Evolutionary Algorithms

We know

- what evolutionary algorithms are and
- how we can design evolutionary algorithms.

What do we want to do now?

What do we do if we design a problem-specific algorithm?

1. prove its correctness
2. analyze its performance: (expected) run time

What does this mean for evolutionary algorithms in the context of optimization?

1. prove that max. f-value in population converges to global max. of f for $t \to \infty$
2. analyze how long this takes on average: expected optimization time

Analysis of Evolutionary Algorithms

What kind of evolutionary algorithms do we want to analyze?

clearly all kinds of evolutionary algorithms

more realistic very simple evolutionary algorithms
at least as starting point

For what kind of problems do we want to do analysis?

clearly all kinds of problems

more realistic very simple problems — “toy problems”
at least as starting point

On "Toy Problems"

better term example problems

Why should we care?

- support analysis, help to develop analytical tools
- are easy to understand, are clearly structured
- present typical situations in a paradigmatic way
- make important aspects visible
- act as counter examples
- help to discover general properties
- are important tools for further design and analyis

Upper bounds with f-based partitions

Method of f-based partitions works well with plus-selection.

Definition

Let $f\colon \{0,1\}^n \to \mathbb{R}$. A partition $L_0, L_1, \ldots, L_k$ of $\{0,1\}^n$ is called f-based partition iff the following holds.

1. $\forall i, j \in \{0, \ldots, k\} \colon \forall x \in L_i \colon \forall y \in L_j \colon \ (i < j \Rightarrow f(x) < f(y))$
2. $L_k = \{x \in \{0,1\}^n \mid f(x) = \max\{f(y) \mid y \in \{0,1\}^n\}\}$

Often the trivial f-based parition works well.

$k := |\{f(x) \mid x \in \{0,1\}^n\}| - 1$

$\{f(x) \mid x \in \{0,1\}^n\} = \{f_0, f_1, \ldots, f_k\}$ with $f_0 < f_1 < \cdots < f_k$

for $i \in \{0, 1, \ldots, k\} \colon L_i := \{x \in \{0,1\}^n \mid f(x) = f_i\}$

Example: (1+1) EA on OneMax

$$\textsc{OneMax}\colon \{0,1\}^n \to \mathbb{R} \text{ with } \textsc{OneMax}(x) := \sum_{i=1}^{n} x_i$$

The (1+1) EA

1. **Initialization**
   Choose $x \in \{0,1\}^n$ uniformaly at random.
2. **Mutation**
   $y :=$ mutate(x); (standard bit mutations, $p_m = 1/n$)
3. **Selection**
   If $f(y) \geq f(x)$, Then $x := y$.
4. **"Stoppping Criterion"**
   Continue at line 2.

Method: f-based partitions

Key Observation:
(1+1) EA leaves each fitness layer at most once.

Lower bound on the probability to leave L_i:

$$s_i := \min_{x \in L_i} \sum_{j=i+1}^{k} \sum_{y \in L_j} p_m^{\mathsf{H}(x,y)} \cdot (1-p_m)^{n-\mathsf{H}(x,y)}$$

Upper bound on the expected time needed to leave L_i:
$\mathsf{E}\,(\text{time to leave } L_i) \leq 1/s_i$

Upper bound on the expected optimization time:

$$\mathsf{E}\left(T_{\text{(1+1) EA},f}\right) \leq \sum_{i=0}^{k-1} 1/s_i$$

Upper Bound: (1+1) EA on OneMax

Use trivial OneMax-based partition.

To leave L_i, flip exactly 1 out of $n-i$ 0-bits.

$s_i \geq \binom{n-i}{1} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$

$$\begin{aligned} \mathsf{E}\left(T_{\text{(1+1) EA},\textsc{OneMax}}\right) &\leq \sum_{i=0}^{n-1} \frac{en}{n-i} = en \cdot \sum_{i=1}^{n} \frac{1}{i} \\ &< en\ln(n) + en \\ &= O(n \log n) \end{aligned}$$

Linear Functions

Observation $\textsc{OneMax}(x) = \sum_{i=1}^{n} x[i]$

is of the form $f(x) = w_0 + \sum_{i=1}^{n} w_i \cdot x[i]$

Definition $f\colon \{0,1\}^n \to \mathbb{R}$ is called linear

if f is of the form $f(x) = w_0 + \sum_{i=1}^{n} w_i \cdot x[i]$

Are all linear functions like $\textsc{OneMax}$?

Definition different extreme example

$\textsc{BinVal}\colon \{0,1\}^n \to \mathbb{R}$ with

$$\textsc{BinVal}(x) = \sum_{i=1}^{n} 2^{n-i} \cdot x[i]$$

Upper bound for $\mathrm{E}\left(T_{(1+1)\ \mathrm{EA},\textsc{BinVal}}\right)$

Consider trivial fitness levels

$\forall i \in \{0, 1, \ldots, 2^n - 1\} : L_i := \{x \in \{0,1\}^n \mid \textsc{BinVal}(x) = i\}$

without considering s_i at best upper bound $\geq 2^n - 1$ achievable

Observation for good upper bounds number of fitness levels needs to be small

Try more clever fitness levels

$\forall i \in \{0, 1, \ldots, n-1\}$:

$$L_i := \left\{ x \in \{0,1\}^n \setminus \left(\bigcup_{j=0}^{i-1} L_j \right) \mid \textsc{BinVal}(x) < \sum_{j=0}^{i} 2^{n-1-j} \right\}$$

Upper bound for $\mathsf{E}\left(T_{(1+1)\ \mathsf{EA},\textsc{BinVal}}\right)$ (II)

$$\forall i \in \{0, 1, \ldots, n-1\}\colon$$
$$L_i := \left\{ x \in \{0,1\}^n \setminus \left(\bigcup_{j=0}^{i-1} L_j \right) \mid \textsc{BinVal}(x) < \sum_{j=0}^{i} 2^{n-1-j} \right\}$$

obvious $\quad s_i \geq \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{en}$

Theorem $\quad \mathsf{E}\left(T_{(1+1)\ \mathsf{EA},\textsc{BinVal}}\right) \leq en^2$

$\square$

Upper bounds for linear functions

Theorem f linear $\Rightarrow \mathrm{E}\left(T_{(1+1)\text{ EA},f}\right) = O(n^2)$

Proof $f(x) = \sum_{i=1}^{n} w_i x[i]$ mit $w_1 \geq w_2 \geq \cdots \geq w_n$

Definition fitness levels for $i \in \{0, 1, \ldots, n-1\}$

$$L_i := \left\{ x \in \{0,1\}^n \setminus \left(\bigcup_{j=0}^{i-1} L_j \right) \mid f(x) < \sum_{j=1}^{i+1} w_j \right\}$$

$$L_n := \{1^n\}$$

Observation in order to leave L_i:
sufficient to mutate left-most 0-bit

thus $\mathrm{E}\left(T_{(1+1)\text{ EA},f}\right) \leq en^2$ □