Island Models Meet Rumor Spreading

Benjamin Doerr
École Polytechnique
Laboratoire d'Informatique (LIX)
Palaiseau, France

Philipp Fischbeck
Hasso Plattner Institute
Potsdam, Germany

Clemens Frahnow
Hasso Plattner Institute
Potsdam, Germany

Tobias Friedrich
Hasso Plattner Institute
Potsdam, Germany

Timo Kötzing
Hasso Plattner Institute
Potsdam, Germany

Martin Schirneck
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

Island models in evolutionary computation solve problems by a careful interplay of independently running evolutionary algorithms on the island and an exchange of good solutions between the islands. In this work, we conduct rigorous run time analyses for such island models trying to simultaneously obtain good run times and low communication effort.

We improve the existing upper bounds for the communication effort (i) by improving the run time bounds via a careful analysis, (ii) by setting the balance between individual computation and communication in a more appropriate manner, and (iii) by replacing the usual communicate-with-all-neighbors approach with randomized rumor spreading, where each island contacts a randomly chosen neighbor. This epidemic communication paradigm is known to lead to very fast and robust information dissemination in many applications. Our results concern islands running simple (1+1) evolutionary algorithms, we regard d-dimensional tori and complete graphs as communication topologies, and optimize the classic test functions OneMax and LeadingOnes.

CCS CONCEPTS

•**Theory of computation** → **Evolutionary algorithms;** *Communication complexity; Distributed algorithms;* •**General and reference** → General conference proceedings;

1 INTRODUCTION

To speed up evolutionary algorithms, *island models* can be used as a means of distributing the work load over many computing nodes. Each island runs a simple evolutionary algorithm, occasionally sharing information with other nodes. One of the most common ways of sharing information is to send a copy of the best-so-far solution to other islands, a process called *migration*. Many applications of this paradigm are known to be successful [1, 2, 4].

One main choice for designing an efficient algorithm following the island model is to choose the way migration is carried out. For example, the islands can be equipped with a neighborhood structure, determining for each island, which other islands to migrate its individuals to; this is referred to as the *migration topology*. Dense migration topologies, such as the complete graph, lead to a fast spread of good solutions at the price of a high communication overhead. The impact of the migration topology on algorithm performance has been analyzed both experimentally [19] and theoretically [14]. Another choice lies in the frequency of migration. A frequent approach is to introduce a parameter τ indicating that once every τ generations all islands engage in communication with all neighbors. A high value of τ can thus save on the communication overhead, at the price of delays in the spread of new good individuals. Setting the migration interval correctly is a challenge for designing efficient island algorithms [17]. It has been noted that island models are also particularly useful for dynamic optimization problems [16] and when employing crossover [18]. An overview of practical concerns of research in the area of island models can be found in [2]; for an overview of theoretical work, see [20].

In this work, we will consider λ islands running a (1+1) EA, a standard evolutionary algorithm (EA) considered in theoretical analyses [10]. For various migrations topologies (such as d-dimensional tori and the complete graph) and migration intervals τ, we are interested in the expected time until some island evolves the optimal solution for the given fitness function, of which we consider the two standard functions OneMax and LeadingOnes. It is not surprising that in this simple setting of unimodal fitness functions, fast migration topologies, such as the complete graph, perform best in terms of the number of generations, while performing badly in terms of communication [14]. We improve the analysis especially pertaining to the *combined costs* (number of generations plus number of communications per island) in the following ways.

First, we analyze the run time of the island models carefully. We see that, for the number of generations that any of the topologies require on OneMax, the dependence on τ is not linear, but, surprisingly, logarithmic. For the complete topology we further improve the bounds on the number of generations by making a detailed analysis of different optimization phases and employing a variable drift theorem; we show this analysis to be tight by providing matching lower bounds. Second, we use the parameter τ to avoid communication overhead. By finding the right balance between individual computation of the islands and spreading the information to the neighbors, we see that the combined costs for the complete graph on OneMax are as low as $O(n \log\log n)$, using $\lambda = \ln n$ islands and a migration interval of $\tau = \ln n$. Similarly, we obtain combined costs for the complete graph on LeadingOnes

GECCO '17, Berlin, Germany

978-1-4503-4920-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071206

Table 1: Overview of Results. The optimization times of the island model on λ islands with migration interval τ on fitness functions OneMax and LeadingOnes of length-n bit strings. Shown are the run times for the push protocol on complete graphs as well as for the d-dimensional torus and the complete graph using broadcast communication. The best optimization time is the parallel run time for the optimal choice of parameters λ and τ. The best combined costs refer to the parameters that minimize the sum of the optimization time and communication costs.

Objective	Model	Optimization Time	Best Optimization Time	Best Combined Costs
OneMax	Push	$O\left(\frac{n\log n}{\lambda} + n\log\tau\right)$	$O(n)$	$O(n)$
	d-Torus	$O\left(\frac{n\log n}{\lambda} + n\log\tau\right)$	$O(n)$	$O(n)$
	Complete	$\Theta\left(\frac{n\log n}{\lambda} + \frac{n\tau}{\log\lambda}\log\left(\frac{\log\lambda}{\tau}\right)\right)$, if $\tau = o(\log\lambda)$ $\Theta\left(\frac{n\log n}{\lambda} + n\right)$, if $\tau = \Theta(\log\lambda)$ $\Theta\left(\frac{n\log n}{\lambda} + n\log\left(\frac{\tau}{\log\lambda}\right)\right)$, if $\tau = \omega(\log\lambda)$	$\Theta\left(n\frac{\log\log n}{\log n}\right)$	$\Theta(n\log\log n)$
LeadingOnes	Push	$O\left(\frac{n^2}{\lambda} + n\tau\log\left(\frac{n}{\tau}\right)\right)$	$O(n\log n)$	$O(n\log n)$
	d-Torus	$O\left(\frac{n^2}{\lambda} + n^{\frac{d+2}{d+1}}\tau^{\frac{d}{d+1}}\right)$	$O\left(n^{\frac{d+2}{d+1}}\right)$	$O\left(n^{\frac{d+2}{d+1}}\right)$
	Complete	$\Omega\left(\frac{n^2}{\lambda} + \frac{n}{\log^2 n}\tau\right)$; $O\left(\frac{n^2}{\lambda} + n\tau\right)$	$\Omega\left(\frac{n}{\log^2 n}\right)$; $O(n)$	$\Omega\left(\frac{n^{3/2}}{\log n}\right)$; $O\left(n^{3/2}\right)$

of $O(n^{3/2}) \cap \Omega(n^{3/2}/\log n)$, using $\lambda = \sqrt{n}$ islands and a migration interval of $\tau = \sqrt{n}$ for the positive bound. Finally, we question the method of broadcasting the information to all available neighbors. Instead, we propose to employ the *push protocol*, known from the area of *epidemic algorithms* or *rumor spreading*, where in each communication round each island chooses one neighbor uniformly at random to send the best individual to. It is known that for the complete topology the process requires logarithmically many communication rounds until all islands are informed [6, 8]. This is significantly faster than the ring and torus topologies considered previously (and also faster than d-dimensional tori in general), while the communication overhead is still constant per island and communication round (compared with the linear overhead of complete topologies). By proving lower bounds on the performance of the complete topology, we show that the push protocol is superior even to broadcast communication in some settings.

In Table 1 we give an overview of our results. Section 2 introduces the island models and test functions more formally. Section 3 concerns the Push Protocol; in Section 4 we give run time bounds for tori; finally, Section 5 concerns the complete topology.

2 ISLAND MODELS

In this paper we examine the maximization of pseudo-Boolean functions $f : \{0,1\}^n \to \mathbb{R}_0^+$ on bit strings $\mathbf{x} = \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_n$ of length n. We interpret the value $f(\mathbf{x})$ as the *fitness* of the individual $\mathbf{x}$. A fitness function is called *unimodal* if every non-optimal bit string has a Hamming-neighbor of higher fitness. We investigate

$$\textsc{OneMax}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i, \quad \textsc{LeadingOnes}(\mathbf{x}) = \sum_{i=1}^{n}\prod_{j=1}^{i} \mathbf{x}_j$$

as prototypes of unimodal functions with $n+1$ different values. The main difference between these two functions is the number of improving Hamming-neighbors. While every bit string $\mathbf{x}$ with $\textsc{OneMax}(\mathbf{x}) = i < n$ has $n - i$ neighbors of higher fitness, the improving neighbor w.r.t. LeadingOnes is unique.

We employ the *island model* as a common framework for distributed evolutionary computation, cf. [14, 18, 19]. Suppose an undirected graph $G = (V, E)$, the *migration topology*, on $\lambda = |V|$ vertices to be given. Every vertex, called *island*, marks an independent instance of the (1+1) Evolutionary Algorithm using standard bit mutation. Prior to the first iteration all islands are initialized uniformly at random, after that they operate in lockstep. Occasionally, governed by a *migration protocol*, the islands share copies of their currently best solutions along the edges of G. A maximum-fitness migrant replaces the solution of a receiving island if the fitness of the former is not smaller than that of the latter. Ties among incoming migrants (with maximum fitness) are broken uniformly at random. We employ migration periodically every τ rounds, the *migration interval*. The simplest migration protocol is a broadcast of the currently best solution to all neighboring islands. This leads to Algorithm 1. Here $\mathbf{x}^{(j)}$ denotes the best individual on island j.

We are mainly interested in two measures of complexity. First, we count the number of generations until an optimal individual is sampled for the first time, we call this random variable the *optimization time* and denote it with T. Second, we count the messages sent during the migration phases leading up to an optimal solution (line 10 in Algorithm 1). We adopt an amortized view on the *communication costs* as we only account for the average number of messages *per island*. Let C denote this average. Observe that even in the case of a deterministic migration protocol, C is a random variable. We refer to the sum $T + C$ as the *combined costs*. This implicitly assumes that generating and evaluating a new individual is as expensive as sending a message to one neighboring island.

Algorithm 1: Island model with migration topology $G = (V, E)$ on λ islands and migration interval τ.

```
t ← 0;
for 1 ≤ j ≤ λ in parallel do
    x^(j) ← solution drawn u.a.r. from {0,1}^n;
repeat
    t ← t + 1;
    for 1 ≤ j ≤ λ in parallel do
        y^(j) ← flip each bit of x^(j) independently w/ prob. 1/n;
        if f(y^(j)) ≥ f(x^(j)) then x^(j) ← y^(j) ;
        if t mod τ = 0 then
            Send x^(j) to all islands k with {j,k} ∈ E;
            N = {x^(i) | {i,j} ∈ E};
            M = {x^(i) ∈ N | f(x^(i)) = max_{x∈N} f(x)};
            y^(j) ← solution drawn u.a.r. from M;
            if f(y^(j)) ≥ f(x^(j)) then x^(j) ← y^(j) ;
until termination condition met;
```

However, as the theorems below quantify both measures separately all results can easily be extended to the weighted case.

In this work we derive asymptotic bounds on the expectations $E[T]$ and $E[C]$ for several migration topologies and protocols. To distinguish the two fitness functions also in notation, we let $E[T_{\mathrm{OM}}]$ and $E[C_{\mathrm{OM}}]$ stand for the respective cost measures when optimizing OneMax, and $E[T_{\mathrm{LO}}]$ and $E[C_{\mathrm{LO}}]$ for LeadingOnes. All bounds will be in terms of n, λ, and τ simultaneously. More formally, we regard $\lambda = \lambda(n)$ and $\tau = \tau(n)$ as positive, non-decreasing, integer-valued functions and characterize the univariate asymptotics of the expected costs w.r.t. n for arbitrary choices of λ and τ.

2.1 The Spreading Time

Our main tool to establish upper bounds on the expected optimization time is a fitness level argument [21]. We say that Algorithm 1 is on *fitness level i* if the maximum fitness over all islands equals i. OneMax and LeadingOnes both induce $n + 1$ fitness levels. Due to the elitist selection the level never decreases and T measures the number of rounds until the algorithm enters level n. We split T in the partial optimization times $(T_i)_{0 \le i < n}$, where T_i is the time needed to leave level i. T_i crucially depends on the number of islands whose individual has the currently best fitness i. By preferring fitter individuals, migration helps to spread good solutions so that more islands can effectively contribute to the overall progress. The ability of a topology to speed up computation through migration is quantified in the notion of the *spreading time*.

Definition 2.1. Suppose a migration topology $G = (V, E)$ and a natural number $1 \le k \le \lambda$ is given. For a vertex $v \in V$, let $S_v(k)$ be the number of communication steps needed to inform at least k islands starting from v. The *spreading time* of G is the function $S(k) = \max_v S_v(k)$.

Communication steps happen only during migration phases (i.e., once every τ rounds). The number of generations that pass until a good solution is sufficiently widespread is thus by a factor τ larger than the spreading time.

The following lemma is also implicitly given in [14, Lemma 1].

Lemma 2.2. *Let p_i be (a lower bound on) the probability that the* (1+1) EA *samples an individual of fitness larger than i from one of fitness exactly i and $1 \le \lambda_i \le \lambda$. Then,*

$$E[T_i] \le 1 + \tau S(\lambda_i) + \frac{1}{p_i \lambda_i}.$$

Proof. After $\tau S(\lambda_i)$ iterations, λ_i islands have adopted a solution of maximum fitness i via migration. The expected waiting time until one of the λ_i islands creates a solution of larger fitness results in an upper bound on $E[T_i]$. The probability of not finding a better solution in one round is at most $(1 - p_i)^{\lambda_i}$, thus the waiting time is

$$\frac{1}{1 - (1 - p_i)^{\lambda_i}} \le 1 + \frac{1}{p_i \lambda_i}. \qquad \square$$

The spreading time S is non-decreasing, so it worsens the estimate when λ_i gets larger. On the other hand, if more islands share a good solution, the probability to complete the current level increases. The extreme value of $\lambda_i = 1$ completely eliminates the influence of the spreading time (as $S(1) = 0$), but in turn also bars migration from contributing to the optimization process. The result below allows us to choose λ_i independently for every fitness level to balance out these two opposing trends. It is an immediate consequence of the linearity of expectation.

Corollary 2.3. *Let $(\lambda_i)_{0 \le i < n}$ be any sequence of integers between* 1 *and* λ*, then*

$$E[T] \le n + \sum_{i=0}^{n-1} \left(\tau S(\lambda_i) + \frac{1}{p_i \lambda_i} \right).$$

In case of a randomized migration protocol the spreading time is no longer a deterministic function but a random variable, parameterized by the target number λ_i of islands to reach. However, if we replace $S(\lambda_i)$ with its expectation the reasoning above stays valid.

3 PUSH PROTOCOL

We start the analysis with a probabilistic approach to migration, namely, the *push protocol.* The migration interval is fixed at value τ, but the transmission itself is randomized. Every island chooses a neighbor uniformly and sends its current best solution to it. Using probabilistic communication is a robust way to save on the communication costs, even in densely connected migration topologies. We prove this for the complete graph K_λ on λ vertices.

The push protocol is well-analyzed in literature. To bound the expected spreading time, we use the following adaption of a more general result by Doerr and Künnemann [8]. Let $\mathrm{ld}\, x$ denote the base-2 logarithm of x.

Lemma 3.1. [8, Lemma 3.3] *Consider the complete graph K_λ as migration topology using the push protocol. Then, there is a constant $c \ge 1$ such that $E[S(k)] \le \mathrm{ld}\, k$ for all $1 \le k \le \lambda/c$.*

Theorem 3.2. *Consider the complete graph K_λ as the migration topology using the push protocol. Then,*

(1) $E[T_{\mathrm{OM}}] = \mathrm{O}\Big(\frac{n \log n}{\lambda} + n \log \tau\Big)$;

(2) $E[C_{\mathrm{OM}}] = \mathrm{O}\Big(\frac{n \log n}{\lambda \tau} + \frac{n \log \tau}{\tau}\Big)$.

Proof. (2) can be obtained from (1) and the fact that every island sends exactly one message every τ generations.

We now prove part (1). A standard computation shows that the probability p_i (formally defined in Lemma 2.2) to find an improving Hamming-neighbor of an individual $\mathbf{x}$ with $\text{OneMax}(\mathbf{x}) = i$ is at least $(n-i)/(en)$. We want to use (the randomized version of) Corollary 2.3 in the proof and thus define a sequence $(\lambda_i)_{0\le i<n}$, its members represent the minimum target number of islands to which we want to distribute the best solution. Let constant c be as in Lemma 3.1 and let

$$\lambda_i = \begin{cases} 1, & \text{if } i < \left(1-\frac{1}{\tau}\right)n; \\ \frac{n}{\tau(n-i)}, & \text{if } \left(1-\frac{1}{\tau}\right)n \le i < \left(1-\frac{c}{\lambda\tau}\right)n; \\ \frac{\lambda}{c}, & \text{otherwise.} \end{cases}$$

Let L_1 denote the lower limit and L_2 the upper limit of the range of i defined in the second case of above equation. From this sequence we get the following upper bounds on the spreading times in level i,

$$E[S(\lambda_i)] \le \begin{cases} 0, & \text{if } i < L_1; \\ \text{ld}\left(\frac{n}{\tau(n-i)}\right), & \text{if } L_1 \le i < L_2; \\ \text{ld}\left(\frac{\lambda}{c}\right), & \text{otherwise.} \end{cases}$$

Intuitively speaking, while the fitness $i < L_1$ is small, a single island is capable of making significant progress on its own and does not require any migration. The middle range is designed such that the sum $\tau E[S(\lambda_i)] + 1/(p_i\lambda_i) \le \tau\,\text{ld}\,\lambda_i + en/(\lambda_i(n-i))$ stemming from Lemma 2.2 is minimized (up to constant factors). This balances the time needed to spread good solutions with the waiting time to complete the level. If $i > L_2$ is already quite large, we need a lot of generations to make further progress, it is thus beneficial to inform (almost) all islands in the meantime. We tacitly assume $\lambda/c > 1$. Otherwise, λ is constant and we get the usual $O(n\log n)$ bound.

Applying Corollary 2.3 to the sequence $(\lambda_i)_i$ gives

$$E[T_{\text{OM}}] \le n + \sum_{i=0}^{L_1-1}\left(\frac{en}{n-i}\right) + \sum_{i=L_2}^{n-1}\left(\tau\,\text{ld}\,\lambda + \frac{cen}{\lambda(n-i)}\right) + \sum_{i=L_1}^{L_2-1}\left(\tau\,\text{ld}\left(\frac{n}{\tau(n-i)}\right) + \frac{en}{n-i}\left(\frac{\tau(n-i)}{n}\right)\right).$$

We handle the partial sums separately.

$$P_1 = \sum_{i=0}^{L_1-1}\left(\frac{en}{n-i}\right) = en\sum_{j=n-L_1+1}^{n}\frac{1}{j} \le en\ln\left(\frac{n}{n-L_1}\right) + en.$$

The last inequality is due to the estimate $\sum_{j=k}^{n} 1/j \le \ln(n/(k-1))+1$, with $k > 1$, of the harmonic series. Substituting $n - L_1 = n/\tau$ gives $P_1 \le en(\ln\tau + 1)$.

$$P_2 = \sum_{i=L_2}^{n-1}\left(\tau\,\text{ld}\,\lambda + \frac{cen}{\lambda(n-i)}\right) = (n-L_2)\tau\,\text{ld}\,\lambda + ce\frac{n}{\lambda}\left(\sum_{j=1}^{n-L_2}\frac{1}{j}\right) \le (n-L_2)\tau\,\text{ld}\,\lambda + ce\frac{n}{\lambda}(\ln(n-L_2)+1).$$

With $n - L_2 = n/(\lambda\tau)$ we get

$$P_2 \le \frac{n\,\text{ld}\,\lambda}{\lambda} + ce\frac{n}{\lambda}\left(\ln\left(\frac{cn}{\lambda\tau}\right)+1\right).$$

Regarding the third part, we have

$$P_3 = \sum_{i=L_1}^{L_2-1}\left(\tau\,\text{ld}\left(\frac{n}{\tau(n-i)}\right) + \frac{en}{n-i}\left(\frac{\tau(n-i)}{n}\right)\right) = \tau\left(\sum_{j=n-L_2+1}^{n-L_1}\text{ld}\left(\frac{n}{\tau j}\right)\right) + (L_2-L_1)e\tau.$$

We insert $L_2 - L_1 = (1-c/\lambda)\cdot n/\tau \le n/\tau$ and bound the sum by an integral,

$$P_3 \le \tau\left(\int_{n/(\lambda\tau)}^{n/\tau}\text{ld}\left(\frac{n}{\tau j}\right)\mathrm{d}j\right) + en \le (e+1)n - \frac{n\,\text{ld}\,\lambda}{\lambda}.$$

As a result, the $(n\,\text{ld}\,\lambda)/\lambda$ terms in P_2 and P_3 cancel out and

$$E[T_{\text{OM}}] \le P_1 + P_2 + P_3 = O\left(n\log\tau + \frac{n}{\lambda}\log\left(\frac{n}{\lambda\tau}\right)\right) = O\left(n\log\tau + \frac{n\log n}{\lambda}\right). \qquad \square$$

For the push protocol on a complete graph a linear parallel optimization time can be enforced by setting $\lambda = \Omega(\log n)$ and τ a constant. This parameter setting also minimizes the expected combined costs $E[T_{\text{OM}} + C_{\text{OM}}]$. Observe that the communication costs are always dominated by the optimization time. However, when choosing the migration interval $\tau = \Omega(n)$ (e.g. to reduce communication costs even more) the island model behaves like a single (1+1) EA and the influence of migration is diminished.

Lässig and Sudholt [14] consider, instead of deterministic migration intervals, a *migration probability p*, with which any two neighboring islands communicate. This loosely corresponds to a migration interval of $\tau = 1/p$. While Theorem 3.2.(1) gives a logarithmic dependence on τ, the corresponding bound from [14, Theorem 18] gives a linear dependence in $1/p$.

Theorem 3.3. *Consider the complete graph K_λ as the migration topology using the push protocol. Then,*

(1) $E[T_{\text{LO}}] = O\left(\frac{n^2}{\lambda} + n\tau\log\left(\frac{n}{\tau}\right)\right)$;

(2) $E[C_{\text{LO}}] = O\left(\frac{n^2}{\lambda\tau} + n\log\left(\frac{n}{\tau}\right)\right)$.

Proof. It is clearly enough to show (1). The proof follows the same ideas as that of Theorem 3.2 but is somewhat easier. The reason being that for LeadingOnes the bound on the probability p_i does not depend on the current level, and it is always at least $1/en$. Let again $c \ge 1$ be such that $S(k) \le \text{ld}\,k$ whenever $k \le \lambda/c$. We choose $\lambda_i = \min\{n/\tau, \lambda/c\}$ for all $0 \le i \le n-1$. Assume for the moment that the λ_i defined that way are all larger than 1. We split the analysis into two cases. First, suppose that the above minimum is n/τ. From Corollary 2.3 we get

$$E[T_{\text{LO}}] \leq n+n\left(\tau S\left(\frac{n}{\tau}\right)+\frac{en}{n/\tau}\right) \leq n\tau\left(\text{ld}\left(\frac{n}{\tau}\right)+2e\right)=\text{O}\left(n\tau\log\left(\frac{n}{\tau}\right)\right).$$

Now suppose $\lambda_i = \lambda/c$, then

$$E[T_{\text{LO}}] \leq n+n\left(\tau S\left(\frac{\lambda}{c}\right)+\frac{cen}{\lambda}\right) \leq n\left(\tau\,\text{ld}\left(\frac{\lambda}{c}\right)+\frac{cen}{\lambda}+1\right).$$

Using the assumption $\lambda/c \leq n/\tau$ gives the claimed bound.

If n/τ is smaller than 1, we choose $\lambda_i = 1$ instead for all i. This corresponds to the migration interval τ being too large to benefit the optimization. However, also the influence of the spreading time S is reduced to zero. The optimization time degenerates to the usual $\text{O}(n^2)$ generations of a single (1+1) EA (which can be seen by another application of Corollary 2.3). This can only happen if $\tau = \Omega(n)$. The observation that this implies $\text{O}(n^2) = \text{O}(n\tau\log(n/\tau))$ completes the proof. □

For LeadingOnes, an expected optimization time of $\text{O}(n\log n)$ can be reached by setting $\lambda = \Omega(n/\log n)$ and $\tau = \Theta(1)$. The same parameter setting applies to $E[T_{\text{LO}} + C_{\text{LO}}]$.

The bound given in Theorem 3.3 is never worse than the $\text{O}(n^2/\lambda + n\tau\log\lambda)$ proven in [14] and for all reasonable parameter settings they are equivalent.

4 MULTIDIMENSIONAL TORI

In this section we investigate the broadcast model. Every τ generations all islands send their best solution to all their neighbors simultaneously. Now the spreading time is a deterministic function. Also, the communication costs are functionally determined by the optimization time and the structure of the underlying graph. Let $d(G)$ denote the average degree of a given migration topology G, then random variables C and T differ by a factor $d(G)/\tau$. Consequently, we focus again on bounding the optimization time.

As a proof of concept we consider the d-dimensional torus as topology. It can be constructed from a (finite) d-grid by connecting the outermost vertices through wrapping edges. More formally, fix two integers d and ℓ at least 1 and define the vertex set $V = \{0,\dots,\ell-1\}^d$. $E = \{\{u,v\} \in \binom{V}{2} \mid \exists i\colon (u_i - v_i = 1 \mod \ell) \wedge (\forall j \neq i : u_j = v_j)\}$ is the edge set. Symbol u_i denotes the i-th component of vector u. In dimension one this definition gives a bidirectional ring and in two dimensions the usual torus.

We would like to point out that throughout this section d is regarded as a constant independent of n.

A characteristic property of d-tori is that the spreading time is in the order of the d-th root, provided that only a constant fraction of the nodes needs to be informed.

Lemma 4.1. *Consider the d-dimensional torus as the migration topology using broadcast communication. Define constant $c = 1$, if $d = 1$, and $c = 4^d(d-1)^{d-1}$ otherwise. Then, for every $1 \leq k \leq \lambda/c$, $S(k) \leq 2d\sqrt[d]{k}$.*

Proof. Let integers d and ℓ be as defined above. First assume $d \geq 2$. As long as no wrapping edges are involved, the collection of informed nodes make up a d-dimensional diamond shape. W.l.o.g. it is centered at node $(0,\dots,0) \in V$. This polytope is bounded by 2^d $(d-1)$-dimensional faces consisting of exactly the islands that have still uninformed neighbors. They are the only ones contributing to the rumor spreading in the next round. In order to avoid double counting, we only consider a single face, namely the one pointing in the "direction" of vector $(1,\dots,1)$. After $t \geq 0$ communication steps this face consists of exactly the points $(a_1, a_2, \dots, a_d)$ satisfying $a_1 + a_2 + \cdots + a_d = t$. Basic combinatorics tells us that there are $\binom{d+t-1}{d-1}$ many of them.

Let the *growth rate* [3], $F(t)$, be the total number of informed nodes after t communication steps (i.e., the size of the whole diamond).

$$F(t) \geq \sum_{i=0}^{t}\binom{d+i-1}{d-1} \geq \sum_{i=0}^{t}\left(\frac{i}{d-1}\right)^{d-1} \geq \frac{1}{(d-1)^{d-1}}\left(\frac{t}{2}\right)^d.$$

This implies that the spreading time $S(k) = \min_t\{t \mid F(t) \geq k\}$ is at most $2(d-1)^{\frac{d-1}{d}}\sqrt[d]{k}$, which is smaller than the claimed bound.

The condition of not using the wrapping edges is surely satisfied for the first $\ell/2$ steps. During this period at least

$$F\left(\frac{\ell}{2}\right) \geq \frac{1}{(d-1)^{d-1}}\left(\frac{\ell}{4}\right)^d = \frac{\lambda}{c}$$

nodes can be informed. Here, we used that $\ell^d = \lambda$ by construction.

The result for $d = 1$ can be derived by elementary means. It is easy to see that $F(t) = 2t+1$ for the bidirectional ring and thus $S(k) = \lceil (k-1)/2 \rceil$. This imposes no further conditions on k. □

Theorem 4.2. *Consider the d-dimensional torus as the migration topology using broadcast communication. Then,*

(1) $E[T_{\text{OM}}] = \text{O}\left(\frac{n\log n}{\lambda} + n\log\tau\right)$;

(2) $E[C_{\text{OM}}] = \text{O}\left(\frac{n\log n}{\lambda\tau} + n\frac{\log\tau}{\tau}\right)$;

(3) $E[T_{\text{LO}}] = \text{O}\left(\frac{n^2}{\lambda} + n^{\frac{d+2}{d+1}}\tau^{\frac{d}{d+1}}\right)$;

(4) $E[C_{\text{LO}}] = \text{O}\left(\frac{n^2}{\lambda\tau} + n^{\frac{d+2}{d+1}}\tau^{-\frac{1}{d+1}}\right)$.

We omit the proof due to space limitations. The results can be obtained using the same techniques as in the proofs of Theorem 3.2 and Theorem 3.3. Note that the d-torus is a $2d$-regular graph and d a constant, which implies $E[C] = \text{O}(E[T]/\tau)$.

Theorem 4.2.(1) improves on the bounds in [14, Theorem 7] for OneMax (with a migration probability equal to $1/\tau$) in showing that the dependency of the optimization time on τ is logarithmic instead of $\sqrt{\tau}$ for the ring or $\tau^{2/3}$ for the torus. Part (3) generalizes their bounds for LeadingOnes to arbitrary dimensions $d \geq 1$.

We get an expected optimization time of $E[T_{\text{OM}}] = \text{O}(n)$ with $\lambda = \Omega(\log n)$, and $E[T_{\text{LO}}] = \text{O}\left(n^{\frac{d+2}{d+1}}\right)$ with $\lambda = \Omega\left(n^{\frac{d}{d+1}}\right)$. For both cases a constant migration interval τ is best. These bounds extend to the respective combined optimization costs.

5 THE COMPLETE GRAPH

We now cover the special case of broadcast communication on a complete graph as the migration topology. This setting differs from all examples above in that the spreading time degenerates into a step function. That means in the vast majority of iterations the islands compute their local improvements in total isolation. However, periodically *all* islands obtain a *globally* best solution in a network-spanning communication effort, only to be left alone for another phase of τ generations.

If the migration interval is too large, namely, if $\tau > en$, one can expect every island to find its own improving mutation w.r.t. LeadingOnes (and even more so for OneMax) between migrations. Then, inter-node communication is obsolete, eroding the characteristics of an island model. This can also be seen from the Theorems 3.2 through 4.2 as for $t = \Omega(n)$ the parallel optimization times exceed the run time of a simple (1+1) EA on the same fitness function. Consequently, we assume $\tau \leq en$ throughout this section.

For the other extreme of $\tau = 1$, it has been pointed out that the island model using broadcast on a complete graph is very similar to the (1+λ) EA [14]. The only distinction is that different islands can store different solutions of the same maximum fitness. Tight run time bounds for the (1+λ) EA on OneMax are known [9]. Hence, by characterizing the optimization time of the island model we can precisely quantify the influence of the migration interval τ.

In this section we give tight bounds for the expected optimization time for the OneMax fitness function and upper and lower bounds for LeadingOnes. The expected communication costs can be obtained from this value by multiplying with a factor $\Theta(\lambda/\tau)$ since every islands sends $(\lambda - 1)$ messages every τ iterations. Although these bounds will be proven for the K_λ as migration topology, the lower bounds extend to any connected graph. This is due to the fact that additional informed islands can only benefit the optimization and no topology spreads solutions faster than the complete graph.

5.1 OneMax

Theorem 5.1. *Consider the complete graph K_λ as the migration topology using broadcast communication and a migration interval $\tau \leq en$. If $\tau = o(\log\lambda)$, we have*

$$E[T_{\mathrm{OM}}] = \Theta\left(\frac{n\log n}{\lambda} + \frac{n\tau}{\log\lambda}\log\left(\frac{\log\lambda}{\tau}\right)\right).$$

If $\tau = \Theta(\log\lambda)$, we have

$$E[T_{\mathrm{OM}}] = \Theta\left(\frac{n\log n}{\lambda} + n\right).$$

If $\tau = \omega(\log\lambda)$, we have

$$E[T_{\mathrm{OM}}] = \Theta\left(\frac{n\log n}{\lambda} + n\log\left(\frac{\tau}{\log\lambda}\right)\right).$$

In the remainder of this section we prove the various bounds given in the theorem, starting with the upper bounds. Prior to this we need the following two lemmas. Due to space limitations we omit their proofs.

Lemma 5.2. *Suppose a positive integer n, probability $0 < p < 1$, and a constant $\varepsilon > 0$ to be given. Let $c = (1-p)^n$ and let $X \sim \mathrm{Bin}(n,p)$ be a binomially distributed random variable. If $\ln(c/\varepsilon) \geq enp$,*

$$P\left[X \geq \frac{\ln(c/\varepsilon)}{\ln(\ln(c/\varepsilon)/(np\varepsilon))}\right] \geq \varepsilon.$$

Lemma 5.3 ([5, Prop. 9]). *Consider the (1+1) EA with mutation probability p on OneMax and suppose the current search point has k 0-bits remaining. If $k \leq 0.6n$, then the probability that the search point of the next iteration has less than $k/2$ 0-bits is $\exp(-\Omega(k))$.*

Proof of Theorem 5.1. *Upper Bound.* Since we aim to optimize OneMax, we consider the number of remaining 0-bits in a solution as the *distance* to the optimum. We divide the optimization process into three phases depending on the minimal distance of all islands. The cutoff points between the phases are at $d_0 = \min\{n, n\ln(\lambda)/(2\tau)\}$ and $d_1 = n/(\tau\ln\lambda)$. Note that the first phase is only relevant if $2\tau \geq \ln\lambda$. In a first step, we estimate the time it takes for Algorithm 1 to achieve a distance of at most d_0. This is bounded by the time to reach a distance of at most d_0 on a *single* island and an additional phase of τ iterations until this solution is sent to all islands. Regarding a single island, we can resort to the well-known run time bounds for the (1+1) EA, cf. e.g. [7]. They imply that the time for an island to reach distance d_0 is $O(n\log(n/d_0)) = O(n\log(\tau/\log\lambda))$, if $2\tau \geq \ln\lambda$, and 0 otherwise.

Before considering the second phase between d_0 and d_1, we analyze the third phase of optimization from d_1 all the way to the optimum. To that end, suppose that one island has a non-optimal solution of distance $d \leq d_1$. Using Corollary 2.3 adapted to the landscape of the d_1 fitness levels in question, we see that the expected remaining optimization time is at most

$$\sum_{d=1}^{d_1}\left(1 + \tau + \frac{en}{d\lambda}\right) = O\left(\tau d_1 + \frac{n\log d_1}{\lambda}\right) = O\left(\frac{n}{\log\lambda} + \frac{n\log n}{\lambda}\right).$$

Now we turn to the more involved phase of optimization between distances d_0 and d_1. If we consider a point in time where migration just occurred, all islands now have an individual with fitness d (where $d_0 > d > d_1$) and τ rounds of evolution without migration will follow. The probability that a specific 1-bit is gained during any iteration is at least $1/en$. Thus, the probability that it is not gained during τ iterations is at most $(1 - 1/en)^\tau \leq 1 - \tau/2en$, using $\tau \leq en$. As a result, the number of gained bits in τ iterations is dominated by a binomially distributed random variable with parameters d and $\tau/2en$. We use Lemma 5.2 to derive a bound on the possible improvements of λ islands during τ iterations. We put $\varepsilon = 1/\lambda$ so that we can conclude that, with constant probability, at least one island will have made the progress described in the lemma. Note that the c in Lemma 5.2 may be $o(1)$ if $d \geq 2en/\tau$; in this case we only consider $2en/\tau$ many missing bits, giving c constant. Note that the lemma requires $\ln(c\lambda) > d\tau/(2en)$. Thus, for small λ, we might need to restrict to an even smaller number d of bits (but again only by a constant factor). In any way, there is a constant c' such that

$$h(d) = c'\ \ln\lambda \Big/ \ln\left(\frac{n\ln\lambda}{\tau d}\right)$$

is a lower bound on the expected progress in the phase between subsequent migrations. Since we are only interested in asymptotic bounds, we will omit this constant in what follows.

We now use the Variable Drift Theorem [13, Theorem 4.6] with h as a bound to the improvement. Observe that h is monotonically increasing in d. We get that the expected time to reach distance d_1 when starting from d_0 is at most

$$\tau\int_{d_1}^{d_0}\frac{\mathrm{d}x}{h(x)} = \frac{\tau}{\ln\lambda}\int_{d_1}^{d_0}\ln\left(\frac{n\ln\lambda}{\tau x}\right)\mathrm{d}x = -\frac{\tau}{\ln\lambda}\int_{d_1}^{d_0}\ln\left(\frac{\tau x}{n\ln\lambda}\right)\mathrm{d}x.$$

We use integration by substitution with $y = \frac{\tau x}{n \ln \lambda}$ and arrive at

$$-\frac{\tau}{\ln \lambda}\frac{n \ln \lambda}{\tau}\int_{\tau d_1/n \ln \lambda}^{\tau d_0/n \ln \lambda} \ln y \, \mathrm{d}y = -n\int_{1/(\ln \lambda)^2}^{\tau d_0/n \ln \lambda} \ln y \, \mathrm{d}y.$$

The integral of the natural logarithm is $y \mapsto y(\ln(y) - 1)$. In the case of $d_0 = n$, i.e., $\tau < \ln \lambda$, we thus resolve the integral to

$$-n\Big[y(\ln(y)-1)\Big]_{1/(\ln \lambda)^2}^{\tau/\ln \lambda} = \frac{n}{\ln \lambda}\left(\tau\left(\ln\left(\frac{\ln \lambda}{\tau}\right)+1\right) - \frac{2\ln\ln\lambda + 1}{\ln \lambda}\right),$$

in accordance with the claimed bound. If $\tau \geq \ln \lambda$, we get

$$-n\Big[y(\ln(y)-1)\Big]_{1/(\ln \lambda)^2}^{1} = n\left(1 - \frac{2\ln\ln\lambda+1}{(\ln\lambda)^2}\right),$$

which is $\mathrm{O}(n)$. Combined with the bounds already established above and the additive term $1/h(d_1) = O(1)$ from the variable drift theorem, this gives the desired upper bounds on the expected optimization time in the three cases stated in the theorem.

Lower Bound. It is straightforward to prove a lower bound of $\Omega((n \log n)/\lambda)$ from the observation that the *unary unbiased black-box complexity* of OneMax is $\Omega(n \log n)$ [15]. The (1+1) EA needs an expected number of $\Omega(n \log n)$ fitness evaluations to optimize OneMax. So λ copies of it need at least $\Omega((n \log n)/\lambda)$ generations to provide this many evaluations.

For the remaining terms we will give phases of the optimization which have the claimed run time as a lower bound. In order to see that none of these phases is skipped, we use Lemma 5.3. First, we examine the case of $\tau = \omega(\log \lambda)$. We show that the expected time it takes until any island samples a solution with at most $(n \log \lambda)/\tau$ 0-bits for the first time is $\Omega(n \log(\tau/\log \lambda))$, which establishes the bound in this case. Again, we consider the progress of a single island in the τ iterations between migrations. Suppose the current solution has distance $d \geq (n \log \lambda)/\tau$ to the optimum. The expected progress between subsequent generations is at most d/n, cp. [23, Lemma 6.7]. Therefore, within τ iterations the expected progress is at most $d\tau/n \geq \log \lambda$. For any constant $C \geq 6$, the probability of progress at least $Cd\tau/n$ is at most $2^{-Cd\tau/n} \leq 2^{-C\log\lambda} = \lambda^{-C}$, using a standard Chernoff bound. In particular, the probability that there is one among the λ islands which makes a progress of at least $Cd\tau/n$ is λ^{-C+1}. This shows that the expected progress of the *best* out of the λ islands between migrations is $\mathrm{O}(d\tau/n)$. Using a Multiplicative Drift Theorem for lower bounds from [22], we get that the time the island model takes to optimize a randomly initialized bit string into one with at most $n \log \lambda/\tau$ 0-bits is

$$\tau \cdot \frac{n}{\tau} \cdot \left(\ln n - \ln\left(\frac{n \log \lambda}{\tau}\right)\right) = \Omega\left(n \log\left(\frac{\tau}{\log \lambda}\right)\right).$$

Next we consider the case of $\tau = \Theta(\log \lambda)$, where we want to show a bound of $\Omega(n)$. To that end, we measure the time Algorithm 1 takes to get from distance $n(\ln \lambda)/\tau$ to distance $n(\ln \lambda)/(2\tau)$ from the optimum. The reasoning is similar as above. Suppose the number of bits set to 0 in the currently best individual is still $d \geq n(\ln \lambda)/(2\tau)$. In expectation the fitness on this island improves by at most $d\tau/n \geq (\ln \lambda)/2$ within τ rounds and for any $C \geq 6$, the probability that the island makes progress of at least $Cd\tau/n$ is at most $2^{-Cd\tau/n} \leq \lambda^{-C/2}$. Using the assumption $d \leq n(\ln \lambda)/\tau$, we get an expected progress of $\mathrm{O}(\log \lambda)$ over all λ islands and τ iterations. The Additive Drift Theorem in [12] implies $\Omega(n)$ steps are needed to find a solution of distance at most $n(\ln \lambda)/(2\tau)$.

Finally, suppose $\tau = o(\log \lambda)$. We want to show a lower bound of $\Omega(\frac{n\tau}{\log \lambda} \log \frac{\log \lambda}{\tau})$. This time we consider the range between distance $n(\ln \lambda)/\tau$ and n/τ to the optimum. Suppose the current best individual of an island has $d \geq n/\tau$ 0-bits. The expected progress in τ iterations of this island is at most $d\tau/n$. We define

$$h(d) = \ln \lambda \Big/ \ln\left(\frac{n \ln \lambda}{\tau d}\right)$$

and abbreviate $r = \ln(n(\ln \lambda)/(d\tau))$, that is, $h(d) = (\ln \lambda)/r$. The Chernoff bound argument is a bit more involved but still shows that for any C sufficiently large, the probability that one island makes progress of at least $C\,h(d)$ is at most

$$\begin{aligned}\exp\left(-\ln\left(\frac{n}{ed\tau}C\,h(d)\right)C\,h(d)\right) &= \exp\left(-\ln\left(\frac{Cn\ln\lambda}{e\tau dr}\right)C\,\frac{\ln\lambda}{r}\right)\\ &\leq \exp\left(-\ln\left(\frac{n\ln\lambda}{\tau d}\right)C\,\frac{\ln\lambda}{2r}\right) = \exp\left(-C\,\frac{\ln\lambda}{2}\right) = \lambda^{-\frac{C}{2}}.\end{aligned}$$

Once again we conclude that the maximum progress of λ islands in τ iterations is $\mathrm{O}(h(d))$. With a Variable Drift Theorem for lower bounds [11] we can employ the same integration method as for the upper bound to get a matching run time. □

The tight bounds on the expected optimization time translate to the following optimal parameter setting. To minimize the parallel optimization time (for λ bounded from above by a polynomial) one should choose $\lambda = n^{\Theta(1)}$ and τ a constant to obtain $E[T_{\mathrm{OM}}] = \Theta(n(\log\log n)/\log n)$. If, however, the total optimization costs $E[T_{\mathrm{OM}} + C_{\mathrm{OM}}]$ for both the computation and communication are to be optimized, the ideal choice is $\lambda = \Theta(\log n)$ and $\tau = \Theta(\log n)$ which leads to a bound of $\Theta(n \log\log n)$.

5.2 LeadingOnes

An upper bound on the optimization time of LeadingOnes on the complete graph can be obtained from the results in [14] or an easy application of Corollary 2.3.

Theorem 5.4. *Consider the complete graph K_λ as the migration topology using broadcast communication. Then,*

$$E[T_{\mathrm{LO}}] = \mathrm{O}\left(\frac{n^2}{\lambda} + n\tau\right).$$

We proceed to prove a lower bound on the expected optimization time. The main difficulty in applying fitness-level arguments to *lower* bounds is the possibility that the optimization process may skip several levels with a single improvement. We handle this issue by combining a poly-logarithmic number of consecutive levels to one block; skipping a block then is unlikely. This technique has already been used in [3]. We omit the proof.

Lemma 5.5. *Consider the complete graph K_λ optimizing* LeadingOnes *using broadcast communication and a migration interval $\tau \leq en$. If $\lambda = \mathrm{poly}(n)$, there is a constant $c > 0$ such that the probability of any island finding a fitness improvement of more than $c\ln^2 n$ between consecutive migrations is $o(1/n)$.*

THEOREM 5.6. *Under the conditions of Lemma 5.5, we have*

$$E[T_{\mathrm{LO}}] = \Omega\left(\frac{n^2}{\lambda} + \frac{n}{\log^2 n}\tau\right).$$

PROOF. $\Omega(n^2/\lambda)$ clearly is a lower bound on the optimization time, again from the unbiased black-box complexity of LEADINGONES, cp. [15]. The second term stems from the time the islands spend on finding independent improvements between migrations. We prove this bound by coupling several random variables.

Let the constant $c > 0$ be as in Lemma 5.5 and, for integers $0 \leq j \leq (n+1)/(c\ln^2 n)$, define the *j-th fitness block* as the fitness levels from $j(c\ln^2 n)$ to $(j+1)(c\ln^2 n) - 1$. Note that optimization on all islands starts in block 0 with probability superpolynomially close to 1, by Chernoff bounds. We define a simplified random process of *block discovery* that models the computation on a single island but, at the same time, incorporates the beneficial influence of migration. There are two ways for the island to discover a block. Every mutation flipping the left-most 0-bit and none of the leading 1s, called an *essential step*, leads to the discovery of a new block. Migration, happening every τ iterations, grants another one.

Let T' stand for the random variable denoting the number of rounds the process needs to discover all $(n+1)/(c\ln^2 n) + 1$ blocks. Lemma 5.5 implies that the probability of the original optimization skipping a full block in $n\tau$ iterations is $o(1)$. Hence, T_{LO} first-order stochastically dominates T' and any lower bound on $E[T']$ extends to $E[T_{\mathrm{LO}}]$ (if it does not exceed $n\tau$).

To establish the estimate $E[T'] = \Omega(n\tau/\log^2 n)$, it is enough to prove the existence of a constant $c' > 0$ such that

$$\lim_{n\to\infty} P\left[T' \leq c'\frac{n\tau}{\ln^2 n}\right] = 0.$$

We fix some real number $c' < 1/(c(e+1))$. The reason for this choice will become apparent in the following discussion. For the moment, it is sufficient that it ensures $C = (1/c) - c' > 0$.

Define $t = c'n\tau/\ln^2 n$. The discovery process guarantees $t/\tau = c'n/\ln^2 n$ blocks via migration. If $T' \leq t$ shall hold, the essential steps have to make up for the remaining blocks. Let X denote the number of essential steps during t rounds. X, in turn, is dominated by a binomially distributed variable $Y \sim \mathrm{Bin}(t, 1/n)$. That is,

$$P[T' \leq t] \leq P\left[X > \frac{n+1}{c\ln^2 n} - \frac{t}{\tau}\right] \leq P\left[Y > C\frac{n}{\ln^2 n}\right].$$

It is best to split the remaining argument into two cases depending on the limit behavior of $E[Y] = t/n = c'\tau/\ln^2 n$. First, suppose the expected value is bounded for all n, then $\mathrm{Var}[Y] = E[Y](1-(1/n))$ is bounded as well. By Chebyshev's inequality,

$$\lim_{n\to\infty} P\left[Y > C\frac{n}{\ln^2 n}\right] \leq \lim_{n\to\infty} \frac{\mathrm{Var}[Y]}{\left(C\frac{n}{\ln^2 n} - E[Y]\right)^2} = 0.$$

In case $E[Y]$ diverges, we define

$$1 + \delta = \frac{Cn}{E[Y]\ln^2 n} = \frac{\frac{1}{c} - c'}{c'} \cdot \frac{n}{\tau}.$$

The assumptions $\tau \leq en$ and $c' < 1/(c(e+1))$ together imply $\delta > 0$. Using Chernoff bounds, we finally arrive at

$$P\left[Y > C\frac{n}{\ln^2 n}\right] = P\big[Y > (1+\delta)E[Y]\big] \leq \exp\left(-\frac{\delta}{3}E[Y]\right).$$

The right member of the inequality converges to 0. □

The upper and lower bounds for LEADINGONES yield the following optima. To minimize $E[T_{\mathrm{LO}}]$ choose $\lambda = \Theta(n)$ and a constant τ, resulting in $O(n)$. The lower bound $\Omega(n/\log^2 n)$ follows immediately because τ cannot be sub-constant. For the combined costs $E[T_{\mathrm{LO}} + C_{\mathrm{LO}}]$ set $\lambda = \Theta(\sqrt{n})$ and $\tau = \Theta(\sqrt{n})$ to get $O(n^{3/2})$. The lower bound $\Omega(n^{3/2}/\log n)$ follows from the following observation. Set $\lambda = \Theta(\sqrt{n}\log n)$ and $\tau = \Theta(\sqrt{n}\log n)$. All terms of the lower bound are $\Omega(n^{3/2}/\log n)$, and changing any parameter will increase either of the terms.

ACKNOWLEDGMENTS

This work has been partially supported by the X-DGA contract.

REFERENCES

[1] E. Alba. Parallel evolutionary algorithms can achieve super-linear performance. *Information Processing Letters*, 82:7–13, 2002.
[2] E. Alba, G. Luque, and S. Nesmachnow. Parallel metaheuristics: Recent advances and new trends. *International Transactions in Operational Research*, 20:1–48, 2013.
[3] G. Badkobeh, P. K. Lehre, and D. Sudholt. Black-box complexity of parallel search with distributed populations. In *Proc. of FOGA'15*, pages 3–15, 2015.
[4] E. Cantú-Paz. A survey of parallel genetic algorithms. *IlliGAL Report*, 97003, 1997.
[5] B. Doerr. Optimal parameter settings for the (1+(λ, λ)) genetic algorithm. *CoRR*, abs/1604.01088, 2016.
[6] B. Doerr and M. Fouz. Asymptotically optimal randomized rumor spreading. In *Proc. of ICALP'11*, pages 502–513, 2011.
[7] B. Doerr, D. Johannsen, and C. Winzen. Multiplicative drift analysis. *Algorithmica*, 64:673–697, 2012.
[8] B. Doerr and M. Künnemann. Tight analysis of randomized rumor spreading in complete graphs. In *Proc. of ANALCO'14*, pages 82–91, 2014.
[9] B. Doerr and M. Künnemann. Optimizing linear functions with the $(1+\lambda)$ evolutionary algorithm - different asymptotic runtimes for different instances. *Theoretical Computer Science*, 561:3–23, 2015.
[10] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.
[11] C. Gießen and C. Witt. Optimal mutation rates for the (1+λ) EA on onemax. In *Proc. of GECCO'16*, pages 1147–1154, 2016.
[12] J. He and X. Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35, 2004.
[13] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, 2010.
[14] J. Lässig and D. Sudholt. General upper bounds on the runtime of parallel evolutionary algorithms. *Evolutionary Computation*, 22:405–437, 2014.
[15] P. K. Lehre and C. Witt. Black-Box Search by Unbiased Variation. *Algorithmica*, 64:623–642, 2012.
[16] A. Lissovoi and C. Witt. On the utility of island models in dynamic optimization. In *Proc. of GECCO'15*, pages 1447–1454, 2015.
[17] A. Mambrini and D. Sudholt. Design and analysis of schemes for adapting migration intervals in parallel evolutionary algorithms. *Evolutionary Computation*, 23:559–582, 2015.
[18] F. Neumann, P. S. Oliveto, G. Rudolph, and D. Sudholt. On the effectiveness of crossover for migration in parallel evolutionary algorithms. In *Proc. of GECCO'11*, pages 1587–1594, 2011.
[19] M. Ruciński, D. Izzo, and F. Biscani. On the impact of the migration topology on the island model. *Parallel Computing*, 36:555–571, 2010.
[20] D. Sudholt. Parallel evolutionary algorithms. In *Springer Handbook of Computational Intelligence*, pages 929–959. 2015.
[21] I. Wegener. Theoretical aspects of evolutionary algorithms. In *Proc. of ICALP'01*, pages 64–78, 2001.
[22] C. Witt. Optimizing linear functions with randomized search heuristics - the robustness of mutation. In *Proc. of STACS'12*, pages 420–431, 2012.
[23] C. Witt. Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Combinatorics, Probability and Computing*, 22:294–318, 2013.