06-27818, 06-28209, 06-27819, 06-28211

Nature Inspired Search and Optimisation

21 – Runtime Analysis

Jon Rowe

Per Kristian Lehre

Evolutionary Algorithms and Computer Science

Goals of design and analysis of algorithms

1. correctness
   "does the algorithm always output the correct solution?"
2. computational complexity
   "how many computational resources are required?"

For Evolutionary Algorithms (General purpose)

1. convergence
   "Does the EA find the solution in finite time?"
2. time complexity
   "how long does it take to find the optimum?"
   (time = n. of fitness function evaluations)

Brief history

Theoretical studies of Evolutionary Algorithms (EAs), albeit few, have always existed since the seventies [Goldberg, 1989];

- Early studies were concerned with explaining the *behaviour* rather than analysing their performance.
- *Schema Theory* was considered fundamental;
  - First proposed to understand the behaviour of the simple GA [Holland, 1992];
  - It cannot explain the performance or limit behaviour of EAs;
  - Building Block Hypothesis was controversial [Reeves and Rowe, 2002];
- No Free Lunch [Wolpert and Macready, 1997]
  - Over all functions...
- Convergence results appeared in the nineties [Rudolph, 1998];
  - Related to the time limit behaviour of EAs.

Convergence

Definition

- Ideally the EA should find the solution in finite steps with probability **1** (visit the global optimum in finite time);
- If the solution is held forever after, then the algorithm converges to the optimum!

Convergence

Definition

- Ideally the EA should find the solution in finite steps with probability **1** (visit the global optimum in finite time);
- If the solution is held forever after, then the algorithm converges to the optimum!

Conditions for Convergence ([Rudolph, 1998])

1. *There is a positive probability to reach any point in the search space from any other point*
2. *The best found solution is never removed from the population (elitism)*

Convergence

Definition

- Ideally the EA should find the solution in finite steps with probability **1** (visit the global optimum in finite time);
- If the solution is held forever after, then the algorithm converges to the optimum!

Conditions for Convergence ([Rudolph, 1998])

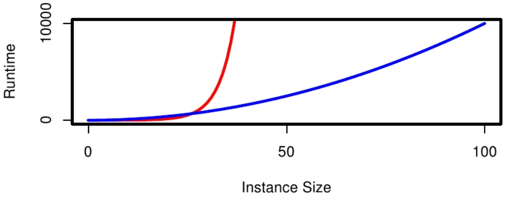
1. *There is a positive probability to reach any point in the search space from any other point*
2. *The best found solution is never removed from the population (elitism)*

- Canonical GAs using mutation, crossover and proportional selection Do Not converge!
- *Elitist* variants Do converge!

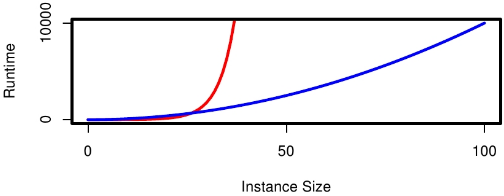
In practice, is it interesting that an algorithm converges to the optimum?

- Most EAs visit the global optimum in finite time (RLS does not!)
- How much time?

Computational Complexity of EAs



Computational Complexity of EAs

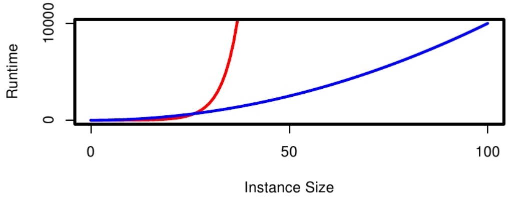


Generally means predicting the resources the algorithm requires:

- Usually the computational time: the number of primitive steps;
- Usually grows with size of the input;
- Usually expressed in asymptotic notation;

Exponential runtime: Inefficient algorithm
Polynomial runtime: "Efficient" algorithm

Computational Complexity of EAs



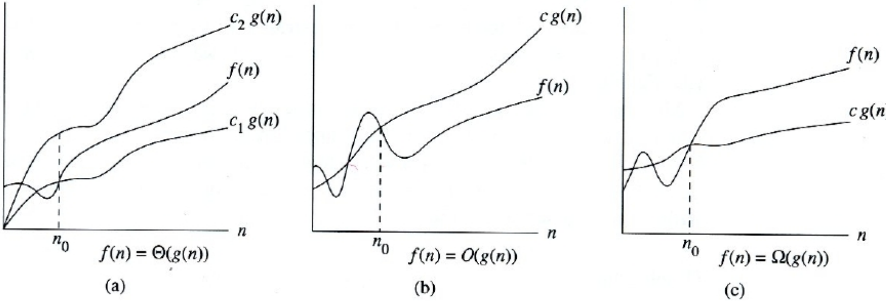
However (EAs):

1. In practice the time for a fitness function evaluation is much higher than the rest;
2. EAs are **randomised algorithms**
   - They do not perform the same operations even if the input is the same!
   - They do not output the same result if run twice!

Hence, the runtime of an EA is a random variable T_f.
We are interested in:

1. Estimating $E(T_f)$, the expected runtime of the EA for f;
2. Estimating $P(T_f \leq t)$, the success probability of the EA in t steps for f.

Asymptotic notation



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

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

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

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

Goals

Understand how the runtime depends on:

- characteristics of the problem
- parameters of the algorithm

In order to:

- explain the success or the failure of these methods in practical applications,
- understand which problems are optimized (or approximated) efficiently by a given algorithm and which are not
- guide the choice of the best algorithm for the problem at hand,
- determine the optimal parameter settings,
- aid the algorithm design.