More Precise Runtime Analyses of Non-elitist EAs in Uncertain Environments

Per Kristian Lehre 1 & Xiaoyu Qin 2

School of Computer Science
University of Birmingham
United Kingdom

14 July 2021

1 p.k.lehre@cs.bham.ac.uk
2 xxq896@cs.bham.ac.uk
Outline

- **Background**
  - Uncertainty models (Prior & posterior noise model, \textit{DYNBV})
  - Non-elitist EAs with 2 tournament selection

- **Our results**
  - General tool for non-elitist EAs on uncertain optimisation
  - Comparison with previous results

- **Conclusion**
Uncertainty models

- **Noisy optimisation**
  - **Prior noise model** \( f^n(x) = f(x') \), where \( x' \) is obtained from \( x \)
    - **One-bit noise model** \((q)\) (Droste 2004):
      - flip one bit in \( x \) with prob. \( q \)
    - **Bit-wise noise model** \((p)\) (Qian et al. 2019):
      - flip each bit of \( x \) independently with prob. \( p \)
  - **Posterior noise model**
    - **Gaussian noise model** \((\sigma)\) (Gießen & Kötzing 2016):
      \[ f^n(x) = f(x) + \mathcal{N}(0, \sigma^2) \]

- **Dynamic optimisation**
  - **Random weights linear function** (Lengler & Schaller 2018)
  - **Dynamic Binary Value (DynBV)** (Lengler & Riedi 2020):
    \[ f^t(x) = \sum_{i=1}^n 2^{n-i} x_{\pi_t(i)} \]
    where \( \pi_t : [n] \rightarrow [n] \) is uniformly sampled in \( t \)-th generation.
Non-elitist EAs with 2-tournament selection

Algorithm 1

1: for $t = 0, 1, 2, \ldots$ until $x^*$ found do
2:   for $i = 1$ to $\lambda$ do
3:     $x_1 \leftarrow P_t(i_1)$ where $i_1 \sim \text{Unif}([\lambda])$
4:     $x_2 \leftarrow P_t(i_2)$ where $i_2 \sim \text{Unif}([\lambda])$
5:     if $f^n(x_1) \geq f^n(x_2)$ then $z \leftarrow x_1$ else $z \leftarrow x_2$
6:     $P_{t+1}(i) \leftarrow y = \text{Mutation}(z, \chi)$
7:   end for
8: end for

The fitness bias is $\theta \in (0, 1/2]$, if

$$f(x_1) > f(x_2) \iff \Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2} \Pr(f^n(x_1) = f^n(x_2)) \geq \frac{1}{2} + \theta$$
General tool

**Theorem**

*If there exists a partition $(A_0, A_1, \ldots, A_m)$ and a constant $\zeta \in (0, 1)$, such that Algorithm 1 with mutation rate $\frac{\chi}{n} \in (0, \ln(1 + 2\theta\zeta)/n)$ satisfies,*

(C1) $\Pr(\text{Mutation}(z, \chi) \in A_{\geq j+1} \mid z \in A_j) \geq h_j(\chi) \geq h_{\min}(\chi)$,

(C2) $x_1 \in A_{\geq j}$ and $x_2 \in A_{< j}$

$\implies \Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2} \Pr(f^n(x_1) = f^n(x_2)) \geq \frac{1}{2} + \theta$

(C3) $\lambda > \frac{c}{\theta^2} \ln \left(\frac{m}{\theta^2 h_{\min}(\chi)}\right)$ for sufficiently large enough constant $c$,

*then*

$$E[T] = O\left(\frac{1}{\theta^2} \sum_{j=0}^{m-1} \left(\lambda \ln \left(\frac{1}{h_j(\chi)}\right) + \frac{1}{h_j(\chi)}\right)\right).$$

Proved by the level-based theorem (Corus et al. 2018).
### Application of general tool (condition (C2))

<table>
<thead>
<tr>
<th>Model</th>
<th>Problem</th>
<th>Fitness bias $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-bit ($q$)</td>
<td><strong>OneMax</strong>                                      $\theta &gt; 1/2 - q/(2(1 - q/2)) - o(1)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>LEADINGOnes</strong>                                $\theta &gt; 1/2 - q(1 - q/2)$</td>
<td></td>
</tr>
<tr>
<td>Bit-wise ($p$)</td>
<td><strong>OneMax</strong>                                      $\theta &gt; \frac{9(1-2p)}{128\sqrt{2}pn+32}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>LEADINGOnes</strong>                                $\theta &gt; (1/2 - 3p/2) e^{-3np}$</td>
<td></td>
</tr>
<tr>
<td>Gaussian ($\sigma$)</td>
<td><strong>OneMax</strong>                          $\theta &gt; 1/(6 + 48\sigma/\pi)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>LEADINGOnes</strong>                               $\theta &gt; 1/(6 + 48\sigma/\pi)$</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td><strong>DYNBV</strong>                                       $\theta &gt; 1 + 1/(2n)$</td>
<td></td>
</tr>
</tbody>
</table>

Pessimistically assume $x_1 = 1^{j+1}00^n-(j+1)-1$ and $x_2 = 1^j01^{n-j-1}$ s.t. $f(x_1) = j + 1$ and $f(x_2) = j$, then

$$\Pr(f^n(x_1) > f^n(x_2)) + \frac{1}{2} \Pr(f^n(x_1) = f^n(x_2))$$

$$\geq \frac{1}{2} + \theta > (1 - q)^2 + (1 - q)q \left(1 - \frac{1}{n}\right)$$

$$+ q(1 - q) \frac{1 - (j + 1)}{n} + \frac{q^2}{2}$$

$$> \frac{1}{2} + \frac{1}{2} - q \left(1 - \frac{q}{2}\right)$$
Comparison to previous results for Algorithm 1:

<table>
<thead>
<tr>
<th>Model</th>
<th>Prob.</th>
<th>Previous Results</th>
<th>This paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-bit</td>
<td>OM</td>
<td>$O(n \log(n) \log \log(n))$</td>
<td>$O(n \log(n))$</td>
</tr>
<tr>
<td>$q \in [0, 1)$</td>
<td>LO</td>
<td>-</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Bit-wise</td>
<td>OM</td>
<td>-</td>
<td>$O(n^{4c+3} \log^2(n))$ for $p = 1/2 - 1/n^c$</td>
</tr>
<tr>
<td>$p \geq 0$</td>
<td>LO</td>
<td>-</td>
<td>$O(n^{12c+1} \log^2(n) + n^{9c+2})$ for $p = c \log(n)/n$</td>
</tr>
<tr>
<td>Gaussian</td>
<td>OM</td>
<td>$O(\sigma^7 n \log(n) \log \log(n))$</td>
<td>$O(\sigma^4 n \log^2(n))$</td>
</tr>
<tr>
<td>$\sigma \in \text{poly}(n)$</td>
<td>LO</td>
<td>$O(\sigma^7 n \log(n) \log \log(n) + \sigma^6 n^2)$</td>
<td>$O(\sigma^4 n \log^2(n) + \sigma^3 n^2)$</td>
</tr>
<tr>
<td>-</td>
<td>DYNBV</td>
<td>-</td>
<td>$O(n^5 \log^2(n))$</td>
</tr>
</tbody>
</table>

3 (Dang & Lehre 2015)
4 Assuming using specific mutation rates and population sizes
## Comparison to other algorithms:

<table>
<thead>
<tr>
<th>Model</th>
<th>Prob.</th>
<th>((1+1)) EA (^a)</th>
<th>Best Alg.</th>
<th>This paper (^b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-bit ((q))</td>
<td>OM</td>
<td>(\text{poly}(n)) for (q \in O(\log(n)/n))</td>
<td>(O(n \log^2(n))) (^c)</td>
<td>(O(n \log(n))) for (q \in [0, 1])</td>
</tr>
<tr>
<td></td>
<td>LO</td>
<td>(\text{poly}(n)) for (q \in O(\log(n)/n^2))</td>
<td>(O(n^2)) (^d)</td>
<td>(O(n^2)) for (q \in [0, 1])</td>
</tr>
<tr>
<td>Bit-wise ((p))</td>
<td>OM</td>
<td>(\text{poly}(n)) for (p \in O(\log(n)/n^2))</td>
<td>(\text{poly}(n)) (^e)</td>
<td>(O(n^{4c+3} \log^2(n))) for (p = 1/2 - 1/n^c)</td>
</tr>
<tr>
<td></td>
<td>LO</td>
<td>(\text{poly}(n)) for (p \in O(\log(n)/n^3))</td>
<td>(O(n^{32c+5})) (^e)</td>
<td>(O(n^{12c+1} \log^2(n) + n^{9c+2})) for (p = c \log(n)/n)</td>
</tr>
<tr>
<td>Gaussian ((\sigma))</td>
<td>OM</td>
<td>(O(n \log(n))) for (\sigma^2 \leq 1/(4 \log(n)))</td>
<td>(O(\sigma^4 n \log^2(n))) (^f)</td>
<td>(O(\sigma^4 n \log^2(n))) for (\sigma^2 \in \text{poly}(n))</td>
</tr>
<tr>
<td></td>
<td>LO</td>
<td>(O(n^2)) for (\sigma^2 \leq 1/(12en^2))</td>
<td>(O(\sigma^2 n^4)) (^e)</td>
<td>(O(\sigma^4 n \log^2(n) + \sigma^3 n^2)) for (\sigma^2 \in \text{poly}(n))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(O(n \log(n)))</td>
<td>(O(n^5 \log^2(n)))</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\)(Gießen & Kötzing 2016, Sudholt 2020)

\(^b\)Assuming using specific mutation rates and population sizes

\(^c\)(\(\mu+1\)) EA (Gießen & Kötzing 2016)

\(^d\)UMDA (Lehre & Nguyen 2019)

\(^e\)(1+1) EA with resampling (Qian et al. 2018, 2019)

\(^f\)cGA (Friedrich et al. 2016)
Conclusion

- **General tool** for analysing non-elitist EAs under uncertainty
- More precise results than previous results
- Non-elitist population-based EAs can beat the current state of the art algorithms in some setting
- Lower mutation rate & larger population size required if higher uncertain level imposed
- First result to show that population-based EAs optimise DynBV in polynomial from any start point
- Future work:
  - Room for improvement: avoid overly pessimistic assumptions about the population, e.g., $x_1$ only slightly better than $x_2$
  - Negative analysis: what happens for mutation rate $\chi/n > \ln(1 + 2\theta\zeta)/n$
  - ...
Title: More Precise Runtime Analyses of Non-elitist EAs in Uncertain Environments

Authors: Per Kristian Lehre & Xiaoyu Qin

{p.k.lehre, xxq896}@cs.bham.ac.uk

Presenter: Xiaoyu Qin

Connect to me via LinkedIn:


