Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland

CS4618 Artificial Intelligence I

Today: Analysing Mutation
Applying RSH

Thomas Jansen

November 16th

Plans for Today

1. Analysing Mutation
   - Motivation
2. Global Mutations Excel
   - Shortcuts
3. Local Mutations Excel
   - Traps
4. Research
   - Intuition and Counterexample
5. Applying RSH
   - Problem Modelling
6. Summary
   - Summary & Take Home Message

Comparing Mutation Operators

Remember (1+1) EA and RLS are algorithmically similar
they only differ in the mutation operator

RLS	1-bit flips
(1+1) EA	flip each bit with probability $1/n$

$\rightsquigarrow$ in expectation 1 bit flipped for both

We have seen

- on OneMax both algorithms in time $O(n \log n)$
- on $f_1(x) = \begin{cases} \text{OneMax}(x) & \text{if } |x| \neq n-1 \\ 0 & \text{otherwise} \end{cases}$

  (1+1) EA in time $O(n^2)$
  RLS in finite time only with probability 2^{-n+1}
- useful proof method: f-based partitions
- useful visualisation of functions
  - plotted over number of 1-bits (not always possible)
  - as 'map' over search space

On Comparing Mutation Operators

So we have compare local mutations (RLS)
with global mutations ((1+1) EA)
and proved that global mutations outperform local mutations
when local optima a involved

Observation completely pointless and completely obvious

More useful question Can global mutations outperform local mutations on functions without local optima?

Interlude: A Useful Tool

Long k-Paths (introduced by Horn, Goldberg, and Deb (1994))

Definition

- long k-path of dimension 1
  $P_k^1 = (0, 1)$
- long k-path of dimension n
  $P_k^n = (0^k v_1, 0^k v_2, \ldots, 0^k v_l, 10^{k-1} v_l, 1^2 0^{k-2} v_l, \ldots, 1^k v_l, 1^k v_{l-1}, \ldots, 1^k v_1)$
  if $P_k^{n-k} = (v_1, v_2, \ldots, v_l)$

Properties

- length $(k+1) \cdot 2^{(n-1)/k} - k + 1$
- $\forall p \in P_k^n \colon \forall i \in \{1, 2, \ldots, k-1\} \colon \exists_1$successor $y \in P_k^n \colon$ $\mathrm{H}(p, y) = i$

Long Path Function

Example

$$f_2(x) := \begin{cases} n^2 + p & \text{if } x = v_p \in P_2^n \\ n^2 - n \cdot \left(\sum_{i=1}^{2} x[i]\right) - \sum_{i=3}^{n} x[i] & \text{otherwise} \end{cases}$$

Easy to prove $\mathrm{E}\left(T_{\mathrm{RLS}, f_2}\right) = \Omega(n \cdot |P_2^n|) = \Omega\left(n \cdot 2^{n/2}\right)$

Idea $\mathrm{Prob}\,(\text{hit path in first half}) \geq 1/2 = \Omega(1)$

after that specific 1-bit mutations needed ✓

Observe $\mathrm{E}\left(T_{(1+1)\ \mathrm{EA}, f_2}\right) = O(n^3)$ (due to Rudolph (1996))

Idea taking $\leq n$ shortcuts suffices

for each shortcut 2-bit mutation suffices

(formal proof with f_2-based partition) ✓

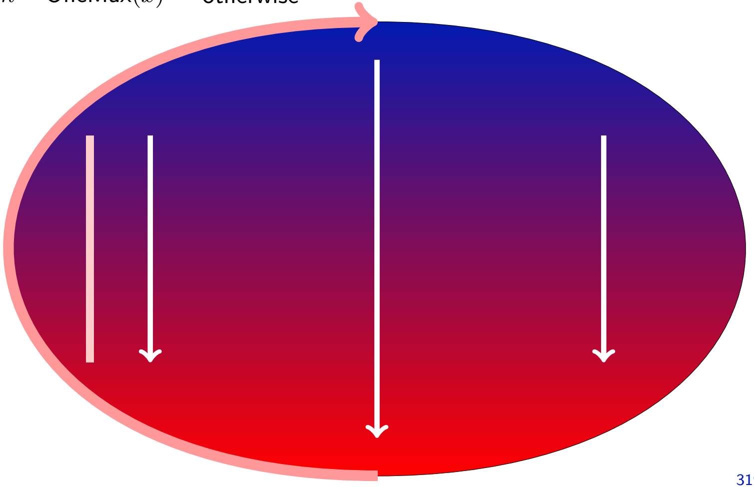
What we have so far

- global and local mutations can result in almost equal performance
  on $\textsc{OneMax}$
  (1+1) EA in time $O(n \log n)$, RLS in time $O(n \log n)$
- global mutations outperform local mutations when local optima are involved
  on $f_1(x) = \begin{cases} \textsc{OneMax}(x) & \text{if } |x| \neq n-1 \\ 0 & \text{otherwise} \end{cases}$
  (1+1) EA in time $O(n^2)$
  RLS in finite time only with probability 2^{-n+1}
- global mutations can exploit shortcuts that are not available to local mutations
  on a long path function
  (1+1) EA in time $O(n^3)$, RLS in time $\Omega\left(n \cdot 2^{n/2}\right)$

Can global mutations be very much slower than local mutations?

Trapping Global Mutations

$$f_3(x) := \begin{cases} n + 2i & \text{if } x = 1^i 0^{n-i} \\ 3n - 1 & \text{if } x = 1^i 0^j 1 0^k 1 0^{n-i-j-k-2} \\ & \text{with } n/4 \le i \le 3n/4 \text{ and } j > 0 \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases}$$



RLS on f_3

$$f_3(x) := \begin{cases} n + 2i & \text{if } x = 1^i 0^{n-i} \\ 3n - 1 & \text{if } x = 1^i 0^j 1 0^k 1 0^{n-i-j-k-2} \\ & \text{with } n/4 \leq i \leq 3n/4 \text{ and } j > 0 \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases}$$

Theorem

$\text{Prob}\left(T_{\text{RLS},f_3} = O(n^2)\right) = 1 - 2^{-\Omega(n)}$

Proof

Observation $\text{Prob}\left(\text{trap not found}\right) = 1 - 2^{-\Omega(n)}$ ✔

Observation

$\text{E}\left(T_{\text{RLS},f_3} \mid \text{trap not found}\right) = O(n \log n) + \text{E}\left(\text{time on ridge}\right)$

RLS on the Ridge

$$f_3(x) := \begin{cases} n + 2i & \text{if } x = 1^i 0^{n-i} \\ 3n - 1 & \text{if } x = 1^i 0^j 1 0^k 1 0^{n-i-j-k-2} \\ & \text{with } n/4 \leq i \leq 3n/4 \text{ and } j > 0 \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases}$$

$$\mathrm{E}\left(\text{time on ridge}\right) = \sum_{i=0}^{n-1} \mathrm{E}\left(\text{time with } |x| = i\right)$$

$$\leq \sum_{i=0}^{n-1} \left(\frac{1}{n}\right)^{-1} = n \cdot \sum_{i=0}^{n-1} 1 = n^2$$

Thus $\mathrm{E}\left(T_{\text{RLS},f_3} \mid \text{trap not found}\right)$
$= O(n \log n) + \mathrm{E}\left(\text{time on ridge}\right)$
$= O(n \log n) + O(n^2) = O(n^2)$

□

(1+1) EA on f_3

$$f_3(x) := \begin{cases} n + 2i & \text{if } x = 1^i 0^{n-i} \\ 3n - 1 & \text{if } x = 1^i 0^j 1 0^k 1 0^{n-i-j-k-2} \\ & \text{with } n/4 \leq i \leq 3n/4 \text{ and } j > 0 \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases}$$

Theorem

$$\text{Prob}\left(T_{\text{(1+1) EA}, f_3} \geq n^{n/4}\right) = 1 - 2^{-\Omega(n)}$$

Proof

Observation $\quad \text{E}\left(T_{\text{(1+1) EA}, f_3} \mid \text{trapped}\right) \geq \left(\left(\frac{1}{n}\right)^{n/4}\right)^{-1} = n^{n/4}$

Needed $\quad \text{Prob}\left(\text{enter trap}\right)$

On Entering the Trap

$$f_3(x) := \begin{cases} n + 2i & \text{if } x = 1^i 0^{n-i} \\ 3n - 1 & \text{if } x = 1^i 0^j 1 0^k 1 0^{n-i-j-k-2} \\ & \text{with } n/4 \leq i \leq 3n/4 \text{ and } j > 0 \\ n - \text{OneMax}(x) & \text{otherwise} \end{cases}$$

$\text{Prob}\left(\text{(1+1) EA enters trap}\right)$
$= \text{Prob}\left(\text{(1+1) EA enters trap 'in the beginning'}\right)$
$+\text{Prob}\left(\text{(1+1) EA enters trap 'when on the ridge'}\right)$

like for RLS

$\text{Prob}\left(\text{(1+1) EA enters trap 'in the beginning'}\right) = 2^{-\Omega(n)}$ negligible

Observation $\text{Prob}\left(\text{improve on ridge}\right) = \Theta\left(\frac{1}{n}\right)$

$\text{Prob}\left(\text{enter trap}\right) \geq \binom{n/4}{2} \left(\frac{1}{n}\right)^2 \cdot \left(1 - \frac{1}{n}\right)^{n-2}$

$\geq \frac{(n/4)\cdot(n/4-1)}{2} \cdot \frac{1}{n^2} \cdot \frac{1}{e} = \Theta(1)$

Thus in each step on the ridge $\text{Prob}\left(\text{enter trap}\right) = 1 - O(1/n)$

Thus $\text{Prob}\left(\text{don't enter trap}\right) = 1 - 2^{-\Omega(n)}$ □

One Word on Research

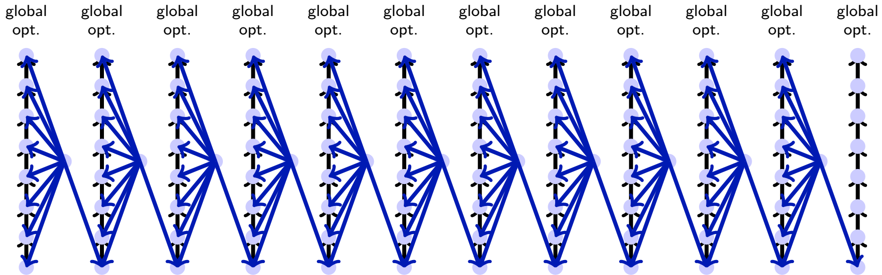
When do local and global mutations perform similar?

Idea when there are no local optima and no traps

Fact wrong

Idea for proving this wrong

Think random walk on graph with
single steps for RLS, double steps for (1+1) EA



Embedding the Counter-Example in $\{0,1\}^n$

Definition

For $n \in \mathbb{N}$ with $n \geq 16$

$n_1 := 4 \cdot \lfloor n/8 \rfloor$, $k := \left\lfloor \sqrt{n_1 - 1} \right\rfloor$, $n_2 := k^2 + 1$, $n_3 := n - n_1 - n_2$.

For $x = x_1 x_2 x_3 \in \{0,1\}^n$ (with $\forall i \in \{1,2,3\}\colon x_i \in \{0,1\}^{n_i}$)

$$f_4(x) := \begin{cases} 2^n \cdot n^2 & \text{if } x_1 = 1^{n_1} \text{ or } x_2 = v_l \\ i \cdot n^2 + j & \text{if } x_1 = 1^j 0^{n_1 - j},\ x_2 = v_i \\ i \cdot n^2 + n + n_1 - j_1 & \text{if } x_1 = 1^{j_1} 0^{j_2} 1 0^{j_3} 1 0^{n_1 - j_1 - j_2 - j_3 - 2},\ x_2 = v_i, \\ & n_1/4 \leq j_1 + j_2 \leq n_1/2,\ j_1 + j_2 + j_3 \geq (3/4) n_1 \\ i \cdot n^2 + 2n & \text{if } x_1 = 0^j 1 0^{n_1 - j - 1},\ n_1/4 \leq j \leq n_1/2, \\ & x_2 = v_i,\ i \text{ odd} \\ i \cdot n^2 - 1 & \text{if } x_1 = 0^j 1 0^{n_1 - j - 1},\ n_1/4 \leq j \leq n_1/2, \\ & x_2 = v_i,\ i \text{ even} \\ i \cdot n^2 + 2n & \text{if } x_1 = 0^j 1 0^{n_1 - j - 1},\ j > n_1/2, \\ & x_2 = v_i,\ i \text{ even} \\ i \cdot n^2 - 1 & \text{if } x_1 = 0^j 1 0^{n_1 - j - 1},\ j > n_1/2, \\ & x_2 = v_i,\ i \text{ odd} \\ n - |x| & \text{otherwise} \end{cases}$$

Result and Open Problem

Theorem

$\forall x \in \{0,1\}^n\colon \mathrm{E}\left(T_{\mathsf{RLS},f_4} \mid x_0 = x\right) = O(n^2)$

$\mathrm{Prob}\left(T_{(1+1)\ \mathsf{EA},f_4} = 2^{\Omega(\sqrt{n})}\right) = 1 - 2^{-\Omega(\sqrt{n})}$

For a proof see B. Doerr, T. Jansen, C. Klein (GECCO 2008): Comparing global and local mutations on bit strings.

Still open For what class of functions do RLS and (1+1) EA have similar performance?

More precisely 'Find X such that Y.'

Instantiations of X

- sufficient conditions
- necessary and sufficient conditions
- interesting function classes
- relevant function classes

Instantiations of Y

- $\mathrm{E}\left(T_{\mathsf{RLS},f}\right) = \Theta(t(n))$ $\Rightarrow \mathrm{E}\left(T_{(1+1)\ \mathsf{EA},f}\right) = \Theta(t(n))$
- $\mathrm{E}\left(T_{\mathsf{RLS},f}\right) = O(t(n))$ $\Rightarrow \mathrm{E}\left(T_{(1+1)\ \mathsf{EA},f}\right) = O(t(n))$
- $\mathrm{E}\left(T_{\mathsf{RLS},f}\right) = O(t(n))$ $\Rightarrow \mathrm{E}\left(T_{(1+1)\ \mathsf{EA},f}\right) = O(p(t(n)))$, p poly.

Applying Randomised Search Heuristics

Situation have difficult optimisation problem
have no good problem-specific algorithm
have desire to apply standard randomised search heuristic

Observation encoding of the problem required
so that it matches the standard setting
of the randomised search heuristics

Main Ingredients

- problem-wise
  - set of potential solutions P
  - quality measure $g\colon P \to \mathbb{R}$
- RSH-wise
  - standard search space S (e. g., $S = \{0,1\}^n$)
  - fitness function $f\colon S \to \mathbb{R}$

Observation one way of solving this gap
design mapping $m\colon S \to P$
delivers $f(s) = g(m(s))$

More Freedom in the Design

We have — one way of solving this gap
design mapping $m\colon S \to P$
delivers $f(s) = g(m(s))$

Observation — more freedom in design by
designing mapping $m\colon S \to P \cup I$
(I set of invalid solutions)
delivers $f(s) = g(m(s))$ if $m(s) \in P$

Remark — optimisation problems with constraints
introduce I naturally

often — specific optimisation problem
easier to represent as constrained case of more general problem

Design Guidelines

Setting

have problem by means of
potential solutions P and quality measure $g\colon P \to \mathbb{R}$
have standard RSH
with search space S and fitness $f\colon S \to \mathbb{R}$
design encoding by
mapping $m\colon S \to P \cup I$ gives $f(s) = g(m(s))$ for $m(s) \in P$

Design Guidelines

- $m(s)$ must be computable efficiently, for all $s \in S$
- ideally, m should be bijective
- if not possible, try to make m surjective
- if not possible, have $s \in S$ with $m(s) = p$ for all important p
- make m 'even': $\left|m^{-1}(p)\right|$ roughly equal for all $p \in P$
- $s \approx s'$ should imply $m(s) \approx m(s')$ (and vice versa)
- make finding P from I as easy as possible

Summary & Take Home Message

Things to remember

- Local mutations may systematically miss shortcuts.
- Global mutations can get trapped.
- Telling when local and global mutations are equal is an open problem.
- Applying standard RSH to optimisation problems requires modelling.

Take Home Message

- Equal expected behaviour is very much different from equal behaviour.
- Be careful when something is intuitively clear.