Heuristic Optimization

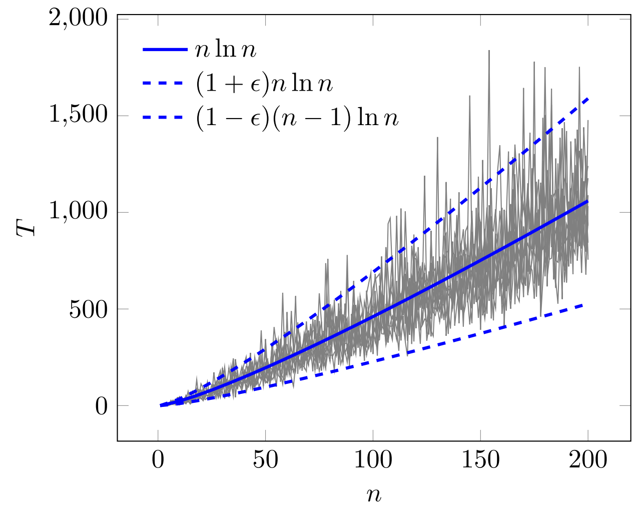
Lecture 6

Algorithm Engineering Group
Hasso Plattner Institute, University of Potsdam

19 May 2015

Runtime analysis – RLS on OneMax

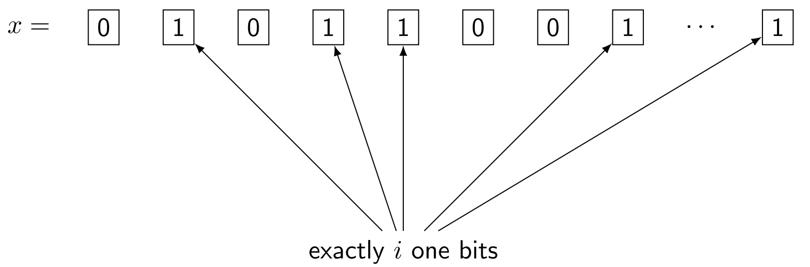
10 trials of $n \in \{1, \ldots, 200\}$.



We want to rigorously understand this behavior.

Runtime analysis – RLS on OneMax

Let's suppose: during the execution of RLS the current string x looks like this:



Let's look into

- p_i: probability that RLS makes an improving move from x
- T_i: time until RLS makes an improving move from x

Runtime analysis – RLS on OneMax

0	1	2	3	4	5		
0	0	0	0	0	0	$p_0 = \frac{6}{6}$	$E(T_0) = \frac{6}{6}$
0	0	0	0	0	1	$p_1 = \frac{5}{6}$	$E(T_1) = \frac{6}{5}$
0	0	1	0	0	1	$p_2 = \frac{4}{6}$	$E(T_2) = \frac{6}{4}$
1	0	1	0	0	1	$p_3 = \frac{3}{6}$	$E(T_3) = \frac{6}{3}$
1	0	1	0	1	1	$p_4 = \frac{2}{6}$	$E(T_4) = \frac{6}{2}$
1	1	1	0	1	1	$p_5 = \frac{1}{6}$	$E(T_5) = \frac{6}{1}$

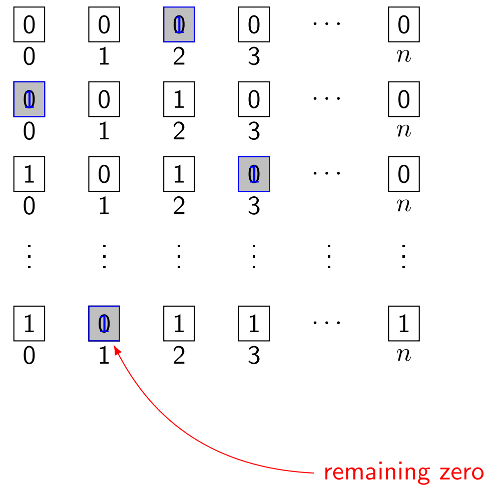
Runtime analysis – RLS on OneMax

Runtime

T is the random variable that counts the number of steps (function evaluations) taken by RLS until the optimum is generated.

$$\begin{aligned} E(T) &= E(T_0) + E(T_1) + \cdots + E(T_5) \\ &= 1/p_0 + 1/p_1 + \cdots + 1/p_5 \\ &= \sum_{i=0}^{5} \frac{1}{p_i} = \sum_{i=0}^{5} \frac{6}{i+1} = 6 \sum_{i=1}^{6} \frac{1}{i} = 6 \cdot 2.45 = 14.7 \end{aligned}$$

Runtime analysis – RLS on OneMax



$p_0 = \frac{n}{n}$ $\quad E(T_0) = \frac{n}{n}$

$p_1 = \frac{n-1}{n}$ $\quad E(T_1) = \frac{n}{n-1}$

$p_2 = \frac{n-2}{n}$ $\quad E(T_2) = \frac{n}{n-2}$

$\vdots$

$p_{n-1} = \frac{1}{n}$ $\quad E(T_{n-1}) = \frac{n}{1}$

Coupon collector process

Suppose there are n different kinds of coupons. We must collect all n coupons during a series of trials.

In each trial, exactly one of the n coupons is drawn, each one equally likely. We must keep drawing in each trial until we have collected each coupon at least once.

Starting with zero coupons, what is the exact number of trials needed before we have all n coupons?

Theorem (Coupon collector theorem)

Let T be the number of trials until all n coupons are collected. Then

$$\begin{aligned} E(T) &= \sum_{i=0}^{n-1} \frac{1}{p_{i+1}} = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=0}^{n-1} \frac{1}{i} \\ &= n \cdot H_n = n(\log n + \Theta(1)) = n \log n + O(n) \end{aligned}$$

Coupon collector process: concentration bounds

What is the probability that $T > n \ln n + O(n)$?

Theorem (Coupon collector upper bound)

Let T be the number of trials until all n coupons are collected. Then

$$\Pr(T \geq (1+\epsilon) n \ln n) \leq n^{-\epsilon}$$

Proof.

Probability of choosing a specific coupon: $1/n$.
Probability of not choosing a specific coupon: $1 - 1/n$.
Probability of not choosing a specific coupon for t rounds: $(1-1/n)^t$
Probability that one of the n coupons is not chosen in t rounds: $n \cdot (1-1/n)^t$ (union bound)
Let $t = cn \ln n$,

$$\Pr(T \geq cn \ln n) \leq n(1-1/n)^{cn \ln n} \leq ne^{-c \ln n} = n \cdot n^{-c} = n^{-c+1}$$

□

Coupon collector process: concentration bounds

Theorem (Coupon collector lower bound) (Doerr, 2011)

Let T be the number of trials until all n coupons are collected. Then

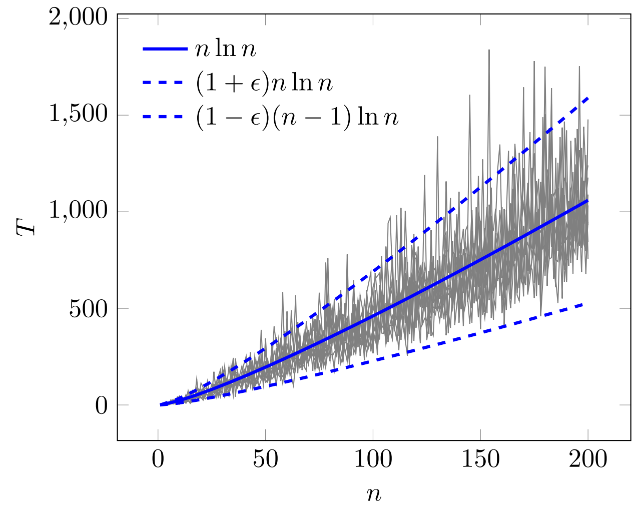
$$\Pr(T < (1-\epsilon)(n-1)\ln n) \leq e^{-n^{\epsilon}}$$

Corollary

Let T be the time for RLS to optimize OneMax. Then,

$$E(T) = \Theta(n \log n)$$
$$\Pr(T \geq (1+\epsilon)n \ln n) \leq n^{-\epsilon}$$
$$\Pr(T < (1-\epsilon)(n-1)\ln n) \leq e^{-n^{-\epsilon}}$$

Runtime analysis – RLS on OneMax



What about **(1+1) EA**? Can we use Coupon Collector? Why/why not?

Fitness levels

Observation: fitness during optimization is always monotone increasing

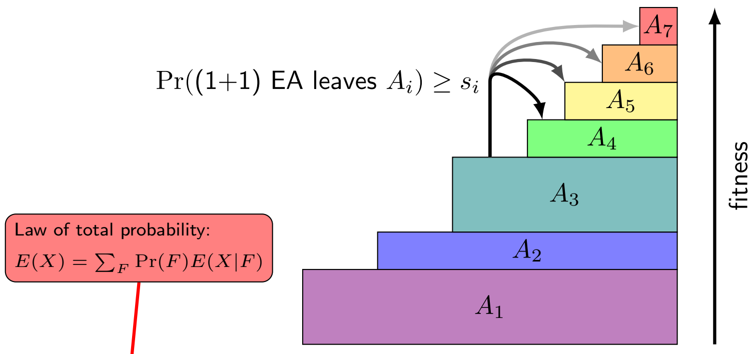
Idea: partition the search space $\{0,1\}^n$ into m sets $A_1, \ldots A_m$ such that

1. $\forall i \neq j: \quad A_i \cap A_j = \emptyset$
2. $\bigcup_{i=1}^{m} A_i = \{0,1\}^n$
3. for all points $a \in A_i$ and $b \in A_j$, $f(a) < f(b)$ if $i < j$

We require A_m to contain only optimal search points

Procedure: for each level A_i, bound the probability of leaving a level A_i for a higher level A_j, $j > i$.

Fitness levels



- $p(A_i)$ be the probability that a random chosen point belongs to A_i
- s_i be the probability to leave level A_i for level A_j with $j > i$

$$E(T) \leq \sum_{i=1}^{m-1} p(A_i) \cdot \left(\frac{1}{s_i} + \cdots + \frac{1}{s_{m-1}}\right) \leq \left(\frac{1}{s_1} + \cdots + \frac{1}{s_{m-1}}\right) = \sum_{i=1}^{m-1} \frac{1}{s_i}$$

Figure adapted from D. Sudholt, Tutorial 2011

Runtime analysis – (1+1) EA on OneMax

Theorem

The expected runtime of the (1+1) EA on OneMax is $O(n \log n)$.

Proof

We partition $\{0,1\}^n$ into disjoint sets $A_0, A_1, \ldots, A_n$ where x is in A_i if and only if it has i zeros ($n-i$ ones).

To escape A_i, it suffices to flip a single zero and leave all other bits unchanged.

Thus, $s_i \geq \frac{i}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{i}{en}$, and $\frac{1}{s_i} \leq \frac{en}{i}$.

We conclude

$$E(T) \leq \sum_{i=1}^{m-1} \frac{1}{s_i} \leq \sum_{i=1}^{n} \frac{en}{i} = en \cdot H_n = O(n \log n).$$

□

Runtime analysis – (1+1) EA on OneMax

This gives only an upper bound. Maybe the (1+1) EA can be much quicker. For example it could be $O(n)$ or even something like $O(n \log \log n)$.

Runtime analysis – (1+1) EA on OneMax

Theorem (Droste, Jansen, Wegener 2002)

The expected runtime of the (1+1) EA on OneMax is $\Omega(n \log n)$.

Lemma

The probability that the (1+1) EA needs at least $(n-1)\ln n$ steps is at least a constant c.

Runtime analysis – (1+1) EA on OneMax

Proof of Lemma.

The initial solution has **at most** $n/2$ one bits with probability **at least** $1/2$.

There is a constant probability that in $(n-1)\ln n$ steps one of the remaining zero bits does not flip:

- Probability a particular bit doesn't flip in t steps: $(1-1/n)^t$
- Probability it flips **at least once** in t steps: $1-(1-1/n)^t$
- Probability $n/2$ bits flip at least once in t steps: $(1-(1-1/n)^t)^{n/2}$
- Probability at least one of the $n/2$ bits does not flip in t steps: $1-[1-(1-1/n)^t]^{n/2}$.

Set $t=(n-1)\ln n$. Then

$$\begin{aligned} 1-[1-(1-1/n)^t]^{n/2} &= 1-[1-(1-1/n)^{(n-1)\ln n}]^{n/2} \\ &\geq 1-[1-(1/e)^{\ln n}]^{n/2} \\ &= 1-[1-1/n]^{n/2} \\ &= 1-[1-1/n]^{n\cdot 1/2} \geq (1-(2e))^{-1/2} = c. \end{aligned}$$

□

Runtime analysis – (1+1) EA on OneMax

Theorem (Droste, Jansen, Wegener 2002)

The expected runtime of the (1+1) EA on OneMax is $\Omega(n \log n)$.

Proof

Expected runtime:

$$E(T) = \sum_{t=1}^{\infty} t \Pr(T = t) \geq (n-1)\ln n \cdot \Pr(T \geq (n-1)\ln n)$$
$$\geq (n-1)\ln n \cdot c = \Omega(n \log n).$$

by previous lemma

□

Upper bound given by fitness levels is tight.

Fitness levels

There are several more advanced results that use the fitness levels technique:

Expected runtime of the (1+λ) EA on LeadingOnes is $O(\lambda n + n^2)$ (Jansen et al., 2005)

Expected runtime of the (μ+1) EA on LeadingOnes is $O(\mu n \log n + n^2)$ (Witt, 2006)

Fitness levels for proving lower bounds (Sudholt, 2010).

Non-elitist populations (Lehre, 2011).

Drift Analysis

Consider a process moving towards/away from a goal (possibly stochastically).

Model this as a sequence of numbers $X_0, X_1, \ldots$ where

$$X_t := \text{ distance from the goal at time } t.$$

$(10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0)$ ← $E(X_t - X_{t+1}) = \begin{cases} 0 & \text{if } X_t = 0 \\ 1 & \text{otherwise} \end{cases}$

$(10, 9, 8, 9, 8, 7, 6, 5, 4, 5, \ldots, 0)$ ← ???

Definition

The drift of a process at time t is the expected decrease in distance from a goal:

$$E(X_t - X_{t+1})$$

Drift analysis allows us to relate the *drift* to the *time to reach the goal*.

Drift Analysis – Deterministic Process

Consider a process that moves as follows. In each step,

- With probability 1, move one step toward the goal.

Starting at distance n, how many steps until the goal is reached? n

Drift is $E(X_t - X_{t+1}) = 1$ as long as $X_t > 0$.

Expected time to reach the goal:

$$E(T) = \frac{\textit{maximum distance}}{\textit{drift}} = \frac{n}{1} = n.$$

Drift Analysis – Stochastic Process

Consider a process that moves as follows:

- with probability $3/5$, move one step toward the goal,
- with probability $2/5$, move one step *away* from the goal.

Starting at distance n, how many steps until the goal is reached? $5n$

$$X_t - X_{t+1} = \begin{cases} 0 & \text{if } X_t = 0, \\ 1 & \text{if } X_t \neq 0, \text{ with probability } 3/5, \\ -1 & \text{if } X_t \neq 0, \text{ with probability } 2/5, \end{cases}$$

Drift is

$$E(X_t - X_{t+1}) = \frac{3}{5} \cdot 1 + \frac{2}{5} \cdot (-1) = \frac{3-2}{5} = \frac{1}{5}.$$

Expected time to reach the goal:

$$E(T) = \frac{\textit{maximum distance}}{\textit{drift}} = \frac{n}{1/5} = 5n.$$

Drift Analysis

Theorem (He and Yao, 2001)

Let $\{X_t : t \geq 0\}$ be a Markov process over $\mathbb{R}_0^+$. Let $T := \min\{t \geq 0 : X_t = 0\}$. If there exists $\delta > 0$ such that at any time step $t \geq 0$ and at any state $X_t > 0$, the following condition holds:

$$E(X_t - X_{t+1} \mid X_t > 0) \geq \delta,$$

then

$$E(T \mid X_0 > 0) \leq \frac{X_0}{\delta} \text{ and } E(T) \leq \frac{E(X_0)}{\delta}$$

Example: (1+1) EA on OneMax:

$$E(X_t - X_{t+1} \mid X_t > 0) \geq 1 \cdot \frac{i}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{i}{en} \geq \frac{1}{en} = \delta$$

$$E(T \mid X_0 > 0) \leq \frac{E(X_0)}{\delta} \leq \frac{n/2}{1/(en)} = O(n^2).$$

Obviously not tight!

Drift Analysis

Observation: we don't have to use the distance directly!

Idea: progress toward goal depends on distance from goal. We can use a potential function.

Let $X_t = \ln(i+1)$ where i is the number of zeros in the bitstring.

$$E(X_t - X_{t+1} \mid X_t > 0) \geq \ln(i+1) \cdot \frac{i}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{\ln(i+1)}{en} \geq \frac{\ln(2)}{en} = \delta$$

$$E(T \mid X_0 > 0) \leq \frac{X_0}{\delta} \leq \frac{\ln(n+1)}{\ln(2)/en} = O(n \log n).$$

Drift Analysis

Drift analysis has many powerful variants:

- Multiplicative Drift (Doerr et al., 2010)
- Negative Drift (Oliveto and Witt, 2011)
- Drift Analysis for Stochastic Populations (Lehre, 2010)
- Variable Drift (Johannsen 2010)

Refinements allow for

- Upper and lower bounds on expectation
- Tail inequalities

Further reading

Pietro Oliveto and Xin Yao. *A Gentle Introduction to the Time Complexity Analysis of Evolutionary Algorithms:*[2]
http://www.cs.bham.ac.uk/~olivetps/images/Oliveto2012Tutorial.pdf

Frank Neumann and Carsten Witt, *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity.* Natural Computing Series, Springer, 2010.
http://www.bioinspiredcomputation.com/

Anne Auger and Benjamin Doerr (editors). *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific, 2011.

Thomas Jansen, *Analyzing Evolutionary Algorithms. The Computer Science Perspective.* Springer, 2013.

[2] Lectures 5&6 are based in part on these slides (with permission).

References

Benjamin Doerr, "Analyzing Randomized Search Heuristics: Tools from Probability Theory." Chapter 1 of *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific, 2011.

Benjamin Doerr, Daniel Johannsen and Carola Winzen (2010). "Multiplicative drift analysis." In *Proceedings of the Twelfth Annual Conference on Genetic and Evolutionary Computation*, pages 1449–1456. ACM.

Stefan Droste, Thomas Jansen and Ingo Wegener (2002). "On the analysis of the (1+1) evolutionary algorithm." *Theoretical Computer Science*, 276(1-2):51–81.

Thomas Jansen, Ken A. De Jong and Ingo Wegener (2005). "On the choice of the offspring population size in evolutionary algorithms." *Evolutionary Computation*, 13(4):413–440.

Daniel Johannsen (2010). "Random Combinatorial Structures and Randomized Search Heuristics." *PhD thesis*, Universität des Saarlandes.

Per Kristian Lehre (2010). "Negative drift in populations." In *Proceedings of the Eleventh International Conference on Parallel Problem Solving from Nature*, pages 244–253.

Per Kristian Lehre (2011). "Fitness-levels for non-elitist populations." In *Proceedings of the Thirteenth Annual Conference on Genetic and Evolutionary Computation*, pages 20752082. ACM.

Pietro S. Oliveto and Carsten Witt (2011). "Simplified drift analysis for proving lower bounds inevolutionary computation." *Algorithmica*, 59(3):369–386. Erratum: http://arxiv.org/abs/1211.7184
Dirk Sudholt (2010). "General lower bounds for the running time of evolutionary algorithms." In *Proceedings of the Eleventh International Conference on Parallel Problem Solving from Nature*, pages 124133. Springer.

Witt, C. (2006). "Runtime analysis of the $(\mu+1)$ ea on simple pseudo-boolean functions evolutionary computation." In *Proceedings of the Eigth Annual Conference on Genetic and Evolutionary Computation*, pages 651658. ACM