega) \cdot r \\ &= \sum_{r \in \mathbb{R}} r \cdot \sum_{\omega: X(\omega)=r} P(\omega) \\ &= \sum_{r \in \mathbb{R}} r \cdot P(X = r). \end{aligned}$$

Exercise (*linearity* of  $E$ ):

1.  $E(X + Y) = E(X) + E(Y)$ ;
2.  $E(rX) = rE(X)$ .

## Some Theorems

### Products of independent random variables

Let  $X, Y$  be independent random variables. We have  $E(XY) = E(X)E(Y)$ .

### Markov's Inequality

Let  $X$  be a random variable with  $P(X < 0) = 0$ . For all  $a > 0$  we have

$$P(X \geq a) \leq \frac{E(X)}{a}.$$

## Variance

$$\text{Var}(X) = E((X - E(X))^2).$$

### Variance

Let  $X, Y$  be independent random variables. We have

1.  $\text{Var}(X) = E(X^2) - E(X)^2$ ; and
2.  $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$ .

## Binomial Distribution

- $X$  has **Bernoulli distribution** with parameter  $p \in [0, 1]$ :  $P(X = 1) = p$  and  $P(X = 0) = 1 - p$ .
- $(X_i)_{i \leq n}$  i.i.d. Bernoulli-distributed with parameter  $p$ :  $\sum_{i=1}^n X_i$  is a **Binomial distribution** with parameters  $n$  and  $p$  ( $B(n, p)$ ).
- $X$  has **geometric distribution** with parameter  $p \in (0, 1]$  if:

$$\forall k \geq 0 : P(X = k) = (1 - p)^k p.$$

## Runtime analysis

### RANDOMSEARCH

```
Choose  $x$  uniformly at random from  $S$ ;  
while stopping criterion not met do  
  Choose  $y$  uniformly at random from  $S$ ;  
  if  $f(y) \geq f(x)$  then  $x \leftarrow y$ ;  
end
```

### We now have the tools to analyze this!

Suppose w.l.o.g., there is a unique maximum solution  $x^* \in S$  (if there are more, it can only be faster).

Consider a run of the algorithm  $(x^{(0)}, x^{(1)}, \dots)$  where  $x^{(t)}$  is the solution generated in the  $t$ -th iteration.

## Runtime analysis

Define the random variable  $X_t$  for  $t \in \mathbb{N}_0$  as

$$X_t = \begin{cases} 1 & \text{if } x^{(t)} = x^*, \\ 0 & \text{otherwise;} \end{cases}$$

So  $X_t$  has a **Bernoulli distribution** with parameter  $p = 1/|S|$ .

Let  $T$  be the smallest  $t$  for which  $X_t = 1$ .

Then  $T$  is a **geometrically distributed** random variable.

**Expected runtime:**  $E(T) = 1/p = |S|$

## Runtime analysis

**Success probability:**  $\Pr(T \leq k) = 1 - (1 - p)^k$

For example,

$$\Pr(T \leq |S|) = 1 - (1 - 1/|S|)^{|S|} \geq 1 - 1/e \approx 0.6321$$

Constant chance that it takes  $|S|$  steps to find the solution.

Let  $S = \{0, 1\}^n$ . Let's bound the success probability before  $2^{\epsilon n}$  for some constant  $0 < \epsilon < 1$ .

$$\begin{aligned}\Pr(T \leq 2^{\epsilon n}) &= 1 - (1 - 2^{-n})^{2^{\epsilon n}} \leq 1 - \underbrace{(1 - 2^{-n})^{2^{\epsilon n}}}_{\text{see HW 2, Exercise 2a}} \\ &= 2^{-n(1-\epsilon)} = 2^{-\Theta(n)}\end{aligned}$$

So the probability that random search is successful before  $2^{\Theta(n)}$  steps is vanishing quickly (faster than every polynomial) as  $n$  grows.

## Runtime analysis

Let's consider more interesting cases. . .

Recall from Project 1:

### (1+1) EA

Choose  $x$  uniformly at random from  $\{0, 1\}^n$ ;

**while** *stopping criterion not met* **do**

$y \leftarrow x$ ;

**foreach**  $i \in \{1, \dots, n\}$  **do**

        | With probability  $1/n$ ,  $y_i \leftarrow (1 - y_i)$ ;

**end**

**if**  $f(y) \geq f(x)$  **then**  $x \leftarrow y$ ;

**end**

In each iterations, how many bits flip in expectation?

What is the probability exactly one bit flips?

What is the probability exactly two bits flip?

What is the probability that no bits flip?

## Runtime Analysis

### Theorem (Droste et al., 2002)

The expected runtime of the (1+1) EA for an arbitrary function  $f: \{0, 1\}^n \rightarrow \mathbb{R}$  is  $O(n^n)$ .

#### Proof.

Without loss of generality, suppose  $x^*$  is the unique optimum and  $x$  is the current solution.

Let  $k = |\{i : x_i \neq x_i^*\}|$ .

Each bit flips (resp., does not flip) with probability  $1/n$  (resp., with probability  $1 - 1/n$ ).

## Runtime Analysis

In order to reach the global optimum **in the next step** the algorithm has to mutate the  $k$  bits and leave the  $n - k$  bits alone.

The probability to create the global optimum in the next step is

$$\left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k} \geq \left(\frac{1}{n}\right)^n = n^{-n}.$$

Assuming the process has not already generated the optimal solution, in expectation we wait  $O(n^n)$  steps until this happens. ■

**Note:** we are simply overestimating the time to find the optimal *for any arbitrary pseudo-Boolean function*.

**Note:** The upper bound is *worse* than for RANDOMSEARCH. In fact, there are functions where RANDOMSEARCH is *guaranteed* to perform better than the (1+1) EA.

## Initialization

**Recall from Project 1:**  $\text{ONEMAX}: \{0, 1\}^n \rightarrow \mathbb{R}, x \mapsto |x|$ ;

**How good is the initial solution?**

Let  $X$  count the number of 1-bits in the initial solution.  $E(X) = n/2$ .

**How likely to get **exactly**  $n/2$ ?**

$$\Pr(X = n/2) = \binom{n}{n/2} \frac{1}{2^{n/2}} \left(1 - \frac{1}{n}\right)^{n/2}$$

For  $n = 100$ ,  $\Pr(X = 50) \approx 0.0796$

## Initialization (Tail Inequalities)

**How likely is the initial solution no worse than  $(3/4)n$ ?**

### Markov's Inequality

Let  $X$  be a random variable with  $P(X < 0) = 0$ . For all  $a > 0$  we have

$$\Pr(X \geq a) \leq \frac{E(X)}{a}.$$

$$E(X) = n/2; \text{ then } \Pr(X \geq (3/4)n) \leq \frac{E(X)}{(3/4)n} \leq 2/3$$

## Initialization (Tail Inequalities)

Let  $X_1, X_2, \dots, X_n$  be independent Poisson trials each with probability  $p_i$ ;  
For  $X = \sum_{i=1}^n X_i$ , the expectation is  $E(X) = \sum_{i=1}^n p_i$ .

### Chernoff Bounds

- for  $0 \leq \delta \leq 1$ ,  $\Pr(X \leq (1 - \delta)E(X)) \leq e^{\frac{-E(X)\delta^2}{2}}$ .
- for  $\delta > 0$ ,  $\Pr(X > (1 + \delta)E(X)) \leq \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^{E(X)}$ .

E.g.,  $p_i = 1/2$ ,  $E(X) = n/2$ , fix  $\delta = 1/2 \rightarrow (1 + \delta)E(X) = (3/4)n$ ,

$$\Pr(X > (3/4)n) \leq \left(\frac{e^{1/2}}{(3/2)^{(3/2)}}\right)^{n/2} = c^{-n/2}.$$

## Initialization (Tail Inequalities):

A simple example

Let  $n = 100$ . How likely is the initial solution no worse than  $\text{ONEMAX}(x) = 75$ ?

$\Pr(X_i) = 1/2$  and  $E(X) = 100/2 = 50$ .

**Markov:**  $\Pr(X \geq 75) \leq \frac{50}{75} = \frac{2}{3}$ .

**Chernoff:**  $\Pr(X \geq (1 + 1/2)50) \leq \left(\frac{\sqrt{e}}{(3/2)^{(3/2)}}\right)^{50} < 0.0054$ .

**In reality,**  $\Pr(X \geq 75) = \sum_{i=75}^{100} \binom{100}{i} 2^{-100} \approx 0.0000002818141$ .