Escaping Local Optima with Non-Elitist Evolutionary Algorithms

Duc-Cuong Dang[1], **Anton Eremeev**[2], **Per Kristian Lehre**[3]
[1]Southampton Business School, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom
[2]Sobolev Institute of Mathematics SB RAS, 13, Pevtsov str., Omsk, 644099, Russia
[3]School of Computer Science, University of Birmingham, Edgbaston, B15 2TT Birmingham, United Kingdom
d.c.dang@soton.ac.uk, eremeev@ofim.oscsbras.ru, p.k.lehre@cs.bham.ac.uk

Abstract

Most discrete evolutionary algorithms (EAs) implement elitism, meaning that they make the biologically implausible assumption that the fittest individuals never die. While elitism favours exploitation and ensures that the best seen solutions are not lost, it has been widely conjectured that non-elitism is necessary to explore promising fitness valleys without getting stuck in local optima. Determining when non-elitist EAs outperform elitist EAs has been one of the most fundamental open problems in evolutionary computation. A recent analysis of a non-elitist EA shows that this algorithm does not outperform its elitist counterparts on the benchmark problem JUMP.

We solve this open problem through rigorous runtime analysis of elitist and non-elitist population-based EAs on a class of multi-modal problems. We show that with 3-tournament selection and appropriate mutation rates, the non-elitist EA optimises the multi-modal problem in expected polynomial time, while an elitist EA requires exponential time with overwhelmingly high probability.

A key insight in our analysis is the non-linear selection profile of the tournament selection mechanism which, with appropriate mutation rates, allows a small sub-population to reside on the local optimum while the rest of the population explores the fitness valley. In contrast, we show that the comma-selection mechanism which does not have this non-linear profile, fails to optimise this problem in polynomial time.

The theoretical analysis is complemented with an empirical investigation on instances of the set cover problem, showing that non-elitist EAs can perform better than the elitist ones. We also provide examples where usage of mutation rates close to the error thresholds is beneficial when employing non-elitist population-based EAs.

Introduction

Evolutionary algorithms (EAs) are heuristic optimisation methods inspired by evolution. Departing from nature – where even the fittest individuals eventually die – EAs were early on designed with *elitism* (Cavicchio 1970; De Jong 1975), where some of the fittest individuals are always copied unchanged into the next generation. Both from a practical and theoretical point of view, elitism was considered a necessary mechanism to ensures monotone progress. Elitism was thought necessary to saturate good solutions (Beyer 1997), and Rudolph used elitism as the essential property to guarantee that the canonical genetic algorithm (CGA) converges to the optimum (Rudolph 1994).

In contrast, the less frequently employed *non-elitist* EAs produce the new generation solely from the offspring of the old generation, and the old generation is then discarded. Non-elitist EAs are highly inefficient if their mutation rates and selection pressure have not been set appropriately. In particular, if the mutation rate exceeds the *error threshold*, a quantity which depends on the selective pressure, then the EA will require at least exponential time to locate any unique global optimum (Lehre 2010). Error thresholds (or mutation-selection balance) were originally studied in population genetics (Wilke 2005), virology (Biebricher and Eigen 2005), and later introduced to evolutionary computation (Ochoa 2006). Non-elitist EAs with mutation rates below the error threshold optimise many pseudo-Boolean problems in expected polynomial time (Corus et al. 2018). For certain problems, the mutation rate must be neither too high nor too low to ensure polynomial runtime (Lehre and Yao 2012).

Our aim is to contribute to the long-standing open problem in evolutionary computation, to determine whether non-elitism can lead to more efficient search than elitism (Jägerskupper and Storch 2007; Doerr 2020). An attractive hypothesis, which we call the *escape hypothesis*, is that non-elitism helps EAs escape local optima. Some believe that if the population contains a local optimum, a non-elitist EA may allow a sub-population sufficient time to explore less fit fitness valleys around the optimum, to eventually discover better solutions. In contrast, the locally optimal individual in an elitist EA are thought to take over the population too quickly. To progress, the elitist EA will require a long time to discover a fitter solution through a low probability mutation (or crossover) event.

So far, there has been no convincing evidence that the escape hypothesis is true. To test the hypothesis, one could attempt to analyse the runtime of elitist and non-elitist EAs on multi-modal problems with well known structure.

The CLIFF problem has local optima which can be reached easily through hill-climbing, and which are all separated from the optimum by a fitness valley of Hamming width k. The non-elitist (1,λ) EA with population size $\lambda \geq 5\log(n)$ optimises CLIFF (for $k = \lfloor n/3 \rfloor$) in expected time $O(e^{5\lambda})$, whereas the elitist (1+λ) EA needs at least $n^{n/4}$ function eval-

uations with overwhelmingly high probability (Jägerskupper and Storch 2007). This result is non-convincing for two reasons. Both algorithms require astronomical runtime on this problem, and the EAs have a (parent) population size of one. Clearly, these algorithms cannot help us understand how less fit sub-populations can survive.

The Strong Selection Weak Mutation (SSWM) model from population genetics was studied in the context of evolutionary computation (Paixão et al. 2017). The authors proved a speed-up of $e^{\Omega(d)}$ of SSWM over the expected runtime $\Theta(n^d)$ of the (1+1) EA on the CLIFF function. Here, the quantity d refers both to the "height" of the local optima and their distance to the global optimum. Based on the construction of LONGPATH, the VALLEYPATH function was introduced in (Oliveto et al. 2018). It was shown that the non-elitist algorithms with population of size 1, such as the SSWM and Metropolis, are able to cross a valley of deceptive fitness, and their ability to escape the current local optimum depends on the depth of the valley. This is in contrast to the (1+1) EA in which case the ability to escape crucially depends on the width of the valley. Both of these results concern evolutionary models with population size one. However, these results give no insights into the population-dynamics of non-elitist EAs.

The JUMP problem has a similar (but not identical) structure to the CLIFF problem. Doerr proved that the non-elitist (μ, λ) EA below the error threshold has asymptotically the same optimisation time on the JUMP problem, as the elitist (μ+λ) EA (Doerr 2020). (Above the error threshold, the (μ, λ) EA requires exponential time.) While these algorithms are true population-based EAs, they fail to show any difference in performance between the algorithms. The author argues that the (μ,λ) EA operates in one of two modes. Above the error threshold, the algorithm easily leaves the local optimum, but fails to find the global optimum efficiently as proved earlier (Lehre 2010). Below the error threshold, the algorithm has a similar behaviour to the elitist (μ+λ) EA, and has a difficulty in leaving the optimum.

This dichotomy of the (μ,λ) EA was already well known and studied several years earlier (Dang and Lehre 2016b). For the PEAK problem, it was shown that the (μ, λ) EA below the error threshold requires exponential time to escape the local optimum. Above the error threshold, the algorithm escapes the local optimum quickly, however fails to obtain the global optimum in exponential time. The authors proved that a non-elitist EA using tournament selection (rather than (μ,λ)-selection) and self-adaptation to dynamically adjust mutation rates, would optimise PEAK in expected polynomial time. This result does not fully validate the escape hypothesis because the efficiency of the algorithm on this problem relies partly on the self-adaptation mechanism. While self-adaptation can be a highly effective mechanism for controlling mutation rates in non-elitist EAs, (Case and Lehre 2020), we are here interested in understanding the benefit of non-elitism alone.

Our Contributions

- Solving a decades-long open problem, we prove for the first time that non-elitist population-based EAs can outperform elitist ones. In particular, we construct a problem class FUNNEL where the escape hypothesis holds: The elitist (μ+λ) EA gets trapped on a local optimum, and needs exponential time with overwhelmingly high probability (Theorem 4). A non-elitist EA with 3-tournament selection, mutation rate close to but below the error threshold, and sufficiently large population size, escapes the local optimum by exploring a fitness value, and optimises the problem in expected time $\mathrm{O}\big(n\lambda\log(\lambda)+n^2\log(n)\big)$ (Theorem 9).
- We demonstrate that the capability of non-elitist EAs to escape local optima depends on the choice of selection mechanism. The non-elitist EA becomes inefficient on FUNNEL if 3-tournament selection is replaced with (μ, λ)-selection (Theorem 6). This is the first time a such a difference has been observed between these two selection mechanisms.
- Further demonstrating the importance of parameterisation of non-elitist EAs, we show that the non-elitist EA becomes inefficient on FUNNEL if the mutation rate deviates too much from the error threshold (Theorem 11).
- We provide preliminary empirical evidence that the non-elitist EA with tournament selection outperforms elitist EAs on instances of the Set Cover problem. The experiments also indicate that the best performance occurs when the mutation rate is close to the error threshold.

Preliminaries

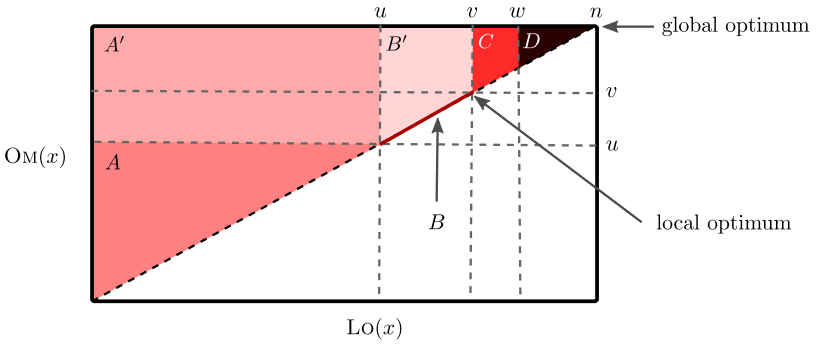
The natural and base-2 logarithms are denoted $\ln(\cdot)$, and $\log(\cdot)$ respectively. For any $n \in \mathbb{N}$, define $[n] := \{1, \dots, n\}$. The Hamming distance is denoted by $\mathrm{H}(\cdot,\cdot)$ and the Iverson bracket by $[\cdot]$. Given a partition of a search space $\mathcal{X}$ into m ordered "levels" $(A_1, \dots, A_m)$, we define for any $j \in [m]$, $A_{\geq j} := \cup_{i=j}^{m} A_i$. A *population* is a vector $P \in \mathcal{X}^\lambda$, the i-th individual of P is denoted $P(i)$. Given $x \in \mathcal{X}$, define $\mathrm{H}(x, P) := \min_{j\in[|P|]}\{\mathrm{H}(P(j), x)\}$, and for $A \subseteq \mathcal{X}$, we let $|P \cap A| := |\{i \mid P(i) \in A\}|$, i. e. the number of individuals of P belonging to A.

All non-elitist EAs with *unary variation operators* can be cast in the framework of Algorithm 1 (Dang and Lehre 2016a). A new population P_{t+1} is generated by independently sampling λ individuals from an existing population P_t according to a selection mechanism p_{sel}, then by perturbing each of the selected individuals with a unary variation operator p_{mut}. The fitness function $g : \mathcal{X} \to \mathbb{R}$ is implicitly defined through p_{sel}.

In *k-tournament selection*, a set S of k random numbers are drawn independently and uniformly from $[\lambda]$ then p_{sel}

Algorithm 1 (Dang and Lehre 2016a)

Require: Initial population $P_0 \in \mathcal{X}^\lambda$, parameter $\chi \in [0, n]$.
1: **for** $t \in \mathbb{N}$ until a termination cond. is met **do**
2: **for** $i = 1$ to λ **do**
3: Sample $I_t(i) \sim p_{\mathrm{sel}}(P_t)$, and set $x := P_t(I_t(i))$.
4: Sample $x' \sim p_{\mathrm{mut}}(x, \chi)$, and set $P_{t+1}(i) := x'$.
5: **end for**
6: **end for**



$$\textsc{Funnel}(x) := \begin{cases} \textsc{Lo}(x) & \text{if } w < \textsc{Lo}(x) \leq n & (D) \\ \textsc{Lo}(x) + u - v & \text{if } v < \textsc{Lo}(x) \leq w & (C) \\ \textsc{Lo}(x) + w - v & \text{if } u < \textsc{Lo}(x) \leq v \text{ and } \textsc{Lo}(x) = \textsc{Om}(x) & (B) \\ -n & \text{if } u < \textsc{Lo}(x) \leq v \text{ and } \textsc{Lo}(x) < \textsc{Om}(x) & (B') \\ \textsc{Lo}(x) & \text{if } \textsc{Om}(x) \leq u & (A) \\ -\textsc{Om}(x) & \text{otherwise} & (A') \end{cases}$$

Figure 1: Structure and definition of FUNNEL function with parameters u, v, w. The darker the shade the higher region fitness.

returns $\operatorname{argmax}_{i \in S}\{f(P_t(i))\}$. In (μ, λ)-selection (comma-selection), the set of indices $S = [\lambda]$ is first sorted according to $f(P_t(i))$, then p_{sel} returns $S[i]$ where $i \sim \text{Unif}([\mu])$. We consider the standard *bitwise mutation operator* as p_{mut} and it is configured by a parameter $\chi \in (0, n/2]$ so that for any pair of bitstrings $x, x' \in \{0,1\}^n$, the probability of obtaining x' from x is $\Pr(x' = p_{\text{mut}}(x, \chi)) = (\chi/n)^{\text{H}(x,x')}(1-\chi/n)^{n-\text{H}(x,x')}$.

In the elitist $(\mu{+}\lambda)$ EA (see Algorithm 2 in the supplementary material[1]), a new offspring population P is created by selecting individuals uniformly from P_t and perturbing them with p_{mut}. The surviving population P_{t+1} of the next generation then composes of the μ best individuals among both parent and offspring populations $P_t \cup P$. The analysis presented in this paper can be seen as comparing the three selection mechanisms: the *tournament* and *comma selections*, i. e. line 3 in Algorithm 1, and the *plus selection*, i. e. line 6 in Algorithm 2.

We consider the optimisation of the function $f(x) = \textsc{Funnel}(x)$ defined in Fig. 1 with parameters u, v, w. Here, $\textsc{Lo}(x) := \sum_{i=1}^{n}\prod_{j=1}^{i} x(j)$ and $\textsc{Om}(x) := \sum_{i=1}^{n} x(i)$ are the well-known LEADINGONES and ONEMAX functions. In this paper, we assume $u \geq (n/2)(1+\varepsilon)$ for a constant $\varepsilon > 0$. The label to the right of each case corresponds to the set of search points validating the part of the function, e. g. $A := \{x \in \{0,1\}^n \mid u \geq \textsc{Om}(x)\}$, Fig. 1 illustrates these sets on $\textsc{Lo}(x)$ and $\textsc{Om}(x)$ values.

Lemma 1. *The function f satisfies $-n = f(B') < f(A') < f(A) < f(C) < f(B) < f(D) \leq n$.*

By this hierarchy of the search space and assuming $w - v = \Omega(n)$, we observe that B is a difficult part of the search space for any algorithm to start in as it is a narrow region of highly fit solutions, with a local optimum on its end-point. If the whole population of $(\mu{+}\lambda)$ EA is in B, then to progress further, an individual in D must be sampled from one in B and the probability of such event is $n^{-\Omega(n)}$. Similarly, we can prove that if the μ best individuals of (μ, λ) EA are in B and the selection-mutation balance is strong enough (required to optimise functions with unique optimum (Dang and Lehre 2016b)), then the individuals created in C (eg. from mutating one in B) cannot reproduce, thus creating an individual in D also takes exponential time. On the other hand, there is no such difficulty for a non-elitist population with tournament selection under the right setting. Using 3-tournament, we will show that the number of individuals in B is always limited, thus enabling those created in C to evolve towards D.

Inefficiency of Plus and Comma Selection

In this section, we will study the efficiency of both $(\mu{+}\lambda)$ EA and (μ, λ) EA. The result for the former is complete in the sense that we are able to prove that the population will eventually end up quickly with all individuals in B once one is discovered. An analogous analysis for (μ, λ) EA requires a weak assumption that the μ best individuals reach B quickly. Experiments indicate that the assumption holds with high probability, but proving it is left for future work.

The following two lemmas show that it is unlikely that $(\mu{+}\lambda)$ EA optimises the B-region before the entire population is on B.

Lemma 2. *Assume that $u \geq (n/2)(1+\varepsilon)$ for any constant $\varepsilon > 0$, $v - u = \Omega(n)$ and define $\alpha := (v-u)/n$. Assume that $(\mu{+}\lambda)$ EA with $\lambda = \text{poly}(n)$ and $\chi \leq n/2$ has obtained an individual in B for the first time in generation t_0. Define $X_t := \max_{i \in [\lambda]} \textsc{Lo}(P_{t_0+t}(i))$ and assume that $u \leq X_0 \leq v - (2/3)\alpha n$. Let $T := \min\{t \in \mathbb{N} \mid X_t \geq v - (1/3)\alpha n\}$. Then, $\Pr(T \leq n^{1-2\delta}) = e^{-\Omega(n^\delta)}$ for any*

[1] Available at: https://gitlab.com/d2cmath/ea-solve-scp

constant $\delta \in (0, 1/2)$.

Proof. We say that *failure event 1* occurs if any individual in the initial population P_0 does not belong to region A. By a Chernoff bound and a union bound, the probability of failure event 1 is $e^{-\Omega(n)}$.

If failure event 1 does not occur, then no future individual will be accepted from region B'. Furthermore, to mutate any search point x in region A into region $C \cup D$, it is necessary to flip all of the at least αn 0-bits within the first v bit-positions of x. We say that *failure event 2* occurs if this happens to any individual within the first $n^{1-2\delta}$ generations. By a union bound, the probability of failure event 2 is no more than $\lambda n^{1-2\delta} \left(\frac{\chi}{n}\right)^{\alpha n} \leq \lambda n^{1-2\delta} \left(\frac{1}{2}\right)^{\alpha n} = e^{-\Omega(n)}$.

We now assume that failure events 1 and 2 did not occur. For a lower bound, we assume that any selected individual x in generation $t + t_0$ will belong to region B, and will have $\textsc{Lo}(x) = X_t$. Due to the definition of B, the probability of mutating an individual $x \in B$ with $\textsc{Lo}(x) \leq v - n^\delta$ into an individual $x' \in B$ with $\textsc{Lo}(x') - \textsc{Lo}(x) \geq n^\delta$ is less than $(\chi/n)^{n^\delta} = 2^{-\Omega(n^\delta)}$, because it is necessary to flip at least n^δ specific bit-positions. We call any such mutation within the first $n^{1-2\delta}$ generations *failure event 3*. By a union bound, the probability of failure event 3 is no more than $\lambda n^{1-2\delta} e^{-\Omega(n^\delta)} = e^{-\Omega(n^\delta)}$. If none of the three failure event occurs, then for all $t \leq n^{1-2\delta}$, it holds $X_t - X_0 \leq tn^\delta \leq n^{1-\delta} < (1/3)\alpha n$ for n sufficiently large. □

The following Lemma 3 gives a rough bound on the upgrade time for the (μ+λ) EA in region B. For example, by letting X_t be the number of individuals on B, the stochastic process $(X_t)_{t \geq 0}$ matches the one in Lemma 3 with $\delta = (1-\varepsilon)e^{-\chi} \leq (1-\chi/n)^n$ for any constant $\varepsilon > 0$. Setting $z = n^\delta$ in the lemma, and further assuming that $\lambda/\mu = \Omega(1)$ and that there are no fitter individuals in the population, the probability that the population has not converged on B within $\mathrm{O}\big(n^\delta \log \mu\big)$ generations is less than $e^{-\Omega(n^\delta)}$.

Lemma 3. *Assume a stochastic process $(X_t)_{t \in \mathbb{N}}$ where $X_0 \geq 1$ and for some constant $\delta \in (0, 1)$, for all $t \in \mathbb{N}$, $X_{t+1} = \min(\mu, X_t + Y_{t+1})$ where $Y_{t+1} \sim \mathrm{Bin}(\lambda, \delta X_t/\mu)$. Then for any $z \geq 0$, $\Pr(X_\tau < \mu) \leq se^{-z}$ where $\tau := \lceil \frac{8\mu}{\delta\lambda} \rceil sz$ and $s := \lceil \log(\mu)/\log(1 + \frac{\delta\lambda}{2\mu}) \rceil$.*

Proof. To simplify the analysis, we assume that $X_t := X_t + Y_{t+1}$ for all $X_t \leq \mu$. This will not change the probability of the event $X_\tau < \mu$. We call an iteration *successful* if $Y_{t+1} \geq X_t\delta\lambda/(2\mu)$. We consider s phases, where phase $i \in [s]$ lasts until a successful iteration occurs or if there have been $r := \lceil 8\mu z/(\delta\lambda) \rceil$ unsuccessful iterations. We say that a phase is successful if it has a successful iteration.

Assume all phases until phase i have been successful. Then, for any iteration t in phase $i+1$ it holds $X_t \geq (1+\delta\lambda/(2\mu))^i$. By a Chernoff bound,

$$\Pr\left(X_{t+1} \leq (1 + \delta\lambda/(2\mu))^{i+1} \mid X_t\right)$$
$$\leq \Pr\left(Y_{t+1} \leq E\left[Y_{t+1}\right]/2 \mid X_t\right) \leq e^{-\frac{\delta\lambda X_t}{8\mu}} \leq e^{-\frac{\delta\lambda}{8\mu}} =: p.$$

The probability that all of the r iterations in phase i are unsuccessful is $p^r = e^{-z}$. By a union bound, the probability that any of the s phases is unsuccessful is less than se^{-z}. If all phases are successful, then $X_\tau \geq \mu$ for iteration $\tau = sr$. □

We can now show that the (μ+λ) EA is inefficient.

Theorem 4. *The runtime of the (μ+λ) EA with population sizes $\mu, \lambda \in \mathrm{poly}(n)$, $\lambda/\mu = \Omega(1)$, and mutation rate parameter $\chi = O(1)$ on function f with $v - u = \Omega(n)$, $w - v = \Omega(n)$ and $u \geq (1+\varepsilon)n/2$ for any constant $\varepsilon > 0$ satisfies $\Pr(T \leq e^{cn^d}) \leq e^{-\Omega(n^d)}$ for some constants $c, d > 0$.*

Proof. We will prove a stronger statement that with probability $1 - e^{-\Omega(n^d)}$ during the first e^{cn^d} function evaluations none of the search point in $C \cup D$ is created, where $c, d \in (0, 1)$ are constants which will be defined later. Because $u \geq (1 + \varepsilon)n/2$, then by a Chernoff and union bound, the probability that the whole initial population is in A is at least $1 - \mu e^{-\Omega(n)} = 1 - e^{-\Omega(n)}$. From these individuals in A, it is possible to create those in $C \cup D$ by either direct mutation, or through creating individuals in B.

In the case of direct mutation, to create an individual in $C \cup D$ from A, at least $v - u = \Omega(n)$ 0-bits must be flipped and the probability of such an event is $n^{-\Omega(n)}$ for the mutation rate $\chi/n = O(n^{-1})$. Thus by a union bound, the probability that no search point in $C \cup D$ is created by a direct mutation from A within the first e^{cn^d}/λ generations of the algorithm is $1 - e^{cn^d} n^{-\Omega(n)} = 1 - e^{-\Omega(n)}$.

In the case of mutating an individual in B, we note that if the whole population is in B, then to create a search point in $C \cup D$ it is necessary to mutate from B to D. By an argument similar to the one above, with probability $1 - e^{-\Omega(n)}$ no search point in D is created that way within e^{cn^d}/λ generations afterwards. Therefore, to complete the proof it suffices to show that once an individual in B appears for the first time in the population, it will take over the population faster than the time needed for an individual in $C \cup D$ to appear, or more pessimistically, than the time needed for an individual in B with a close distance to $C \cup D$ to appear in the population.

By Lemma 3 with $z = n^\delta$ and $\delta = e^{-\chi}/2 < (1 - \frac{\chi}{n})^n$ for sufficiently large n, with probability $1 - e^{-\Omega(n^\delta)}$ the take over time is less than $c' n^\delta \log \mu$ generations for some constant c'. On the other hand, when an individual x in B is created for the first time from one in A, it holds with probability $1 - n^{-\Omega(n)}$ that $\textsc{Lo}(x) \leq u + (v - u)/3$ since at least $(v - u)/3 = \Omega(n)$ bits have to be flipped.

Define the constant $d := \delta/2$ and recall that $\delta = e^{-\chi}/2 < 1/2$. Applying Lemma 2 gives that the probability that the number of generations required to create an individual in B with at least $u + (2/3)(v - u)$ leading 1-bits exceeds n^{1-2d} is $1 - e^{-\Omega(n^d)}$. Note that $\log \mu < n^{1-2d-\delta}/c' = n^{1-2\delta}/c' =$ for sufficiently large n. The probability that the population is taken over by search points in B before it gets close to $C \cup D$ is $1 - e^{-\Omega(n^d)}$. In overall, by a union bound, the algorithm will need more than e^{cn^d} function evaluations to create a search point in $C \cup D$ and this holds with probability $1 - e^{-\Omega(n^d)}$. □

We now explain informally why the non-elitist (μ,λ)-selection mechanism behaves differently than tournament selection. Let P be a population sorted according to fitness. We define $\beta(\gamma) = \beta(\gamma, P)$ for $\gamma \in [0,1]$ as the probability of selecting an individual at least as fit as the individual with rank $\gamma\lambda$ in P. Note that $\beta(\gamma)$ is a cumulative probability function which depends on the selection mechanism used. In the case of (μ,λ)-selection, $\beta(\gamma)$ is a piece-wise linear function (Lehre 2011) where $\beta(\gamma) = \gamma\lambda/\mu$ if $\gamma \leq \mu/\lambda$ and $\beta(\gamma) = 1$ otherwise. In contrast, for k-tournament selection, β is a non-linear function $\beta(\gamma) = 1-(1-\gamma)^k$ (Lehre 2011). Assume that the current population contains $\gamma\lambda$ individuals on a given subset B of the search space, and that when the mutation operator is applied to a solution in B, it produces an offspring in B with probability p_0. Note that p_0 depends on the mutation rate χ and the structure of B. If the subset B represents promising new search points, we want this fraction to grow, which requires that $\beta(\gamma)p_0 > \gamma(1+\delta)$ when γ is small for some small constant $\delta > 0$. If B is a local optimum, we do not want the individuals in B to take over the population, so that less fit sub-populations have the chance to explore fitness valleys. We would therefore also require that $\beta(\gamma)p_0 < \gamma(1-\delta)$ when γ is large. It is easy to see that the non-linear β-function for tournament selection displays both properties: there exists a ψ such that $\beta(\gamma)p_0 \leq \gamma$ for $\gamma \in (\psi, 1)$, and $\beta(\gamma)p_0 \geq \gamma$ for $\gamma \in [0, \psi)$. In contrast, the piece-wise linear function β for (μ, λ)-selection can only display either one of the two properties exclusively.

In Algorithm 1, define $R_t(i) := \sum_{j\in[\lambda]}[I_t(j) = i]$, i.e. the number of times $P_t(i)$ is selected as parent at generation t. Then, $E[R_t(i)]$ is called the *reproductive rate* of the i-th individual of the population at time t (Lehre 2011).

Lemma 5 (Theorem 3 in (Dang and Lehre 2016b))**.** *Algorithm 1 with mutation parameter χ and an anytime upper bound α_0 of the reproductive rate satisfying $\chi \geq \ln\alpha_0 + \delta$ for some constant $\delta > 0$ has runtime such that $\Pr(T \leq e^{cn}) \leq e^{-\Omega(n)}$ for some constant $c > 0$ on any function with unique optimum x^* assuming that $H(P_0, x^*) > bn$ holds for some constant $b \in (0,1)$ with probability $1 - e^{-\Omega(n)}$.*

For (μ, λ)-selection of the $(\mu,\ \lambda)$ EA, each individual among the μ best of the population has the same probability $1/\mu$ of being selected as parent. Thus, they have the same reproductive rate λ/μ, which is also the highest of the population. Therefore, if the mutation parameter $\chi > \ln(\lambda/\mu)$, the lemma above implies that the $(\mu,\ \lambda)$ EA is inefficient on any function with a unique optimum, and also on f.

On the other hand, if the mutation parameter is below $\ln(\lambda/\mu)$, assuming that the population starts with the μ best individuals in B, it can be shown by induction that with high probability the next generations still have the μ best individuals from B. This implies that the other individuals of the population, particularly those in C which are closer to D, indeed have zero reproductive rate and cannot evolve.

Theorem 6. *For any constant $\delta > 0$, the $(\mu,\ \lambda)$ EA with mutation parameter satisfying $\chi = O(1)$ and $\chi \notin [\ln(\lambda/\mu) - \delta, \ln(\lambda/\mu) + \delta]$, with a population of size $\lambda = \Omega(n)$ has runtime T, such that $\Pr(T \leq e^{cn}) \leq e^{-\Omega(n)}$ on function f with $w - v = \Omega(n)$ and $n - w = \Omega(n)$ for some constant $c > 0$, assuming that the μ best individuals of P_0 are in the B region of f.*

Proof. The result for $\chi \geq \ln(\lambda/\mu) + \delta$ follows directly from Lemma 5 by noting that the μ best individuals of P_0 are still at distance $n - w$ away from the global optimum 1^n.

When $\chi \leq \ln(\lambda/\mu) - \delta$, it suffices to prove by induction that with high probability, the μ best individuals are still search points in B, i.e. non-optimal points, during the first $e^{c\lambda}$ generations for some constant c sufficiently small. Since the μ best individuals of P_0 are in B, it is certain that the selected parents to produce offspring in the next generation are from B. Then, to create an offspring in B, it suffices to not modify any bit from the parent, the corresponding probability is $\left(1 - \frac{\chi}{n}\right)^n \geq (1-\sigma)e^{-\chi} \geq \frac{\mu(1-\sigma)e^{\delta}}{\lambda}$ for any constant $\sigma \in (0,1)$, if n is large enough. Choosing σ so that $\frac{1+\sigma}{1-\sigma} = e^{\delta}$, this probability is at least $(1+\sigma)(\mu/\lambda)$. By a Chernoff bound, the probability that the population has less than μ individuals from B in the next generation is $e^{-\Omega(\lambda)} = e^{-\Omega(n)}$. For those individuals to be the best of the population, no individual in D must be created. The probability of creating a D-individual by mutating a B-individual is $n^{-\Omega(w-v)} = n^{-\Omega(n)}$. By a union bound, this event occurs with probability no more than $\lambda \cdot n^{-\Omega(n)} = n^{-\Omega(n)}$ within one generation. By induction and a union bound, with probability $1 - e^{-\Omega(n)}$ the μ best individuals are those from B during the next e^{cn} generations for some sufficiently small constant c. □

Efficiency of Tournament Selection

To analyse tournament selection, we will apply an analytical tool called the level-based theorem (Corus et al. 2018). The theorem is applicable to any population-based process where the individuals in P_{t+1} are sampled independently from the same distribution $D(P_t)$ parameterised by P_t (see Algorithm 3 in the supplementary material). For example, in Algorithm 1 we have $D = p_{\text{mut}} \circ p_{\text{sel}}$.

Theorem 7 (Theorem 1 in (Corus et al. 2018))**.** *Consider Algorithm 3 with population size λ. Given a partition $(A_1, \ldots, A_m)$ of $\mathcal{X}$, define $T := \min\{t\lambda \mid |P_t \cap A_m| > 0\}$, where for all $t \in \mathbb{N}$, $P_t \in \mathcal{X}^\lambda$ is the population in generation t. If there exist $z_1, \ldots, z_{m-1}, \delta \in (0,1]$, and $\gamma_0 \in (0,1)$ such that for any population $P \in \mathcal{X}^\lambda$, $y \sim D(P)$, any $j \leq m-1$, and any $\gamma \leq \gamma_0$*

(G1) *If $|P \cap A_{\geq j}| \geq \gamma_0\lambda$, then $\Pr(y \in A_{\geq j+1}) \geq z_j$,*

(G2) *If $|P \cap A_{\geq j}| \geq \gamma_0\lambda$ and $|P \cap A_{\geq j+1}| \geq \gamma\lambda$, then $\Pr(y \in A_{\geq j+1}) \geq (1+\delta)\gamma$,*

(G3) $\lambda \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{z_*\delta^2}\right)$, *where $z_* := \min_j z_j$,*

then $E[T] \leq \left(\frac{8}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda\ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right) + \frac{1}{z_j}\right)$.

Note that the original theorem in (Corus et al. 2018) requires a weaker condition on (G2) which does not need to hold for the level A_{m-1}. This can be useful in some applications but irrelevant for our case. In fact, we critically need (G2) to hold for the levels of the C region, i.e. the valley below B that can lead to optimality. For this purpose, we first

prove a variant of Lemma 3 in (Dang and Lehre 2016b) for 3-tournament selection because this will allow us to restrict the number of individuals in B.

Lemma 8. *Given any subset $B \subset \mathcal{X}$, let $Y_t := |P_t \cap B|$ be the number of individuals in generation $t \in \mathbb{N}$ of Algorithm 1 with p_{sel} being 3-tournament, that belong to B. If there exist three parameters $\sigma, \varepsilon \in (0,1)$ and $\rho \in (0, 1-\sigma)$ satisfying that for any $t \in [t_0, t_1]$, if $x = P_t(p_{\text{sel}}(P_t))$ and $y = p_{\text{mut}}(x)$ then $\Pr(y \in B \mid x \in B) \leq \rho$, and $\Pr(x \notin B \wedge y \in B) \leq \sigma\psi - \varepsilon$ where $\psi := (1/2)\left(3 - \sqrt{(4-3\rho-4\sigma)/\rho}\right)$, then for any $t \in [t_0, t_1]$, $\Pr(Y_t \geq \max(\psi\lambda, (1-\varepsilon/2)^{t-t_0} Y_{t_0})) \leq (t-t_0)\cdot e^{-\Omega(\lambda)}$.*

Proof. Let $\gamma := Y_t/\lambda \leq 1$. It can be shown that the parameter ψ is monotonically increasing wrt ρ when $\rho \in (0, (4/3)(1-\sigma))$. Under the assumption $0 < \rho < 1-\sigma$, the parameter ψ is therefore bounded by $\psi < 1$. Given the definition of γ, we also have $\max(\gamma, \psi) \leq 1$.

For an upper bound, assume that the individuals in B are fitter than any other individual, then

$$\begin{aligned}
&\Pr(x \in B \wedge y \in B)\\
&= \Pr(x \in B)\Pr(y \in B \mid x \in B)\\
&\leq (1-(1-\gamma)^3)\rho\\
&\leq (1-(1-\max(\psi,\gamma))^3)\rho\\
&= \max(\psi,\gamma)(3-\max(\psi,\gamma)(3-\max(\psi,\gamma)))\rho\\
&\leq \max(\psi,\gamma)(3-\psi(3-\psi))\rho\\
&= \max(\psi,\gamma)(1-\sigma).
\end{aligned}$$

In the last inequality above, we used that the function $h(x) = x(3-x)$ increases monotonically on the interval $[0,1)$. Thus, using that $\max(\psi,\gamma) < 1$, the probability of producing an individual in B is

$$\begin{aligned}
\Pr(y \in B) &= \Pr(x \in B \wedge y \in B) + \Pr(x \notin B \wedge y \in B)\\
&\leq \max(\psi,\gamma)(1-\sigma) + \sigma\psi - \varepsilon\\
&\leq \max(\psi,\gamma)(1-\sigma) + \max(\psi,\gamma)\sigma - \varepsilon\\
&\leq \max(\psi,\gamma)(1-\varepsilon) =: p_s.
\end{aligned}$$

Hence Y_{t+1} is stochastically dominated by the random variable $Z \sim \text{Bin}(\lambda, p_s)$. It now follows by a Chernoff bound that

$$\begin{aligned}
&\Pr(Y_{t+1} \geq \max(\psi\lambda, Y_t(1-\varepsilon/2)))\\
&\quad\leq \Pr(Z \geq \max(\psi\lambda, Y_t(1-\varepsilon/2))\\
&\quad\leq \Pr\left(Z \geq E[Z]\left(1 + \frac{\varepsilon}{2(1-\varepsilon)}\right)\right)\\
&\quad\leq \exp\left(-\frac{\varepsilon^2 \max(\psi\lambda, Y_t)}{12(1-\varepsilon)}\right) \leq e^{-\frac{\varepsilon^2\psi\lambda}{12(1-\varepsilon)}}.
\end{aligned}$$

The proof is completed by induction with respect to t and a union bound. □

Our main result of the section is the following.

Theorem 9. *There exists a constant $c > 0$ such that the expected runtime of Algorithm 1 with 3-tournament as p_{sel}, population size $\lambda \geq c\log(n)$, mutation parameter $\chi = 1.09812$ on function f with $w/n \leq 3/4$ is $O(n\lambda\log(\lambda) + n^2\log(n))$.*

To prove the result, we use the following partition of the search space into $m = 2n + u + 2$ levels:

$$U_j := \begin{cases} B' & \text{if } j = -n-u-1\\ \{x \in A' \mid \textsc{Om}(x) = u+j\} & \text{if } j \in [-n-u, -1]\\ \{x \in A \mid \textsc{Lo}(x) = j\} & \text{if } j \in [0, u],\\ \{x \in B \mid \textsc{Lo}(x) = j\} & \text{if } j \in [u+1, v],\\ \{x \in C \mid \textsc{Lo}(x) = j\} & \text{if } j \in [v+1, w], \text{ and}\\ \{x \in D \mid \textsc{Lo}(x) = j\} & \text{if } j \in [w+1, n]. \end{cases}$$

Particularly, when j falls into the appropriate range we will also use X_j to refer to the internal level U_j of X, i. e. $U_j \subseteq X$, here X can be either A', B', A, B, C, or D. A similar notation also uses for $X_{\geq j}$, e. g. $C_{\geq w-1} = U_{\geq w-1} = C_{w-1} \cup D$.

The following lemma ensures that condition (G2) of Theorem 7 is satisfied in region C.

Lemma 10 (Condition (G2))**.** *There exist constants $c, \delta, \gamma_0 > 0$ and a failure event $\mathcal{F}$ with probability $e^{-\Omega(\lambda)}$, such that unless the failure event occurs, for any population $P_t \in \mathcal{X}^\lambda$ with $t \leq e^{c\lambda}$ of Algorithm 1 with p_{sel} being 3-tournament, mutation parameter $\chi := 1.09812$ running on f with $w/n \leq 3/4$, for any $\gamma \in (0, \gamma_0)$, if $|P_t \cap C_{\geq j+1}| = \gamma\lambda$ and $y = p_{\text{mut}}(P_t(p_{\text{sel}}(P_t)))$ then $\Pr(y \in U_{\geq j+1}) \geq \gamma(1+\delta)$.*

Note that Theorem 9 assumes a very specific value 1.09812 for the mutation parameter χ. For k-tournament selection this value is close to the natural logarithm of the tight upper bound by Bernoulli's inequality on the reproductive rate of the best individual $\lambda(1-(1-1/\lambda)^k) \leq \lambda(1-(1-k/\lambda)) = k$, e. g. if $k = 3$ then $\ln(3) - 1.09812$ is positive, but close to 0. It is essential for the non-elitist populations to escape the local optima, and this will not occur if the mutation rate is too low. To complete the picture, we now consider 3-tournament selection for χ below $\ln(3)$. The following theorem implies that given the likely assumption that a large fraction of the population reaches region B, if the mutation rate is below $\chi \leq \ln(\xi) \approx 0.647461$, then the algorithm will not be able to optimise the problem in polynomial time.

Theorem 11. *There exist a constant $\psi \in (0,1)$ such that if Algorithm 1 with 3-tournament selection, and mutation rate $\chi \leq \ln(\xi(1-\delta))$ for $\xi := \frac{4}{3} + \frac{1}{\sqrt{3}} - \delta$ and any constant $\delta \in (0,1)$ obtains a population with at least $\psi\lambda$ individuals in the B-region, then $\Pr(T \leq e^{cn}) \leq e^{-\Omega(\lambda)}$ for a small constant $c > 0$.*

Experiments

To extend our theoretical analysis and observations to practical problems, we carried out some preliminary experiments with the non-elitist EAs on several instances of the Set Cover Problem (SCP) from the OR-library (Beasley 1990), representing the CYC and CLR families (Grossman and Wool 1997) and the Stein (Fulkerson, Nemhauser, and Trotter 1974) family of hard unicost benchmarks. The problem series CYC and CLR are related to the combinatorial questions posed by Paul Erdős in (Erdős 1990) and (Erdős 1963). The Stein series is based on Steiner triple systems, which have applications e. g. in the statistical design of experiments.

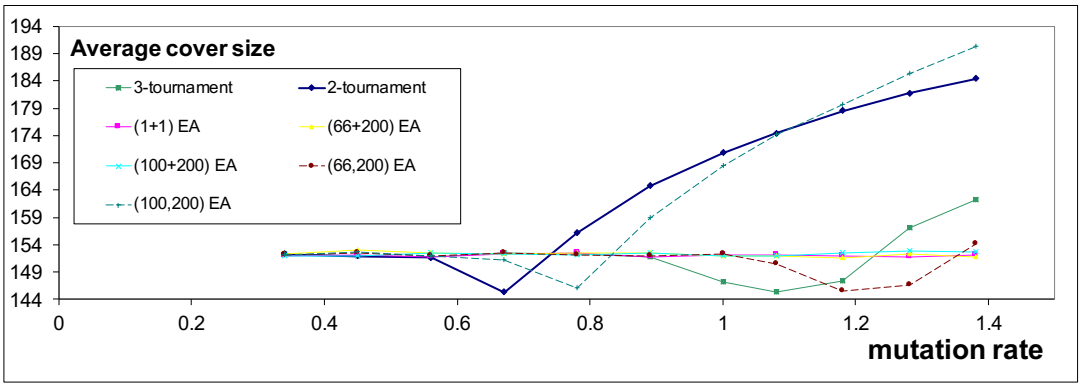


Figure 2: Average cover size as a function of mutation rate χ on CYC.7.

SCP is a constraint combinatorial optimisation problem, and we use the penalty method to reduce it to the optimisation of a pseudo-Boolean function. We assume that each bit x_i of a solution corresponds to a covering subset, so that the value of $x_i = 1$ means that subset i is included into the solution, then we assign a unit of penalty for each uncovered element. An uncovered element can easily be covered with one new covering subset, so the objective of a solution repaired from infeasibility in a unicost instance is at most the number of ones, plus the penalty value. The practical performance of this penalty method in the case of the Independent Set Problem was demonstrated in (Borisovsky and Zavolovskaya 2003).

In this paper, we experimented with seven algorithms: the (1+1) EA, (100+200) EA, (66+200) EA, (100, 200) EA, (66, 200) EA, and the non-elitist EAs with tournament selection of size 2 and 3. The population parameters used in comma selection were chosen so that the estimates of the error threshold for mutation, i. e. $\ln(\lambda/\mu)$, approximately match those in tournament selection, i. e. $\ln(k)$ from our theory. The same parameters were then carried over to plus selection. The algorithms were implemented in C++ using its standard library for the random number generation. They were compiled, then called from a Python program running on a server machine with AMD EPYC 7502 processors, Ubuntu 20.04 OS, GCC 9.3.0 and Python 3.8.3. Each algorithm was allowed the same budget of 2×10^8 function calls to evaluate its solutions, and each setting on each instance were tested with 40 replications of the run using different random seeds.

To illustrate an example of the output results, Fig. 2 shows the average cover sizes in the experiments for instance CYC.7, where $n = 448$. The graphs of non-elitist EAs with tournament selection are passing close to the best-known solution size 144 when $\chi = 0.67$ if the tournament size is $k = 2$, and when $\chi = 1.08$ if $k = 3$. These χ-values are slightly less than $\ln(2)$ and $\ln(3)$, the error threshold rates for tournament sizes 2 and 3. The average performances of the elitist (μ+λ) EA are analogous one to another, and much worse when compared to the two tournament settings. These observations match perfectly with what our developed theory would suggest. The EAs with (μ, λ)-selection mechanism performs only slightly worse than the non-elitist EAs with tournament selection. This is interesting and worthy of further investigations in the sense that the induced landscape may not be typically as hard as the one studied in our theoretical results.

On the instance CYC.6 where $n = 192$, the average solution cost behaves similarly, attaining the optimal value of 60 at $\chi = 0.67$ and $\chi = 1.08$ for tournament sizes 2 and 3, respectively. Fig. 3 shows the frequency of obtaining the optimum (of size 60) and the 95% confidence intervals of the probability to find the optimum, computed using the Bernoulli distribution quantiles. (For better visibility, only results for selected algorithms are shown.) Again we remark that the non-elitist EA with mutation rate slightly below the error threshold significantly outperforms the elitist (1+1) EA. However, relatively small deviations of χ from the error threshold lead to dramatic drops in performance.

Using the same experimental setting and tuneable parameters, we tested the seven algorithms on instances CLR.10, CLR.11, Stein.81 and Stein.135. In general, these experiments demonstrated similar tendencies to those observed on CYC.6 and CYC.7, as discussed above, with one exception for the instance Stein.81. On this instance, the best settings for the mutation rates are significantly higher than the error threshold and the elitist algorithm (1+1) EA is quite competitive (see Fig. 4). The plots with confidence intervals, and the source code with the instruction on how to reproduce the experiments are provided in the supplementary material.

Conclusion

Despite promises that diverse populations in EAs can cope with multi-modal problems, overcoming local optima remains one of the major challenges in evolutionary computation. We suggest this problem is related to the biologically implausible, but prevalent practice, of enforcing elitism.

Solving a long-standing open problem, we show how and when non-elitist population-based EAs without any additional enhancements can cope with local optima. We have proved that non-elitist EAs with the right set of operators, here with tournament selection and bitwise mutation, and under the right setting, i. e. mutation rate below but close to the error threshold, is able to both explore fitness valleys and escape the local optima. This leads to the efficient optimisation on a class of benchmark functions with a local optimum. However elitist EAs or ill-configured non-elitist

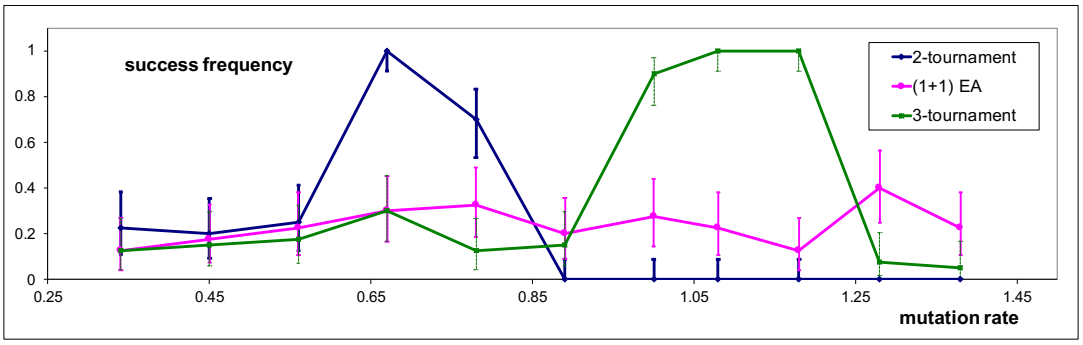


Figure 3: Frequency of successful runs and confidence intervals for success probability on CYC.6.

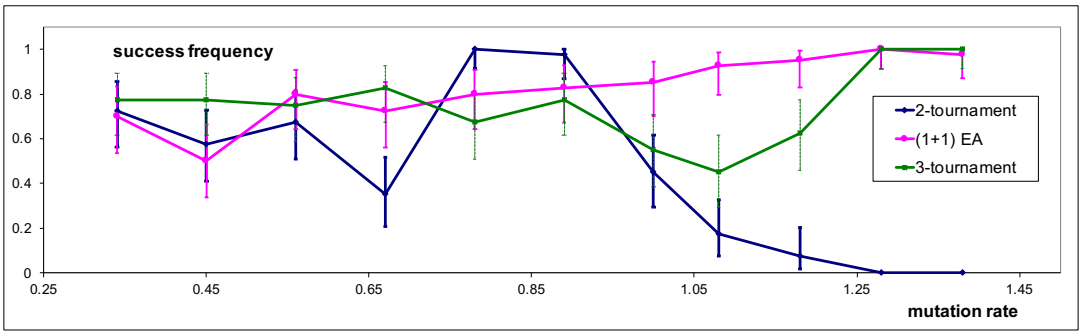


Figure 4: Frequency of successful runs and confidence intervals for success probability on Stein.81.

EAs, e. g. employing (μ,λ)-selection or too low mutation rate, are shown to stagnate at the local optimum, resulting in super-polynomial runtime.

Preliminary experiments conducted on instances of the Set Cover problem demonstrated the possibility to extend our theoretical investigation to practical problems. In particular, we demonstrate that on some well-known benchmark instances of diverse origins it may be recommended to use mutation rates close to the error threshold when employing non-elitist population-based EAs. Moreover, given appropriate mutation rates, these algorithms tend to outperform the elitist EAs in a similar experimental setting. One exception observed in the case of Stein.81 instance implies that these recommendations are not universal and further research is required to understand how the fitness landscape properties influence the runtime of the EA.

The analysis should be extended to wider ranges of selection mechanisms, such as exponential ranking selection, and mutation operators, such as with heavy-tailed mutation rates (Doerr et al. 2017). Future work should compare non-elitist EAs with elitist EAs employing diversity mechanisms, such as stochastic ageing (Oliveto and Sudholt 2014). The performance of EAs are here compared in terms of their runtime. In the future, it would be interesting to compare the quality of the best solutions obtained by elitist and non-elitist EAs within a fixed number of function evaluations (Jansen and Zarges 2014).

Acknowledgements

Eremeev was supported by program of fundamental scientific research of the Russian Academy of Sciences, I.5.1, project 0314-2019-0019. Lehre was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1). The first and third authors would like to thank Tiago Paixão for useful discussions about non-elitism in population genetics.

References

Beasley, J. E. 1990. OR-Library: Distributing Test Problems by Electronic Mail. *The Journal of the Operational Research Society* 41(11): 1069–1072.

Beyer, H.-G. 1997. G4.2 Design optimization of a linear accelerator using evolution strategy: solving a TSP-like optimization problem. In *Handbook of Evolutionary Computation*. IOP Publishing Ltd.

Biebricher, C. K.; and Eigen, M. 2005. The error threshold. *Virus Research* 107(2): 117–127. doi:10.1016/j.virusres.2004.11.002.

Borisovsky, P. A.; and Zavolovskaya, M. S. 2003. Experimental Comparison of Two Evolutionary Algorithms for the Independent Set Problem. In *Proceedings of EvoWorkshop 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM (EvoWorkshop'2003)*, Applications of Evolutionary Computing, 154–164. Berlin, Heidelberg: Springer-Verlag.

Case, B.; and Lehre, P. K. 2020. Self-adaptation in non-Elitist Evolutionary Algorithms on Discrete Problems with Un-

known Structure. *IEEE Transactions on Evolutionary Computation* 24(4): 650–663. doi:10.1109/TEVC.2020.2985450.

Cavicchio, D. J. 1970. *Adaptive Search Using Simulated Evolution*. Ph.D. thesis, University of Michigan, Ann Arbor, MI.

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

Dang, D.-C.; and Lehre, P. K. 2016a. Runtime Analysis of Non-elitist Populations: From Classical Optimisation to Partial Information. *Algorithmica* 75: 428–461. doi:10.1007/s00453-015-0103-x.

Dang, D.-C.; and Lehre, P. K. 2016b. Self-adaptation of Mutation Rates in Non-elitist Populations. In *Proceedings of the 2016 Conference on Parallel Problem Solving from Nature (PPSN 2016)*, 803–813. Cham: Springer. doi:10.1007/978-3-319-45823-6_75.

De Jong, K. A. 1975. *An analysis of the behavior of a class of genetic adaptive systems.* Ph.D. thesis, University of Michigan, USA. AAI7609381.

Doerr, B. 2020. Does comma selection help to cope with local optima? In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO 2020)*, 1304–1313. New York, NY, USA: ACM.

Doerr, B.; Le, H. P.; Makhmara, R.; and Nguyen, T. D. 2017. Fast genetic algorithms. In *Proceedings of the 2017 Genetic and Evolutionary Computation Conference (GECCO' 2017)*, 777–784. New York, NY, USA: ACM.

Erdős, P. 1963. On a combinatorial problem. *Nordisk Matematisk Tidskrift* 11(1): 5–10.

Erdős, P. 1990. On some of my favourite problems in graph theory and block designs. *Le Matematiche* 45(1): 61–74.

Fulkerson, D. R.; Nemhauser, G. L.; and Trotter, L. E. 1974. *Two computationally difficult set covering problems that arise in computing the 1-width of incidence matrices of Steiner triple systems*, 72–81. Berlin, Heidelberg: Springer.

Grossman, T.; and Wool, A. 1997. Computational experience with approximation algorithms for the set covering problem. *European Journal of Operational Research* 101(1): 81–92.

Jägerskupper, J.; and Storch, T. 2007. When the Plus Strategy Outperforms the Comma Strategy and When Not. In *2007 IEEE Symposium on Foundations of Computational Intelligence*, 25–32.

Jansen, T.; and Zarges, C. 2014. Performance analysis of randomised search heuristics operating with a fixed budget. *Theoretical Computer Science* 545: 39 – 58. doi:10.1016/j.tcs.2013.06.007.

Lehre, P. K. 2010. Negative Drift in Populations. In *Proceedings of the 2010 Conference on Parallel Problem Solving from Nature (PPSN'2010)*, 244–253. Berlin, Heidelberg: Springer.

Lehre, P. K. 2011. Fitness-Levels for Non-Elitist Populations. In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference (GECCO 2011)*, 2075–2082. ACM. doi:10.1145/2001576.2001855.

Lehre, P. K.; and Yao, X. 2012. On the impact of mutation-selection balance on the runtime of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 16(2): 225–241. doi:10.1109/TEVC.2011.2112665.

Ochoa, G. 2006. Error Thresholds in Genetic Algorithms. *Evolutionary Computation* 14(2): 157–182. ISSN 1063-6560, 1530-9304. URL https://www.mitpressjournals.org/doi/abs/10.1162/evco.2006.14.2.157.

Oliveto, P. S.; Paixão, T.; Heredia, J. P.; Sudholt, D.; and Trubenová, B. 2018. How to Escape Local Optima in Black Box Optimisation: When Non-elitism Outperforms Elitism. *Algorithmica* 80(5): 1604–1633. doi:10.1007/s00453-017-0369-2.

Oliveto, P. S.; and Sudholt, D. 2014. On the runtime analysis of stochastic ageing mechanisms. In *Proceedings of the 2014 Genetic and Evolutionary Computation Conference (GECCO 2014)*, 113–120. New York, NY, USA: ACM. doi:10.1145/2576768.2598328.

Paixão, T.; Heredia, J. P.; Sudholt, D.; and Trubenová, B. 2017. Towards a Runtime Comparison of Natural and Artificial Evolution. *Algorithmica* 78(2): 681–713. doi:10.1007/s00453-016-0212-1.

Rudolph, G. 1994. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks* 5(1): 96–101.

Wilke, C. O. 2005. Quasispecies theory in the context of population genetics. *BMC Evolutionary Biology* 5(1): 44. doi:10.1186/1471-2148-5-44.