Runtime Analysis of Evolutionary Algorithms: Basic Introduction[1]

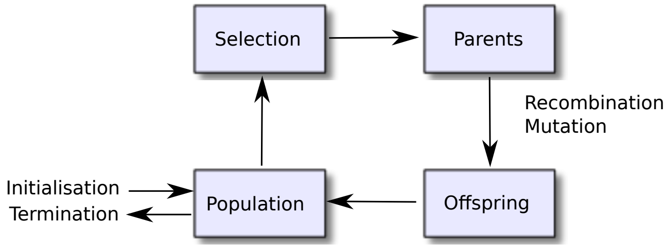
Introductory Tutorial at GECCO 2018

Per Kristian Lehre
University of Birmingham
Birmingham B15 2TT, UK
P.K.Lehre@cs.bham.ac.uk

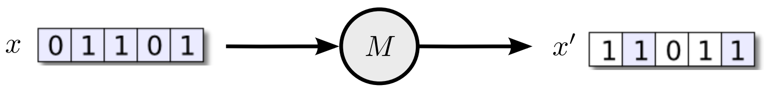
Pietro S. Oliveto
University of Sheffield
Sheffield S1 4DP, UK
P.Oliveto@sheffield.ac.uk

[1] For the latest version of these slides, please go to http://www.cs.bham.ac.uk/~lehrepk/gecco2018

Evolutionary Algorithms



Bitwise Mutation



for $i = 1$ **to** n **do**
  with probability χ/n
    $x'_i := 1 - x_i$
  otherwise
    $x'_i := x_i$
return x'

Linear Ranking Selection [Goldberg and Deb, 1991]

$\alpha : [0,1] \to [0,\infty)$ a *ranking function* if

$$\int_0^1 \alpha(y)dy = 1$$

Prob. of selecting individual with rank $\leq \gamma$ is

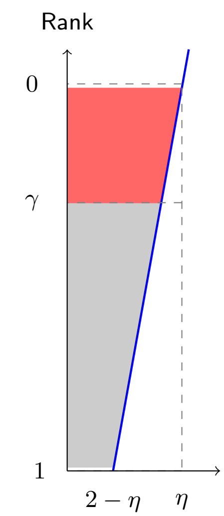
$$\beta(\gamma) := \int_0^\gamma \alpha(y)dy$$

Linear ranking selection is obtained for

$$\alpha(\gamma) := \eta - 2(\eta - 1)\gamma,$$

where $\eta \in (1,2)$ specifies **selection pressure**.

$$\int_0^\gamma \alpha(y)dy = \gamma(\eta(1-\gamma) + \gamma)$$



Example - Linear Ranking Selection

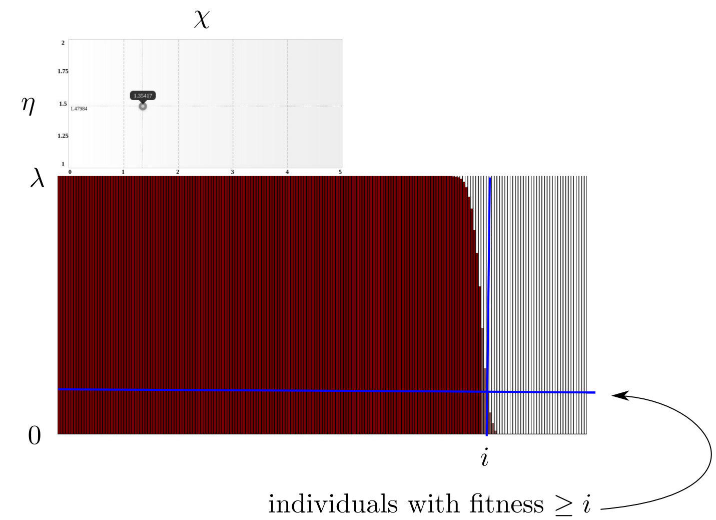
for $t = 0$ to ∞ **do**
  Sort current population P_t according to fitness f, st
  $f(P_t(0)) \geq f(P_t(1)) \geq \cdots \geq f(P_t(\lambda - 1))$.
  for $i = 0$ to $\lambda - 1$ **do**
    (Selection)
    Sample index r st. $\Pr(r \leq \gamma\lambda) = \gamma(\eta(1-\gamma)+\gamma)$.
    $P_{t+1}(i) := P_t(r)$.
    (Mutation)
    Flip each bit position in $P_{t+1}(i)$ with prob. χ/n.

Problem

- Is it possible to predict the behaviour of this and other EAs?
- Can we choose the parameters λ, η, and χ so that the EA optimises f efficiently, e.g.

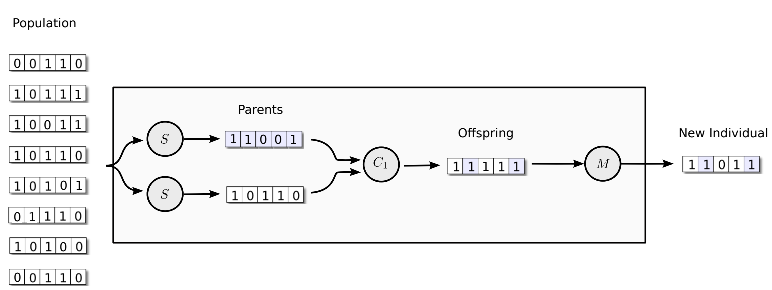
$$\text{ONEMAX}(x) := \sum_{i=1}^{n} x_i$$

Interactive Visualisation of Evolutionary Algorithm



`http://www.cs.bham.ac.uk/~lehrepk/ea/`

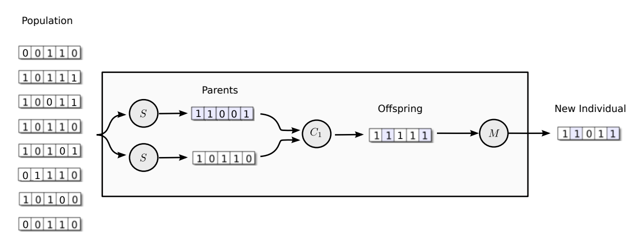
Outline of an Evolutionary Algorithm[2]



1: **initialise** a population P_0 of λ individuals uniformly at random.
2: **for** $t = 0, 1, 2, \ldots$ until termination condition **do**
3:   **evaluate** the individuals in population P_t.
4:   **for** $i = 1$ to λ **do**
5:     **select** two parents from population P_t.
6:     **recombine** the two parents.
7:     **mutate** the offspring and add it to population P_{t+1}.

[2] Pseudo-code adapted from Eiben and Smith [2003].

A Model of Population-based EAs

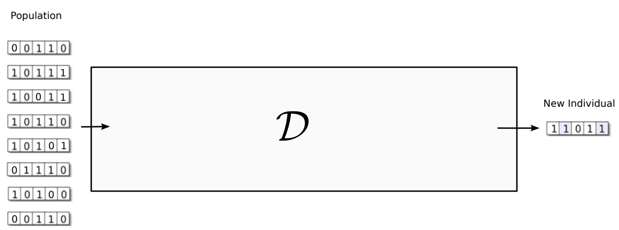


Wide range of evolutionary algorithms...

- selection mechanisms (ranking selection, (μ, λ)-selection, tournament selection, ...)
- fitness models (deterministic, stochastic, dynamic, partial, ...)
- variation operators
- search spaces (e.g. bitstrings, permutations, ...)

We will describe many of these with a general mathematical model.

A Model of Population-based EAs



Require: Search space $\mathcal{X}$ and random operator $\mathcal{D} : \mathcal{X}^\lambda \to \mathcal{X}$

$P_0 \sim \mathrm{Unif}(\mathcal{X}^\lambda)$

for $t = 0, 1, 2, \ldots$ until termination condition **do**

  for $i = 1$ to λ **do**

    $P_{t+1}(i) \sim \mathcal{D}(P_t)$

Aims and Goals of this Tutorial

- The scope of this tutorial is restricted to
  - population-based evolutionary algorithms, with finite parent– and offspring population sizes > 1,
  - using non-elitist selection mechanisms
- This tutorial will provide an overview of
  - the goals of runtime analysis of EAs
  - selected, generally applicable techniques
- You should attend if you wish to
  - theoretically understand the behaviour and performance of the EAs you design
  - familiarise yourself with some of the techniques used
  - pursue research in the area
- enable you or enhance your ability to
  1. understand theoretically population-dynamics of EAs on different problems
  2. perform time complexity analysis of population-based EAs on common toy problems
  3. have the basic skills to start independent research in the area

Outline

Evolutionary Algorithms are Algorithms

Criteria for evaluating algorithms

1. Correctness
   - Does the algorithm always give the correct output?
2. Computational Complexity
   - How much computational resources does the algorithm require to solve the problem?

Same criteria also applicable to evolutionary algorithms

1. Correctness.
   - Discover global optimum in finite time?
2. Computational Complexity.
   - Time (number of fitness function evaluations) is the most often studied computational resource.

Runtime Analysis of Population-based EAs

Definition
Given any target subset $B(n) \subset \{0,1\}^n$ (e.g. optima), let

$$T_{B(n)} := \min_{t \in \mathbb{N}} \{t\lambda \mid P_t \cap B(n) \neq \emptyset\}$$

be the first time[3] the population contains an individual in $B(n)$.

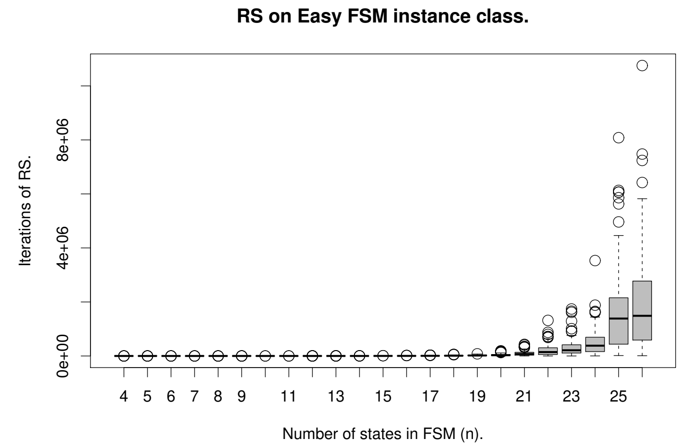
Problem
Show how

- $\mathbf{E}\left[T_{B(n)}\right]$ (the expected runtime)
- $\Pr\left(T_{B(n)} \leq k\right)$ (the "success" probability)

depend on the mapping $\mathcal{D}$.

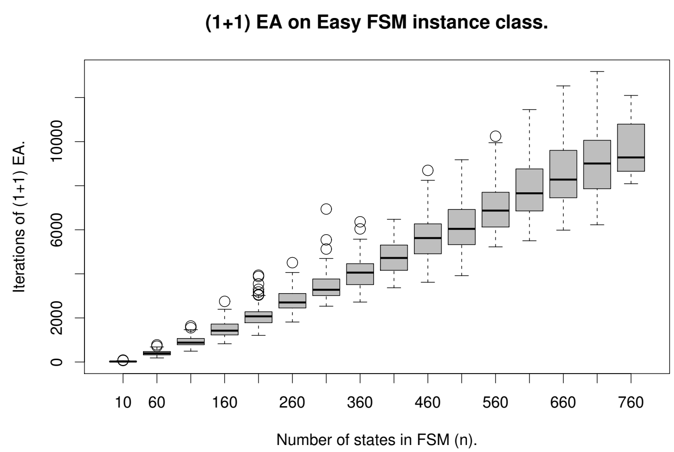
[3] We here count time as the number of search points that have been sampled since the start of the algorithm. For a typical $\mathcal{D}$ that models an EA, this corresponds to the number of times the fitness function is evaluated.

Runtime as a function of problem size



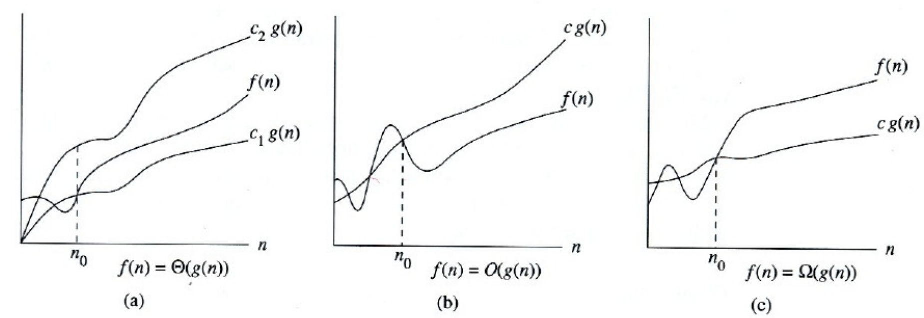
- Exponential $\Longrightarrow$ Algorithm impractical on problem.
- Polynomial $\Longrightarrow$ Possibly efficient algorithm.

Runtime as a function of problem size



- Exponential $\Longrightarrow$ Algorithm impractical on problem.
- Polynomial $\Longrightarrow$ Possibly efficient algorithm.

Asymptotic notation



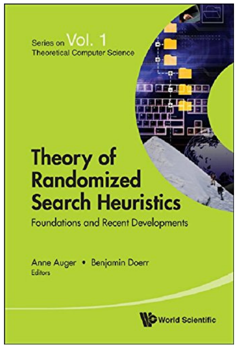
$f(n) \in O(g(n)) \iff \exists$ constants $c, n_0 > 0$ st. $0 \leq f(n) \leq cg(n) \quad \forall n \geq n_0$

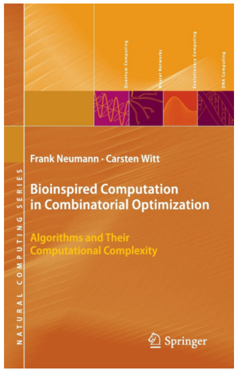
$f(n) \in \Omega(g(n)) \iff \exists$ constants $c, n_0 > 0$ st. $0 \leq cg(n) \leq f(n) \quad \forall n \geq n_0$

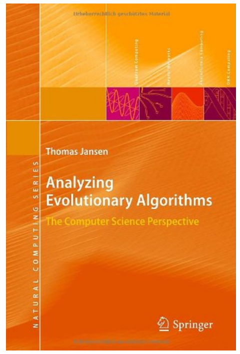
$f(n) \in \Theta(g(n)) \iff f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$

$$f(n) \in o(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Runtime Analysis of Evolutionary Algorithms







Approaches to Runtime Analysis of Populations

- Infinite population size
- Markov chain analysis He and Yao [2003]
- No parent population, or monomorphic populations
  - (1+1) EA
  - (1+λ) EA Jansen, Jong, and Wegener [2005]
  - (1,λ) EA Rowe and Sudholt [2012]
- Fitness-level techniques
  - (μ+1) EA Witt [2006]
  - (N+N) EAs Chen, He, Sun, Chen, and Yao [2009]
  - non-elitist EAs with unary variation operators Lehre [2011b], Dang and Lehre [2014]
- Classical drift analysis
  - Fitness proportionate selection Neumann, Oliveto, and Witt [2009], Oliveto and Witt [2014, 2015]
- Family trees
  - (μ+1) EA Witt [2006]
  - (μ+1) IA Zarges [2009]
- Multi-type branching processes Lehre and Yao [2012]
  - Negative drift theorem for populations Lehre [2011a]
- Level-based analysis Corus, Dang, Eremeev, and Lehre [2014]

Level-based Theorem[4]

[4] Corus, Dang, Eremeev, and Lehre [2017]

Level-based Analysis

Problem

- Search space $\mathcal{X}$ and a "target region" $B \subseteq \mathcal{X}$ (e.g. optima)
- A population-process $(P_t)_{t\in\mathbb{N}} \in \mathcal{X}^\lambda$, induced by an operator $\mathcal{D}$ which describes our EA including the fitness function
- What is the expected time to reach the target set B?
  More precisely, how does $\mathbf{E}[T_B]$ depend on $\mathcal{D}$ and λ?

Level-based theorem

- Gives upper bound on $\mathbf{E}[T_B]$ if $\mathcal{D}$ satisfies some conditions.
- The theorem is both highly general and precise.
  The upper bounds are nearly tight Corus et al. [2017].
- Used to *analyse* the runtime of EAs, often in complex settings
  - e.g., genetic algorithms, estimation of distribution algorithms, self-adaptation, noisy optimisation, rugged fitness landscapes...
- and to *design* efficient EAs for a given problem
  - e.g., choose appropriate parameterisation of the algorithm, design GA for shortest paths problem (Corus & Lehre, MIC'15)

Outline - Level-based Theorem[5]

1. Definition of levels of search space
2. Definition of "current level" of population
3. Statement of theorem and its conditions
4. Recommendations for how to apply the theorem
5. Some example applications
6. Derivation of special cases
   - Mutation-only EAs
   - Crossover
   - Mutation-only EAs with uncertain fitness (e.g. noise)

[5] It is out of scope of this tutorial to present the proof of this theorem. The proof uses drift analysis with a distance function that takes into account the current level, as well as the number of individuals above the current level.

Level Partitioning of Search Space $\mathcal{X}$

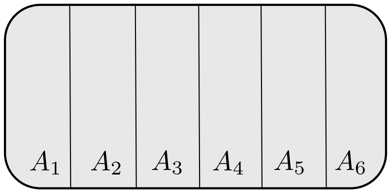
Definition

$(A_1, \dots, A_m)$ is a level-partitioning of search space $\mathcal{X}$ if

- $\bigcup_{j=1}^{m} A_j = \mathcal{X}$ (together, the levels cover the search space)
- $A_i \cap A_j = \emptyset$ whenever $i \neq j$ (the levels are nonoverlapping)
- the last level A_m covers the optima for the problem

We write $A_{\geq j}$ to denote everything in level j and higher, i.e.,

$$A_{\geq j} := \bigcup_{i=j}^{m} A_i.$$



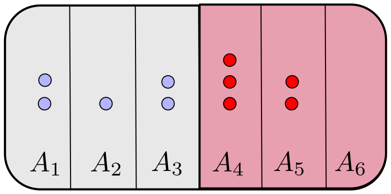
Notation

- For any population $P = (y_1, \dots, y_\lambda) \in \mathcal{X}^\lambda$ and $j \in [m]$, let

$$|P \cap A_{\geq j}| \; := \; |\{i \mid x_i \in A_{\geq j}\}|,$$

i.e, the number of individuals in P that is in subset $A_{\geq j}$.

Example



$|P \cap A_{\geq 4}| = 5$ where $A_{\geq 4}$ corresponds to the red region.

Current level of a population P wrt $\gamma_0 \in (0, 1)$

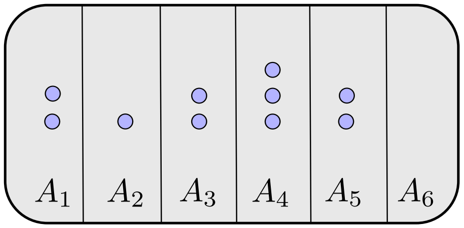
Definition

The unique integer $j \in [m-1]$ such that

$$|P \cap A_{\geq j}| \geq \gamma_0 \lambda > |P \cap A_{\geq j+1}|$$

Example

Current level wrt $\gamma_0 = \frac{1}{2}$ is ....4.



Level-based theorem (informal version)

If the following three conditions are satisfied

- **(G1)** it is always possible to sample above the current level
- **(G2)** the proportion of the population above the current level increases in expectation
- **(G3)** the population size is large enough

then the expected time to reach the last level is upper bounded by

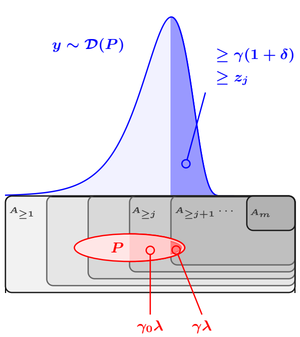
$$\mathbf{E}[T_B] \leq \cdots$$

Level-based Theorem[6] (1/2) (setup)

- Given a level-partitioning $(A_1, \ldots, A_m)$ of $\mathcal{X}$
- $m-1$ *upgrade probabilities* $z_1, \ldots, z_{m-1} \in (0,1]$ and $z_{\min} := \min_i z_i$
- a parameter $\delta \in (0,1)$, and
- a constant $\gamma_0 \in (0,1)$,

[6] This version of the theorem simplifies some of the conditions at the cost of a slightly less precise bound on the runtime.

Level-based Theorem (2/2) [Corus, Dang, Eremeev, and Lehre, 2017]



If for all populations $P \in \mathcal{X}^\lambda$, an individual $y \sim \mathcal{D}(P)$ has

$$\Pr(y \in A_{\geq j+1}) \geq z_j, \qquad \text{(G1)}$$
$$\Pr(y \in A_{\geq j+1}) \geq \gamma(1+\delta), \qquad \text{(G2)}$$

where $j \in [m-1]$ is the current level of population P, i.e.,

$$|P \cap A_{\geq j}| \geq \gamma_0\lambda > |P \cap A_{\geq j+1}| = \gamma\lambda,$$

and the population size λ is bounded from below by

$$\lambda \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{z_{\min}\delta^2}\right), \qquad \text{(G3)}$$

then the algorithm reaches the last level A_m in expected time

$$\mathbf{E}[T_{A_m}] \leq \left(\frac{8}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda\ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right)+\frac{1}{z_j}\right).$$

Suggested recipe for application of level-based theorem

1. Find a partition $(A_1, \ldots, A_m)$ of $\mathcal{X}$ that reflects the state of the algorithm, and where A_m consists of all goal states.
2. Find parameters γ_0 and δ and a configuration of the algorithm (e.g., mutation rate, selective pressure) such that whenever $|P \cap A_{\geq j+1}| = \gamma\lambda > 0$, condition (G2) holds, i.e.,
$$\Pr(y \in A_{\geq j+1}) \geq \gamma(1+\delta)$$
3. For each level $j \in [m-1]$, estimate a lower bound $z_j \in (0,1)$ such that whenever $|P \cap A_{\geq j+1}| = 0$, condition (G1) holds, i.e.,
$$\Pr(y \in A_{\geq j+1}) \geq z_j$$
4. Calculate the sufficient population size λ from condition (G3).
5. Read off the bound on expected runtime.

Simple Example to Illustrate Theorem

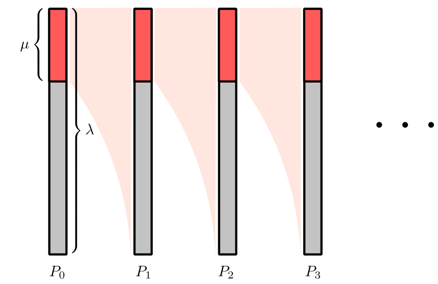
Problem

- search space $\mathcal{X} = \{1, \cdots, m\}$
- fitness function $f(x) = x$ (to be maximised)

Evolutionary Algorithm

for $t = 0, 1, 2, \ldots$ until termination condition **do**
  for $i = 1$ to λ **do**
    Select a parent x from P_t using (μ, λ)-selection
    Obtain y by mutating x
    Set i-th offspring $P_{t+1}(i) = y$

(μ, λ)-selection mechanism

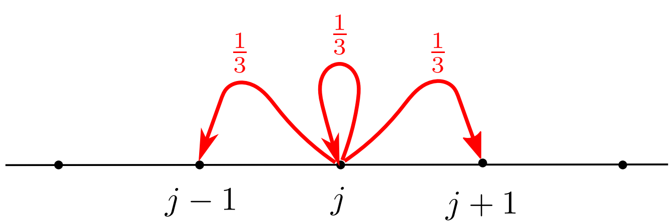


1. Sort the current population $P = (x_1, \ldots, x_\lambda)$ such that

$$f(x_1) \geq f(x_2) \geq \ldots \geq f(x_\lambda)$$

2. **return** $\mathrm{Unif}(x_1, \ldots, x_\mu)$

A simple mutation operator...



$$\Pr\left(V(x) = y\right) = \begin{cases} \frac{1}{3} & \text{if } y \in \{x-1, x, x+1\} \\ 0 & \text{otherwise.} \end{cases}$$

Step 1: Level-partition

Problem

- search space $\mathcal{X} = \{1, \cdots, m\}$
- fitness function $f(x) = x$ (to be maximised)

Level-partition of $\mathcal{X}$

$$A_j := \{j\}$$

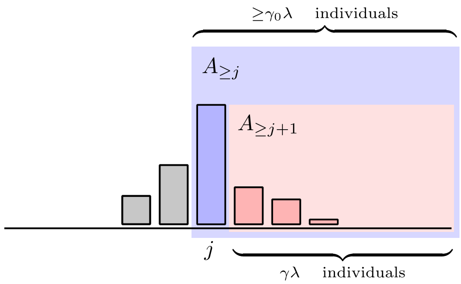
thus

$$A_{\geq j} = \bigcup_{i=j}^{m} A_j = \{j, j+1, \ldots, m\}.$$

Properties of a Population at Level j

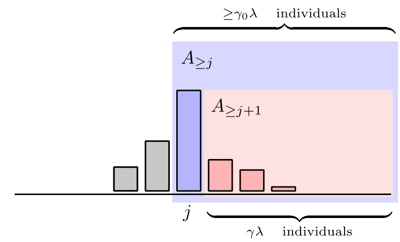
- Assume that the current level of the population P is j, i.e.,

$$\gamma\lambda = |P \cap A_{\geq j+1}| < \gamma_0\lambda \leq |P \cap A_{\geq j}| \qquad (1)$$



- (μ, λ) selects parent u.a.r. among best μ individuals
- by choosing parameter $\gamma_0 := \mu/\lambda$, assumption (1) implies
  - $\mathbf{Pr}\left(\text{select parent in } A_{\geq j}\right) = 1$
  - $\mathbf{Pr}\left(\text{select parent in } A_{\geq j+1}\right) = \frac{\gamma\lambda}{\mu}$

Condition (G2)

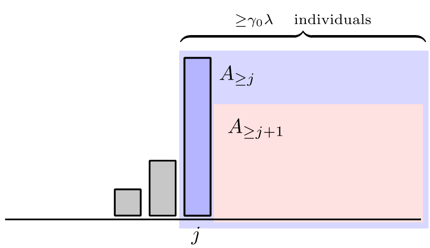


Assuming that $\frac{\lambda}{\mu} = \frac{9}{4} = \frac{1+\frac{1}{2}}{1-\frac{1}{3}}$

$$\begin{aligned}
&\mathbf{Pr}\left(y \in A_{\geq j+1}\right) \\
&\quad\geq \mathbf{Pr}\left(\text{select parent in } A_{\geq j+1}\right) \cdot \mathbf{Pr}\left(\text{do not downgrade}\right) \\
&\quad\geq \gamma \cdot \frac{\lambda}{\mu} \cdot \left(1 - \frac{1}{3}\right) = \gamma\left(1 + \frac{1}{2}\right). \\
&\quad\geq \gamma(1+\delta)
\end{aligned}$$

$\Longrightarrow$ Condition (G2) satisfied for $\delta = 1/2$.

Condition (G1)



$$\begin{aligned}
&\mathbf{Pr}\left(y \in A_{\geq j+1}\right) \\
&\quad\geq \mathbf{Pr}\left(\text{select parent in } A_{j}\right) \cdot \mathbf{Pr}\left(\text{upgrade offspring to } A_{\geq j+1}\right) \\
&\quad\geq 1 \cdot \frac{1}{3} \\
&\quad= z_j > 0
\end{aligned}$$

$\Longrightarrow$ Condition (G1) satisfied by choosing $z_j := \frac{1}{3}$ for all $j \in [m-1]$.

Condition (G3) - Sufficiently Large Population

Recall that $\gamma_0 = \mu/\lambda = 4/9$ and $\delta = 1/2$ and $z_{\text{min}} = \min_j z_j = \frac{1}{3}$

$$\begin{aligned}
&\left(\frac{4}{\gamma_0\delta^2}\right) \ln\left(\frac{128m}{z_{\text{min}}\delta^2}\right) \\
&= 36\ln(1536m) \\
&< 36(\ln(m) + 8) \leq \lambda
\end{aligned}$$

Hence, it is sufficient to choose a population size

$$\lambda \geq 36(\ln(m) + 8)$$

to satisfy condition (G3).

Example: Summary

We have shown that if $\lambda \geq 36(\ln(m)+8)$ and $\mu = 4\lambda/9$

- (G1) is satisfied for $z_j = 1/3$ for all $j \in [m-1]$
- (G2) is satisfied for $\delta = 1/2$, and
- (G3) is satisfied

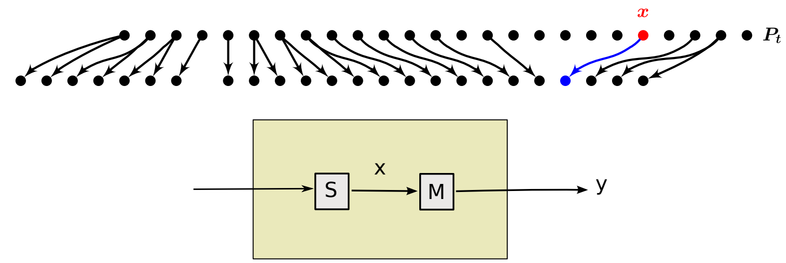
hence, by the level-based theorem, the expected running time of the EA is no more than

$$\left(\frac{8}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda \ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right)+\frac{1}{z_j}\right)$$

$$< \left(\frac{8}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda \ln\left(\frac{6}{z_j}\right)+\frac{1}{z_j}\right)$$

$$= 32\sum_{j=1}^{m-1}(\lambda\ln(18)+3) < 100m\lambda.$$

Population-Selection Variation Algorithm (PSVA)



for $t = 0$ to ∞ **do**
  for $i = 1$ to λ **do**
    Sample i-th parent x according to select(P_t)
    Sample i-th offspring $P_{t+1}(i)$ according to mutate(x)

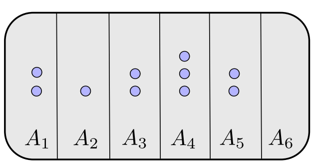
Measuring Selective Pressure

Definition (Cumulative selection probability)
For any population P of λ individuals, where the levels of the individuals are in decreasing order $\ell_0 \geq \ell_1 \geq \cdots \geq \ell_{\lambda-1}$, define for all $\gamma \in (0,\gamma_0)$

$$\zeta(\gamma,P) := \mathbf{Pr}\left(\text{select}(P) \in A_{\geq \ell_{\lceil\gamma\lambda\rceil}}\right),$$

(i.e., prob. of not selecting a worse individual than the $\gamma\lambda$-ranked).

Example



ℓ_0	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	ℓ_8	ℓ_9
5	5	4	4	4	3	3	2	1	1

$$\zeta(1/10,P) = \mathbf{Pr}\left(\text{select}(P) \in A_{\geq \ell_1}\right) = \mathbf{Pr}\left(\text{select}(P) \in A_{\geq 5}\right)$$

$$\zeta(3/10,P) = \mathbf{Pr}\left(\text{select}(P) \in A_{\geq \ell_3}\right) = \mathbf{Pr}\left(\text{select}(P) \in A_{\geq 4}\right)$$

Corollary for PSVA

If for any level $j \in [m-1]$ and all search points $x \in A_{\geq j}$,

(C1) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq s_j \geq s_{\text{min}}$

(C2) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j}\right) \geq p_0$

and for all non-optimal populations $P \in (\mathcal{X} \setminus A_m)^\lambda$ and $\gamma \in (0,\gamma_0]$

(C3) $\zeta(\gamma,P) \geq \frac{(1+\delta)\gamma}{p_0}$

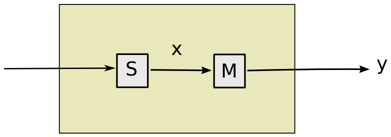
and the population size λ satisfies

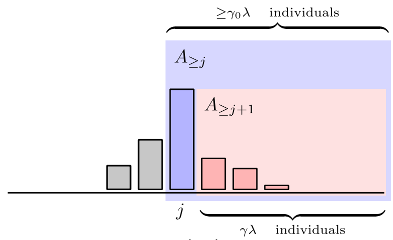
(C4) $\lambda \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{\gamma_0 s_{\text{min}}\delta^2}\right)$

then the expected time to reach the last level A_m is less than

$$\left(\frac{8}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda \ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j\delta\lambda}\right)+\frac{1}{\gamma_0 s_j}\right).$$

Proof of Corollary: (C2) & (C3) $\implies$ (G2)

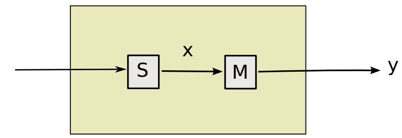


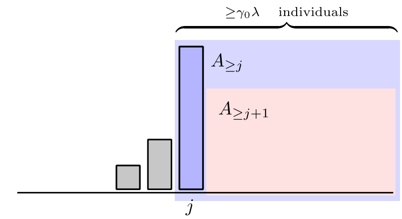


If $|P \cap A_{\geq j}| \geq \gamma_0 \lambda > |P \cap A_{\geq j+1}| =: \gamma\lambda$ and $y \sim \mathcal{D}(P)$ then

$$\begin{aligned}
&\Pr\left(y \in A_{\geq j+1}\right) \\
&\quad \geq \Pr\left(x \in A_{\geq j+1}\right) \Pr\left(y \in A_{\geq j+1} \mid x \in A_{\geq j+1}\right) \\
&\qquad \text{(i.e., select } x \text{ from level } j+1 \\
&\qquad \text{and do not downgrade it)} \\
&\quad \geq \zeta(\gamma, P) p_0 \\
&\quad \geq \gamma(1+\delta).
\end{aligned}$$

Proof of Corollary: (C1) & (C3) $\implies$ (G1)





If $|P \cap A_{\geq j}| \geq \gamma_0 \lambda$ and $y \sim \mathcal{D}(P)$

$$\begin{aligned}
\Pr\left(y \in A_{\geq j+1}\right) &\geq \Pr\left(x \in A_{\geq j}\right) \Pr\left(y \in A_{\geq j+1} \mid x \in A_{\geq j}\right) \\
&\qquad \text{(i.e., select } x \text{ from } A_{\geq j} \text{ and "upgrade" it)} \\
&\geq \zeta(\gamma_0, P) s_j \\
&\geq \gamma_0 (1+\delta) s_j / p_0 \\
&= z_j > 0
\end{aligned}$$

Example Application

$$\textsc{LeadingOnes}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

Partition into $n+1$ levels

$$A_j := \{x \in \{0,1\}^n \mid x_1 = \cdots = x_{j-1} = 1 \land x_j = 0\}$$

Example Application

(μ,λ) EA with bit-wise mutation rate χ/n on LeadingOnes.
For any const. $\delta \in (0,1)$ and large n, no bits mutated with prob.

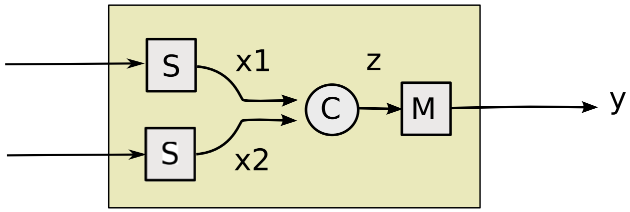
$$\left(1 - \frac{\chi}{n}\right)^n > \frac{1-\delta}{e^\chi}.$$

If $x \in A_{\geq j}$, $\lambda/\mu > e^\chi \left(\frac{1+\delta}{1-\delta}\right)$ and $\lambda > c'' \ln(n)$ then

- (C1) $\Pr\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq \dfrac{\chi(1-\delta)}{n e^\chi}$ $=: s_j =: s_{\min}$
- (C2) $\Pr\left(\text{mutate}(x) \in A_{\geq j}\right) \geq \dfrac{1-\delta}{e^\chi}$ $=: p_0$
- (C3) $\zeta(\gamma, P) \geq \gamma\lambda/\mu > \gamma e^\chi \left(\dfrac{1+\delta}{1-\delta}\right)$ $= \gamma(1+\delta)/p_0$
- (C4) $\lambda > c'' \ln(n)$ $> c \ln(m/s_{\min})$

then $\mathbf{E}[T] = \mathcal{O}\left(\sum_{j=1}^{m-1} \lambda \ln\left(\frac{\lambda}{1+s_j\lambda}\right) + \frac{1}{s_j}\right) = \mathcal{O}(n\lambda \ln(\lambda) + n^2)$

Genetic Algorithms with Crossover



Definition (Genetic Algorithm)

for $t = 0, 1, 2, \ldots$ until termination condition **do**
  for $i = 1$ to λ **do**
    Select parents x_1 and x_2 from population P_t acc. to p_{sel}
    Create z by applying a crossover operator to x_1 and x_2.
    Create y by applying a mutation operator to y.

Corollary for Genetic Algorithms

If for any level $j \in [m-1]$ and all search points $x \in A_{\geq j}$

(C1) $\Pr\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq s_j \geq s_{\text{min}}$

(C2) $\Pr\left(\text{mutate}(x) \in A_{\geq j}\right) \geq p_0$

and for all $u \in A_{\geq j}$ and $v \in A_{\geq j+1}$

(C3) $\Pr\left(\text{crossover}(u, v) \in A_{\geq j+1}\right) \geq \varepsilon_1$

and for all non-optimal populations $P \in (\mathcal{X} \setminus A_m)^\lambda$ and $\gamma \in (0, \gamma_0]$

(C4) $\zeta(\gamma, P) \geq \gamma \sqrt{\frac{1+\delta}{p_0 \varepsilon_1 \gamma_0}}$

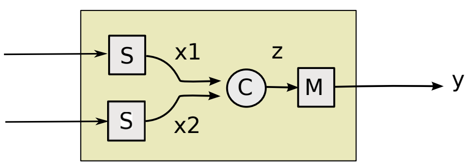
and the population size λ satisfies

(C5) $\lambda \geq \left(\frac{4}{\gamma_0 \delta^2}\right) \ln\left(\frac{128m}{\gamma_0 \delta^2 s_{\text{min}}}\right)$

then the expected time to reach the last level A_m is less than

$$\left(\frac{8}{\delta^2}\right) \sum_{j=1}^{m-1} \left(\lambda \ln\left(\frac{6\delta\lambda}{4 + \gamma_0 s_j \delta\lambda}\right) + \frac{1}{\gamma_0 s_j}\right).$$

Example application – (μ,λ) GA on Onemax



$$\text{ONEMAX}(x) := \sum_{i=1}^{n} x_i.$$

(μ,λ) Genetic Algorithm (GA)

for $t = 0, 1, 2, \ldots$ until termination condition **do**
  for $i = 1$ to λ **do**
    Select a parent x from population P_t acc. to (μ, λ)-selection
    Select a parent y from population P_t acc. to (μ, λ)-selection
    Apply uniform crossover to x and y, i.e. $z :=$ crossover(x, y)
    Create $P_{t+1}(i)$ by flipping each bit in z with probability χ/n.

Theorem

If $\lambda > c\ln(n)$ for a sufficiently large constant $c > 0$, and $\frac{\lambda}{\mu} > 2e^{\chi}(1+\delta)$ for any constant $\delta > 0$, then the expected runtime of (μ,λ) GA on ONEMAX *is $O(n\lambda)$.*

Partition of Search Space into Levels

Partition into $m := n + 1$ levels $A_0, \ldots, A_n$

$$A_j := \{x \in \{0,1\}^n \mid \text{ONEMAX}(x) = j\}$$

Condition (C1) and (C2)

Given any search point $x \in A_{\geq j}$,

- to remain at the same level, it is sufficient to not flip any bits

$$\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j}\right) \geq \left(1 - \frac{\chi}{n}\right)^n \geq \frac{1-\delta}{e^\chi} =: p_0.$$

- to reach a higher level, it suffices to flip a zero-bit into a one-bit and leave the other bits unchanged, i.e.,

$$\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq (n-j)\frac{\chi}{n}\left(1 - \frac{\chi}{n}\right)^{n-1} \geq \frac{\chi(n-j)(1-\delta)}{ne^\chi} =: s_j.$$

Example application – (μ,λ) GA on Onemax

If $\lambda/\mu > \dots$ and $\lambda > c\ln(n)$ and $x \in A_{\geq j}$ then

(C1) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq \frac{\chi(n-j)(1-\delta)}{ne^\chi} =: s_j$ ✓

(C2) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j}\right) \geq \frac{1-\delta}{e^\chi} =: p_0$ ✓

and for all $u \in A_{\geq j}$ and $v \in A_{\geq j+1}$

(C3) $\mathbf{Pr}\left(\text{crossover}(u,v) \in A_{\geq j+1}\right) \geq \varepsilon_1 > 0$

and for all non-optimal populations $P \in (\mathcal{X} \setminus A_m)^\lambda$ and $\gamma \in (0, \gamma_0]$

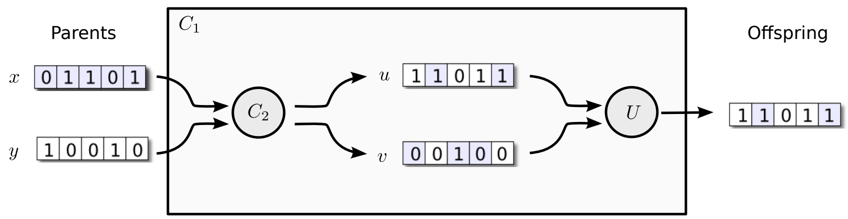
(C4) $\zeta(\gamma, P) \geq \frac{\gamma\lambda}{\mu} \geq \gamma\sqrt{\frac{1+\delta}{p_0\varepsilon_1\gamma_0}}$

and the population size λ satisfies

(C5) $\lambda > c\ln(n) \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{\gamma_0\delta^2 s_{\text{min}}}\right)$ ✓

- (C5) holds if the constant $c > 0$ is large enough ($m = n + 1$)
- Remains to show that (C3) and (C4) can be satisfied
  - Need to determine the parameter ε_1.
  - Need to determine a lower bound for the ratio λ/μ.

Condition (C3) – (μ,λ) GA on OneMax



Proof.
Assume that $x \in A_{\geq j+1}$ and $y \in A_{\geq j}$, and w.l.o.g. that $|u| \geq |v|$

$$\begin{aligned} 2j + 1 &\leq |x| + |y| \\ &= |u| + |v| \\ &\leq 2|u|. \end{aligned}$$

Therefore $\mathbf{Pr}\left(u \in A_{\geq j+1}\right) = 1$ and

$$\mathbf{Pr}\left(\text{crossover}(x,y) \in A_{\geq j+1} \mid x \in A_{\geq j+1} \text{ and } y \in A_{\geq j}\right) \geq \frac{1}{2} =: \varepsilon.$$

□

Example application – (μ,λ) GA on Onemax

If $\lambda/\mu > 2e^\chi\left(\frac{1+\delta}{1-\delta}\right)$ for any const. $\delta > 0$, and $\lambda > c\ln(n)$

(C1) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j+1}\right) \geq \frac{\chi(n-j)(1-\delta)}{ne^\chi} =: s_j$ ✓

(C2) $\mathbf{Pr}\left(\text{mutate}(x) \in A_{\geq j}\right) \geq \frac{1-\delta}{e^\chi} =: p_0$ ✓

(C3) $\mathbf{Pr}\left(\text{crossover}(u,v) \in A_{\geq j+1}\right) > 1/2 =: \varepsilon_1 > 0$ ✓

(C4) $\beta(\gamma) \geq \frac{\gamma\lambda}{\mu} \geq \gamma\sqrt{\frac{1+\delta}{p_0\varepsilon_1\gamma_0}}$ ✓

(C5) $\lambda > c\ln(n) \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{\gamma_0\delta^2 s_{\text{min}}}\right)$ ✓

We have all the necessary parameters, and would like to find a simple expression for the expected runtime

$$\left(\frac{8}{\delta^2}\right)\left(\lambda\sum_{j=0}^{n-1}\ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j\delta\lambda}\right) + \sum_{j=0}^{n-1}\frac{1}{\gamma_0 s_j}\right).$$

Bounding the first term (first attempt, imprecise)

$$\sum_{j=0}^{n-1} \ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j \delta\lambda}\right) < \sum_{j=0}^{n-1} \ln\left(\frac{6\delta\lambda}{4}\right) = \mathcal{O}(n\ln(\lambda)).$$

- This upper bound is imprecise because it does not exploit that the upgrade probabilities s_j are large for small j.

Bounding the first term (second attempt, more precise)

$$\sum_{j=0}^{n-1} \ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j \delta\lambda}\right) < \sum_{j=0}^{n-1} \ln\left(\frac{6}{\gamma_0 s_j}\right)$$

using $\ln(a)+\ln(b)=\ln(ab)$ and defining $c := \frac{6e^{\chi}}{\gamma_0(1-\delta)\chi}$

$$= \ln\left(\prod_{j=0}^{n-1}\frac{cn}{n-j}\right) = \ln\left(\frac{(cn)^n}{n!}\right)$$

and using the lower bound $n! > (n/e)^n$

$$< \ln\left(\frac{(cn)^n e^n}{n^n}\right) = n\ln(ec) = \mathcal{O}(n).$$

Bounding the second term

Recall the definition of the n-th Harmonic number

$$H_n := \sum_{i=1}^{n}\frac{1}{i} = \mathcal{O}(\ln(n)).$$

The second term can therefore be bounded as

$$\sum_{j=0}^{n-1}\frac{1}{\gamma_0 s_j} = \mathcal{O}\left(\sum_{j=0}^{n-1}\frac{n}{n-j}\right) = \mathcal{O}(n\ln(n))$$

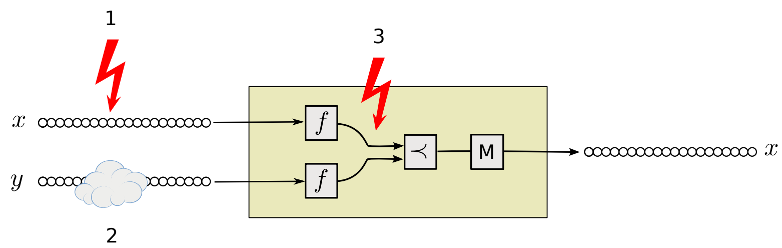
Final result

Theorem

If $\lambda > c\ln(n)$ for a sufficiently large constant $c>0$, and $\frac{\lambda}{\mu} > 2e^{\chi}(1+\delta)$ for any constant $\delta > 0$, then the expected runtime of (μ,λ) GA on OneMax is

$$\left(\frac{8}{\delta^2}\right)\left(\lambda\sum_{j=0}^{n-1}\ln\left(\frac{6\delta\lambda}{4+\gamma_0 s_j\delta\lambda}\right) + \sum_{j=0}^{n-1}\frac{1}{\gamma_0 s_j}\right)$$
$$= \mathcal{O}(n\lambda) + \mathcal{O}(n\ln n) = \mathcal{O}(n\lambda).$$

Uncertainty in Comparison-based PSVAs



Sources of uncertainty

1. Droste noise model (Droste, 2004)
2. Partial evaluation
3. Noisy fitness (Prügel-Bennet, Rowe, Shapiro, 2015)

Sufficient with mutation rate $\delta/(3n)$ and

$$\Pr\left(x \text{ choosen} \mid f(x) > f(y)\right) \geq \frac{1}{2} + \delta \quad \text{with } 1/\delta \in \text{poly}(n)$$

Dang and Lehre [2015] and Dang and Lehre [2016]

Lower Bounds

Lower Bounds

Problem

Consider a sequence of populations $P_1, \ldots$ over a search space $\mathcal{X}$, and a target region $A \subset \mathcal{X}$ (e.g., the set of optimal solutions), let

$$T_A := \min\{\ \lambda t \ \mid \ P_t \cap A \neq \emptyset\ \}$$

We would like to prove statements on the form

$$\Pr\left(T_A \leq t(n)\right) \leq e^{-\Omega(n)}. \qquad (2)$$

- i.e., with overwhelmingly high probability, the target region A has not been found in $t(n)$ evaluations
- lower bounds often harder to prove than upper bounds
- will present an easy to use method that is applicable in many situations

Algorithms considered for lower bounds

Definition (Non-elitist EA with selection and mutation)

for $t = 0, 1, 2, \ldots$ until termination condition **do**
  for $i = 1$ to λ **do**
    Select parent x from population P_t according to p_{sel}
    Flip each position in x independently with probability χ/n.
    Let the i-th offspring be $P_{t+1}(i) := x$.
    (i.e., create offspring by mutating the parent)

Assumptions

- population size $\lambda \in \text{poly}(n)$, i.e. not exponentially large
- bitwise mutation with probability χ/n, but no crossover.
- results hold for any non-elitist selection scheme p_{sel} that satisfy some mild conditions to be described later.

Reproductive rate[7]

Definition
For any population $P = (x_1, \ldots, x_\lambda)$ let $p_{\text{sel}}(x_i)$ be the probability that individual x_i is selected from the population P

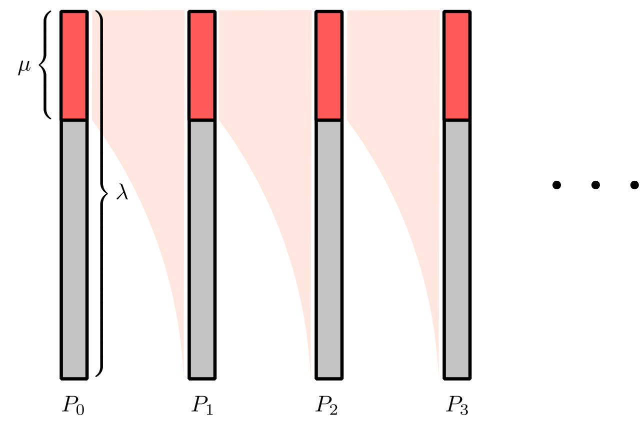
- The reproductive rate of individual x_i is $\lambda \cdot p_{\text{sel}}(x_i)$.
- The reproductive rate of a selection mechanism is bounded from above by α_0 if

$$\forall P \in \mathcal{X}^\lambda, \quad \forall x \in P \quad \lambda \cdot p_{\text{sel}}(x) \leq \alpha_0$$

(i.e., no individual gets more than α_0 offspring in expectation)

[7] The reproductive rate of an individual as defined here corresponds to the notion of "fitness" as used in the field of population genetics, i.e., the expected number of offspring.

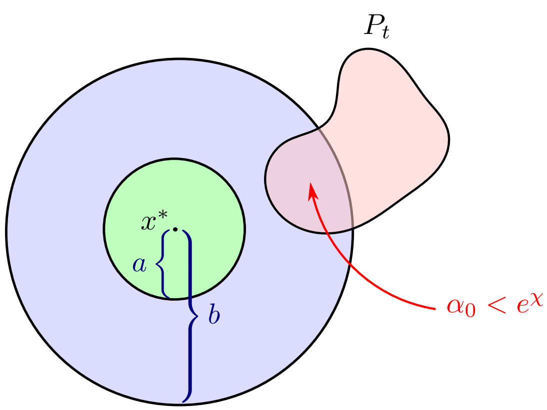
(μ, λ)-selection mechanism



Probability of selecting i-th individual is $p_i \in \{0, \frac{1}{\mu}\}$.

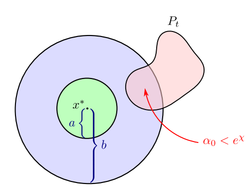
- reproductive rate bounded by $\alpha_0 = \lambda/\mu$

Negative Drift Theorem for Populations (informal)



If individuals closer than b of target has reproductive rate $\alpha_0 < e^\chi$, then it takes exponential time $e^{c(b-a)}$ to reach within a of target.

Negative Drift Thm. for Populations [Lehre, 2011a]



Consider the non-elitist EA with

- population size $\lambda = \text{poly}(n)$
- bitwise mutation rate χ/n for $0 < \chi < n$

let $T := \min\{t \mid H(P_t, x^*) \leq a\}$ for any $x^* \in \{0,1\}^n$.

If there are constants $\alpha_0 \geq 1$, $\delta > 0$ and integers $a(n)$ and $b(n) < \frac{n}{\chi}$ where $b(n) - a(n) = \omega(\ln n)$, st.

(C1) If $a(n) < H(x, x^*) < b(n)$ then $\lambda \cdot p_{\text{sel}}(x) \leq \alpha_0$.
(C2) $\psi := \ln(\alpha_0)/\chi + \delta < 1$
(C3) $b(n) < \min\left\{\frac{n}{5}, \frac{n}{2}\left(1 - \sqrt{\psi(2-\psi)}\right)\right\}$

then there exist constants $c, c' > 0$ such that

$$\Pr\left(T \leq e^{c(b(n)-a(n))}\right) \leq e^{-c'(b(n)-a(n))}.$$

The worst individuals have low reproductive rate

Lemma

Consider any selection mechanism which for $x, y \in P$ satisfies

(a) *If $f(x) > f(y)$, then $p_{sel}(x) > p_{sel}(y)$.*
(selection probabilities are monotone wrt fitness)

(b) *If $f(x) = f(y)$, then $p_{sel}(x) = p_{sel}(y)$.*
(ties are drawn randomly)

If $f(x) = \min_{y \in P} f(y)$, then $p_{sel}(x) \leq 1/\lambda$.
(individuals with lowest fitness have reproductive rate ≤ 1)

Proof.

- By (a) and (b), $p_{sel}(x) = \min_{y \in P} p_{sel}(y)$.
- $1 = \sum_{x \in P} p_{sel}(x) \geq \lambda \cdot \min_{y \in P} p_{sel}(y) = \lambda \cdot p_{sel}(x)$.

□

Example 1: Needle in the haystack

Definition

$$\text{NEEDLE}(x) = \begin{cases} 1 & \text{if } x = 1^n \\ 0 & \text{otherwise.} \end{cases}$$

Theorem

The optimisation time of the non-elitist EA with any selection mechanism satisfying the properties above[8] on NEEDLE *is at least e^{cn} with probability $1 - e^{-\Omega(n)}$ for some constant $c > 0$.*

[8] From black-box complexity theory, it is known that NEEDLE is hard for all search heuristics (Droste et al 2006).

Example 1: Needle in the haystack (proof[9])

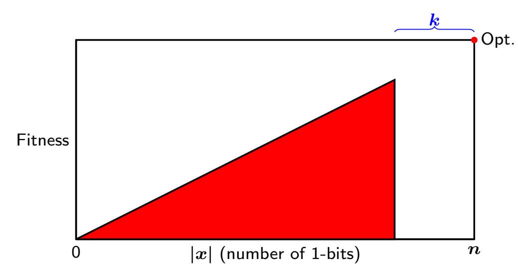
- Apply negative drift theorem with $a(n) := 1$.
- By previous lemma, can choose $\alpha_0 = 1$ for any $b(n)$, hence $\psi = \ln(\alpha)/\chi + \delta = \delta < 1$ for all χ and $\delta < 1$.
- Choosing the parameters $\delta := 1/10$ and $b(n) := n/6$ give

$$\min\left\{\frac{n}{5}, \frac{n}{2}\left(1 - \sqrt{\psi(2-\psi)}\right)\right\} = \frac{n}{5} > b(n).$$

- It follows that $\Pr\left(T \leq e^{c(b(n)-a(n))}\right) \leq e^{-\Omega(n)}$.

[9] For simplicity, we assume that $\chi \leq 6$, thus $b(n) = n/6 \leq n/\chi$ holds.

Exercise: Optimisation time on Jump$_k$



Recipe

- $a(n) = 1$
- $b(n) = k$
- $\alpha_0 = 1$ as before
- small δ

$$\text{JUMP}_k(x) := \begin{cases} 0 & \text{if } n - k \leq |x| < n, \\ |x| & \text{otherwise.} \end{cases}$$

When the best individuals have low reproductive rate

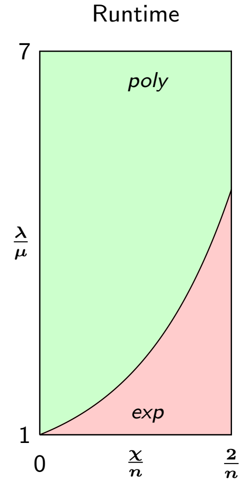
Remark

- The negative drift conditions hold trivially if $\alpha_0 < e^{\chi}$ holds for all individuals.

Example (Insufficient selective pressure)

Selection mechanism	Parameter settings
Linear ranking selection	$\eta < e^{\chi}$
k-tournament selection	$k < e^{\chi}$
(μ,λ)-selection	$\lambda < \mu e^{\chi}$
Any in cellular EAs	$\Delta(G) < e^{\chi}$

Mutation-selection balance



Example

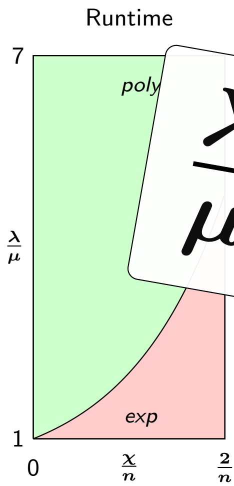
The runtime T of a non-elitist EA with

- (μ, λ)-selection
- bit-wise mutation rate χ/n
- population size $\lambda > c\log(n)$

on LeadingOnes has expected value

$$\mathbf{E}[T] = \begin{cases} e^{\Omega(n)} & \text{if } \lambda < \mu e^{\chi} \\ O(n\lambda\ln(\lambda) + n^2) & \text{if } \lambda > \mu e^{\chi} \end{cases}$$

Mutation-selection balance



$$\frac{\lambda}{\mu} > e^{\chi}$$

Example

The runtime T of a non-elitist EA with

- (μ, λ)-selection
- bit-wise mutation rate χ/n
- population size $\lambda > c\log(n)$

on LeadingOnes has expected value

$$\mathbf{E}[T] = \begin{cases} e^{\Omega(n)} & \text{if } \lambda < \mu e^{\chi} \\ O(n\lambda\ln(\lambda) + n^2) & \text{if } \lambda > \mu e^{\chi} \end{cases}$$

Other Example Applications

Expected runtime of EA with bit-wise mutation rate χ/n

Selection Mechanism	High Selective Pressure	Low Selective Pressure
Fitness Proportionate	$\nu > f_{\max}\ln(2e^{\chi})$	$\nu < \chi/\ln 2$ and $\lambda \geq n^3$
Linear Ranking	$\eta > e^{\chi}$	$\eta < e^{\chi}$
k-Tournament	$k > e^{\chi}$	$k < e^{\chi}$
(μ, λ)	$\lambda > \mu e^{\chi}$	$\lambda < \mu e^{\chi}$
Cellular EAs		$\Delta(G) < e^{\chi}$
OneMax	$O(n\lambda)$	$e^{\Omega(n)}$
LeadingOnes	$O(n\lambda\ln(\lambda) + n^2)$	$e^{\Omega(n)}$
Linear Functions	$O(n\lambda\ln(\lambda) + n^2)$	$e^{\Omega(n)}$
r-Unimodal	$O(r\lambda\ln(\lambda) + nr)$	$e^{\Omega(n)}$
Jump$_r$	$O(n\lambda + (n/\chi)^r)$	$e^{\Omega(n)}$

Fitness proportional selection + crossover Oliveto and Witt [2014, 2015]

Definition (Simple Genetic Algorithm (SGA) (Goldberg 1989))

for $t = 0, 1, 2, \ldots$ until termination condition **do**
  for $i = 1$ to λ **do**
    Select two parents x and y from P_t proportionally to fitness
    Obtain z by applying uniform crossover to x and y with $p = 1/2$
    Flip each position in z independently with $p = 1/n$.
    Let the i-th offspring be $P_{t+1}(i) := x$.
    (i.e., create offspring by crossover followed by mutation)

Application to OneMax

Expected Behaviour

- Backward drift due to mutation close to the optimum
- no positive drift due to crossover
- selection too weak to keep positive fluctuations

Difficulties When Introducing Crossover:

- Variance of offspring distribution
- # flipping bits due to mutation Poisson-distributed $\to$ variance $O(1)$
- # of one-bits created by crossover binomially distributed according to Hamming distance of parents and $1/2 \to$ deviation $\Omega(\sqrt{n})$ possible

Negative Drift Theorem With Scaling

Let X_t, $t \geq 0$, random variable describing a stochastic process over finite state space $S \subseteq \mathbb{R}$;

If there $\exists$ interval $[a, b]$ and, possibly depending on $\ell := b - a$, bound $\epsilon(\ell) > 0$ and scaling factor $r(\ell)$ st.

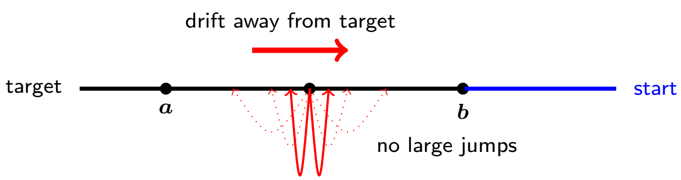
(C1) $E(X_{t+1} - X_t \mid X_0, \ldots, X_t \ \wedge \ a < X_t < b) \ \geq \ \epsilon$,

(C2) $\mathrm{Prob}(|X_{t+1} - X_t| \geq jr \mid X_0, \ldots, X_t \ \wedge \ a < X_t) \ \leq \ e^{-j}$ for $j \in \mathbb{N}_0$,

(C3) $1 \leq r \leq \min\{\epsilon^2 \ell, \sqrt{\epsilon\ell/(132\log(\epsilon\ell))}\}$.

then

$$\Pr\left(T \leq e^{\epsilon\ell/(132r^2)}\right) = O(e^{-\epsilon\ell/(132r^2)}).$$



Potential Function

For drift theorem, capture whole population in one value: For $X = \{x_1, \ldots, x_\mu\}$ let $g(X) := \sum_{i=1}^{\mu} e^{\kappa \textsc{OneMax}(x_i)}$.

Problem: maybe $r(\ell) = \Omega(\sqrt{\ell})$

Solution

Find bits that are “converged” within population, i.e., either ones or zeros only.

Diversity

X_t: # individuals with 1 in some fixed position at time t

Assume uniform selection (and no mutation). Then:

- The probability crossover produces an individual with 1 in the fixed position is ($X_t = k$):
- $\frac{k}{\mu} \cdot \frac{k}{\mu} + 2 \cdot \frac{1}{2} \cdot \frac{k(\mu-k)}{\mu^2} = \frac{k}{\mu}$
- $\{X_t\} \approx B(\mu, k/\mu) \rightsquigarrow E(X_t \mid X_{t-1} = k) = k$ (martingale)
- But random fluctuations $\rightsquigarrow$ absorbing state 0 or μ due to the variance ($E(T_{0\vee\mu}) = O(\mu \log \mu)$ [drift analysis]).
- Progress by crossover is at most $n^{1/2+\epsilon}$ w.o.p. (Chernoff Bounds when ones are $n/2$).
- If $\mu \leq n^{1/2-\epsilon}$ a bit has converged to 0 before optimum is found w.o.p.

Diversity

X_t: # individuals with 1 in some fixed position at time t

Assume uniform selection (and no mutation). Then:

- The probability crossover produces an individual with 1 in the fixed position is ($X_t = k$):
- $\frac{k}{\mu} \cdot \frac{k}{\mu} + 2 \cdot \frac{1}{2} \cdot \frac{k(\mu-k)}{\mu^2} = \frac{k}{\mu}$
- $\{X_t\} \approx B(\mu, k/\mu) \rightsquigarrow E(X_t \mid X_{t-1} = k) = k$ (martingale)
- But random fluctuations $\rightsquigarrow$ absorbing state 0 or μ due to the variance

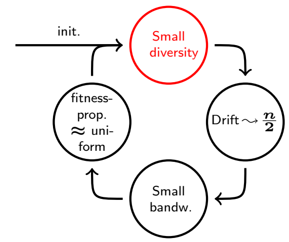
Compare fitness-prop. and uniform selection:

- Basically no difference for small population *bandwidth* (difference of best and worst OneMax-value in pop.)
- $E(X_t \mid X_{t-1} = k) = k \pm 1/(7\mu)$

Result

Let $\mu \le n^{1/8-\epsilon}$ for an arbitrarily small constant $\epsilon > 0$. Then with probability $1 - 2^{-\Omega(n^{\epsilon/9})}$, the SGA on OneMax does not create individuals with more than $(1+c)\frac{n}{2}$ or less than $(1-c)\frac{n}{2}$ one-bits, for arbitrarily small constant $c > 0$, within the first $2^{n^{\epsilon/10}}$ generations. In particular, it does not reach the optimum then.

Overall Proof Structure



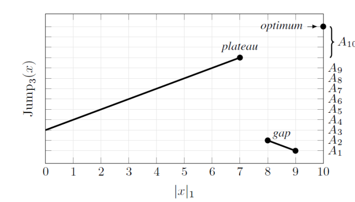
Not a loop, but in each step only exponentially small failure prob.

Steady-state (μ+1) GA

Definition ((μ+1) GA)

$P_0 \leftarrow \mu$ individuals, uniformly at random from $\{0,1\}^n$
for $t = 1, 2, \ldots$ until termination condition **do**
  Select x and y from P_t unif. at random with replacement
  Obtain z by applying uniform crossover to x and y with $p = 1/2$
  Mutate each position in z independently with $p = c/n$
  Select one element from P with lowest fitness and remove it.

Crossover allows faster escape from local optima Dang, Friedrich, Kötzing, Krejca, Lehre, Oliveto, Sudholt, and Sutton [2017]



Expected Runtimes ($k > 2$)

- (μ+1) EA with $p_m = 1/n$: $\Theta(n^k)$ (i.e., no crossover);
- (μ+1) GA with $p_m = 1/n$: $O(n^{k-1}\log n)$ [$\mu = \Theta(n)$];
- (μ+1) GA with $p_m = (1+\delta)/n$ is $O(n^{k-1})$ [$\mu = \Theta(\log n)$].

The interplay between mutation and crossover can create diversity on the top of the plateau; Then crossover + mutation can take advantage of the diversity to jump more quickly.

Summary

- Runtime analysis of evolutionary algorithms
  - mathematically rigorous statements about EA performance
  - most previous results on simple EAs, such as (1+1) EA
  - special techniques developed for population-based EAs
- Level-based method Corus et al. [2014]
  - EAs analysed from the perspective of EDAs
  - Upper bounds on expected optimisation time
  - Example applications include crossover and noise
- Negative drift theorem Lehre [2011a]
  - reproductive rate vs selective pressure
  - exponential lower bounds
  - mutation-selection balance
- Diversity + Bandwidth analysis for fitness proportional selection
  - analysis of crossover
  - low selection pressure
  - exponential lower bounds
- Speed-up via crossover for steady state GAs to escape local optima

Acknowledgements

- Dogan Corus, University of Sheffield, UK
- Duc-Cuong Dang, University of Nottingham, UK
- Anton Eremeev, Omsk Branch of Sobolev Institute of Mathematics, Russia
- Carsten Witt, DTU, Lyngby, Denmark

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE) and by the EPSRC under grant no EP/M004252/1.

References I

Tianshi Chen, Jun He, Guangzhong Sun, Guoliang Chen, and Xin Yao. A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(5):1092–1106, Oct. 2009. ISSN 1083-4419. doi: 10.1109/TSMCB.2008.2012167.

Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. Level-based analysis of genetic algorithms and other search processes. In *Parallel Problem Solving from Nature - PPSN XIII - 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings*, pages 912–921, 2014. doi: 10.1007/978-3-319-10762-2_90. URL http://dx.doi.org/10.1007/978-3-319-10762-2_90.

Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. Level-based analysis of genetic algorithms and other search processes. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2017. ISSN 1089-778X. doi: 10.1109/TEVC.2017.2753538.

Duc-Cuong Dang and Per Kristian Lehre. Refined Upper Bounds on the Expected Runtime of Non-elitist Populations from Fitness-Levels. In *Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 1367–1374, 2014. ISBN 9781450326629. doi: 10.1145/2576768.2598374.

Duc-Cuong Dang and Per Kristian Lehre. Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, pages 62–68, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3434-1. doi: 10.1145/2725494.2725508. URL http://doi.acm.org/10.1145/2725494.2725508.

Duc-Cuong Dang and Per Kristian Lehre. Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica*, 75(3):428–461, 2016. doi: 10.1007/s00453-015-0103-x. URL https://doi.org/10.1007/s00453-015-0103-x.

Duc-Cuong Dang, Tobias Friedrich, Timo Kötzing, Martin S. Krejca, Per Kristian Lehre, Pietro Simone Oliveto, Dirk Sudholt, and Andrew M. Sutton. Escaping local optima using crossover with emergent diversity. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2017. ISSN 1089-778X. doi: 10.1109/TEVC.2017.2724201.

Agoston E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003. ISBN 3540401849.

References II

David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.

Jun He and Xin Yao. Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 145(1-2):59–97, 2003.

Thomas Jansen, Kenneth A. De Jong, and Ingo Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440, 2005. doi: 10.1162/1063656057774666921.

Per Kristian Lehre. Negative drift in populations. In *Proceedings of Parallel Problem Solving from Nature - (PPSN XI)*, volume 6238 of *LNCS*, pages 244–253. Springer Berlin / Heidelberg, 2011a.

Per Kristian Lehre. Fitness-levels for non-elitist populations. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, (GECCO 2011), pages 2075–2082, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0557-0.

Per Kristian Lehre and Xin Yao. On the impact of mutation-selection balance on the runtime of evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 16(2):225 –241, April 2012. ISSN 1089-778X. doi: 10.1109/TEVC.2011.2112665.

Frank Neumann, Pietro Simone Oliveto, and Carsten Witt. Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation (GECCO 2009)*, pages 835–842, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9. doi: http://doi.acm.org/10.1145/1569901.1570016.

Pietro S. Oliveto and Carsten Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 545:2–19, 2014.

Pietro S. Oliveto and Carsten Witt. Improved time complexity analysis of the simple genetic algorithm. *Theoretical Computer Science*, 605:21–41, 2015.

Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the $(1,\lambda)$ ea. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, GECCO '12, pages 1349–1356, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1177-9.

Carsten Witt. Runtime Analysis of the $(\mu + 1)$ EA on Simple Pseudo-Boolean Functions. *Evolutionary Computation*, 14(1):65–86, 2006.

Christine Zarges. On the utility of the population size for inversely fitness proportional mutation rates. In *FOGA 09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 39–46, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-414-0. doi: http://doi.acm.org/10.1145/1527125.1527132.