Analysis of a Simple Evolutionary Algorithm for Minimization in Euclidean Spaces

Jens Jägersküpper

Dept. of Computer Science
University of Dortmund
Germany

Evolutionary Algorithms (EAs)

- **heuristic approach to optimization**
- based upon the idea to “evolve” one or several (candidate) solutions in an optimization task — hopefully, towards a global optimum
- mimic mutation, recombination, and selection
- often applied in practical optimization, typically to engineering tasks
- **unfortunately, almost never designed with their (theoretical) analysis in mind**

Black-Box Optimization (BBO)

- usually, EAs just evaluate the function f to be minimized
- no further conditions (like differentiability) are put on f

⇒ "Black-Box Optimization" scenario:

knowledge about f can solely be gathered by f-evaluations at certain points in the search space

- if f is given only implicitly (e. g. by simulations), BBO algorithms are mandatory
- in BBO "**runtime**" is defined as the **# f-evaluations**

Evolution Strategies (ESs)

- commonly subsume **EAs for continuous optimization**
- in a more specific sense:
  a certain kind of EAs developed for $\mathbb{R}^n$
  by Rechenberg, Schwefel et al. in the 1960s
- applied in real-world optimization — often successfully
- originally for unconstrained optimization
  (constraints $\longrightarrow$ penalty function)

Practice

- we cannot expect BBO methods (like ESs) to compete with classical optimization methods (like Newton's)
- classical methods often cannot be applied in practice due to missing knowledge about f
- EAs, in particular ESs, are applied . . .

  sometimes successfully and sometimes without success

⇒ when and why do EAs (not) work?

⇒ **we need a theory of EAs**

Theory

- up to now results on EAs for continuous optimization
  - either are of empirical nature (experimental)
  - or use a simplified model of the stochastic process
  - or deal with (global) convergence
- **EAs should be analyzed as algorithms**

⇒ **how does the runtime (asymptotically) depend on n** w. r. t. the approximation quality

- for discrete search spaces, this approach has been taken successfully since the mid 1990s
- now: **first result on an EA for continuous optimization**

Overview

- the concrete EA investigated
- the function scenario
- the results obtained
- rough idea of one of the proofs

(1+1) ES minimizing $f : \mathbb{R}^n \to \mathbb{R}$

- uses a single-individual population
- applies solely mutation to generate a single offspring
- uses “elitist selection”

initialize $c \in \mathbb{R}^n$ with a starting point
REPEAT
1. choose the mutation vector $m \in \mathbb{R}^n$ randomly
2. generate the mutant: $x := c + m$
3. selection: if $f(x) \leq f(c)$ then $c := x$

UNTIL stopping criterion met

⇒ **the (1+1) ES is a randomized hill-climber**

(1+1) ES minimizing $f : \mathbb{R}^n \to \mathbb{R}$

- the stopping criterion is important in practice,
- yet we investigate the (1+1) ES as an infinite process as
- **we are interested in the asymptotic runtime w. r. t. the quality of the approximation**,

  namely, in the RV defined as the # steps until c is in the ε-neighborhood of the optimum of a unimodal function

Gauss Mutations

- date back to the 1960s (Rechenberg)
- yet still very common in practice

Definition.
Let each component of $z \in \mathbb{R}^n$ be independently standard normal distributed. Then $m \in \mathbb{R}^n$ is called a “Gauss mutation” if its distribution equals the one of $\lambda \cdot z$ for a $\lambda \in \mathbb{R}_{>0}$.

Proposition.
A Gauss mutation $m \in \mathbb{R}^n$ is **isotropically distributed**, i. e.,
$m/|m|$ is uniformly distributed upon the unit hyper-sphere,
$|m|$ is independent of $m/|m|$.

Mutation Adaptation for the (1+1) ES

- the “strength” of the mutation must somehow scale with the approximation quality

⇒ the distribution of the mutation vector m must be adapted during the optimization process

- rule of thumb: “The closer c is to the optimum,
  the smaller the ‘step length’ $|m|$.”

$|m|$ denotes m’s length in Euclidean space (L_2-norm)

1/5-rule for Mutation Adaptation

- proposed by Rechenberg (mid 1960s)
- based upon experiments and rough calculations
- **idea:** adapt the scalar λ in “$m := \lambda \cdot z$”
  according to the progress of the optimization
- (here) the 1/5-rule reads

  “Keep λ unchanged for n steps;
  if in more than $n/5$ of the respective last n steps
  ‘$c := x$’ has been executed (‘successful mutation’),
  then λ is doubled, otherwise λ is halved.”

Function Scenario

- for a reasonable theoretical analysis, assumptions on f must be made
- the probably most often investigated function in the field of EAs is SPHERE(c) := $\sum_{i=1}^{n} c_i^{\,2} = |c|^2$
- the results hold for all unimodal $f : \mathbb{R}^n \to \mathbb{R}$ satisfying

$$|x - x_{\mathrm{opt}}| < |y - x_{\mathrm{opt}}| \quad \Rightarrow \quad f(x) < f(y)$$

- the concrete scenario to be investigated:

> **(1+1) ES**
> **minimizing f (e. g. SPHERE)**
> **using Gauss mutations**
> **adapted according to the 1/5-rule**

Results

Theorem (Upper Bound).

Given an initialization satisfying $\lambda^{(0)} = \Theta(|c^{(0)} - x_{\mathrm{opt}}|/n)$,

in the considered scenario the expected runtime until

$$\frac{|c^{(t)} - x_{\mathrm{opt}}|}{|c^{(0)} - x_{\mathrm{opt}}|} \le 2^{-k} \text{ is } O(k \cdot n) \text{ for } k = \mathrm{poly}(n).$$

Theorem (Lower Bound).

For any mutation adaptation, as long as the (1+1) ES uses

isotropic mutations to minimize f, the expected runtime until

$$\frac{|c^{(t)} - x_{\mathrm{opt}}|}{|c^{(0)} - x_{\mathrm{opt}}|} \le 2^{-k} \text{ is } \Omega(k \cdot n).$$

Results

Corollary (Asymptotic Optimality).

In our function scenario, Gauss mutations in combination with the 1/5-rule result in the (1+1) ES having asymptotically optimal runtime (w. r. t. isotropic mutations).

Corollary ("Robustness").

In the considered scenario, any "α-rule" results in same asymptotic runtime, where

α is a constant in $(0, 1/2)$ and

λ is adapted every $\Theta(n)$ steps

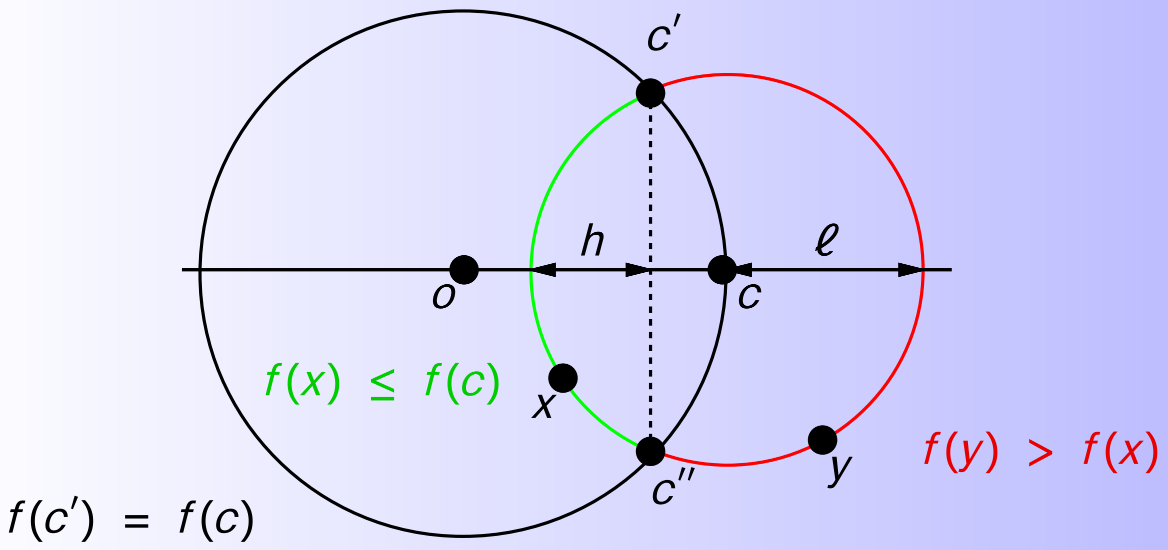
using an arbitrary constant greater than 1 for its scaling.

Techniques applied

- analysis of a single mutation
  - decomposition of m's distribution / deferred decisions
  - Laplace's method
    (asymptotic approximation of integrals)
- analysis of a run of the (1+1) ES
  - Chernoff bounds
    (to show that the adaptation works)
  - a modification of Wald's equation
    (for the lower bound)

Analysis of a single mutation (f = SPHERE)

As m is isotropically distributed, we may assume that ℓ is chosen according to the distribution of $|m|$ first. Then the mutant is uniformly distributed upon the "mutation sphere" $M := \{x \in \mathbb{R}^n \mid |x - c| = \ell\}$



Analysis of a single mutation (2)

$$\text{Prob}\{ f(x) \le f(c) \mid |m| = \ell \} = \frac{(n-1)\text{-volume of part of } M}{(n-1)\text{-volume of } M} = \frac{\int_0^{\arccos(1-h/\ell)} (\sin\beta)^{n-2} d\beta}{\int_0^{\pi} (\sin\beta)^{n-2} d\beta}$$

Lemma 1. For any constant ε such that $0 < \varepsilon < 1/2$

$$\text{Prob}\{ f(x) \le f(c) \} \ge \varepsilon \quad \Leftrightarrow \quad |m| = O(|c|/\sqrt{n})$$

$$\text{Prob}\{ f(x) \le f(c) \} \le \varepsilon \quad \Leftrightarrow \quad |m| = \Omega(|c|/\sqrt{n})$$

$\Rightarrow$ the 1/5-rule aims at $|m| = \Theta(|c|/\sqrt{n})$

Upper bound on the expected runtime

Lemma 2.

$$|m^{(t)}| = \Theta(|c^{(t)}|/\sqrt{n}) \;\Rightarrow\; \text{Prob}\left\{\frac{|c^{(t+1)}|}{|c^{(t)}|} = 1 - \Theta(1/n)\right\} = \Omega(1)$$

Lemma 3.

If the mutation adaptation results in $E[\,|m|\,] = \Theta(|c|/\sqrt{n})$ at the beginning of a phase of n steps, then the distance from the optimum is expected to decrease by a constant fraction.

show that adaptation works (condition in Lemma 3 is met)

$\Rightarrow$ Markov yields the upper bound on the expected runtime

Conclusion

- also for continuous search spaces, an asymptotic analysis of the (expected) runtime of EAs (w. r. t. the approximation quality) is possible
- insight into how EAs work is gained
- hopefully, more results will follow promoting the “de-mystification” of EAs and expanding the theory of EAs, in particular w. r. t. continuous optimization