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

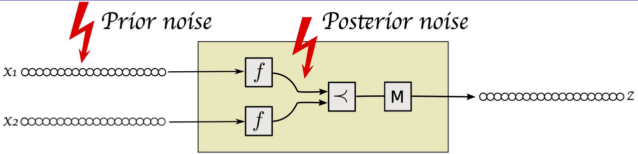
14 July 2021

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

Outline

- **Background**
  - Uncertainty models (Prior & posterior noise model, 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

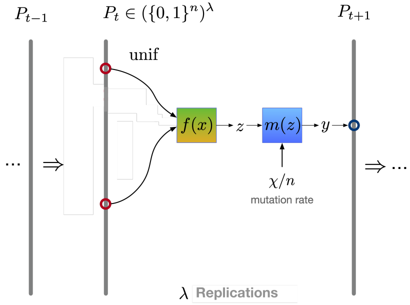


- Noisy optimisation
  - Prior noise model $f^n(x) = f(x')$, where x' is obtained from x
    - One-bit noise model (q) (Droste 2004):
      flip one bit in x with prob. q
    - Bit-wise noise model (p) (Qian et al. 2019):
      flip each bit of x independently with prob. p
  - Posterior noise model
    - Gaussian noise model (σ) (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 $\pi_t : [n] \to [n]$ is uniformly sampled in t-th generation.

Non-elitist EAs with 2-tournament selection



Algorithm 1

```
for t = 0, 1, 2, ... until x* found do
    for i = 1 to λ do
        x_1 ← P_t(i_1) where i_1 ∽ Unif([λ])
        x_2 ← P_t(i_2) where i_2 ∽ Unif([λ])
        if f^n(x_1) ≥ f^n(x_2)
            then z ← x_1 else z ← x_2
        P_{t+1}(i) ← y = MUTATION(z, χ)
    end for
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* $\chi/n \in (0, \ln(1+2\theta\zeta)/n)$ *satisfies,*

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

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

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

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

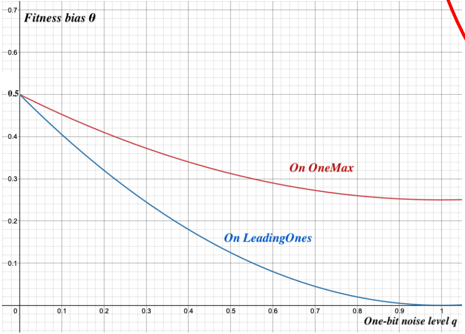
then

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

Proved by the level-based theorem (Corus et al. 2018).

Application of general tool (condition (C2))

Model	**Problem**	**Fitness bias θ**
One-bit (q)	OneMax	$\theta > 1/2 - q/2(1-q/2) - o(1)$
	LeadingOnes	$\theta > 1/2 - q(1-q/2)$
Bit-wise (p)	OneMax	$\theta > \frac{9(1-2p)}{128\sqrt{2pn}+32}$
	LeadingOnes	$\theta > (1/2 - 3p/2)\, e^{-3np}$
Gaussian (σ)	OneMax LeadingOnes	$\theta > 1/(6 + 48\sigma/\pi)$
-	DynBV	$\theta > 1 + 1/(2n)$



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

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

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

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

$$> \frac{1}{2} + \frac{1}{2} - q\left(1 - \frac{q}{2}\right)$$

Comparison to previous results for Algorithm 1:

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

[3](Dang & Lehre 2015)
[4]Assuming using specific mutation rates and population sizes

Comparison to other algorithms:

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

[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 $\chi/n > \ln(1+2\theta\zeta)/n$
  - $\cdots$

Thank you

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:

Reference I

Corus, D., Dang, D.-C., Eremeev, A. V. & Lehre, P. K. (2018), 'Level-Based Analysis of Genetic Algorithms and Other Search Processes', *IEEE Transactions on Evolutionary Computation* **22**(5), 707–719.

Dang, D.-C. & Lehre, P. K. (2015), Efficient Optimisation of Noisy Fitness Functions with Population-based Evolutionary Algorithms, *in* 'Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII - FOGA '15', ACM Press, Aberystwyth, United Kingdom, pp. 62–68.

Droste, S. (2004), Analysis of the (1+1) EA for a Noisy OneMax, *in* 'Genetic and Evolutionary Computation – GECCO 2004', Vol. 3102, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1088–1099.

Friedrich, T., Kotzing, T., Krejca, M. S. & Sutton, A. M. (2016), 'The Compact Genetic Algorithm is Efficient under Extreme Gaussian Noise', *IEEE Transactions on Evolutionary Computation* pp. 1–1.

Gießen, C. & Kötzing, T. (2016), 'Robustness of Populations in Stochastic Environments', *Algorithmica* **75**(3), 462–489.

Lehre, P. K. & Nguyen, P. T. H. (2019), Runtime analysis of the univariate marginal distribution algorithm under low selective pressure and prior noise, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference', ACM, Prague Czech Republic, pp. 1497–1505.

Lengler, J. & Riedi, S. (2020), 'Runtime analysis of the (mu+1)-EA on the Dynamic BinVal function', *arXiv:2010.13428 [cs]* . arXiv: 2010.13428.

Lengler, J. & Schaller, U. (2018), The (1+1)-EA on Noisy Linear Functions with Random Positive Weights, *in* '2018 IEEE Symposium Series on Computational Intelligence (SSCI)', pp. 712–719.

Qian, C., Bian, C., Jiang, W. & Tang, K. (2019), 'Running Time Analysis of the (1+1)-EA for OneMax and LeadingOnes Under Bit-Wise Noise', *Algorithmica* **81**(2), 749–795.

Qian, C., Yu, Y., Tang, K., Jin, Y., Yao, X. & Zhou, Z.-H. (2018), 'On the Effectiveness of Sampling for Evolutionary Optimization in Noisy Environments', *Evolutionary computation* **26**, 237–267.

Sudholt, D. (2020), 'Analysing the Robustness of Evolutionary Algorithms to Noise: Refined Runtime Bounds and an Example Where Noise is Beneficial', *Algorithmica* .