“Global” Analyses

Summer Lecture Series 2002

Thomas Jansen

`tjansen.gmu.edu or Thomas.Jansen@cs.uni-dortmund.de`

Before we start: A Word on “Global”

Global can have many different meanings.

We use it in the sense of global in time.

Why?

Because it is shorter than

concerned with a substantial part of a run and not only with a very limited number of steps.

Nothing else is intended.

Overview

- Convergence
- Expected Optimization Time
- Local vs. Global Analysis
- Conclusions

Overview

- Convergence
  - What is Convergence?
  - What is Convergence to a Function Value?
  - What is Gene Convergence?
  - What is Premature Convergence?
  - Is a GA a Function Optimizer?
- Expected Optimization Time
- Local vs. Global Analysis
- Conclusions

Overview

- Convergence
- Expected Optimization Time
  - What is Expected Optimization Time?
  - Why should we care?
  - Proof Methods
- Local vs. Global Analysis
- Conclusions

Overview

- Convergence
- Expected Optimization Time
- Local vs. Global Analysis
  - Local Performance Measures
  - Example: Local Measures Can Be Misleading
- Conclusions

Overview

- Convergence
- Expected Optimization Time
- Local vs. Global Analysis
- Conclusions
  - Summary
  - “Take Home Message”

Convergence — What is Convergence?

"Convergence" as known from analysis not sufficient.

(X_n) sequence of random variables
L random variable

When do we say that (X_n) converges to L?

Convergence — What is Convergence?

"Convergence" as known from analysis not sufficient.

(X_n) sequence of random variables
L random variable

When do we say that (X_n) converges to L?

(X_n) converges almost surely to L

$:\Leftrightarrow$ Prob $\left(\lim_{n\to\infty} |X_n - L| = 0\right) = 1$

Other definitions of convergence known.

Convergence — What is Convergence to an f-Value?

Consider any evolutionary algorithm.
F_t: best function value in the t-th generation

(F_t) is sequence of random variables

constant c is (degenerated) random variable

Thus: We can ask whether (F_t) converges to c.

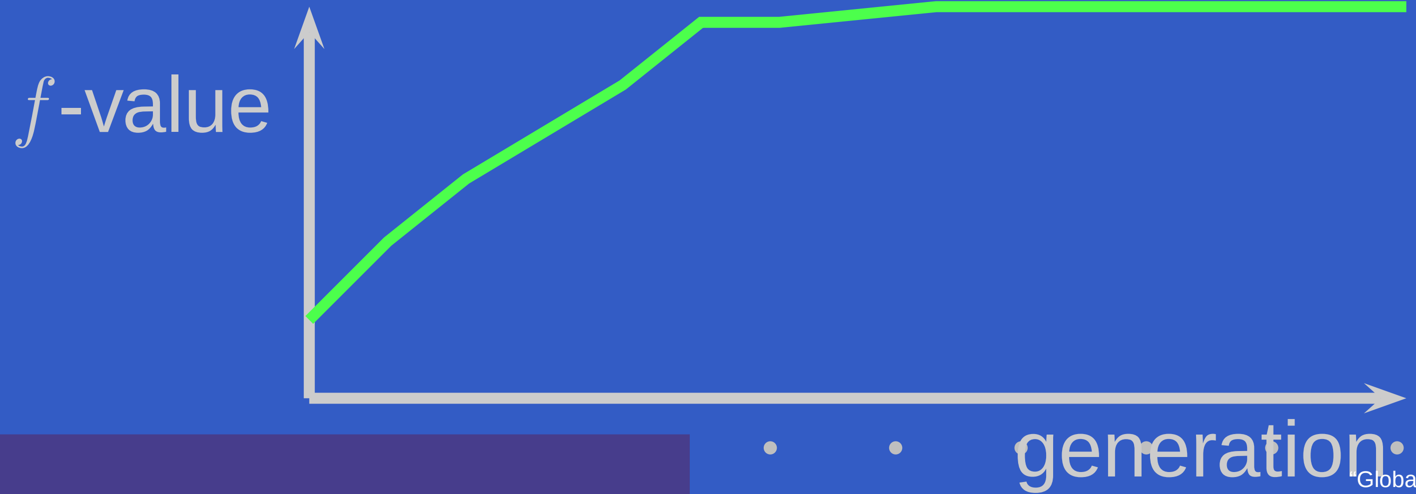
Convergence — What is Convergence to an f-Value?

Consider any evolutionary algorithm.
F_t: best function value in the t-th generation

(F_t) is sequence of random variables

constant c is (degenerated) random variable

Thus: We can ask whether (F_t) converges to c.



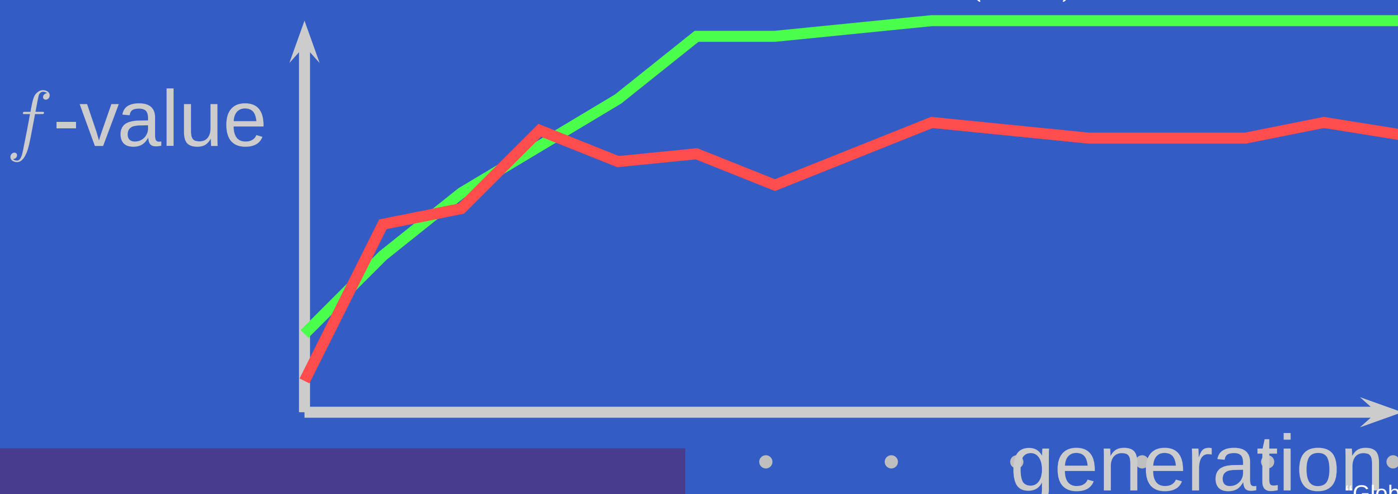
Convergence — What is Convergence to an f-Value?

Consider any evolutionary algorithm.
F_t: best function value in the t-th generation

(F_t) is sequence of random variables

constant c is (degenerated) random variable

Thus: We can ask whether (F_t) converges to c.



Convergence — What is Convergence to an f-Value?

Consider any evolutionary algorithm.
F_t: best function value in the t-th generation

(F_t) is sequence of random variables

constant c is (degenerated) random variable

Thus: We can ask whether (F_t) converges to c.

Convergence — What is Gene Convergence?

Consider EA with population size $\mu > 1$
and binary representation with length n.

$$\forall i \in \{1, \ldots, n\}\colon\ B_t^{(i)} := \left(\sum_{j=1}^{\mu} x_t^{(i)}[j] \right) / \mu$$

average bit value at position i in generation t

$(B_t^{(i)})$ is random variable.

Convergence — What is Gene Convergence?

Consider EA with population size $\mu > 1$
and binary representation with length n.

$$\forall i \in \{1, \ldots, n\} \colon B_t^{(i)} := \left(\sum_{j=1}^{\mu} x_t^{(i)}[j] \right) / \mu$$

average bit value at position i in generation t

$(B_t^{(i)})$ is random variable.

We say the i-th bit converges if $(B_t^{(i)})$ converges almost surely to 0 or 1.

Convergence — What is Gene Convergence?

We say the population converges if all bits converge.

Convergence — What is Gene Convergence?

We say the population converges if all bits converge.

Caution: Often:

"i-th bit is converged" if $B_t^{(i)} \in \{0, 1\}$ for some t.

"population is converged" if this holds for all $B_t^{(i)}$.

Convergence — What is Gene Convergence?

We say the population converges if all bits converge.

Caution: Often:

"i-th bit is converged" if $B_t^{(i)} \in \{0,1\}$ for some t.

"population is converged" if this holds for all $B_t^{(i)}$.

In GAs mutation sometimes considered not important.

Crossover cannot change anything on converged bits.

Convergence — What is Premature Convergence?

Gene Convergence

Without

Convergence to an Optimal Function Value

is called

Premature Convergence.

Convergence — Is a GA a Function Optimizer?

An algorithm is called a function optimizer
if it optimizes any function with probability 1.

Is a GA a function optimizer?

Convergence — Is a GA a Function Optimizer?

An algorithm is called a function optimizer
if it optimizes any function with probability 1.

Is a GA a function optimizer?

That depends on the GA.

The “canonical GA” is not a function optimizer.

The “canonical GA” maintaining the best solution found is a function optimizer.

Conditions for Convergence (I)

Consider generational GA with population size μ, binary representation, string length n, crossover, mutation, and selection.
Model as Markov chain:

- population plus “best-so-far” is state
- size of state space: $2^{(\mu+1)\cdot n}$
- model crossover, mutation, selection as matrices C, M, S
- get transition matrix as $C \cdot M \cdot S$

Conditions for Convergence (II)

Some definitions: Markov chain with transition matrix P

- For states i, j we say $i \to j$ if there exists $m \in \mathbb{N}$ such that $P^m[i, j] > 0$.
- A state i is essential, if for all states j we have $(i \to j) \Rightarrow (j \to i)$.
- P is irreducible if $\forall$ states i, j we have $i \to j$.
- P is diagonal-positive if all diagonal elements are positive.

Conditions for Convergence (III)

Results:

- $P = CMS$ irreducible and diag.-positive $\Rightarrow$ state i sub-optimal $\Leftrightarrow$ i not essential
- $P = CMS$ with C, M, S diag.-positive and M irreducible $\Rightarrow$ Markov chain converges to global optimum

In simple words:

If a GA keeps track of "best-so-far" and each state is reachable via a sequence of generations, then the GA is a function optimizer.

Overview

- Convergence $\surd$
- Expected Optimization Time
- Local vs. Global Analysis
- Conclusions

Overview

- Convergence √
- Expected Optimization Time
  - What is Expected Optimization Time?
  - Why should we care?
  - Proof Methods
- Local vs. Global Analysis
- Conclusions

What is Expected Optimization Time?

Consider EA maximizing some function f.

Let T denote the number of steps (function evaluations) before some x with $f(x) = \max\{f(y)\}$ belongs to the current population for the first time.

$\mathrm{E}\,(T)$ is called the expected optimization time.

Why should we care?

Suppose you want to do optimization.

Is Convergence to an Optimal f-Value an Issue?

Why should we care?

Suppose you want to do optimization.

Is Convergence to an Optimal f-Value an Issue?

Perhaps in theory, but not in practice.

Why should we care?

Suppose you want to do optimization.

Is Convergence to an Optimal f-Value an Issue?

Perhaps in theory, but not in practice.

Crucial: When is optimization efficient?

Thus, expected optimization time is most important measure.

Methodology

- Consider simplified algorithms.
- Consider simplified example problems.
- Give bounds for expected optimization time or similar measures.
- Analyze these bounds for growing dimension of the search space.
- Do not use unproven assumptions or further simplifications.

The (1+1) EA

Maximize $f\colon \{0,1\}^n \to \mathbb{R}$.

1. Initialization
   Choose $x \in \{0,1\}^n$ uniformly at random.
2. Mutation
   Create $y \in \{0,1\}^n$ by bit-wise mutation of x with mutation probability $1/n$.
3. Selection
   If $f(y) \geq f(x)$, replace x by y.
4. Continue at 2.

Proof Methods

- f-based partitions: simple, intuitive, sometimes surprisingly powerful
- typical run: generalization of f-based partitions
- expected advance: powerful method for lower bounds
- potential method: powerful, not very intuitive, difficult to use

Method: f-based Partitions (I)

$P_1, P_2, \ldots, P_l$ partition of search space $\{0,1\}^n$ with

- $\forall i \in \{1,\ldots,l\}\colon P_i \neq \emptyset$
- $P_l = \{x \in \{0,1\}^n \mid f(x) = \max\{f(y) \mid y \in \{0,1\}^n\}\}$
- $\forall i \in \{1,\ldots,l-1\}, x \in P_i, y \in P_{i+1}\colon f(x) < f(y)$

Method: f-based Partitions (I)

$P_1, P_2, \ldots, P_l$ partition of search space $\{0,1\}^n$ with

- $\forall i \in \{1, \ldots, l\} \colon P_i \neq \emptyset$
- $P_l = \{x \in \{0,1\}^n \mid f(x) = \max\{f(y) \mid y \in \{0,1\}^n\}\}$
- $\forall i \in \{1, \ldots, l-1\}, x \in P_i, y \in P_{i+1} \colon f(x) < f(y)$

$$s_{x,y} := \left(\frac{1}{n}\right)^{\mathsf{H}(x,y)} \cdot \left(1 - \frac{1}{n}\right)^{n-\mathsf{H}(x,y)}$$

Method: f-based Partitions (II)

Upper bound: $s_i := \min \left\{ \sum_{y \in P_j, j > i} s_{x,y} \mid x \in P_i \right\}$

$$\mathsf{E}\,(T) \leq \sum_{i=1}^{l-1} \frac{1}{s_i}$$

Method: f-based Partitions (II)

Upper bound: $s_i := \min \left\{ \sum_{y \in P_j, j > i} s_{x,y} \mid x \in P_i \right\}$

$$\mathsf{E}(T) \leq \sum_{i=1}^{l-1} \frac{1}{s_i}$$

Lower bound: $S_i := \max \left\{ \sum_{y \in P_j, j > i} s_{x,y} \mid x \in P_i \right\}$

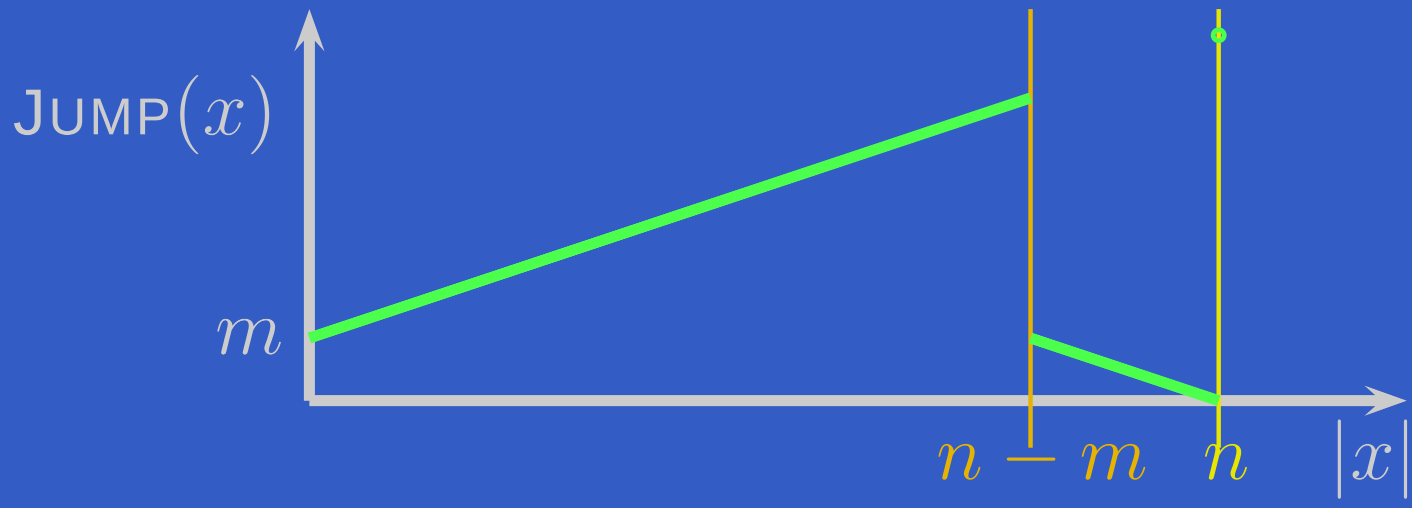
$$\mathsf{E}(T) \geq \max \left\{ \frac{|P_i|}{2^n} \cdot \frac{1}{S_i} \mid 1 \leq i < l \right\}$$

Method: f-based Partitions (III)

Example: Upper bound for JUMP_m, $m \in \{1, \ldots, n\}$

$$\text{JUMP}_m(x) := \begin{cases} m + |x| & \text{if } (|x| \leq n - m) \vee (|x| = n) \\ n - |x| & \text{otherwise} \end{cases}$$

with $|x| = \text{ONEMAX}(x)$



Method: f-based Partitions (IV)

$$i \in \{1, \ldots, n\} \colon P_i := \{x \in \{0,1\}^n \mid \textsf{JUMP}_m(x) = i\}$$

$$P_{n+1} := \{1^n\}$$

Method: f-based Partitions (IV)

$$i \in \{1,\ldots,n\}\colon P_i := \{x \in \{0,1\}^n \mid \textsf{JUMP}_m(x) = i\}$$
$$P_{n+1} := \{1^n\}$$

$$i \notin \{n-m, n\}\colon s_i \geq \binom{n-i}{1}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$
$$s_{n-m} \geq \left(\frac{1}{n}\right)^m \left(1-\frac{1}{n}\right)^{n-m} \geq \frac{1}{en^m}$$

Method: f-based Partitions (IV)

$$i \in \{1, \dots, n\}\colon P_i := \{x \in \{0,1\}^n \mid \textsf{JUMP}_m(x) = i\}$$
$$P_{n+1} := \{1^n\}$$

$$i \notin \{n-m, n\}\colon s_i \geq \binom{n-i}{1} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$
$$s_{n-m} \geq \left(\frac{1}{n}\right)^m \left(1 - \frac{1}{n}\right)^{n-m} \geq \frac{1}{en^m}$$

$$\mathsf{E}\,(T) \leq en^m + \sum_{i=1}^{n} \frac{en}{n-i} = O(n^m + n \log n)$$

Method: Typical Run (I)

Describe a “typical run” in phases by

- partition $P_1, \ldots, P_l$
- upper and lower bounds $t_1, \ldots, t_l, T_1, \ldots, T_l$

Find upper bounds on failure probabilities $p_1, \ldots, p_l$.

Method: Typical Run (I)

Describe a “typical run” in phases by

- partition $P_1, \ldots, P_l$
- upper and lower bounds $t_1, \ldots, t_l, T_1, \ldots, T_l$

Find upper bounds on failure probabilities $p_1, \ldots, p_l$.

$$\text{Prob}\left(T \leq \sum_{i=1}^{l} t_i\right) \leq \sum_{i=1}^{l} p_i \qquad \text{Prob}\left(T \geq \sum_{i=1}^{l} T_i\right) \leq \sum_{i=1}^{l} p_i$$

Method: Typical Run (I)

Describe a "typical run" in phases by

- partition $P_1, \ldots, P_l$
- upper and lower bounds $t_1, \ldots, t_l, T_1, \ldots, T_l$

Find upper bounds on failure probabilities $p_1, \ldots, p_l$.

$$\text{Prob}\left(T \leq \sum_{i=1}^{l} t_i\right) \leq \sum_{i=1}^{l} p_i \qquad \text{Prob}\left(T \geq \sum_{i=1}^{l} T_i\right) \leq \sum_{i=1}^{l} p_i$$

$$\mathrm{E}\left(T\right) \geq \left(1 - \sum_{i=1}^{l} p_i\right) \cdot \sum_{i=1}^{l} t_i$$

Method: Typical Run (II)

Example: Upper bound for Real Royal Road Function R for one-point crossover

$$R_m(x) := \begin{cases} 2n^2 & \text{if } x = 1^n \\ n|x| + b(x) & \text{if } |x| \leq n - m \\ 0 & \text{otherwise} \end{cases}$$

where $|x| = \textsc{OneMax}(x)$ and
$b(x) =$ length of longest block of 1-bits in x

Method: Typical Run (III)

1. Initialization
   Choose pop. of size μ uniformly at random.
2. Crossover
   With prob. p_c create z by one-point crossover, otherwise choose z from population.
3. Mutation
   Create z' by bit-wise mutation of z.
4. Selection
   Add z' to the population. Remove one member with minimal f-value and maximal number of copies.
5. Continue at 2.

Method: Typical Run (IV)

Theorem: For $p_c \leq 1 - \varepsilon$ ($\varepsilon > 0$ constant),
$m \leq \lceil n/3 \rceil$, $\mu \geq m + 1$:
E $(T) = O\left(n^2 \mu m + n^2 \log n + n\mu \log \mu + \mu^2/p_c\right)$

For p_c constant, $\mu = O(n)$: E $(T) = O\left(n^3 \cdot \mu\right)$

Proof Method: Variant of “Typical Run”
Describe typical run with 5 phases.
Find upper bound on expected length for each phase.
Get result by addition.

Method: Typical Run (V)

Phase 1

Assumption: after random initialization
Goal: There is a member of the population with at most $n - m$ ones or exactly n ones.

Expected Length $s(n) + o(1)$

Method: Typical Run (VI)

Phase 2

Assumption: Phase 1 finished
Goal: All members of the population have exactly $n - m$ ones or optimum found.

Expected Length $O(n^2 \cdot \mu/m)$

Proof: Prob (increase number of 1-bits) $= \Omega\left(\frac{m}{en}\right)$

Method: Typical Run (VII)

Phase 3

Assumption: Phase 2 finished
Goal: All members of the population have exactly $n-m$ ones and these ones in one block, or optimum found.

Expected Length: $O(n^2 \log n + n\mu \log \mu)$

Proof: Case 1: All members have $b(x)=i$.
Then 2-bit mutation needed. Prob. for such mutation $\geq (n-m-i)/(en^2)$; sum $O(n^2 \log n)$

Case 2: $\exists x$ with $b(x)=i$ and $j>0$ have larger b-value
Sufficient: Choose one of the j and don't change it.
Prob. for this event $\Omega(\mu/j)$; sum $O(n\mu \log \mu)$.

Method: Typical Run (VIII)

Phase 4

Assumption: Phase 3 finished
Goal: all possible x with $|x| = n - m$ and $b(x) = n - m$ in population, or optimum found

Expected Length: $O(n^2 \mu m)$
Proof: essential: $\mu \geq m + 1 =$#different such strings
2-bit-mutation sufficient, Prob. $= \Omega(1/n^2)$
Prob. choose appropriate parent $\geq 1/\mu$
m such events sufficient

Method: Typical Run (IX)

Phase 5

Assumption: Phase 4 finished
Goal: find optimum

Expected Length: $O(\mu^2/p_c)$
Proof:
Prob (choose $1^{n-m}0^m$ and 0^m1^{n-m} for crossover) $\geq p_c/\mu^2$
Prob (choose crossover position “in the middle”) $\geq 1/3$

$\square$

Method: Expected Advance (I)

Define measure of advance F (f-value, Hamming distance, ...)

F_t: advance after t steps

$$\begin{aligned} \mathsf{E}(T) &\geq t \cdot \mathsf{Prob}(T \geq t) \\ &= t \cdot \mathsf{Prob}(F_t \leq \Delta) \\ &= t \cdot (1 - \mathsf{Prob}(F_t > \Delta)) \end{aligned}$$

Method: Expected Advance (I)

Define measure of advance F (f-value, Hamming distance, ...)
F_t: advance after t steps

$$\begin{aligned} \mathsf{E}(T) &\geq t \cdot \mathsf{Prob}(T \geq t) \\ &= t \cdot \mathsf{Prob}(F_t \leq \Delta) \\ &= t \cdot (1 - \mathsf{Prob}(F_t > \Delta)) \end{aligned}$$

Markov: $\mathsf{Prob}(F_t > \Delta) \leq \mathsf{E}(F_t)/\Delta$

$$\mathsf{E}(T) \geq t \cdot \left(1 - \frac{\mathsf{E}(F_t)}{\Delta}\right)$$

Method: Expected Advance (II)

Example: Lower bound for long-path function

Definition: $n \geq 1$, $k > 1$, $(n-1)/k \in \mathbb{N}$
long k-path of dimension 1: $P_1^k := (0, 1)$
long k-path of dimension $n-k$:
$P_{n-k}^k = (v_1, \ldots, v_l)$
long k-path of dimension n:

$$
\begin{aligned}
P_n^k := (&0^k v_1, 0^k v_2, \ldots, 0^k v_l,\\
&0^{k-1} 1 v_l, 0^{k-2} 1^2 v_l, \ldots, 01^{k-1} v_l, 1^k v_l,\\
&1^k v_{l-1}, 1^k v_1)
\end{aligned}
$$

Method: Expected Advance (III)

Path Properties:
P_n^k contains $(k+1)2^{(n-1)/k} - k + 1$ different points.

$\forall i \in \{1, \ldots, k-1\}:$
x with at least i successors on path:
i-th successor has Hamming distance i and all other points have different Hamming distances

Definition: $\textsc{Path}_k(x) :=$

$$\begin{cases} n^2 + l & \text{if } x \text{ is } l\text{-th point of } P_n^k \\ n^2 - n\sum_{i=1}^{k} x_i - \sum_{i=k+1}^{n} x_i & \text{if } x \notin P_n^k \end{cases}$$

Method: Expected Advance (IV)

Theorem: $(1+1)$ EA on $\text{PATH}_{\sqrt{n-1}}$
$\mathrm{E}(T) = \Omega\left(n^{3/2} \cdot 2^{\sqrt{n}}\right)$

Proof Method:
Prob (first path point not after “bridge”) $\geq \frac{1}{2}$

Give lower bound on expected optimization time when first point is not after the “bridge” with method of expected advance.

Method: Expected Advance (V)

T_i: optimization time when started in i-th path point

E_t: "in t steps no mutation of $\geq \sqrt{n-1}$ bits"

Method: Expected Advance (V)

T_i: optimization time when started in i-th path point

E_t: "in t steps no mutation of $\geq \sqrt{n-1}$ bits"

Prob $(E_t) \geq 1 - t \cdot n^{-\sqrt{n-1}}$

Method: Expected Advance (VI)

d: distance between starting point and optimum

a_j: distance between current point and starting point after j generations

Method: Expected Advance (VI)

d: distance between starting point and optimum
a_j: distance between current point and starting point after j generations

$$\begin{aligned} \mathsf{E}\left(T_i\right) &\geq \mathsf{Prob}\left(E_t\right) \cdot \mathsf{E}\left(T_i \mid E_t\right) \\ &\geq \mathsf{Prob}\left(E_t\right) \cdot t \cdot \mathsf{Prob}\left(a_t < d \mid E_t\right) \\ &= \mathsf{Prob}\left(E_t\right) \cdot t \cdot \left(1 - \mathsf{Prob}\left(a_t \geq d | E_t\right)\right) \end{aligned}$$

Markov:

$$\mathsf{E}\left(T_i\right) \geq \mathsf{Prob}\left(E_t\right) \cdot t \cdot \left(1 - \frac{\mathsf{E}\left(a_t \mid E_t\right)}{d}\right)$$

Method: Expected Advance (VII)

Due to Path Property:

$\mathsf{E}\left(a_t \mid E_t\right) \leq t \cdot \mathsf{E}\left(a_1 \mid E_t\right)$

$$\mathsf{E}\left(T_i\right) \geq t \cdot \mathsf{Prob}\left(E_t\right) \cdot \left(1 - \frac{t \cdot \mathsf{E}\left(a_1 \mid E_t\right)}{d}\right)$$

Method: Expected Advance (VIII)

$$
\begin{aligned}
\mathsf{E}\left(a_1 \mid E_t\right) &= \sum_{i=1}^{\sqrt{n-1}-1} i \cdot \mathsf{Prob}\left(a_1 = i\right) \\
&\leq \sum_{i=1}^{\sqrt{n-1}-1} \frac{i}{n^i} < \sum_{i=1}^{\infty} \frac{i}{n^i} \\
&= \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \frac{1}{n^j} = \frac{n}{(n-1)^2} \leq \frac{2}{n}
\end{aligned}
$$

Method: Expected Advance (VIII)

$$
\begin{aligned}
\mathsf{E}\left(a_1 \mid E_t\right) &= \sum_{i=1}^{\sqrt{n-1}-1} i \cdot \mathsf{Prob}\left(a_1 = i\right) \\
&\leq \sum_{i=1}^{\sqrt{n-1}-1} \frac{i}{n^i} < \sum_{i=1}^{\infty} \frac{i}{n^i} \\
&= \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \frac{1}{n^j} = \frac{n}{(n-1)^2} \leq \frac{2}{n}
\end{aligned}
$$

Plugging in with $t = n^{3/2} \cdot 2^{\sqrt{n}-5}$ yields result. $\square$

Method: Potential Method (I)

Define potential function $\Phi\colon \{0,1\}^n \to \mathbb{R}$ with

$$
\begin{array}{cl}
 & \{x \in \{0,1\}^n \mid \Phi(x) = \max\{\Phi(y) \mid y \in \{0,1\}^n\}\} \\
\begin{array}{c}\supseteq \\ \subseteq\end{array} & \{x' \in \{0,1\}^n \mid f(x') = \max\{f(y') \mid y' \in \{0,1\}^n\}\}
\end{array}
$$

Analyze EA on Φ but with acceptance given by f.

Method: Potential Method (II)

Example: linear functions
Obvious: Suffices to analyze linear functions with positive weights.

$$\Phi(x) := \sum_{i=1}^{n/2} x_i + \sum_{i=(n/2)+1}^{n} x_i$$

generation “successful” $\Leftrightarrow$ child accepted and different from parent
Find upper bound for number of successful generations until $\Phi(x)$ grows.

Overview

- Convergence $\surd$
- Expected Optimization Time $\surd$
- Local vs. Global Analysis
- Conclusions

Overview

- Convergence √
- Expected Optimization Time √
- Local vs. Global Analysis
  - Local Performance Measures
  - Example: Local Measures Can Be Misleading
- Conclusions

Local Performance Measures

- Local performance measures are often easier to estimate.
- Often they come with the promise of "global" predictions via repetition.
- They seem to allow easy comparisons in special cases.

Local Performance Measures

Quality Gain:

$$Q_f^{\text{(1+1) EA}}(x) = \mathsf{E}\left(f\left(x'\right) - f\left(x\right)\right)$$

x current string, x' next current string

Progress Rate:

$$r_f^{\text{(1+1) EA}}(x) = \mathsf{E}\left(\mathsf{H}\left(x, x_{\text{opt}}\right) - \mathsf{H}\left(x', x_{\text{opt}}\right)\right)$$

x current string, x' next current string

Local Performance Measures

Quality Gain:

$$Q_f^{\text{(1+1) EA}}(x) = \mathrm{E}\left(f(x') - f(x)\right)$$

x current string, x' next current string

Progress Rate:

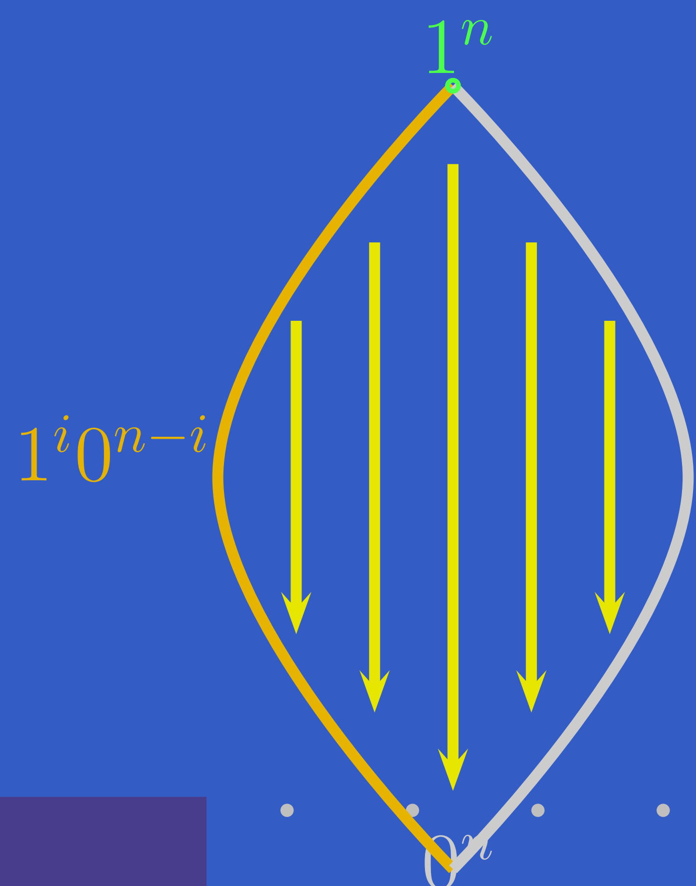
$$r_f^{\text{(1+1) EA}}(x) = \mathrm{E}\left(\mathrm{H}\left(x, x_{\text{opt}}\right) - \mathrm{H}\left(x', x_{\text{opt}}\right)\right)$$

x current string, x' next current string

Obvious: “bigger is better”

(Counter-)Example Function

$$\text{SPC}(x) := \begin{cases} 2n & \text{if } x = 1^n \\ n & \text{if } x = 1^i 0^{n-i}, 0 \le i < n \\ n - \textsc{OneMax}(x) & \text{otherwise} \end{cases}$$



The (1+1)*EA

Maximize $f\colon \{0,1\}^n \to \mathbb{R}$.

1. Initialization
   Choose $x \in \{0,1\}^n$ uniformly at random.
2. Mutation
   Create $y \in \{0,1\}^n$ by bit-wise mutation of x with mutation probability $1/n$.
3. Selection
   If $f(y) > f(x)$, replace x by y.
4. Continue at 2.

Optimization of SPC

(1+1) EA:

E $(T) = O\left(n^3\right)$

success probability in n^4 generations $\geq 1 - e^{-\Omega(n)}$

(1+1)*EA:

E $(T) = n^{\Omega(n)}$

success probability in $n^{n/2}$ generations $\leq e^{-\Omega(n)}$

Quality Gain on SPC

$$Q_{\text{SPC}}^{\text{(1+1)}^*\text{EA}}(x) \;=\; \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') > \text{SPC}(x)}} s_{x,x'} \cdot \left(\text{SPC}(x') - \text{SPC}(x)\right)$$

Quality Gain on SPC

$$Q_{\text{SPC}}^{\text{(1+1)}^*\text{EA}}(x) = \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') > \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x))$$

$$\begin{aligned} Q_{\text{SPC}}^{\text{(1+1) EA}}(x) &= \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') \geq \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x)) \\ &= \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') > \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x)) \end{aligned}$$

Quality Gain on SPC

$$Q_{\text{SPC}}^{\text{(1+1)}^*\text{EA}}(x) = \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') > \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x))$$

$$\begin{aligned} Q_{\text{SPC}}^{\text{(1+1) EA}}(x) &= \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') \geq \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x)) \\ &= \sum_{\substack{x' \in \{0,1\}^n \\ \text{SPC}(x') > \text{SPC}(x)}} s_{x,x'} \cdot (\text{SPC}(x') - \text{SPC}(x)) \end{aligned}$$

$$\Rightarrow \forall x \in \{0,1\}^n : Q_{\text{SPC}}^{\text{(1+1)}^*\text{EA}}(x) = Q_{\text{SPC}}^{\text{(1+1) EA}}(x)$$

Progress Rate on SPC (I)

For all points not on the plateau:

(1+1) EA and $(1+1)^*$EA accept the same strings.

$\Rightarrow$ progress rates equal

Progress Rate on SPC (I)

For all points not on the plateau:

(1+1) EA and (1+1)*EA accept the same strings.

$\Rightarrow$ progress rates equal

On the plateau:

(1+1)$^*EA$ only accepts direct jump to 1^n.

(1+1) EA accepts any step on the plateau.

Progress Rate on SPC (II)

On the plateau:

$$
\begin{aligned}
r^{\text{(1+1)}^*\text{EA}}_{\text{SPC}}\left(1^i 0^{n-i}\right) &= s_{1^i 0^{n-i}, 1^n} \cdot \mathsf{H}\left(1^i 0^{n-i}, 1^n\right) \\
&= \left(\frac{1}{n}\right)^{n-i} \cdot \left(1 - \frac{1}{n}\right)^{i} \cdot (n-i) \\
&\leq \frac{n-i}{n^{n-i}} \in \left\{\frac{1}{n}, \frac{2}{n^2}, \ldots, \frac{n}{n^n}\right\}
\end{aligned}
$$

Progress Rate on SPC (III)

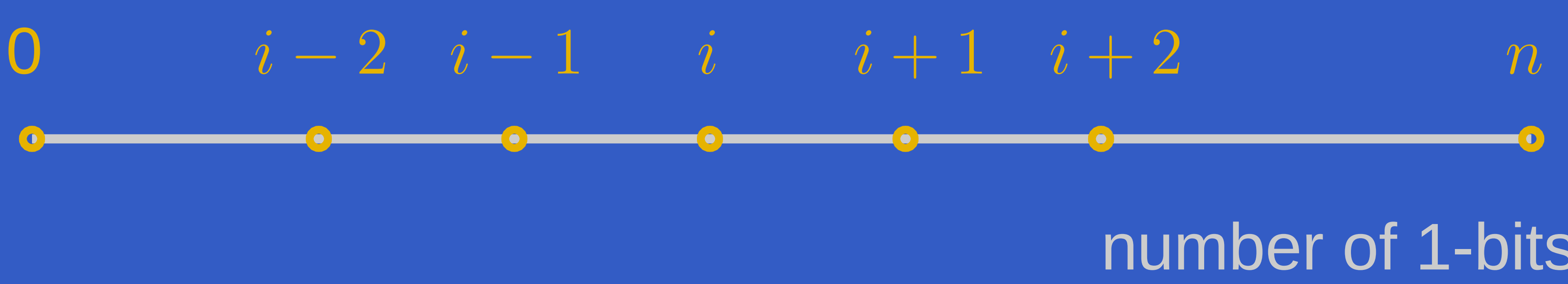
On the plateau:

$$r_{\text{SPC}}^{\text{(1+1) EA}}\left(1^i 0^{n-i}\right) = \sum_{j=0}^{n} s_{1^i 0^{n-i},1^j 0^{n-j}} \cdot (j-i)$$

Progress Rate on SPC (III)

On the plateau:

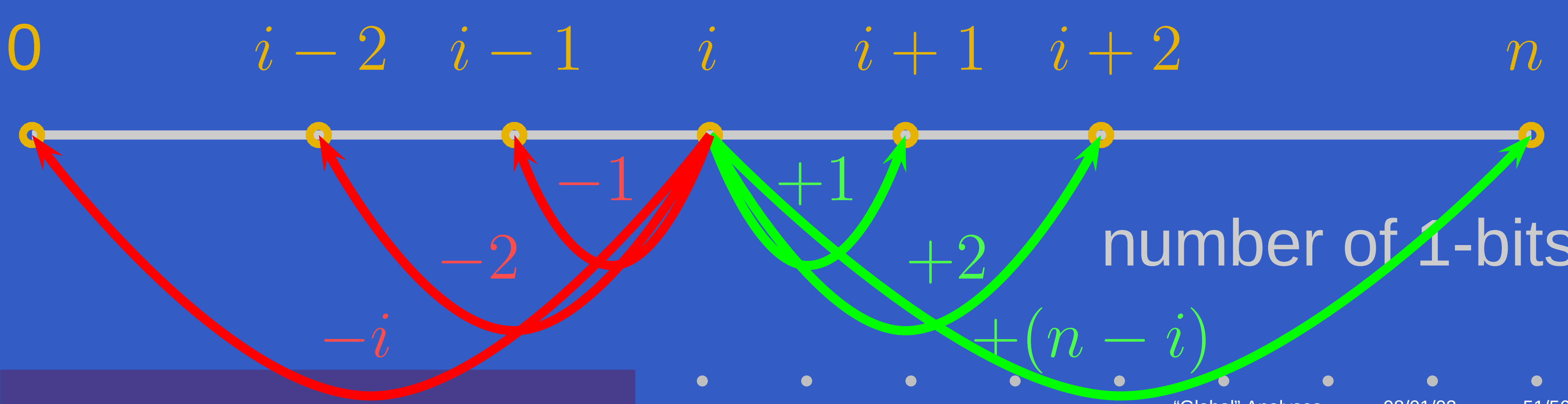
$$r_{\text{SPC}}^{\text{(1+1) EA}}\left(1^i 0^{n-i}\right) = \sum_{j=0}^{n} s_{1^i 0^{n-i},1^j 0^{n-j}} \cdot (j-i)$$



Progress Rate on SPC (III)

On the plateau:

$$r_{\text{SPC}}^{\text{(1+1) EA}}\left(1^i0^{n-i}\right) = \sum_{j=0}^{n} s_{1^i0^{n-i},1^j0^{n-j}} \cdot (j-i)$$



Progress Rate on SPC (IV)

On the plateau:

In the half far from 1^n ($i < n/2$):

$r_{\text{SPC}}^{\text{(1+1) EA}}\left(1^i 0^{n-i}\right) > 0$

only slightly larger then for the (1+1)*EA

In the half near to 1^n ($i > n/2$):

$r_{\text{SPC}}^{\text{(1+1) EA}}\left(1^i 0^{n-i}\right) < 0$

clearly smaller then for the (1+1)*EA

Progress Rate on SPC (IV)

On the plateau:

In the half far from 1^n $(i < n/2)$:

$r^{\text{(1+1) EA}}_{\text{SPC}}\left(1^i 0^{n-i}\right) > 0$

only slightly larger then for the (1+1)*EA

In the half near to 1^n $(i > n/2)$:

$r^{\text{(1+1) EA}}_{\text{SPC}}\left(1^i 0^{n-i}\right) < 0$

clearly smaller then for the (1+1)*EA

Even more misleading then quality gain.

Overview

- Convergence √
- Expected Optimization Time √
- Local vs. Global Analysis √
- Conclusions

Overview

- Convergence √
- Expected Optimization Time √
- Local vs. Global Analysis √
- Conclusions
  - Summary
  - “Take Home Message”

Conclusions — Summary

- most EAs converge to global optimum
- expected optimization time is important measure
- different proof methods and analytical tools available
- stronger methods for populations and crossover needed
- local measures can be very misleading

Conclusions — Take Home Message

- Ask yourself what you are really interested in.

Conclusions — Take Home Message

- Ask yourself what you are really interested in.
- EAs can be analyzed like randomized algorithms.

Conclusions — Take Home Message

- Ask yourself what you are really interested in.
- EAs can be analyzed like randomized algorithms.
- Keep your algorithm and your problem as simple as possible.

Conclusions — Take Home Message

- Ask yourself what you are really interested in.
- EAs can be analyzed like randomized algorithms.
- Keep your algorithm and your problem as simple as possible.
- Try to solve challenging problems.

Conclusions — Take Home Message

- Ask yourself what you are really interested in.
- EAs can be analyzed like randomized algorithms.
- Keep your algorithm and your problem as simple as possible.
- Try to solve challenging problems.
- Be realistic about what you can do.

Conclusions — Take Home Message

- Ask yourself what you are really interested in.
- EAs can be analyzed like randomized algorithms.
- Keep your algorithm and your problem as simple as possible.
- Try to solve challenging problems.
- Be realistic about what you can do.
- Remember: Proofs are nice, even in application oriented work.

Overview

- Convergence √
- Expected Optimization Time √
- Local vs. Global Analysis √
- Conclusions √