UNIVERSITY OF BIRMINGHAM
University of Birmingham Research at Birmingham

Fast non-elitist evolutionary algorithms with power-law ranking selection

Dang, Duc-Cuong; Eremeev, Anton; Lehre, Per Kristian; Qin, Xiaoyu

DOI:
10.1145/3512290.3528873

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Dang, D-C, Eremeev, A, Lehre, PK & Qin, X 2022, Fast non-elitist evolutionary algorithms with power-law ranking selection. in JE Fieldsend (ed.), *GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference.* GECCO: Genetic and Evolutionary Computation Conference, Association for Computing Machinery (ACM), New York, pp. 1372-1380, GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, United States, 9/07/22. https://doi.org/10.1145/3512290.3528873

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
© 2022 ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference, https://doi.org/10.1145/3512290.3528873

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 10. Nov. 2026

Fast Non-elitist Evolutionary Algorithms with Power-law Ranking Selection

Duc-Cuong Dang
University of Passau, Passau, Germany
University of Southampton, Southampton, UK

Anton Eremeev
Sobolev Institute of Mathematics SB RAS, Novosibirsk
Russia

Per Kristian Lehre
University of Birmingham, Birmingham, UK
The Alan Turing Institute, London, UK

Xiaoyu Qin
University of Birmingham, Birmingham, UK

ABSTRACT

Theoretical evidence suggests that non-elitist evolutionary algorithms (EAs) with non-linear selection mechanisms can efficiently overcome broad classes of local optima where elitist EAs fail. However, the analysis assumes a weak selective pressure and mutation rates carefully chosen close to the "error threshold", above which they cease to be efficient. On problems easier for hill-climbing, the populations may slow down these algorithms, leading to worse runtime compared with variants of the elitist (1+1) EA.

Here, we show that a non-elitist EA with power-law ranking selection leads to fast runtime on easy benchmark problems, while maintaining the capability of escaping certain local optima where the elitist EAs spend exponential time in the expectation.

We derive a variant of the level-based theorem which accounts for power-law distributions. For classical theoretical benchmarks, the expected runtime is stated with small leading constants. For complex, multi-modal fitness landscapes, we provide sufficient conditions for polynomial optimisation, formulated in terms of deceptive regions sparsity and fitness valleys density. We derive the error threshold and show extreme tolerance to high mutation rates. Experiments on NK-Landscape functions, generated according to the Kauffman's model, show that the algorithm outperforms the (1+1) EA and the univariate marginal distribution algorithm (UMDA).

CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**.

KEYWORDS

Runtime analysis, Power-law ranking, Selection, Local optima

ACM Reference Format:
Duc-Cuong Dang, Anton Eremeev, Per Kristian Lehre, and Xiaoyu Qin. 2022. Fast Non-elitist Evolutionary Algorithms with Power-law Ranking Selection. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3512290.3528873

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '22, July 9–13, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528873

1 INTRODUCTION

Selection along with mutation are the key ingredients of evolution. Without selection, advantage features cannot be passed on to the next generations, thus there will be no adaptation, and this also holds true for evolutionary algorithms. Early evolutionary algorithms (EAs) were formulated with fitness-proportionate (or roulette-wheel) selection for reproduction [16]. The drawback of such selection is the non-resilience to a scaling of fitness, thus small but perhaps important advantages can be undetected in a large population. This drawback can be addressed with ranking selection [2], where selection probabilities are assigned with respect to the ranks of individuals sorted by fitness. This category includes (μ, λ)-selection (or truncation selection), tournament selection, linear and exponential ranking selections. Characteristics of these operators in one generation was studied intensely in early EAs theory [4, 25].

In this paper, we introduce a new parent selection mechanism, which is simple to describe in terms of cumulative selection probabilities. That is the probability of selecting an individual at rank i or better is $(i/\lambda)^c$ for a population of size λ and here $c \in (0, 1)$ is the parameter of selection. We dub this new selection mechanism the *power-law ranking selection* because from the view probability mass function, the selection probability is a power-law function of the rank. As far as we know, this mechanism has not been studied before, and as we will show that it exposes many interesting properties. To see a flavour of these, one can consider $c = 1/2$ and observe that the best individual has $\sqrt{\lambda}$ expected offspring in the next generation, thus the replication rate of highly fit individuals is extremely high. The cumulative selection function on the other hand is non-linear and suits well to escape local optima [6, 7].

The mechanism also allows non-elitist populations to operate with high mutation rates, and this can be beneficial in some settings. In fact the tolerance limit of the mutation rate which is also known as the *error threshold* [20, 27] can be scaled with the population size. We will assert these properties by means of rigorous runtime analysis and complement them with experiments on both benchmark functions and NK-Landscape functions, randomly generated according to the Kauffman's model [19].

Emerged in the early 2000s, runtime analysis [3] has been a standard approach to study EAs from the theoretical perspective [18]. Early research and also a large body of work in this area has been focused on mutation while making simplifications about

Algorithm 1 Non-elitist EA with unary variation operator [8]

Input: A mutation rate parameter $\chi \in [0, n]$. A population size $\lambda \in \mathbb{N}$ and an initial population $P_0 \in \mathcal{X}^\lambda$ where $\mathcal{X} = \{0,1\}^n$.

1: **for** $t = 0, 1, 2, \ldots$ until the termination condition is met **do**
2: **for** $i = 1$ to λ **do**
3: Sample $I_t(i) \sim p_{\text{sel}}(P_t)$, and set $x := P_t(I_t(i))$.
4: Sample $x' \sim p_{\text{mut}}(x, \chi)$, and set $P_{t+1}(i) := x'$.
5: **end for**
6: **end for**

selection, i. e. using the so-called plus (or elitist) selection within a small population, e. g. resulting in the (1+1) EA. The studies of non-elitist populations are more scarce, however recently some interesting results have been obtained. It has been shown in [11] that asymptotically on Jump the (μ, λ)-selection cannot perform better than the plus selection. In [6], Dang et al. identified the issue as the linearity of the cumulative selection function (see the details in Subsection 2.1), and derived a so-called Funnel function where other non-elitist selection mechanisms like the tournament selection can be efficient. This is generalised to a large class (known as SparseLocalOpt$_{\alpha,\varepsilon}$) in [7], where black-box EAs with plus selection require exponential expected runtime, while non-elitist EAs with the right settings achieve polynomial performance. We will consider these functions and show that the power-law ranking achieves a better performance with a wider range of settings.

The level-based analysis from [21] and its later refinements in [5, 8, 13] have become a general-purpose tool for runtime analysis of non-elitist populations and complex EAs [9, 22, 24]. We will derive a variant of the level-based theorem of [5], adapted to the rapid replication of good individuals, which is observed in the new selection mechanism. The novelty here is that the derived tool allows to derive the runtime bounds precisely up to the leading constants. As we propose a new operator and hence new algorithms, our research is aligned with the development of new theory-founded evolutionary algorithms, such as the (1+(λ,λ)) GA [12] or the fast genetic algorithm using power-law mutation [14]. The most recent results in this direction may be found in [1] and the references therein.

The rest of this paper is structured as follows. We will first provide some basic definitions, including the formal definition of the power-law selection mechanism. The next section analyses the performance of the new mechanism on the standard benchmark functions, and introduces the new variant of the level-based theorem. We then discuss how to tune the algorithms to escape the local optima of SparseLocalOpt$_{\alpha,\varepsilon}$, and the error threshold that the population can tolerate with the new selection. Our experimental section will demonstrate the performance of EAs with the new operator compared with the existing ones on Funnel, and random NK-Landscape functions. We conclude the paper with some final remarks. Due to the space restriction, some proofs are omitted.

2 PRELIMINARIES

For any $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. The search space is $\mathcal{X} := \{0,1\}^n$. A *population* is a vector $P \in \mathcal{X}^\lambda$, the i-th individual of P is denoted $P(i)$. For $A \subseteq \mathcal{X}$, we define $|P \cap A| := |\{i \mid P(i) \in A\}|$. For any logical predicate $\mathcal{P}$, $[\mathcal{P}]$ denotes the Iverson-bracket which equals 1 if $\mathcal{P}$ holds and 0 otherwise. $H(x, y)$ is the Hamming-distance between bitstrings x and y. We consider non-elitist EAs with the outline of Algorithm 1. Here, the population P_{t+1} is generated by independently sampling λ individuals from population P_t according to a probability distribution $p_{\text{sel}}(P_t)$ on $[\lambda]$ (a selection operator), then by perturbing each of the selected individuals using a unary variation (mutation) operator with a probability distribution $p_{\text{mut}}(x, \chi)$ on $\mathcal{X}$. The distribution $p_{\text{sel}}(P_t)$ is parametrised by population P_t (and maybe some tunable selection parameter). The distribution $p_{\text{mut}}(x, \chi)$ is parametrised by the parent genotype x and $\chi \in [0, n]$. We consider the standard *bitwise mutation operator*, such that for any pair of bitstrings $x, x' \in \{0,1\}^n$, the probability of obtaining x' from x is $\Pr(x' = p_{\text{mut}}(x, \chi)) = (\chi/n)^{\mathrm{H}(x,x')}\,(1 - \chi/n)^{n-\mathrm{H}(x,x')}$. Algorithm 1 belongs to a more general class of algorithms outlined by Algorithm 2 [5].

Algorithm 2 Population-based algorithm [5].

Input: A finite state space $\mathcal{X}$, and population size $\lambda \in \mathbb{N}$, a mapping D from $\mathcal{X}^\lambda$ to the space of probability distributions over $\mathcal{X}$, and an initial population $P_0 \in \mathcal{X}^\lambda$.

1: **for** $t = 0, 1, 2, \ldots$ until the termination condition is met **do**
2: **for** $i = 1$ to λ **do**
3: Sample $P_{t+1}(i) \sim D(P_t)$
4: **end for**
5: **end for**

2.1 Power-law Ranking Selection

In ranking selection mechanisms, individuals are sorted according to their fitness in the population, and the expected number of offspring for each parent is defined by its position in this sorted population [2]. In our paper, the EA outline requires independence of all λ selection outcomes, given the current population P_t, which implies that instead of a more general concept of the expected number of offspring, we will talk only about selection probability $p_{\text{sel}}(j, P_t)$ of an individual $P_t(j)$, $j \in [\lambda]$. An individual $P_t(j)$ here has the expected number of offspring $\lambda p_{\text{sel}}(j, P_t)$. In case of equal fitness values of some individuals, we assume that the sorting breaks ties uniformly at random in each call to the selection operator.

We will say that an individual has a rank $\gamma \in (1/\lambda, 1]$, if its position in the sorted population is $\gamma\lambda$, counting from the best individual at position 1. Following [17], we can consider an *assignment function* $\alpha : [0,1] \to [0,1]$, such that $\alpha(\gamma) \geq 0$ for all $\gamma \in [0,1]$ and $\int_0^1 \alpha(x)dx = 1$. Then a ranking selection mechanism may be defined by the probability $\beta(\gamma_1, \gamma_2) := \int_{\gamma_1}^{\gamma_2} \alpha(x)dx$, and the cumulative probability $\beta(\gamma) := \beta(0, \gamma)$ of selecting sufficiently fit individuals with rank at most γ. (Note that in publications on the power-law *mutation*, e.g. [14], symbol β has a different meaning.) By *linearity of the cumulative selection function* here and in [6, 7] we mean that $\beta(\gamma)$ grows linearly in γ until $\beta(\gamma)$ reaches 1.

In practical implementations, we are interested only in λ values $\{\beta(i/\lambda),\ i \in [\lambda]\}$ so that for an individual $P_t(j)$ of rank i/λ we have $p_{\text{sel}}(j, P_t) = \beta(i/\lambda) - \beta((i-1)/\lambda)$. This value can be computed and stored before the main loop of the algorithm. Besides that, sorting of the population may be performed in $O(\lambda \log \lambda)$ time just once at each iteration t so that each subset of equally-fit individuals is placed into the same slot. Then in each call to the selection

operator, like in the proportionate selection (see e.g. [17]), it suffices to perform a binary search and to choose a random element of the located slot in case of a tie. The overall time complexity of selection in each generation is $O(\lambda \log \lambda)$. Alternatively, for mechanisms that have an expression for $\beta^{-1}(\gamma)$, the inverse transform sampling [10] can be used without the need to compute $p_{\text{sel}}(j, P_t)$ explicitly.

While in many publications, the ranking selection is defined by the assignment function $\alpha(x)$, see e.g. [2, 17, 21], in the present paper we consider the case of ranking selection defined by a function $\beta(\gamma) = \gamma^c$ of power-law type with parameter $c \in (0, 1)$. This type of function assigns high probability mass near 0. Its shape may be adjusted for specific problem instances, at the extreme cases approaching the $(1, \lambda)$-selection if $c \to 0$, or the uniform selection if $c \to 1$. Usage of $c > 1$ is not appropriate because it will favour unfit individuals. We also note the inverse $\beta^{-1}(\gamma) = \gamma^{1/c}$.

3 POWER-LAW SELECTION IS FAST ON LEADINGONES AND JUMP

We consider the following two standard benchmark functions (see e.g. [15]). Let $x = (x_1 x_2 \dots x_n) \in \{0,1\}^n$, then $\textsc{LeadingOnes}(x)$ equals to $\sum_{i=1}^{n} \prod_{j=1}^{i} x_j$, and $\textsc{Jump}_r(x)$ is defined as $r + |x|_1 - 1$ if $|x|_1 \le n - r$, or as $n - |x|_1 - 1$, if $n - r < |x|_1 < n$, or set to n, if $|x|_1 = n$. For asymptotic results for power-law ranking, it is straightforward to use the standard level-based theorem of [5] to produce such results for all the benchmark functions. However, to obtain precise upper bounds on the expected optimisation up to the leading constants, we need the following variant which adapts to the fast spreading at non-linear rate of good individuals.

Theorem 1. *Given a partition $(A_1, \dots, A_m)$ 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 of Algorithm 2 in generation t. If there exist $z_1, \dots, z_{m-1}, c \in (0,1]$, $d, \gamma_0 \in (0,1)$, $\delta \in (0, 1/\gamma_0 - 1]$, and $\phi > 0$ such that condition (LB) holds:*

(LB) *if $X \sim \text{Bin}(\lambda, p)$ where $p \ge (i/\lambda)^c (1+\delta)\gamma_0^{1-c}$ and $1 \le i \le \gamma_0\lambda$ then $E\left[\ln\left(\frac{1+X}{1+i}\right)\right] \ge \phi$,*

and for any population $P \in \mathcal{X}^\lambda$, conditions (G1)-(G3) hold:

(G1) *for each level $j \in [m-1]$, if $|P \cap A_{\ge j}| \ge \gamma_0\lambda$, then*

$$\Pr_{y \sim D(P)}\left(y \in A_{\ge j+1}\right) \ge z_j,$$

(G2) *for each level $j \in [m-2]$, and all $\gamma \in (0, \gamma_0]$ if $|P \cap A_{\ge j}| \ge \gamma_0\lambda$ and $|P \cap A_{\ge j+1}| \ge \gamma\lambda$, then*

$$\Pr_{y \sim D(P)}\left(y \in A_{\ge j+1}\right) \ge (1+\delta)\gamma_0^{1-c}\gamma^c,$$

(G3) *and the population size $\lambda \in \mathbb{N}$ satisfies*

$$\lambda \ge \frac{1}{2\gamma_0^2\delta^2} \ln\left(\left(\frac{(1+\delta)\gamma_0^{1-c}}{\phi c} + 2\right)\frac{m-1}{d(1-e^{-\delta})z_*} + \frac{1}{d}\right)$$

where $z_* := \min_{j \in [m-1]}\{z_j\}$,

then with $r_i := \left(\frac{z_i}{(1+\delta)\gamma_0^{1-c}}\right)^{\frac{1}{c}}$ and $q_i := 1 - (1-z_i)^\lambda$ we have

$$E[T] \le \frac{\lambda}{(1-d)(1-e^{-\delta})} \sum_{i=1}^{m-1}\left(\frac{1}{\phi}\ln\left(\frac{1+\lambda}{1+\lambda r_i}\right) + \frac{1}{q_i}\right).$$

The difference with the standard version of [5] is that we require a stronger and non-linear lower bound in (G2). Condition (LB) is introduced to allow the log-transform of the spreading process to be analysed separately, thus we have a *pluggable* level-based theorem. The *drift* (or speed) of this spreading is characterised by parameter ϕ and it only impacts the first component of the expected runtime. Parameter d appears inside the natural log-factor of the lower bound in (G3), and influences the overall expected runtime when the population size is set close to that bound. The rest of the parameters are analogous to the ones in [5], however note that here we allow δ to be large (> 1) and this is crucial for our applications. The proof of Theorem 1, which is omitted due to the space restriction, follows the same steps of drift analysis as in the proof of the standard level-based theorem in [5], however, it uses a different distance function that is able to support large δ and ϕ.

Condition (LB) can be addressed with Lemma 2. The lemma provides a *lower bound* on the expectation of a random variable transformed by a concave function. It is significantly easier to derive an *upper bound* for the same quantity, e.g., using Jensen's inequality.

Lemma 2. *Let $X \sim \text{Bin}(\lambda, p)$ for $p \ge (i/\lambda)^c(1+\delta)\gamma_0^{1-c}$ for any $1 \le i \le \gamma_0\lambda$ with $\gamma_0 = \omega(1/\lambda)$ and any constants $c \in (0,1)$ and $\delta > 0$, it holds that*

$$E\left[\ln\left(\frac{1+X}{1+i}\right)\right] \ge \ln\left(1 + \frac{ab\delta}{1+b}\right) - o(1)$$

where a, b are any constants in $(0,1)$ and $(1, (\gamma_0\lambda)^{1-c}]$ respectively, and the small-o and ω-notation is with respect to the growth of λ.

For the bitwise mutation with the standard parameter $\chi = 1$, we have the following result.

Theorem 3. *Given any constants $\varepsilon, \delta > 0$, $c \in (0,1)$, let $L(c,\delta) := \left(\frac{e^{\frac{2-c}{1-c}}(1+\delta)^{\frac{c}{1-c}}}{1-e^{-\delta}}\right)$. Algorithm 1 using the power-law ranking selection mechanism with parameter c, the standard bitwise mutation with parameter $\chi = 1$ has expected optimisation time no more than*

$$(1+\varepsilon)L(c,\delta)n^2 + O(n\lambda\ln\lambda) \text{ on } \textsc{LeadingOnes}$$
$$(1+\varepsilon)L(c,\delta)n^r + O(n\lambda) \text{ on } \textsc{Jump}_r (r \ge 2)$$

if the population satisfies $\lambda \ge a\ln n, \lambda\ln\lambda \in o(n)$ for $\textsc{LeadingOnes}$, and $\lambda \ge ar\ln n, \lambda \in o(n)$ for $\textsc{Jump}_r$, for the some constant a that may depend on ε, δ.

Proof. We consider the canonical partition $(A_1, \cdots, A_{n+1})$ of the search space, i.e. $A_j := \{x \in \{0,1\}^n \mid f(x) = j-1\}$ and apply Theorem 1, so here $m := n+1$. We first prove the result of $\textsc{LeadingOnes}$, and then the proof for $\textsc{Jump}_r$ follows analogously.

We first show (G2), let $p_0 := \left(1 - \frac{1}{n}\right)^n$ be the probability of not flipping any bit in a mutation, we define $\gamma_0 := \left(\frac{p_0}{1+\delta}\right)^{\frac{1}{1-c}}$ for any arbitrary constant $\delta > 0$, and remark that $p_0\gamma_0^c = (1+\delta)\gamma_0$. To sample an individual in $A_{\ge j+1}$, it suffices to select a parent already in $A_{\ge j+1}$, this occurs with probability γ^c for the power-law ranking selection, then to not flip any of its bits. The overall probability is then at least $\gamma^c p_0 = (1+\delta)\gamma_0^{1-c}\gamma^c$, thus (G2) holds.

Statement (LB) is satisfied by and Lemma 2 for the sampling process described in (G2), i.e. $\gamma = i/\lambda$. Here ϕ is an increasing

function of δ, e. g. choosing $a = 3/4, b = 2$ for the lemma gives $\phi = \ln(1+\delta/2) - o(1)$. This also implies if δ is a constant, so is ϕ.

Regarding (G1), starting from a parent in A_j, the probability of creating the offspring in $A_{\geq j+1}$ by mutation is bounded from below by the probability of flipping exactly the j-th bit while keeping the rest of the bits unchanged. The probability of the event is $\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \geq \frac{1}{en} =: s$, and we define $z_j := (1+\delta)\gamma_0 s$ for all $j \leq n$. This setting satisfies (G1) because the condition assumes that there are at least γ_0 individual at level $A_{\geq j}$, thus with probability γ_0^c one is selected as parent and if the individual is already at level $A_{\geq j+1}$ it suffices to keep the individual unchanged, i. e. with probability at least p_0, otherwise if the individual is at level A_j it suffices to upgrade the individual, i. e. with probability s_j. Overall the probability is at least $\gamma_0^c \min\{p_0, s\} \geq \gamma_0^c p_0 s = (1+\delta)\gamma_0 s$ which is exactly z_j.

It remains to show (G3). Note that $\frac{1}{e} \geq p_0 \geq \left(1-\frac{1}{n_0}\right)\frac{1}{e}$ thus $\left(\frac{1}{(1+\delta)e}\right)^{\frac{1}{1-c}} \geq \gamma_0 \geq \left(\frac{1-1/n_0}{(1+\delta)e}\right)^{\frac{1}{1-c}}$ for any constant $n_0 \geq 1$ and all $n \geq n_0$. All the parameters are bounded by constants, moreover $\ln((m-1)/z_*) = O(\ln(n))$ as $\frac{m-1}{z_*} \leq \frac{en^2}{(1+\delta)\gamma_0} = O(n^2)$, thus for any constant d, there exists an a such that (G3) holds for $\lambda \geq a \ln n$.

All the conditions of Theorem 1 are now satisfied, so the upper bound on the expected optimisation time can be deduced. Remark that with $\lambda \ln \lambda \in o(n)$ the first sum which contains ln-term is dominated by the second sum which contains q_i, because we can bound the first sum with $O(n \ln \lambda)$ by ignoring the λr_i-part in the denominator of the fraction inside the ln-term. Lemma 14 implies $1/q_i \leq 1 + \frac{1}{\lambda z_i}$, thus the second sum is bounded from above by

$$\sum_{i=1}^{n} \frac{1}{q_i} \leq n + \frac{n}{\lambda z_*} \leq n + \frac{en^2}{\lambda(1+\delta)\gamma_0}$$
$$\leq n + \frac{en^2}{\lambda(1+\delta)\left(\frac{1-1/n_0}{(1+\delta)e}\right)^{\frac{1}{1-c}}} = n + \frac{e^{\frac{2-c}{1-c}}(1+\delta)^{\frac{c}{1-c}} n^2}{\lambda(1-1/n_0)^{\frac{1}{1-c}}}.$$

Altogether the expected optimisation time is no more than

$$\frac{e^{\frac{2-c}{1-c}}(1+\delta)^{\frac{c}{1-c}} n^2}{(1-d)(1-1/n_0)^{\frac{1}{1-c}}(1-e^{-\delta})} + O(n\lambda \ln \lambda),$$

and the result follows by noting that the factor $\frac{1}{(1-d)(1-1/n_0)^{\frac{1}{1-c}}}$ can be made less than $1+\varepsilon$ for any $\varepsilon > 0$ by choosing n_0.

To prove the result for $\textsc{Jump}_r$, we use the same setting of the analysis, e. g. $\gamma_0 := \left(\frac{p_0}{1+\delta}\right)^{\frac{1}{1-c}}$. However, we define $z_j := (1+\delta)\gamma_0 s_j$ where $s_j := \frac{n-j+1}{en}$ for the first $n-1$ levels, because this is a lower bound on the probability $\binom{n-j+1}{1}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$ of improving a solution from these levels, and $s_n := 1/en^r$ which is a lower bound on the probability $\left(\frac{1}{n}\right)^r\left(1-\frac{1}{n}\right)^{n-r}$ of jumping from a local optimum in A_n to the global optimum.

The conditions of Theorem 1 hold for the same reason of the setting of the algorithm (but here we have $\lambda \geq ar \ln n$), as in the argument for LeadingOnes and it remains to estimate the upper bound on the expected optimisation time. For the second sum, again using Lemma 14 we get

$$\sum_{i=1}^{n} \frac{1}{q_i} \leq n + \frac{1}{\lambda(1+\delta)\gamma_0}\left(en^r + \sum_{i=1}^{n-1}\frac{en}{\lambda n - i + 1}\right)$$
$$= O\left(n + \frac{n\ln n}{\lambda}\right) + \frac{en^r}{\lambda(1+\delta)\gamma_0}$$
$$= O\left(n + \frac{n\ln\lambda}{\lambda}\right) + \frac{e^{\frac{2-c}{1-c}}(1+\delta)^{\frac{c}{1-c}} n^r}{\lambda(1-1/n_0)^{\frac{1}{1-c}}}.$$

We remark that $r_i = \left(\frac{z_i}{(1+\delta)\gamma_0^{1-c}}\right)^{\frac{1}{c}} = \gamma_0 s_i^{1/c}$, thus for the first sum, using Stirling's formula gives

$$\sum_{i=1}^{n} \ln\left(\frac{1+\lambda}{1+\lambda r_i}\right) = \sum_{i=1}^{n} \ln\left(\frac{1/\lambda+1}{1/\lambda+r_i}\right) \leq \ln\left(\prod_{i=1}^{n}\frac{2}{r_i}\right)$$
$$= \frac{1}{c}\left(\ln\left(\prod_{i=1}^{n}\frac{en}{n-i+1}\right) + \ln\left(en^r\right)\right) + n\ln\left(\frac{2}{\gamma_0}\right)$$
$$= \frac{1}{c}\ln\left(\frac{e^n n^n}{n!}\right) + O(n + r\ln n)$$
$$< \frac{1}{c}\ln\left(\frac{e^n n^n}{\sqrt{2\pi n}(n^n/e^n)e^{\frac{1}{12n+1}}}\right) + O(n+\lambda) = O(n)$$

The result follows by combining these two sums with their respective factors, and we note that the dominating term is n^r of the second sum and that factors related to the free parameters d and n_0 are subsumed into $(1+\varepsilon)$. □

From the proof we notice that the parameter δ only impacts logarithmic population sizes in a critical way, therefore if $\lambda \in \omega(\ln n) \cap o(n)$ the bound holds universally with $\delta > 0$. Particularly, when c is reduced, the leading constant term $L(c,\delta)$ approaches e^2. Our approach also works for OneMax [15], however the leading constant has a long expression thus we omitted that analysis.

We now show that on $\textsc{Jump}_r$, it can be beneficial to have a mutation parameter χ non-constant and greater than 1.

Theorem 4. *There exists a constant $c_\lambda > 0$ such that the expected runtime of Algorithm 1 using power-law ranking selection with constant parameter c, mutation rate $\chi < n^{(1-\varepsilon)/2}$, $\chi = \Omega(1)$ for any constant $\varepsilon \in (0,1)$, and population size λ where $c_\lambda e^{\frac{2\chi}{1-c}}\ln(n) \leq \lambda$, on $\textsc{Jump}_r$ with $r \geq 2$ is $O\left(e^{\chi(1+1/(1-c))}(n/\chi)^r + \lambda(\ln\lambda + n\ln(ne^\chi))\right)$.*

Proof. First note that the lower bound on λ implies

$$\chi \leq \ln\left(\frac{\lambda}{c_\lambda \ln n}\right)\frac{1-c}{2} = o(\ln\lambda). \tag{1}$$

We apply Theorem 1 with the same level-definition as in the proof of Theorem 3, and use the same parameter $\gamma_0 := \left(\frac{p_0}{1+\delta}\right)^{\frac{1}{1-c}}$, where $p_0 := \left(1-\frac{\chi}{n}\right)^n > (1-\chi/n)^n \geq e^{-\chi}(1-n^{-\varepsilon})$ is the probability of not mutating any bits (the last inequality follows by Lemma 13 and $\chi < n^{(1/2)(1-\varepsilon)}$), and δ is an arbitrary positive constant. From the above, it also holds $\gamma_0 = \Omega(e^{-\chi/(1-c)})$. Eq. (1) implies

$$\gamma_0 < e^{-\frac{\chi}{1-c}} = e^{-o(\ln\lambda)} = \omega(1/\lambda),$$

hence γ_0 satisfies the assumptions of Lemma 2.

We first prove (G2) and statement (LB). To produce an individual in the current level, it suffices to select one in the current level, and not mutate any of its bits. The probability of this event is $\gamma^c p_0$, thus

$$\frac{\Pr_{y\sim D(P)}\left(y \in A_{\geq j+1}\right)}{\gamma_0^{1-c}\gamma^c} > \frac{p_0}{\gamma_0^{1-c}} = 1+\delta,$$

i.e., condition (G2) is satisfied. Then, similarly to the proof of Theorem 3, statement (LB) holds by Lemma 2 with the same choices for a and b, and here ϕ is a constant as far as δ is.

For any j, $q_j := 1-(1-z_j)^\lambda \geq 1-e^{-\lambda z_j}$, $z_j := (1+\delta)\gamma_0 s_j$ and

$$s_j \geq \left(\frac{\chi}{n}\right)\left(1-\frac{\chi}{n}\right)^{n-1} > \frac{\chi}{ne^\chi}(1-n^{-\varepsilon}),$$

is a lower bound on the probability of mutating a search point in any level $j \leq n-1$ into a higher level. We then have by Lemma 14 $\sum_{i=1}^{n-1}\frac{1}{q_i} \leq n-1+\frac{n-1}{\lambda z_*} = O\left(n+\frac{n^2e^\chi}{\lambda\chi\gamma_0}\right)$.

For the final level, it is necessary to flip simultaneously r 0-bits, which occurs with probability at least

$$s_n \geq \left(\frac{\chi}{n}\right)^r\left(1-\frac{\chi}{n}\right)^{n-r} > \left(\frac{\chi}{n}\right)^r e^{-\chi}(1-n^{-\varepsilon}).$$

This gives $\frac{1}{q_n} \leq 1+\frac{1}{\lambda z_n} = O\left(\frac{n^r e^\chi}{\lambda\chi^r\gamma_0}\right) = O\left(\frac{n^r e^{\chi(1+1/(1-c))}}{\lambda\chi^r}\right)$

We remark that for $i < n$,

$$r_i = \left(\frac{z_i}{(1+\delta)\gamma_0^{1-c}}\right)^{\frac{1}{c}} = \gamma_0 s_i^{1/c} = \Omega\left(\left(\frac{1}{e^\chi}\right)^{1/(1-c)}\left(\frac{\chi}{ne^\chi}\right)^{1/c}\right)$$

thus the sum $\sum_{i=1}^n \ln\left(\frac{1+\lambda}{1+\lambda r_i}\right)$ is no more than

$$\ln(1+\lambda)+\sum_{i=1}^{n-1}\ln\left(1+\frac{1}{r_i}\right) = O(\ln\lambda + n\ln(ne^\chi)).$$

For condition (G3), note that $\gamma_0^{-2} = O(\exp(\frac{2\chi}{1-c}))$ and that there exists a constant c_λ such that

$$\begin{aligned}
&\frac{1}{2\gamma_0^2\delta^2}\ln\left(\left(\frac{(1+\delta)\gamma_0^{1-c}}{\phi c}+2\right)\frac{m-1}{d(1-e^{-\delta})z_*}+\frac{1}{d}\right)\\
&\leq c_\lambda e^{\frac{2\chi}{1-c}}\ln(n/z_*) < c_\lambda e^{\frac{2\chi}{1-c}}\ln\left(\frac{n(n-\chi)^r e^{\chi(1+1/(1-c))}}{\chi^r}\right)\\
&< c_\lambda e^{\frac{2\chi}{1-c}}((r+1)\ln(n)+\chi) \leq \lambda.
\end{aligned}$$

All conditions of Theorem 1 are satisfied, and the theorem follows by combining these sums with their respective factors, taking into account that d, c, δ are constants. □

4 POWER-LAW SELECTION IS EFFICIENT IN ESCAPING SPARSE LOCAL OPTIMA

The ability of non-elitist populations to escape local optima was studied in [7, 11]. Particularly, [7] characterises the search space by the densities of fitness valleys and of local optima, thus introduced the function class $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$, i. e. Definition 8. Note that the larger the sparsity parameter ε is the denser the local optima are allowed (see Definition 6), thus the harder it is to escape them. Both elitist black-box EAs and non-elitist EAs with (μ,λ)-selection were shown to be inefficient (in the sense of exponential versus polynomial runtime) in escaping local optima even when ε is a small constant. This is in contrast to the efficiency of tournament and linear-ranking selections. We will show that the power-law ranking selection can also escape sparse local optima efficiently.

Definition 5. *For $\alpha \in [0,1]$, a subset $C \subseteq \{0,1\}^n$ is called α-dense if $\forall x \in C$, $|S_1(x)\cap C| \geq \alpha n$.*

Definition 6. *For $\varepsilon \in [0,1]$, a subset $B \subseteq \{0,1\}^n$ is ε-sparse if*

(SP1) $\forall x \in B, \forall r \in [n-1]$, $|S_r(x)\cap B| \leq \varepsilon\cdot\binom{n}{r}$, *and*

(SP2) $\forall x \in \{0,1\}^n \setminus B, \forall r \in [n-1]$, $|S_r(x)\cap B| = O\left(\frac{1}{n}\binom{n}{r}\right)$.

Definition 7. *Given a function $f : \{0,1\}^n \to \mathbb{R}$ and a partition $(A_1,\dots,A_m)$ of $\{0,1\}^n$, a pair (A_i,A_j) is called f-deceptive if $1 \leq i < j \leq m$ and there are elements $x \in A_i$, $y \in A_j$ such that $f(x) \geq f(y)$.*

Definition 8. *An objective function $f : \{0,1\}^n \to \mathbb{R}$ belongs to the problem class $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$ if there exists a partition of $\{0,1\}^n$ into $m \in \text{poly}(n)$, subsets $(A_1,\dots,A_m)$ such that*

- $A_m = \{x \in \{0,1\}^n \mid \forall y \in \{0,1\}^n, f(x) \geq f(y)\}$
- $\forall j \in [m-1], \forall x \in A_j, \exists y \in A_{\geq j+1}$ *s.t.* $\mathrm{H}(x,y) = O(1)$,

and if $(A_{i_1},A_{j_i}),\dots,(A_{i_u},A_{j_u})$ are f-deceptive pairs then

- $\cup_{v=1}^u A_{i_v}$ *is ε-sparse, and*
- $A_{\geq j_v}$ *is α-dense for all $v \in [u]$.*

The following theorem provides the sufficient conditions for which non-elitist EAs with bitwise mutation have expected polynomial runtime on $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$. It requires the non-linearity of the cumulative probability function of the selection.

Theorem 9 (Theorem 12 in [7]). *If there exist constants $\varepsilon,\psi_0,\gamma_0,\delta$, $\alpha \in (0,1)$ such that Algorithm 1 with the bitwise mutation operator with rate χ, and selection mechanism with β, and pop. size λ s.t.*

(SM0) $\beta(0,\gamma) \leq \frac{\gamma}{\frac{\varepsilon}{1-\varepsilon}+\left(1-\frac{\chi}{n}\right)^n}$ *for all $\gamma \in [\psi_0,1]$,*

(SM2a) $\beta(0,\gamma) \geq \frac{\gamma(1+\delta)}{\left(1-\frac{\chi}{n}\right)^n}$ *for all $\gamma \in (0,\gamma_0]$,*

(SM2b) $\beta(\psi,\psi+\gamma) \geq \frac{\gamma(1+\delta)}{\left(1-\frac{\chi}{n}\right)^n(1+\alpha\chi)}$ *for all $\gamma \in (0,\gamma_0]$, $\psi \in [0,\psi_0]$,*

(SM3) $C\ln(n) \leq \lambda \in \text{poly}(n)$ *for a sufficiently large constant C,*

then it has expected polynomial runtime on $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$.

We have the following result for our selection mechanism. Unlike the results in [7], where the characteristic parameters of the search space α,ε have to be specific, here we provide a range of choices where the new selection can yield polynomial optimisation time. Note also that the conditions of Theorem 9 can be formulated as non-linear constraint satisfaction problem, thus this allows the production of results similar to ours for other selections by mean of numerical computation. Using this we will make the comparison with our mechanism in the end of the section.

Theorem 10. *For any constants $\alpha,\varepsilon,\sigma,c$ in $(0,1)$ and $\chi \geq 1$ such that the following equation holds: $\varepsilon = \sigma\left(c(1+\alpha\chi)-1\right)e^{-\chi}$ then Algorithm 1 using the power-law ranking selection mechanism with parameter c, bitwise mutation with parameter χ and population size $\text{poly}(n) \ni \lambda \geq d\ln n$ for some sufficiently large constant $d > 0$ optimises $\textsc{SparseLocalOpt}_{\alpha,\frac{\varepsilon}{1+\varepsilon}}$ in expected polynomial time.*

Proof. The selection mechanism satisfies $\beta(\psi,\psi+\gamma) = (\psi+\gamma)^c - \psi^c$, and we define: $h(\psi,\gamma) := \frac{\beta(\psi,\psi+\gamma)}{\gamma} = \frac{(\psi+\gamma)^c-\psi^c}{\gamma}$, and it follows from Lemma 16 that $h(\psi,\gamma)$ is decreasing in both directions

of ψ and γ, that is, given a fixed pair (ψ, γ), for all $\psi' \geq \psi$ and all $\gamma' \geq \gamma$, it holds that $h(\gamma', \psi') \leq h(\gamma, \psi)$.

The parameters of the analysis are set as $\psi_0 := (\varepsilon + e^{-\chi})^{\frac{1}{1-c}}$, $\gamma_0 := \left(\frac{\delta e^{-\chi}}{1+2\delta}\right)^{\frac{1}{1-c}}$, $\delta := \min\left\{\sigma, \frac{1-\sigma}{\frac{3}{c(1+\alpha\chi)-1}+2\sigma}\right\}$, and let $p_0 := \left(1 - \frac{\chi}{n}\right)^n$.

For $n \geq n_0$ where $n_0 := \frac{\chi}{1-\left(\frac{1+\delta}{1+2\delta}\right)^{1/\chi}}$ it holds that

$$e^{-\chi} > p_0 = \left(1 - \frac{\chi}{n}\right)^{\chi + (n/\chi - 1)\chi} \geq \left(1 - \frac{\chi}{n_0}\right)^{\chi} e^{-\chi} = \frac{1+\delta}{1+2\delta} \cdot e^{-\chi},$$

and the last inequality uses $(1 - 1/r)^{r-1} \geq 1/e$ with r being n/χ.

Note that $\alpha, \varepsilon, \sigma$ and χ are constant, so are the parameters and particularly it holds that δ, ψ_0 and γ_0 are in $(0, 1)$. The reasons for δ and γ_0 are obvious. For ψ_0, we remark that $e^{-\chi} \leq 1/e$ since $\chi \geq 1$. Furthermore, the equation in the statement implies $\varepsilon < \frac{\chi}{e^{\chi}} =: f(\chi)$, and $f(\chi)$ reaches its maximum at $\chi = 1$, so $\varepsilon < f(1) = 1/e$. Combining these gives $\varepsilon + e^{-\chi} < 2/e < 1$, thus $\psi_0 < 1$.

We now verify the conditions of Theorem 9. The first condition (SM0) is satisfied because for all $\gamma \geq \psi_0$:

$$h(0, \gamma) \leq h(0, \psi_0) = \frac{1}{\psi_0^{1-c}} = \frac{1}{\varepsilon + e^{-\chi}} < \frac{1}{\frac{\frac{\varepsilon}{1+\varepsilon}}{1-\frac{\varepsilon}{1+\varepsilon}} + p_0}.$$

We also have, for all $\gamma \leq \gamma_0$:

$$h(0, \gamma) \geq h(0, \gamma_0) = \frac{1}{\gamma_0^{1-c}} = \frac{1+2\delta}{\delta e^{-\chi}} = \frac{1+\delta}{\frac{\delta(1+\delta)e^{-\chi}}{1+2\delta}} > \frac{1+\delta}{p_0},$$

therefore (SM2a) is satisfied.

To show (SM2b), we use Lemmas 16, 15, 17 and the equation in the statement because then they imply,

$$\begin{aligned} h(\psi, \gamma) \geq h(\psi_0, \gamma_0) &> \frac{c}{(\psi_0 + \gamma_0)^{1-c}} > \frac{c}{\psi_0^{1-c} + \gamma_0^{1-c}} \\ &= \frac{1}{\frac{1}{c}\left(\sigma c(1 + \alpha\chi - 1/c)e^{-\chi} + e^{-\chi} + \frac{\delta e^{-\chi}}{1+2\delta}\right)} \\ &= \frac{1+\delta}{\frac{(1+\delta)e^{-\chi}}{1+2\delta}\left(\sigma(1+2\delta)(1+\alpha\chi-1/c) + \frac{1+2\delta}{c} + \frac{\delta}{c}\right)} \\ &\geq \frac{1+\delta}{p_0\left(\sigma(1+2\delta)(1+\alpha\chi-1/c) + \frac{1+3\delta}{c}\right)}. \end{aligned}$$

We notice that the choice of δ implies $\delta \leq \frac{1-\sigma}{\frac{3}{c(1+\alpha\chi)-1}+2\sigma}$. This means $1 - \sigma \geq \frac{3\delta}{c(1+\alpha\chi)-1} + 2\sigma\delta$, or equivalently $\sigma(1+2\delta) \leq 1 - \frac{3\delta}{c(1+\alpha\chi)-1}$ and therefore

$$\begin{aligned} \sigma(1+2\delta)(1+\alpha\chi-1/c) &\leq \left(1 - \frac{3\delta}{c(1+\alpha\chi)-1}\right)(1+\alpha\chi-1/c) \\ &= 1 + \alpha\chi - 1/c - 3\delta/c. \end{aligned}$$

Resuming our previous calculation gives $h(\psi, \gamma) > \frac{1+\delta}{p_0(1+\alpha\chi)}$,thus (SM2b) is satisfied. We have (SM3) by the setting of the population size λ. Since all the conditions are satisfied, the result follows from the statement of Theorem 9. □

Theorem 10 gives a sufficient condition that characterises the subclass of $\textsc{SparseLocalOpt}_{\alpha, \frac{\varepsilon}{1+\varepsilon}}$ in which the power-law ranking selection can be efficient. The condition is essentially:

$$\varepsilon < \frac{c(1+\alpha\chi) - 1}{e^{\chi}}. \tag{2}$$

The parameter σ was introduced to turn this inequality into an equation so that the mutation parameter χ can be quantified. It turns out, (2) is also the necessary condition to apply Theorem 9 for our selection mechanism and this is due to the following result.

Theorem 11. *If inequality* (2) *does not hold for* α, ε, c *in* $(0, 1)$ *and* $\chi > 0$ *then the conditions (SM0) and (SM2b) of Theorem 9 cannot be satisfied simultaneously for Algorithm 1 using the power-law ranking selection mechanism with parameter* c *and bitwise mutation with parameter* χ*, and running on* $\textsc{SparseLocalOpt}_{\alpha, \frac{\varepsilon}{1+\varepsilon}}$.

Proof. To prove the result, it suffices to show that (2) is an implication of the two conditions (SM0) and (SM2b). We use the same notation as the one in the proof of Theorem 10.

Suppose we have found ψ_0, γ_0 and δ such that the two conditions are both satisfied. Specifically we have $h(0, \psi_0) \leq 1/(\varepsilon + p_0)$ by (SM0) and $h(\psi_0, \gamma_0) \geq (1+\delta)/(p_0(1+\alpha\chi))$ by (SM2b). Furthermore, we notice that $h(0, \psi_0) = \frac{1}{\psi_0^{1-c}}$, and $h(\psi_0, \gamma_0) = \frac{(\psi_0+\gamma_0)^c - \psi_0^c}{\gamma_0} < \frac{c}{\psi_0^{1-c}} = ch(0, \psi_0)$ where the inequality is due to Lemma 15. Therefore,

$$\frac{c}{\varepsilon + p_0} \geq ch(0, \psi_0) > h(\psi_0, \gamma_0) > \frac{1}{p_0(1+\alpha\chi)},$$

or equivalently $p_0 c(1 + \alpha\chi - 1/c) > \varepsilon$. Combining this with $p_0 \leq e^{-\chi}$ gives (2), which is now an implication of (SM0) and (SM2b). □

In view of Theorems 10 and 11, given specific values of α and c, we can find the maximal value for the sparsity parameter $\varepsilon/(1+\varepsilon)$ to identify the widest class of landscapes that satisfy the conditions of these theorems. First of all, note that $\varepsilon/(1+\varepsilon)$ is an increasing function so it suffices to maximize the right-hand side of (2) w.r.t. $\chi \in (0, \infty)$. By looking at the differentiation by χ of this right-hand side, it is easy to see that the maximum is obtained at

$$\chi = \frac{(\alpha - 1)\, c + 1}{\alpha c}, \tag{3}$$

which will gives an explicit expression for the maximal $\varepsilon/(1+\varepsilon)$.

In order to compare the ranges of efficiency of the power-law selection to those of the tournament selection, in Figure 1 we provide the plots of the maximal values of ε that satisfy conditions of Theorem 9 for the power-law selection and for the tournament selection with different values of parameter c and tournament size k. The values for tournament selection were found using the non-convex optimisation solver BARON, ran in global optimisation mode. Figure 1 indicates that the tournament selection with $k = 2$ or 4 can handle larger ε given the same density α, compared to the power-law selection with $c = 0.5$ or 0.3. However with $c = 0.9$, which is closer to the uniform selection, the tournament selection yields to the power-law selection for most of the values of α. Note that to overcome a local optimum, the EA needs sufficiently "weak" mutation pressure. The power law selection has an advantage that the value of c can be chosen arbitrary close to 1 (if small α requires that), while the tournament size can not be less than 2.

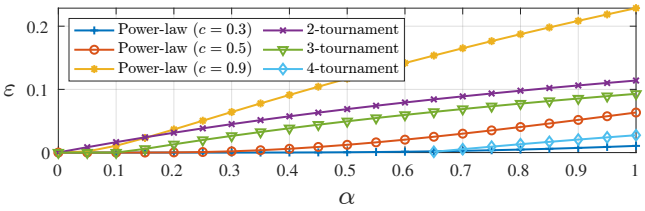

Figure 1: Maximal values of ε within conditions of Theorem 9 for the power-law and tournament selections.

5 ERROR THRESHOLDS

Error thresholds are essential when analysing the efficiency of non-elitist evolutionary algorithms. In particular, non-elitist evolutionary algorithms without crossover with reproductive rate[1] α_0, and mutation rate $\chi/n \geq (1+\delta)\ln(\alpha_0)/n$ for any constant $\delta > 0$ have exponential runtime with overwhelmingly high probability on any fitness function having no more than a polynomial number of optima [20]. At the same time, the runtime of non-elitist EAs often turns from exponential to polynomial when the mutation rate is decreased below the error threshold [5]. In fact, it has been observed that non-elitist EAs tend to perform well on hard problems when the mutation rate is chosen slightly below the error threshold [7].

The probability that power-law selection chooses any individual is upper bounded by $(1/\lambda)^c$. Hence, the reproductive rate of this selection mechanism is upper bounded by λ^{1-c}. However, the negative drift theorem for populations as stated in [20] assumes a population selection-variation algorithm (PSVA) with reproductive rate bounded above by some constant α_0. But by carefully inspecting the proofs, Lemma 2 and Theorem 1 in that paper still hold if for the constant δ in that theorem, the constraint on α_0 is relaxed to $\alpha_0 \leq \frac{\delta}{(2+\delta)(1+\delta)} \cdot d(n)$, where $d(n) = b(n) - a(n)$ is the "distance" in the drift theorem. From these considerations, we derive Theorem 12, which is a special version of Theorem 4 in [20] tailored to EAs using power-law ranking selection. The theorem implies that the error threshold is exceeded for power-law ranking selection when the mutation rate satisfies for an arbitrary constant $\delta \in (0,1)$

$$(1+\delta)(1-c)\ln\lambda \;\leq\; \chi \;\leq\; n/b(n). \tag{4}$$

Here, c refers to the parameter in the power-law distribution. The upper bound $n/b(n)$ where $b(n) = \omega(\ln n)$ is a technical condition which we conjecture is not needed. Interestingly, the error threshold increases logarithmically with the population size λ.

Theorem 12. *Consider Algorithm 1 with power-law ranking selection with parameter c as p_{sel} and bitwise mutation as p_{mut}, and let $a(n)$ and $b(n)$ be positive integers such that $b(n) \leq n/\chi$ and $d(n) := b(n) - a(n) = \omega(\ln n)$. For an $x^* \in \{0,1\}^n$, define $T(n) := \min\{t \mid \min_{j\in[\lambda]} H(P_t(j), x^*) \leq a(n)\}$. If there exist constants $\delta, \delta' \in (0,1)$ such that*

1) $\lambda < \left(\frac{\delta' d(n)}{(2+\delta')(1+\delta')}\right)^{1/(1-c)}$
2) $\psi := (1-c)\ln\lambda/\chi + \delta < 1$
3) $\frac{b(n)}{n} < \min\{1/5, (1/2)(1-\sqrt{\psi(2-\psi)}\}$

then for some constant $C > 0$, $\Pr\left(T(n) \leq e^{Cd(n)}\right) = e^{-\Omega(d(n))}$.

[1]The proof of Theorem 12 provides a formal definition of the reproductive rate.

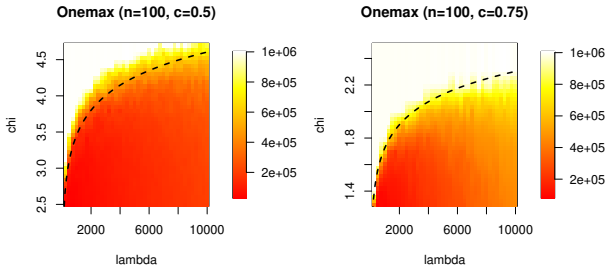

Figure 2: Runtime (function evaluations) on OneMax ($n = 100$) as a function of population size λ and mutation rate χ.

Proof sketch. The theorem is a corollary to Theorem 4 in [20]. We therefore only provide a proof sketch.

Define the reproductive rate of individual i at generation t as $R_t(i) := \sum_{j\in[\lambda]}[I_t(j) = i]$, i.e., the number of times individual i is selected from the population P_t. By the definition of power-law ranking selection, no individual has selection probability higher than $(1/\lambda)^c$. It follows by condition 1 that for all $t \in \mathbb{R}$ and $i \in [\lambda]$,

$$E\left[R_t(i)\right] \leq \lambda^{1-c} =: \alpha_0 < \frac{\delta' d(n)}{(2+\delta')(1+\delta')}. \tag{5}$$

So this α_0 is valid for the generalised variant of Theorem 1 in [20].

Now condition 1 of Theorem 4 in [20] is therefore satisfied. Also, condition 2 of Theorem 4 in [20] follows from condition 2 in this theorem. Finally, conditions 3 are identical in the two theorems. So all conditions of Theorem 4 in [20] hold, and the result follows. □

To illustrate the striking impact of error thresholds on the runtime, Figure 2 shows the runtime of the algorithm on OneMax ($n = 100$) for different mutation rates. For power law parameter $c = 1/2$, the left plot shows for mutation rates $\chi \in \{2.5, 2.6, \ldots, 4.7\}$. For the power-law parameter $c = 3/4$, the right plot shows the outcome for mutation rates $\chi \in \{1.3, 1.4, \ldots, 2.5\}$. Both plots show results for population sizes $\lambda \in \{250, 500, \ldots, 10000\}$. The "heatmaps" are produced by computing the median of 100 independent runs for each value of χ and λ. Runs were capped at 10^6 function evaluations. The dashed lines indicate the error threshold predicted by Eq. (4).

6 EXPERIMENTS

In this section, we empirically analyse the performance of the non-elitist EA with power-law ranking selection on Funnel function and random NK-Landscape functions, and compare it to other EAs.

The Funnel function [6], a multi-modal artificial function, belongs to the SparseLocalOpt problem class. The runtime analyses show that elitist EAs and the (μ, λ) EA require exponential time to discover the optimum with overwhelmingly high probability, while the 3-tournament EA with a mutation rate close to the error threshold, i.e., $\chi/n = 1.09812/n$, can achieve the optimum in $O\left(n^2 \log(n)\right)$ [6]. (The leading constant of the runtime is unknown.)

Figure 3 shows a box plot of runtimes of 100 independent runs of non-elitist EAs using population size $\lambda = 10000\ln(n)$ with tournament and power-law ranking selections on the Funnel function with $u = 0.5n$, $v = 0.6n$, $w = 0.7n$. We use mutation rates close to error thresholds for tournament selection, i.e., $\chi/n = 0.693/n$ and $1.09812/n$ for tournament sizes $k = 2$ and 3, respectively [20]. For

power-law ranking selection, we try selection parameters $c = 0.3$ and 0.8, and using mutation rates $\chi/n = 4.5/n$ and $1.2/n$, respectively. Note that the runtimes are divided by $n^2 \ln(n)$, and the y-axis uses a logarithmic scale. The results imply that the power-law ranking EAs can outperform tournament EAs on the leading constant of runtime on Funnel for the problem size from 100 to 190.

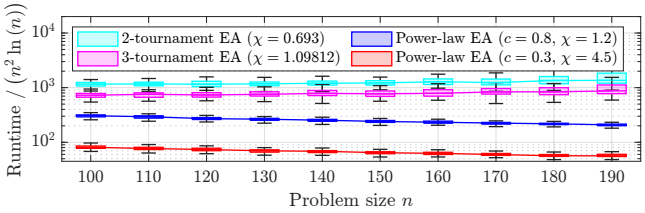

Figure 3: Runtime on Funnel. The y-axis is scaled by $n^2 \ln(n)$.

The NK-Landscape problem [19] is to maximise the function $\sum_{i=1}^{n} f_i(x_i, \ldots, x_{(i+k-1) \mod n})$, where $n, k \in \mathbb{N}$ satisfying $k \leq n$, $i \in [n]$ and $f_i : \{0,1\}^k \to \mathbb{R}$ is a set of sub-functions. Usually functions f_i are given as lookup tables with 2^{k+1} values from $(0, 1)$.

We optimise 100 random NK-Landscape instances for $n = 100$ and $k \in \{5, 10, 15, 20, 25\}$. We first randomly sample 100 instances for each k. Then we run the algorithms on each instance, and record the highest fitness values found in the fitness evaluation budget 10^8. This performance measure was chosen because estimation of the runtime would require practically unacceptable CPU time. For power-law ranking selection, we set the selection parameter $c = 0.8$, the mutation rate $\chi/n = 1.2/n$. We run the $(1 + 1)$ EA with the standard mutation rate $1/n$, the (μ, λ) EA with $\lambda/\mu = 8$ and mutation rate $2.07/n$ [20], the UMDA [26] with $\lambda/\mu = 8$, the 3-tournament EA with mutation rate $1.09812/n$ (this theoretically chosen mutation rate has shown good results in experiment [6, 7]). We set population size $\lambda = 20000$ for all population-based EAs.

Figure 4 shows that the highest fitness values achieved in 10^8 evaluations by the power-law ranking EA are higher than those of $(1 + 1)$ EA and UMDA. The Wilcoxon rank-sum test indicates that the difference is significant with level 0.05 for $k \in \{10, 15, 20, 25\}$. The fitness values achieved by the power-law ranking EA are lower than those of other non-elitist EAs (significance level 0.05).

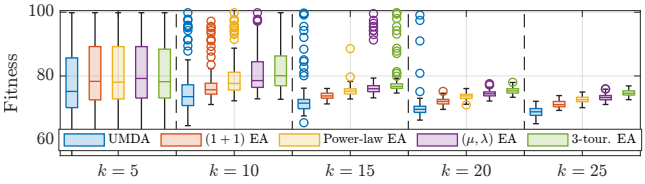

Figure 4: The best found fitness on NK-Landscape instances.

7 CONCLUSION

Recent theoretical work shows that non-elitist population-based EAs using tournament selection or linear ranking selection can optimise multi-modal problems in the $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$ class in expected polynomial time where elitist EAs need exponential time [7]. However, the precise runtime for these non-elitist EAs are not available for most problems (see [11] for an exception). It is possible that due to the large population, and weak selective pressure often assumed, the leading constants could be considerably larger than those for single-individual elitist EAs. On the other hand, (μ, λ)-selection which allows a higher selective pressure lacks the non-linear properties suitable for $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$ [7].

This paper introduces power-law ranking selection to non-elitist EAs. This selection mechanism places an extremely strong selective pressure on the fittest individuals, while retaining the non-linear properties required to optimise $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$ efficiently. Notably, with population size λ, the mutation rate can be chosen $\chi \leq c' \ln(\lambda)$ when c' is a sufficiently small constant. For the traditional benchmark problems LeadingOnes and Jump, the non-elitist EA with power-law ranking selection is fast, with nearly the same performance as hill-climber algorithms that do not use populations. This is significant, because previous non-tight analyses of non-elitist EAs left the possibility open that in practice, using large populations would slow down the algorithms intolerably.

With the strong selective pressure, the algorithm tolerates much higher mutation rates, leading to better runtime on $\textsc{Jump}_r$. Experimental results show that the EA with power-law ranking selection is significantly faster on the multi-modal Funnel problem than non-elitist EAs with tournament selection. Furthermore, power-law selection outperforms UMDA and the (1+1) EA in our experiments on the NK-landscape problem, but yields to the (μ, λ)-selection and 3-tournament selection.

As a technical contribution, we introduced a new level-based theorem that takes into account the strong selective pressure on the fittest individuals. We also derived new sufficient conditions for efficiency on the $\textsc{SparseLocalOpt}_{\alpha,\varepsilon}$ problem class.

Future work should provide more precise runtime bounds for power-law selection on Jump and other functions, and characterise for what fitness landscapes power-law selection outperforms traditional selection mechanisms.

ACKNOWLEDGMENTS

Lehre was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1). Eremeev was funded within the state task of the IM SB RAS, project FWNF-2022-0020.

A VARIOUS RESULTS

Lemma 13 (Lemma 11 in [23]). *For any $\delta \in (0, 1)$ and $\chi > 0$, if $n \geq (\chi + \delta)(\chi/\delta)$, then $\left(1 - \frac{\chi}{n}\right)^n \geq (1 - \delta)e^{-\chi}$.*

Lemma 14 (Lemma 31 in [8]). *For $n \in \mathbb{N}$ and $x \geq 0$, we have $1 - (1 - x)^n \geq 1 - e^{-xn} \geq \frac{xn}{1+xn}$.*

Lemma 15. *For any $c \in (0, 1)$ and any $a, b > 0$ holds*

$$\frac{c}{a^{1-c}} > \frac{(a+b)^c - a^c}{b} > \frac{c}{(a+b)^{1-c}}.$$

Lemma 16. *If $f(x, y) := \frac{(x+y)^c - x^c}{y}$, $c \in (0, 1)$ then $\frac{\partial f}{\partial x}(x, y) < 0$ and $\frac{\partial f}{\partial y}(x, y) < 0$ for all $x, y > 0$.*

Lemma 17. *If $a, b \geq 0$, then $(a + b)^c \leq a^c + b^c$ for any $c \in (0, 1)$.*

Lemma 18 ([20]). *If Algorithm 1 in [20], satisfies conditions 3-5 in Theorem 1 in [20], and conditions 1 and 2 in that theorem for a parameter α_0 (not necessarily constant) where $\alpha_0 \leq \frac{d(n)\delta}{(2+\delta)(1+\delta)}$ then the associated mean matrix M has the Perron root $\rho(M) < \frac{1}{1+\delta/2}$.*

REFERENCES

[1] Denis Antipov, Maxim Buzdalov, and Benjamin Doerr. 2021. Lazy Parameter Tuning and Control: Choosing All Parameters Randomly from a Power-Law Distribution. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 1115–1123.

[2] James Edward Baker. 1989. *An analysis of the effects of selection in genetic algorithms.* Ph.D. Dissertation. Vanderbilt Univertsity.

[3] Hans-Georg Beyer, Hans-Paul Schwefel, and Ingo Wegener. 2002. How to analyse evolutionary algorithms. *Theoretical Computer Science* 287, 1 (2002), 101–130.

[4] Tobias Blickle and Lothar Thiele. 1996. A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation* 4, 4 (1996), 361–394.

[5] Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. 2018. Level-Based Analysis of Genetic Algorithms and Other Search Processes. *IEEE Transactions on Evolutionary Computation* 22, 5 (2018), 707–719.

[6] Duc-Cuong Dang, Anton Eremeev, and Per Kristian Lehre. 2021. Number 14. 12275–12283.

[7] Duc-Cuong Dang, Anton Eremeev, and Per Kristian Lehre. 2021. Non-Elitist Evolutionary Algorithms Excel in Fitness Landscapes with Sparse Deceptive Regions and Dense Valleys. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 1133–1141.

[8] Duc-Cuong Dang and Per Kristian Lehre. 2016. Runtime analysis of non-elitist populations: From Classical Optimisation to Partial Information. *Algorithmica* 75, 3 (2016), 428–461.

[9] Duc-Cuong Dang, Per Kristian Lehre, and Phan Trung Hai Nguyen. 2019. Level-based analysis of the univariate marginal distribution algorithm. *Algorithmica* 81, 2 (2019), 668–702.

[10] Luc Devroye. 1986. *Non-Uniform Random Variate Generation.* Springer-Verlag.

[11] Benjamin Doerr. 2022. Does comma selection help to cope with local optima? *Algorithmica* (2022), 1–35.

[12] Benjamin Doerr, Carola Doerr, and Franziska Ebel. 2015. From black-box complexity to designing new genetic algorithms. *Theoretical Computer Science* 567 (2015), 87–104.

[13] Benjamin Doerr and Timo Kötzing. 2021. Multiplicative Up-Drift. *Algorithmica* 83, 10 (2021), 3017–3058.

[14] Benjamin Doerr, Huu Phuoc Le, Régis Makhmara, and Ta Duy Nguyen. 2017. Fast Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Berlin, Germany) *(GECCO '17)*. Association for Computing Machinery, New York, NY, USA, 777–784.

[15] Stefan Droste, Thomas Jansen, and Ingo Wegener. 2002. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science* 276, 1 (2002), 51–81.

[16] David E. Goldberg. 1989. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley.

[17] David E. Goldberg and Kalyanmoy Deb. 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms *(Foundations of Genetic Algorithms)*, Vol. 1. Elsevier, 69–93.

[18] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms - The Computer Science Perspective.* Springer.

[19] Stuart A Kauffman and Edward D Weinberger. 1989. The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of theoretical biology* 141, 2 (1989), 211–245.

[20] Per Kristian Lehre. 2010. Negative Drift in Populations. In *Parallel Problem Solving from Nature, PPSN XI*, Vol. 6238. Springer Berlin Heidelberg, 244–253.

[21] Per Kristian Lehre. 2011. Fitness-Levels for Non-Elitist Populations *(GECCO '11)*. Association for Computing Machinery, New York, NY, USA, 2075–2082.

[22] Per Kristian Lehre and Phan Trung Hai Nguyen. 2018. Level-Based Analysis of the Population-Based Incremental Learning Algorithm. In *Parallel Problem Solving from Nature - PPSN XV - 15th International Conference, Coimbra, Portugal, September 8-12, 2018, Proceedings, Part II (Lecture Notes in Computer Science)*, Vol. 11102. Springer, 105–116.

[23] Per Kristian Lehre and Pietro S. Oliveto. 2018. Theoretical Analysis of Stochastic Search Algorithms. In *Handbook of Heuristics*, Rafael Martí, Pardalos Panos, and Mauricio G. C. Resende (Eds.). Springer International Publishing, Cham, 1–36.

[24] Per Kristian Lehre and Xiaoyu Qin. 2021. More Precise Runtime Analyses of Non-Elitist EAs in Uncertain Environments. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 1160–1168.

[25] Tatsuya Motoki. 2002. Calculating the Expected Loss of Diversity of Selection Schemes. *Evolutionary Computation* 10, 4 (12 2002), 397–422.

[26] Heinz Mühlenbein and Gerhard Paaß. 1996. From recombination of genes to the estimation of distributions I. Binary parameters. In *Parallel Problem Solving from Nature — PPSN IV*. Springer, 178–187.

[27] Gabriela Ochoa. 2006. Error Thresholds in Genetic Algorithms. *Evolutionary Computation* 14, 2 (06 2006), 157–182.