# Theory of Evolutionary Algorithms

1) History
2) (1+1) EA $\approx$ (1+1) ES
3) LINEAR PSEUDO-BOOLEAN FUNCTIONS
3.1) FITNESS-BASED PARTITIONS
3.2) THE COUPON COLLECTOR PROBLEM
3.2) EXP. MULTIPLICATIVE DISTANCE DECREASE

Exp. time to find the optimum

ONEMAX PROBLEM

---

## History

"DARK AGES" $\longrightarrow$ (ORIGINAL) SCHEMA THEORY

$$1011 \longrightarrow \begin{array}{l} 1*1* \\ 10** \\ ***1 \\ \vdots \end{array}$$

SCHEMATA

A schema is a string on $\{0,1,*\}^n$.

An individual $x \in \{0,1\}^n$ contains the schema $s \in \{0,1,*\}^n$ when $\forall i \in \{1,\dots,n\}$ either $x_i = s_i$ or $s_i = *$.

The lenght of a schema is the maximum distance between two symbols different from "*" plus one.

$\underline{10}**$ 2 $\quad \underline{1**0}$ 4

The order of a schema is the number of symbols different from "*".

THE PROBABILITY OF BREAKING THE SCHEMA IS HIGHER THE HIGHER IS THE ORDER

## Informal results

Longer length → Higher probability of being "broken" by crossover.

Higher order → Higher probability of being "broken" by mutation.

the fitness of a schema S is the average fitness of all individuals containing that schema.

## Schema theorem (Informal)

Short, low-order schemata with a high fitness are preserved by GA.

# EA as MARKOV PROCESSES

"will the EA converge to some stationary distribution where all populations containing the optimum have prob. 1?"

$P(s_i | s_j)$ — if we have mutation this is positive

↳ PROBABILITY OF POPULATION $i$ GIVEN A POPULATION $J$

$M$ a transition matrix between populations

where $M_{i,J}$ is $P(s_i | s_J)$ has all entries that are positive

STANDARD DISTRIBUTION

only st. dist. is the solution of $x = Mx$

**BUT** all populations have positive probability

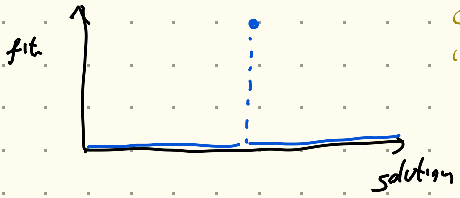
**BUT** if you add elitism...

... you will converge to a pop- containing the optimum.

# RUNTIME ANALYSIS

1) AN EVOLUTIONARY ALGORITHM $A$
2) A PROBLEM OR SET OF PROBLEMS $P$

[What is the expected time for $A$ to reach the optimum for any problem in $P$.]




CAN YOU EXPECT THAT RANDOM SEARCH IS A BETTER CHOICE?

## (1+1) EA

SIMPLE EA THAT YOU CAN THINK

$x \leftarrow \text{random}\{0,1\}^n$
forever {
  OFFSPRING $\leftarrow$ mutate($x$)
  $x \leftarrow$ FITTEST {$x$, offspring}
}

bit flip mutation with prob. $1/n$

# LINEAR PSEUDO-BOOLEAN FUNCTIONS

$f : \{0,1\}^n \longrightarrow \mathbb{R}$ pseudo-Boolean

$x = (x_1, \ldots, x_n) \in \{0,1\}^n$

$f(x) = w_1 x_1 + \ldots + w_n x_n = \sum_{i=1}^{n} w_i x_i$ | linear

with $w_i \in \mathbb{N}$, $w_i \neq 0$

the global optimum is always $\vec{1} = \underbrace{(1, \ldots, 1)}_{n}$

which has fitness $f(\vec{1}) = \sum_{i=1}^{n} w_i$

if $w_i = 1 \ \forall i$ then we have ONEMAX

$f(x) =$ number of 1s in $x$

if $w_1 = 2^0$, $w_2 = 2^1$, $\ldots$, $w_i = 2^{i-1}$ then we have

$f(x) =$ number represented in binary by $x$

## ONEMAX

# Fitness-based partitions

$$S = \{0,1\}^n$$

Take a partition $A_1, \dots, A_m$ of $S$

UNION OF ALL $A_i$ IS $S$

i.e., $\bigcup_{i=1}^{m} A_i = S \qquad \forall i,j \; i \neq j \quad A_i \cap A_j = \emptyset$

YOU CAN SAY TO HAVE A FITNESS BASE PARTITION, WHEN YOU CAN DEFINE:

$$A_i <_f A_j \qquad \forall a \in A_i \; \forall b \in A_j \quad f(a) < f(b)$$

ALL ELEMENTS IN $A_i$ HAVE A FITNESS LOWER THAN $A_J$

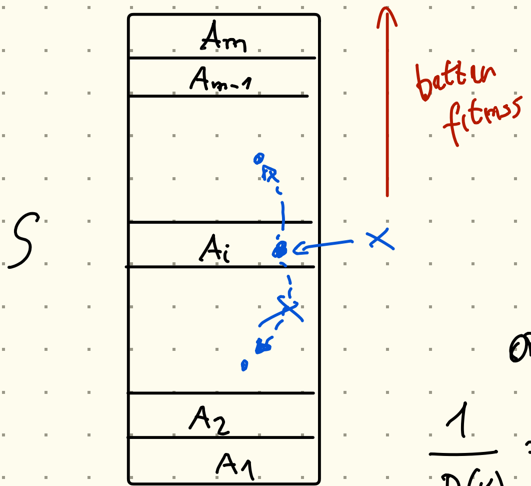
WITH RESPECT TO THE FITNESS

if we have

$$A_1 <_f A_2 <_f \dots <_f A_m$$

with $A_m$ containing <u>only</u> the global optima

then $A_1, \dots, A_m$ is a fitness-based partition




$x \in \{0,1\}^n$

$p(x) =$ prob. of "jumping" to a higher fitness level

OFFSPRING in $A_{i+1} \cup \dots \cup A_m$

$\dfrac{1}{p(x)}$ = expected time to jump to a higher fitness level

$$p_i = \min_{a \in A_i} p(a) \quad \Big\}\ \text{remove dependence from } x$$

$\frac{1}{p_i}$ = upper bound on the exp. time for a solution in $A_i$ to move to a higher fitness level.

worst case: we always jump from $A_i$ to $A_{i+1}$

exp. time bounded above by: $\displaystyle\sum_{i=1}^{m-1} \frac{1}{p_i}$

we only need to find these probabilities.

## The onemax case

$w_i = 1 \qquad f(x) = \sum_{i=1}^{n} x_i$

PARTITION

n+1 sets

$$\left\{\begin{array}{ll} A_0 = \{\vec{0}\} & 0 \\ A_1 = \{00\ldots01, 00\ldots010, \ldots\} & 1 \\ & \vdots \\ A_n = \{\vec{1}\} & n \end{array}\right.$$

(FITNESS column: 0, 1, …, n)

$x \in A_{n-k}$ if $k$ bits of $x$ are set to 0.

**Theorem** The exp. opt. time for a (1+1) EA on the ONEMAX problem is

$$\Theta(n \log n)$$

---

**Proof**

prob. of moving from $A_{n-k}$ to $A_{n-k+1}$

this prob. is $\leq P_{n-k}$

→ prob of flipping one bit from 0 to 1

→ prob. of not flipping any other bit

$$\underbrace{\frac{k}{n}\left(1-\frac{1}{n}\right)^{n-1}}_{\text{exactly one bit was flipped from 0 to 1}} \geq \frac{k}{e \cdot n}$$

$$P_{n-k} \geq \frac{k}{e \cdot n} \rightsquigarrow \overbrace{\frac{1}{P_{n-k}}}^{\text{upper bound on the exp. "waiting" time}} \leq \frac{en}{k}$$

$$\sum_{k=1}^{n} \frac{en}{k} = en \sum_{k=1}^{n} \frac{1}{k} = \Theta(n \log n)$$

∎

Can the <u>real</u> exp. opt. time be below $O(n \log n)$? <u>NO</u> WHY? 1°: WE NEED AT LEAST LINEAR TO FLIP ALL BITS 2°. ...

# Coupon Collector Problem

**Theorem** The exp. opt. time for (1+1) EA on every linear pseudo-Boolean function is $\Omega(n \log n)$.

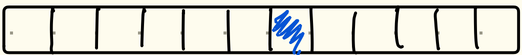
↳ bound from below

## Chernoff Bound

$\frac{n}{2}$ bits set to 1 on expectation

$\frac{2}{3}n$ bits set to 1 has prob. $1-e^{-\Omega(n)}$

$\Rightarrow$ with prob. $1-e^{-\Omega(n)}$ at least $\frac{n}{3}$ of the bits are set to 0.



prob. that after $t$ time steps this position has been flipped

the bit has been flipped at least once.

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

bit not flipped

$$\underbrace{1-\overbrace{\left(1-\left(1-\frac{1}{n}\right)^t\right)^{n/3}}^{\text{we flipped all zero bits at least once}}}_{\text{at least one of the zero bits is still zero}}$$

we select $t = (n-1)\log_2 n$

$$1 - \left(1 - \left(1 - \frac{1}{n}\right)^{(n-1)\log_2 n}\right)^{n/3} \geq$$ MAGIC HERE

$$\geq 1 - e^{-1/3}$$

$$\left(1 - e^{-\Omega(n)}\right)\left(1 - e^{-1/3}\right) \geq 1 - e^{-\Omega(n)} - e^{-1/3} = \Omega(1)$$

thus $\Omega(n \log n)$ is the bound for linear pseudo-Boolean functions ∎

# (1+1) EA on Linear Pseudo-Boolean functions

$\hookrightarrow$ ONEMAX $\to O(n \log n)$

$\hookrightarrow \Omega(n \log n)$

$\Downarrow$

$\Theta(n \log n)$

it holds for all linear p.B. functions

$f: \{0,1\}^n \longrightarrow \mathbb{R}$

$\sim$ linear number of fitness levels ✓

STILL $\Theta(n \log n)$

$f_{bin}: \{0,1\}^n \longrightarrow \mathbb{R} \qquad w_i = 2^{i-1}$

$w_1 = 1 \quad w_2 = 2 \quad w_3 = 4$

$$f_{bin}(\vec{x}) = \sum_{i=1}^{n} 2^{i-1} x_i$$

binary repr. of $x$

WHAT IS THIS?

$p_i$ = prob. of "jumping" to a higher fitness level

$$\left.\begin{array}{ll} 111 & 7 \\ \vdots & \\ 010 & 2 \\ 001 & 1 \\ 000 & 0 \end{array}\right\} 2^n \text{ fitness levels}$$

$$\sum_{i=1}^{m} \frac{1}{p_i}$$

$$\sum_{i=1}^{m} \frac{1}{\underbrace{p_i}_{\text{lucky case of } p_i = 1}} = \sum_{i=1}^{m} 1 = m = 2^n$$

$O(2^n)$

$\Rightarrow$ with a different technique we can find

$$O(n(\log n + \log \underbrace{w_{max}}_{\text{maximum weight}}))$$

# Expected Multiplicative Distance Decrease

$$s \xrightarrow{op} s' \xrightarrow{op} s'' \xrightarrow{op} \dots \xrightarrow{op} s_{opt}$$

distance in fitness $\frac{\text{TOT\_DIST}}{\text{N\_STEP}}$ → ON AVERAGE

THE IDEA IS THAT IF WE FIND A SUBSET OF POSSIBLE MUTATION THAT HAPPENS WITH A CERTAIN PROB. AND PERFORM ALL OF THEM REACHING THE OPTIMUM, WE CAN IMPROVE THE BOUND

$O = \{o_1, o_2, \dots, o_n\}$ set of operations

- each operation has the same probability of being applied.

$$O = \{\text{flip bit } 1, \; \text{flip bit } 2, \; \vdots \; \text{flip bit } n\}$$

$$\boxed{\underbrace{\frac{1}{n}}_{\text{FLIPPING A BIT}} \underbrace{\left(1 - \frac{1}{n}\right)^{n-1}}_{\text{NOT FLIPPING THE OTHERS}}}$$

probability for each operation in $O$

$\forall s \in \{0,1\}^n$ there exists $O' \subseteq O$ such that by performing all operations in $O'$ we reach $s_{opt}$.

$$\begin{array}{cccccc} & 2^0 & 2^1 & 2^2 & 2^3 & 2^4 \\ s = & 0 & 1 & 0 & 0 & 1 \\ & 1 & 2 & 3 & 4 & 5 \end{array}$$

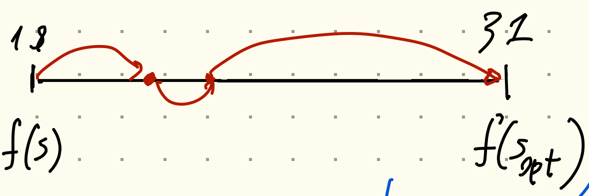
$O' \subseteq O$

$O' = \{$ flip bit 1, flip bit 3, flip bit 4 $\}$

$f(s) = 2^1 + 2^4 = 2 + 16 = 18$

$f(s_{opt}) = 31$

$f(s_{opt}) - f(s) = 31 - 18 = 13$




fitness distance

$$\frac{f(s_{opt}) - f(s)}{n}$$

expected decrease in distance for each operation in $O$.

number of elements in $O$

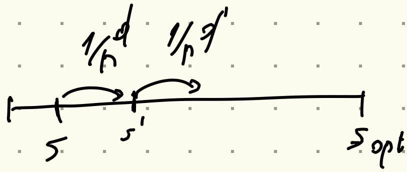
Expected distance from $s_{opt}$ of $s'$ (an offspring of $s$)

$$\left(1 - \frac{1}{n}\right) d = \left(1 - \frac{1}{n}\right)\left(f(s_{opt}) - f(s)\right)$$

$$\left(1-\frac{1}{n}\right)\left(f(s_{opt}) - \underbrace{f(s)}_{\text{dependency from the starting point.}}\right)$$

$$d_{max} = \max_{s \in \{0,1\}^n} \left(f(s_{opt}) - f(s)\right)$$

$$\left(1-\frac{1}{n}\right) d \leq \left(1-\frac{1}{n}\right) d_{max}$$




after t steps

$$\Downarrow$$

$$\left(1-\frac{1}{n}\right)^t d_{max}$$

what happens when $\left(1-\frac{1}{n}\right)^t d_{max} \leq \frac{1}{2}$ ?

- the distance <u>must</u> be an integer
- the distance is 0 at least of the time.

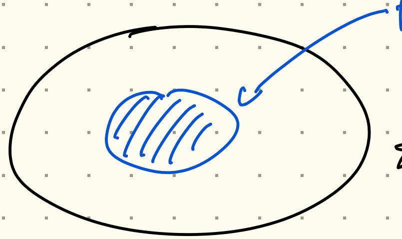
TODO: find a value of $t$ such that $\left(1-\frac{1}{n}\right)^t d_{max} \leq \frac{1}{2}$

$\Rightarrow$ minimum value for $t$ is

$$\underbrace{c}_{\text{constant}} \cdot n \cdot \log d_{max}$$

Expected number of operations from $O$ to reach the optimum is $O(n \cdot \log d_{max})$

(in our case $O(n \log d_{max})$).




$$\overbrace{\alpha}^{\text{prob. of appl. one mutation from } O}\ \underbrace{n}_{\text{number of operations in } O}$$

$\frac{1}{\alpha n} \leftarrow$ Exp. time to apply a mutation from $O$

$$O\left(\frac{1}{\alpha \cancel{n}} \cdot \cancel{n} \cdot \log d_{max}\right) = O\left(\alpha^{-1} \log d_{max}\right)$$

prob of flipping exactly one bit

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

bound for $\alpha$

$$O\left(\alpha^{-1} \log d_{max}\right) = O\left(e \cdot n \cdot \log d_{max}\right) =$$
$$= O\left(n \log d_{max}\right)$$

$$d_{max} = \sum_{i=1}^{n} w_i \leq \sum_{i=1}^{n} w_{max} = n \cdot w_{max}$$

$$w_{max} = \max_{i \in \{1,\dots,n\}} w_i$$

$$O(n \log d_{max}) = O(n (\log(n \cdot w_{max}))) =$$

$$= O(n(\log n + \log w_{max}))$$

the bound we wanted.

The exp. opt. time for (1+1) EA on any linear pseudo-Boolean function is

$$O(n(\log n + \log w_{max}))$$

ONEMAX: $w_{max} = 1$ $\quad O(n \log n)$

BINARY-VALUE: $w_{max} = 2^{n-1}$ $\quad O(n(\log n + \log 2^{n-1}))$

$$= O(n \log n + (n-1))$$

$$= O(n^2)$$