Article:

Lässig, J. and Sudholt, D. orcid.org/0000-0001-6020-1646 (2014) General Upper Bounds on the Runtime of Parallel Evolutionary Algorithms. Evolutionary Computation, 22 (3). pp. 405-437. ISSN 1063-6560

https://doi.org/10.1162/EVCO_a_00114

General Upper Bounds on the Running Time of Parallel Evolutionary Algorithms[*]

Jörg Lässig jlaessig@hszg.de
Department of Electrical Engineering and Computer Science,
University of Applied Sciences Zittau/Görlitz, Germany

Dirk Sudholt d.sudholt@sheffield.ac.uk
Department of Computer Science, University of Sheffield, United Kingdom

Abstract

We present a general method for analyzing the running time of parallel evolutionary algorithms with spatially structured populations. Based on the fitness-level method, it yields upper bounds on the expected parallel running time. This allows to rigorously estimate the speedup gained by parallelization. Tailored results are given for common migration topologies: ring graphs, torus graphs, hypercubes, and the complete graph. Example applications for pseudo-Boolean optimization show that our method is easy to apply and that it gives powerful results. In our examples the performance guarantees improve with the density of the topology. Surprisingly, even sparse topologies like ring graphs lead to a significant speedup for many functions while not increasing the total number of function evaluations by more than a constant factor. We also identify which number of processors lead to the best guaranteed speedups, thus giving hints on how to parametrize parallel evolutionary algorithms.

Keywords

Parallel evolutionary algorithms, runtime analysis, island model, spatial structures

1 Introduction

Due to the increasing number of CPU cores, exploiting possible speedups by parallel computations is nowadays more important than ever. Parallel evolutionary algorithms (EAs) form a popular class of heuristics with many applications to computationally expensive problems [29, 31, 46]. This includes *island models*, also called *distributed EAs*, *multi-deme EAs* or *coarse-grained EAs*. Evolution is parallelized by evolving subpopulations, called *islands*, on different processors. Individuals are periodically exchanged in a process called *migration*, where selected individuals, or copies of these, are sent to other islands, according to a migration topology that determines which islands are neighboring. Also more fine-grained models are known, where neighboring subpopulations communicate in every generation, first and foremost in *cellular EAs* [46].

By restricting the flow of information through spatial structures and/or infrequent communication, diversity in the whole system is increased. Researchers and practitioners frequently report that parallel EAs speed up the computation time, and at the same time lead to a better solution quality [29].

Despite these successes, a long history [5] and very active research in this area [2, 29, 39], the theoretical foundation of parallel EAs is still in its infancy. The impact of even the most basic parameters on performance is not well understood [41].

---

[*] A preliminary version of this paper with parts of the results was published at PPSN 2010 [24].

Past and present research is mostly empirical, and a solid theoretical foundation is missing. Theoretical studies are mostly limited to artificial settings. In the study of *takeover times*, one asks how long it takes for a single optimum to spread throughout the whole parallel EA, if the EA uses only selection and migration, but neither mutation nor crossover [38, 39]. This gives a useful indicator for the speed at which communication is spread, but it does not give any formal results about the running time of evolutionary algorithms with mutation and/or crossover.

One way of gaining insight into the capabilities and limitations of parallel EAs is by means of rigorous running time analysis [15, 47]. By asymptotic bounds on the running time we can compare different implementations of parallel EAs and assess the speedup gained by parallelization in a rigorous manner. Many running time analyses have been presented [4, 18, 34, 35], from simple pseudo-Boolean test functions [8] to NP-hard problems from combinatorial optimization [9, 17, 48, 50].

In [23] the authors presented the first running time analysis of a parallel evolutionary algorithm with a non-trivial migration topology. It was demonstrated for a constructed problem that migration is essential in the following way. A suitably parametrized island model with migration has a polynomial running time while the same model without migration as well as comparable panmictic populations[1] need exponential time, with overwhelming probability. Neumann, Oliveto, Rudolph, and Sudholt [32] presented a similar result for island models using crossover. If islands perform crossover with immigrants during migration, this can drastically speed up optimization. This was demonstrated for a pseudo-Boolean example as well as for instances of the VERTEXCOVER problem [32].

In this work we take a broader view and consider the speedup gained by parallelization in terms of the number of generations, for various common pseudo-Boolean functions and function classes of varying difficulty. A general method is presented for proving upper bounds on the parallel running time of parallel EAs. The latter is defined as the number of generations of the parallel EA until a global optimum is found for the first time. This allows us to estimate the speedup gained by parallelization, defined as the ratio of the expected parallel running time of a single island and the expected running time of an island model with multiple islands (see Section 2 for formal definitions). It also can be used to determine how to choose the number of islands such that the best possible upper bounds on the parallel running time are obtained, while still maintaining an asymptotically optimal speedup.

Our method is based on the *fitness-level method* or *method of f-based partitions*, a simple and well-known tool for the analysis of evolutionary algorithms [8, 47]. The main idea of this method is to divide the search space into sets $A_1, \dots, A_m$, strictly ordered according to fitness values of elements therein. Elitists EAs, i. e., EAs where the best fitness value in the population can never decrease, can only increase their current best fitness. If, for each set A_i we know a lower bound s_i on the probability that an elitist EA finds an improvement, i. e., for finding a new search point in a new best fitness-level set $A_{i+1} \cup \dots \cup A_m$, this gives rise to an upper bound $\sum_{i=1}^{m} 1/s_i$ on the expected running time. The method is described in more detail in Section 2.

In Section 3 we first derive a general upper bound for parallel EAs, based on fitness levels. Our general method is then tailored towards different spatial structures often used in fine-grained or cellular evolutionary algorithms and parallel architectures in general: ring graphs (Theorem 8 in Section 4), torus graphs (Theorem 10 in Section 5),

---

[1] A panmictic population is an unstructured population with no mating restrictions.

hypercubes (Theorem 12 in Section 6) and complete graphs (Theorems 14 and 17 in Section 7).

The only assumption made is that islands run elitist algorithms, and that in each generation each island has a chance of transmitting individuals from its best current fitness level to each neighboring island, independently with probability at least p. We call the latter the *transmission probability*. It can be used to model various stochastic effects such as disruptive variation operators, the impact of selection operators, probabilistic migration, probabilistic emigration and immigration policies, and transient faults in the network. This renders our method widely applicable to a broad range of settings.

1.1 Main Results

Our estimates of parallel running times from Theorems 8, 10, 12, 14, and 17 are summarized in the following theorem, hence characterizing our main results. Throughout this work μ always denotes the number of islands.

Theorem 1. *Consider an island model with μ islands where each island runs an elitist EA. For each island let there be a fitness-based partition $A_1, \dots, A_m$ such that for all $1 \leq i < m$ all points in A_i have a strictly worse fitness than all points in A_{i+1}, and A_m contains all global optima. We say that an island is in A_i if the best search point on the island is in A_i. Let s_i be a lower bound for the probability that in one generation a fixed island in A_i finds a search point in $A_{i+1} \cup \dots \cup A_m$.*

Further assume that for each edge in the migration topology in every iteration there is a probability of at least p that the following holds, independently from other edges and for all $1 \leq i < m$. If the source island is in A_i then after the generation the target island is in $A_i \cup \dots \cup A_m$. Then the expected parallel running time of the island model is bounded from above by

1. $O\left(\frac{1}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}}\right) + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i}$ *for every ring graph or any other strongly connected*[2] *topology (Theorem 8),*
2. $O\left(\frac{1}{p^{2/3}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}}\right) + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i}$ *for every undirected grid or torus graph whose side lengths are at least $\sqrt{\mu}$ in both directions (Theorem 10),*
3. $O\left(\frac{m + \sum_{i=1}^{m-1} \log(1/s_i)}{p}\right) + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i}$ *for the $(\log \mu)$-dimensional hypercube graph (Theorem 12),*
4. $O(m/p) + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i}$ *for the complete topology K_μ, as well as* $O\left(m + \frac{m \log \mu}{\min\{p\mu, 1\}}\right) + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i}$ *(Theorems 14 and 17).*

A remarkable feature of our method is that it can automatically transfer upper bounds for panmictic EAs to parallel versions thereof. The only requirement is that bounds on panmictic EAs have been derived using the fitness-level method, and that the partition $A_1, \dots, A_m$ and the probabilities for improvements $s_1, \dots, s_{m-1}$ used therein are known. Then the expected parallel time of the corresponding island model can be estimated for all mentioned topologies simply by plugging the s_i into Theorem 1. Fortunately, many published runtime analyses use the fitness-level method—either explicitly or implicitly—and the mentioned details are often stated or easy to

---

[2] A directed graph is strongly connected if for each pair of vertices u, v there is a directed path from u to v and vice versa.

derive. Hence even researchers with limited expertise in runtime analysis can easily reuse previous analyses to study parallel EAs.

Further note that we can easily determine which choice of μ, the number of islands, will give an upper bound of order $1/\mu \cdot \sum_{i=1}^{m-1} 1/s_i$—the best upper bound we can hope for, using the fitness-level method. In all bounds from Theorem 1 we have a first term that varies with the topology and p, and a second term that is always $1/\mu \cdot \sum_{i=1}^{m-1} 1/s_i$. The first term reflects how quickly information about good fitness levels is spread throughout the island model. Choosing μ such that the second term becomes asymptotically as large as the first one, or larger, we get an upper bound of $O\Big(1/\mu \cdot \sum_{i=1}^{m-1} 1/s_i\Big)$. For settings where $\sum_{i=1}^{m-1} 1/s_i$ is an asymptotically tight upper bound for a single island, this corresponds to an asymptotic linear speedup. The maximum feasible value for μ depends on the problem, the topology and the transmission probability p.

		(1+1) EA	Ring	Grid/Torus	Hypercube	Complete
OneMax	best μ		$\mu = \Theta(\log n)$	$\mu = \Theta(\log n)$	$\mu = \Theta(\log n)$	$\mu = \Theta(\log n)$
	E (T^{par})	$\Theta(n \log n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
	E (T^{seq})	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$
	E (T^{com})	0	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n (\log n) \log \log n)$	$\Theta(n \log^2 n)$
LO	best μ		$\mu = \Theta(n^{1/2})$	$\mu = \Theta(n^{2/3})$	$\mu = \Theta\left(\frac{n}{\log n}\right)$	$\mu = \Theta(n)$
	E (T^{par})	$\Theta(n^2)$	$\Theta(n^{3/2})$	$\Theta(n^{4/3})$	$\Theta(n \log n)$	$\Theta(n)$
	E (T^{seq})	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$
	E (T^{com})	0	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2 \log^2 n)$	$\Theta(n^3)$
unimodal	best μ		$\mu = \Theta(n^{1/2})$	$\mu = \Theta(n^{2/3})$	$\mu = \Theta\left(\frac{n}{\log n}\right)$	$\mu = \Theta(n)$
	E (T^{par})	$O(dn)$	$O\big(dn^{1/2}\big)$	$O\big(dn^{1/3}\big)$	$O(d \log n)$	$O(d)$
	E (T^{seq})	$O(dn)$	$O(dn)$	$O(dn)$	$O(dn)$	$O(dn)$
	E (T^{com})	0	$O(dn)$	$O(dn)$	$O(dn \log n)$	$O\big(dn^2\big)$
Jump$_k$	best μ		$\mu = \Theta(n^{k/2})$	$\mu = \Theta(n^{2k/3})$	$\mu = \Theta(n^{k-1})$	$\mu = \Theta(n^{k-1})$
	E (T^{par})	$\Theta(n^k)$	$O\big(n^{k/2}\big)$	$O\big(n^{k/3}\big)$	$O(n)$	$O(n)$
	E (T^{seq})	$\Theta(n^k)$	$O\big(n^k\big)$	$O\big(n^k\big)$	$O\big(n^k\big)$	$O\big(n^k\big)$
	E (T^{com})	0	$O\big(n^k\big)$	$O\big(n^k\big)$	$O\big(kn^k \log n\big)$	$O\big(n^{2k-1}\big)$

Table 1: Asymptotic bounds on expected parallel (T^{par}, number of generations) and sequential (T^{seq}, number of function evaluations) running times and expected communication efforts (T^{com}, total number of migrated individuals) for various n-bit functions and island models with μ islands running the (1+1) EA and using migration probability $p = 1$. The number of islands μ was always chosen to give the best possible upper bound on the parallel running time, while not increasing the upper bound on the sequential running time by more than a constant factor. For unimodal functions $d + 1$ denotes the number of function values. See [8] for bounds for the (1+1) EA. Results for Jump$_k$ were restricted to $3 \leq k = O(n/\log n)$ for simplicity. All upper bounds for OneMax and LO stated here are asymptotically tight, as follows from general results in [43].

We give simple examples that demonstrate how our method can be applied. Our examples are from pseudo-Boolean optimization, but the method works in any setting where the fitness-level method is applicable. The simple (1+1) EA is used on each island (see Section 2 for details). Table 1 summarizes the resulting running time bounds for the considered algorithms and problem classes. For simplicity we assume $p = 1$; a more detailed table for general transmission probabilities is presented in the appendix, see Table 3. The number of islands μ was chosen as explained above: to give the smallest possible parallel running time, while not increasing the sequential time, asymptotically. The table also shows the expected *communication effort*, defined as the total number of individuals migrated throughout the run. Details are given in Theorems 8, 10, 12, 14, and 17. Bounds on the expected communication effort follow easily from bounds on the parallel running time using Theorem 2. The functions used in this table are explained in Section 2. Table 3 in the appendix shows all our results for a variable number of islands μ and variable transmission probabilities p.

The method has already found a number of applications and it spawned a number of follow-up papers. After the preliminary version of this work [24] was presented, the authors applied it for various problems from combinatorial optimization: the sorting problem (as maximizing sortedness), finding shortest paths in graphs, and Eulerian cycles [26]. Very recently, Mambrini, Sudholt, and Yao [30] also used it for studying how quickly island models find good approximations for the NP-hard SETCOVER problem. This work has also led to the discovery of simple adaptive schemes for changing the number of islands dynamically throughout the run, see Lässig and Sudholt [25]. These schemes lead to near-optimal parallel running times, while asymptotically not increasing the sequential running time on many examples [25]. These schemes are tailored towards island models with complete topologies, which includes offspring populations as special case. The study of offspring populations in comma strategies is another recent development that was inspired by this work [37].

2 Preliminaries

We are interested in the following performance measures, where the number of islands, μ, in the considered island model is obvious from the context. First we define the *parallel running time* T^{par} as the number of generations until the first global optimum is evaluated. Let T_μ be the number of generations before the island model finds a global optimum for the first time. Then

$$T^{\text{par}} := T_\mu.$$

The *sequential running time* T^{seq} is defined as the number of function evaluations until the first global optimum is evaluated. It thus captures the overall effort across all processors. It is formally defined as

$$T^{\text{seq}} := \mu \cdot T_\mu.$$

In both measures T^{par} and T^{seq} we allow ourselves to neglect the cost of the initialization as this only adds a fixed term to the running times.

The *speedup* is defined as the ratio of expected running times of a single island and an island model with μ islands:

$$\mathrm{E}(T_1) / \mathrm{E}(T_\mu).$$

Our definition of speedup is called *weak orthodox speedup* in Alba's taxonomy [1]. If the speedup is at least of order μ, i. e., if it is $\Omega(\mu)$, we speak of a *linear speedup*. In this work it is generally understood in an asymptotic sense, unless we call it a *perfect linear speedup*.

Note that our notion of speedup is defined with regard to the number of generations. In terms of wall-clock time, when the algorithm is stopped when reaching a global optimum, the execution time of an island model with μ islands can be estimated as

$$\mathrm{Exec}_\mu = T_\mu \cdot (\mathrm{Exec}_\mu^{\mathrm{gen}} + \mathrm{Exec}_\mu^{\mathrm{migr}}) \tag{1}$$

where $\mathrm{Exec}^{\mathrm{gen}}$ is the execution time for one generation (excluding migration) and $\mathrm{Exec}^{\mathrm{migr}}$ is the execution time for migration. The latter depends on the number of islands, the communication topology, and the number of individuals migrated. For sequential algorithms ($\mu = 1$) we have $\mathrm{Exec}_1^{\mathrm{migr}} = 0$. Contrarily, for homogeneous parallel systems we may assume that $\mathrm{Exec}_\mu^{\mathrm{gen}}$ is fixed, that is, independent of μ.

Many common definitions of "speedup" consider the wall-clock time, where the differences in $\mathrm{Exec}^{\mathrm{migr}}$ play a role. Here we consider speedups with regard to the number of generations only, ignoring the differences in $\mathrm{Exec}^{\mathrm{migr}}$. This makes sense in setting where $\mathrm{Exec}^{\mathrm{gen}} \gg \mathrm{Exec}^{\mathrm{migr}}$; for instance, when fitness evaluations are so expensive that they dominate the execution time. Otherwise, the speedups stated here may be optimistic as the overhead induced by migration is ignored. Note, however, that with additional information about $\mathrm{Exec}^{\mathrm{gen}}$ and $\mathrm{Exec}^{\mathrm{migr}}$ and Equation (1) our results easily extend to more sophisticated notions of speedups.

To get a more complete picture of the resources used in a parallel system and to take into account the overhead by communication, we also consider the *communication effort* T^{com}. It is defined as the total number of individuals migrated to other islands during the course of a run. The communication effort therefore captures the total bandwidth used during a run of an island model. It represents an important factor for determining the performance of a parallel EA, alongside the parallel running time.

The expected communication effort is a multiple of the parallel expected running time, with the factor depending on the number of (directed) edges in the topology, the transmission probability p and the number of individuals migrated in each migration event.

The following theorem lists various topologies: a unidirectional ring is a graph consisting of a single directed cycle, whereas a directed ring has undirected edges. Note that an undirected edge can be regarded as two directed edges. In a torus graph all vertices are arranged on a two-dimensional grid, with undirected edges wrapping around (vertices in the top row are neighbored to the ones in the bottom row and vice versa, similarly for the leftmost and rightmost columns). Each vertex in a torus thus has 4 distinct neighbors, provided that the torus has at least 3 rows and at least 3 columns. Hypercubes are formally defined in Section 6, and the complete graph K_μ contains undirected edges between all pairs of nodes.

Theorem 2. *Consider an island model with a directed graph $T = (V, E)$ as topology, such that in each generation along each directed edge migration takes place independently with probability p. Assume that ν individuals are migrated in each migration event. Let T^{par} be the parallel running time of the island model, then the expected communication effort $E\left(T^{\mathrm{com}}\right)$ is*

$$E\left(T^{\mathrm{com}}\right) = p\nu \cdot |E| \cdot E\left(T^{\mathrm{par}}\right). \tag{2}$$

Thereby, if $|V| = \mu$, we have

- $|E| = \mu$ *for a unidirectional ring and* $|E| = 2\mu$ *for a bidirectional ring,*
- $|E| = 4\mu$ *for any torus graph where both sides have length at least* 3,
- $|E| = \mu \log \mu$ *for the* $(\log \mu)$*-dimensional hypercube, and*
- $|E| = \mu(\mu - 1)$ *for the complete graph* K_μ.

Proof. Fix a single edge e, then the expected number of migration events across e equals

$$\sum_{t=0}^{\infty} \text{Prob}\left(T^{\text{par}} = t\right) \cdot t \cdot p = p \cdot \text{E}\left(T^{\text{par}}\right)$$

by definition of $\text{E}\left(T^{\text{par}}\right)$. By linearity of expectations, we can add these values for all edges to get the expected number of migration events across the whole topology. This yields a factor of $|E|$. Additionally multiplying by ν gives the expected communication effort. □

Hence to estimate the expected communication effort it suffices to analyze the expected parallel running time.

In our example applications we consider the maximization of a pseudo-Boolean function $f\colon \{0,1\}^n \to \mathbb{R}$. It is easy to adapt the method for minimization. The number of bits is always denoted by n. The following well known example functions have been chosen because they exhibit different probabilities for finding improvements in a typical run of an EA. For a search point $x \in \{0,1\}^n$ write $x = x_1 \dots x_n$, then $\text{OneMax}(x) := \sum_{i=1}^{n} x_i$ counts the number of ones in x and $\text{LO}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_i$ counts the number of leading ones in x, i. e., the length of the longest prefix containing only 1-bits. A function is called *unimodal* if every non-optimal search point has a Hamming neighbor (i. e., a point with Hamming distance 1 to it) with strictly larger fitness. Observe that LO is unimodal as flipping the first 0-bit results in a fitness increase. For LO every non-optimal point has exactly one Hamming neighbor with a better fitness. For $1 \le k \le n$ we also consider

$$\text{Jump}_k := \begin{cases} k + \sum_{i=1}^{n} x_i, & \text{if } \sum_{i=1}^{n} x_i \le n-k \text{ or } x = 1^n, \\ \sum_{i=1}^{n} (1 - x_i) & \text{otherwise.} \end{cases}$$

This function has been introduced by Droste, Jansen, and Wegener [8] as a function with tunable difficulty as evolutionary algorithms typically have to perform a jump to overcome a gap by flipping k specific bits. It is also interesting because it is one of very few examples where crossover has been proven to be essential [19,22].

Our method for proving upper bounds is based on the fitness-level method [8,47]. The idea is to partition the search space into sets $A_1, \dots, A_m$ called *fitness levels* that are ordered with respect to fitness values. We say that an algorithm is in A_i or on level i if the current best individual in the population is in A_i. An evolutionary algorithm where the best fitness value in the population can never decrease (called an *elitist* EA) can only improve the current fitness level. If one can derive lower bounds on the probability of leaving a specific fitness level towards higher levels, this yields an upper bound on the expected running time.

Theorem 3 (Fitness-level method)**.** *For two sets $A, B \subseteq \{0,1\}^n$ and a fitness function f let $A <_f B$ if $f(a) < f(b)$ for all $a \in A$ and all $b \in B$. Partition the search space into non-empty sets $A_1, A_2, \dots, A_m$ such that $A_1 <_f A_2 <_f \dots <_f A_m$ and A_m only contains global optima. For an elitist EA let s_i be a lower bound on the probability of creating a new offspring in $A_{i+1} \cup \dots \cup A_m$, provided the population contains a search point in A_i. Then the expected number of iterations of the algorithm to find the optimum is bounded from above by*

$$\sum_{i=1}^{m-1} \frac{1}{s_i}\ .$$

In contrast to other methods such as *drift analysis* [16, 21], the fitness-level method is applicable in cases where "easy" and "hard" fitness levels are mixed up, so that the progress towards the optimum cannot reasonably be bounded by a closed formula.

The fitness-level method has also been applied to other elitist optimization methods, including elitist ant colony optimizers [14, 33] and a binary particle swarm optimizer [45]. It gives rise to powerful tail inequalities [51] and it can be used to prove lower bounds as well, when combined with additional knowledge on transition probabilities [43]. Finally, Lehre [27] recently showed that the fitness-level method can be extended towards non-elitist EAs with additional mild conditions on transition probabilities and the population size.

Note that the method only requires a finite *number* of fitness-level sets, not a finite set of fitness values. So in principle the method can be applied to continuous fitness functions as well, provided a suitable discretization is made and the goal is to find the best fitness-level set. This might not be the most practical approach, though.

In the following we apply the fitness-level method to parallel EAs. For the considered EAs we assume that there is a *migration topology*, given by a directed graph. Islands represent vertices of the topology and directed edges indicate neighborhoods between the islands. We often describe undirected graphs for use as migration topology, understanding that for an undirected edge $\{u, v\}$ we have two directed edges (u, v) and (v, u). In other words, though formally the migration topology is a directed graph, we often use the language of undirected graphs to describe it.

Our methods for proving upper bounds require that the islands run elitist evolutionary algorithms. All islands create new offspring independently by mutation and/or recombination among individuals in the island. In every generation there is a chance that migration will send an individual on the current best fitness level to some target island, and that this individual will be included on the target island. This would effectively increase the fitness level of the target island to the current best level (or an even better one). For every pair of connected islands, we call this probability *transmission probability* and denote it p. Note that for any pair of islands, the mentioned transmission events are independent.

The transmission probability can model various settings, where randomness and stochasticity may be involved:

- migrations do not take place in every generation, but only probabilistically with probability p,
- islands do not automatically select individuals on the best fitness level for emigration, but there is a probability of at least p that this happens,
- similarly, islands do not automatically include immigrants on higher fitness levels, but only with probability at least p,
- during migration crossover is performed, and p is a lower bound on the probability that crossover does not disrupt the fitness of an individual on a current best fitness level (if a crossover probability p_c is used, then clearly $p \geq 1 - p_c$),
- the physical architecture suffers from transient faults and p is a lower bound on the probability that migration is executed correctly.

Of course, the transmission probability can also model any combination of the above, in which case the product of all above probabilities gives a lower bound on the transmission probability.

Most of our results also apply when instead of probabilistic migration a fixed migration interval τ is used. This is similar to a migration probability $p = 1/\tau$; in fact, it can be regarded as a derandomized or quasi-random version of probabilistic migration. With a fixed migration interval the variance in the information propagation is reduced, and all islands operate in synchronicity. Probabilistic migrations are asynchronous; this simplifies the analysis as we do not need to keep track on how much time has passed since the last migration. We expect our results for probabilistic migration to transfer to the study of migration intervals. The only notable exception is the case of a complete topology, when the migration probability is rather small (Theorem 17) as there synchronous and asynchronous migrations lead to different effects.

As elaborated above, our method is robust and it applies in various settings, and for various types of EAs simulated on the islands. In our applications for illustrating concrete speedups for test problems, we use a simple (1+1) EA for all islands. The (1+1) EA maintains a single current search point, and in each generation it creates an offspring by mutation. The offspring replaces its parent if its fitness is not worse. The resulting island model is shown in Algorithm 1.

Algorithm 1: Parallel (1+1) EA with μ islands and migration probability p

```
For all $1 \leq i \leq \mu$ choose $x^i \in \{0,1\}^n$ uniformly at random.
repeat
    For all $1 \leq i \leq \mu$ do in parallel
        Create $y^i$ by flipping each bit in $x^i$ with probability $1/n$.
        if $f(y^i) \geq f(x^i)$ then $x^i := y^i$.
        Send a copy of $x^i$ to each neighboring island, independently with prob. $p$.
        Choose $z^i$ with maximum fitness among all incoming migrants.
        if $f(z^i) \geq f(x^i)$ then $x^i := z^i$.
```

3 Proving Upper Bounds for Parallel EAs

3.1 A General Upper Bound

Now we describe how to prove upper bounds on the running time of parallel EAs. In contrast to panmictic EAs, in an island model several islands might participate in the search for improvements from the current-best fitness level. The number of islands may vary over time according to the spread of information.

The following theorem transfers upper bounds for panmictic EAs derived by the fitness-level method into upper bounds for parallel EAs in a systematic way.

Theorem 4 (Fitness-level method for parallel EAs)**.** *Consider a partition of the search space into fitness levels $A_1 <_f A_2 <_f \cdots <_f A_m$ such that A_m only contains global optima. Let s_i be (a lower bound on) the probability that a fixed island running an elitist EA creates a new offspring in $A_{i+1} \cup \cdots \cup A_m$, provided the island contains a search point in A_i. Let μ_t for $t \in \mathbb{N}$ denote (a lower bound on) the number of islands that have discovered an individual in $A_i \cup \cdots \cup A_m$ in the t-th generation after the first island has found such an individual. Then the expected parallel running time of the parallel EA on f is bounded from above by*

$$E\left(T^{\text{par}}\right) \;\leq\; \sum_{i=1}^{m-1} \sum_{t=0}^{\infty} (1 - s_i)^{\sum_{j=1}^{t} \mu_j} \;.$$

Proof. Let T_i denote the random time until the first island finds an individual on a fitness level $i+1, \dots, m$, starting with at least one individual on fitness level i in the whole population. The expected parallel running time can be written as

$$E(T^{\text{par}}) \le \sum_{i=1}^{m-1} \mathrm{E}\left(T_i\right) = \sum_{i=1}^{m-1} \sum_{t=1}^{\infty} \text{Prob}\left(T_i \ge t\right) = \sum_{i=1}^{m-1} \sum_{t=0}^{\infty} \text{Prob}\left(T_i \ge t+1\right).$$

A necessary condition for $T_i \ge t+1$ is that during all t generations after the first individual has reached fitness level i all islands are unsuccessful in finding an improvement. In the j-th of these generations there are at least μ_j islands with individuals in $A_i \cup \dots \cup A_m$. Each island is successful with probability at least s_i. Using that the islands create new offspring independently, the probability of all islands being unsuccessful is at most $(1-s_i)^{\mu_j}$. Thus,

$$\sum_{i=1}^{m-1} \sum_{t=0}^{\infty} \text{Prob}\left(T_i \ge t+1\right) \le \sum_{i=1}^{m-1} \sum_{t=0}^{\infty} \prod_{j=1}^{t} (1-s_i)^{\mu_j} = \sum_{i=1}^{m-1} \sum_{t=0}^{\infty} (1-s_i)^{\sum_{j=1}^{t} \mu_j}. \qquad \square$$

The upper bound from Theorem 4 is very general as it does not restrict the communication among the islands in any way. These aspects are hidden in the definition of the variables μ_t. When looking at one particular fitness level, say level i, we also speak of islands being *informed* if and only if they contain an individual on level i. The variable μ_t then gives the number of informed islands t generations after the first island has become informed by reaching level i.

The spread of information obviously depends on the migration topology, the migration interval, and the selection strategies used to choose migrants that are sent and how migrants are included in the population. The basic method works for all choices of these design aspects. We elaborate on these aspects and then move on to more specific scenarios where we can obtain more concrete results.

3.2 How to Deal with Migration Intervals

With a migration interval of $\tau > 1$ the μ_t-value remains fixed for periods of τ generations, unless further islands are raised towards the current best fitness level by variation. If we pessimistically ignore this effect, then we have for appropriate t that $\mu_t = \mu_{t+1} = \dots = \mu_{t+\tau-1}$. In any case we have $\mu_t \le \mu_{t+1} \le \dots \le \mu_{t+\tau-1}$, hence the sum of μ-values is at least $\sum_{j=1}^{t} \mu_j \ge \tau \sum_{j=1}^{t/\tau} \mu_{(j-1)\tau+1}$. This implies the following simplified upper bound.

Corollary 5. *For a parallel EA with migration interval τ the bound from Theorem 4 simplifies to*

$$E\left(T^{\text{par}}\right) \le \sum_{i=1}^{m-1} \sum_{t=0}^{\infty} (1-s_i)^{\tau \sum_{j=1}^{t/\tau} \mu_{(j-1)\tau+1}}.$$

The values $\mu_{(j-1)\tau}$ can be estimated like the values μ_j in a setting with $\tau = 1$. In order to keep the presentation simple, in the following applications we only consider the case that $\tau = 1$, i.e., migration happens in every generation. This reflects common principles used in fine-grained or cellular evolutionary algorithms. The following considerations can always be combined with the above arguments to handle migration intervals larger than 1.

3.3 Stochastic Communication and Finding Improvements

In order to arrive at more concrete bounds on the parallel running time for common migration topologies, we need to understand how the number of informed islands grows on each fitness level, i. e., the growth curves underlying the μ_j-variables. Note that these variables are random variables in all settings where we have a transmission probability less than 1. This means that getting a closed formula for the expected parallel running time is not easy. In Theorem 4 we cannot simply replace the μ_j-variables by their expectations as by Jensen's inequality [20] this would yield an estimation in the wrong direction (i. e., it would give a lower bound where an upper bound is needed). More work is required in order to arrive at closed formulas for common topologies.

Instead of arguing with the random number of informed islands, it is easier to argue with expected hitting times for the time until a specified number of islands is informed. If we know such expected hitting times, or upper bounds thereof, we can estimate the time until the parallel EA finds a better fitness level.

Lemma 6. *Consider an island model running elitists EAs and fix some fitness level i with success probability s_i for each island. Let $\xi(k)$ denote the random number of generations until at least k islands are informed. Then for every $k \leq \mu$ the expected time until this fitness level is left towards a better one is at most*

$$E(\xi(k)) + 1 + \frac{1}{k} \cdot \frac{1}{s_i}.$$

Proof. The following argument is similar to previous work by Witt on parent populations [49]. After $\mathrm{E}(\xi(k))$ expected generations there are at least k informed islands. Then the probability of leaving the fitness level is at least $1-(1-s_i)^k$ and the expected time is bounded from above by

$$\frac{1}{1-(1-s_i)^k} \leq 1 + \frac{1}{k} \cdot \frac{1}{s_i}, \tag{3}$$

where the inequality was published in [37, Lemma 3] and additionally stated as Lemma 19 in the appendix. Together, this proves the claim. □

A good choice for k is one where $\mathrm{E}(\xi(k)) \approx \frac{1}{k} \cdot \frac{1}{s_i}$ as this is likely to minimize the bound from Lemma 6, at least asymptotically.

The lemma ignores the fact that during the first $\xi(k)$ generations islands can already find improvements. It also ignores that the number of informed islands might grow beyond k after this time. However, we will see that for appropriate choices of k, the lemma can still give near-optimal results. In the first generations the number of informed islands is likely to be too small anyway to yield a significant benefit. In addition, after k islands have been informed this number is large enough to guarantee that improvements are found quickly, for appropriate k.

3.4 Information Propagation in Networks

It remains to estimate the first hitting time for informing a certain number of vertices. Note that this is similar to studying growth curves and takeover times. In fact, $\xi(\mu)$ is the expected time until the whole island model is informed. Growth curves and takeover times have been studied in artificial settings where no variation takes place, see [3, 10–13, 38–40] or recent surveys [29, Chapter 4], [42].

In the following, we refer to our model of transmission probabilities as it is a general model that captures many stochastic components in the dynamic behavior of island models. But at the same time it is simple enough to allow for a theoretical analysis.

Transmission probabilities give rise to a stochastic information propagation process in networks. Each informed vertex in the network independently tries to inform all its neighbors in every iteration, and information is successfully transmitted across any of these edges with probability p. This process was studied by Rowe, Mitavskiy, and Cannings [36], who considered the *propagation time* as the time until all vertices in the network are informed. They presented bounds for interesting graph classes as well as a general upper bound of

$$\frac{8\,\mathrm{diam}(G) + 8\log n}{p(1-e^{-1})}$$

for the propagation time on an undirected graph G. Thereby $\mathrm{diam}(G)$ denotes the *diameter* of G, defined as the maximum number of edges on any shortest path between two vertices in the graph.

Interestingly, the same probabilistic process also underlies the way randomized search heuristics find shortest paths in weighted undirected graphs. Doerr, Happ, and Klein [6,7] showed that the (1+1) EA can find shortest paths in graphs by simulating the Bellman-Ford algorithm. The task is to find shortest paths from a source v^* to all other vertices. For vertices whose shortest paths have few edges, shortest paths are found quickly. In our language these vertices would be called *informed*. If u is informed and the graph contains an edge $\{u, v\}$, then v can become informed with a fixed probability during a lucky mutation, if the shortest path from v^* to v contains u. This way, shortest paths propagate through the graph in the same fashion as information does. The same can be observed for ant colony optimizers [44].

Doerr, Happ, and Klein [6] independently used a different argument for bounding the expected propagation time. Fix a shortest path in the graph, leading from v^* to some fixed vertex v. In every generation there is a chance of informing the first uninformed vertex on the path, until eventually the information reaches v. If the path has at least $\log n$ edges, the time until v is informed is highly concentrated. Using tail bounds, the probability of significantly exceeding the expectation is very small. This allows us to apply a union bound for all considered vertices v.

Following the proof of [6, Lemma 3], we get the following lemma. An advantage over the general bound from [36] is that it not only bounds the propagation time for the whole network. It also bounds expected hitting times for informing smaller numbers of vertices.

Lemma 7. *Consider propagation with transmission probability p on any undirected graph where initially a single vertex v^* is informed. For $i \in \mathbb{N}_0$ let V_i contain all vertices v whose shortest path from v^* to v contains i edges. Let $n_k := \sum_{i=1}^{k} |V_i|$. The probability of not having informed n_k vertices in time $\lambda k/p$, $\lambda \geq 2$, is at most*

$$n_k \cdot \exp\left(-\frac{(\lambda-1)^2}{2\lambda} \cdot k\right) \leq n_k \cdot \exp\left(-\frac{\lambda k}{8}\right).$$

The expected time until n_k vertices are informed is at most

$$\frac{\frac{c}{c-1} \cdot \max\{2k, 8\ln(cn_k)\}}{p}$$

for every $c > 1$.

Proof. The first claim follows from the proof of Lemma 3 in [6] and the fact that $(\lambda - 1)^2/\lambda \geq \lambda/4$ for $\lambda \geq 2$.

If $(8/k) \cdot \ln(cn_k) \geq 2$ we use $\lambda := (8/k) \cdot \ln(cn_k)$ and have that after $\lambda k/p$ iterations the probability of not having informed all vertices is at most

$$n_k \cdot \exp\left(-\ln(cn_k)\right) = \frac{1}{c}.$$

If not, we repeat the argument with another phase of $\lambda k/p$ iterations. As each phase is successful with probability at least $1 - 1/c$, the expected propagation time is at most

$$\frac{1}{1-1/c} \cdot \frac{\lambda k}{p} = \frac{\frac{c}{c-1} \cdot 8\ln(cn_k)}{p}.$$

If $(8/k) \cdot \ln(cn_k) < 2$ then $k/4 > \ln(cn_k)$. The first statement with $\lambda := 2$ then gives a probability bound of

$$n_k \cdot \exp\left(-\frac{k}{4}\right) \leq n_k \cdot \exp\left(-\ln(cn_k)\right) \leq \frac{1}{c}$$

and using the same arguments as before we get a time bound of

$$\frac{1}{1-1/c} \cdot \frac{\lambda k}{p} = \frac{\frac{c}{c-1} \cdot 2k}{p}.$$

□

Note that putting $k := \mathrm{diam}(G)$ and $c = 2$, we get a bound of

$$\max\left\{\frac{4\,\mathrm{diam}(G)}{p}, \frac{16\ln(2n)}{p}\right\} \leq \frac{4\,\mathrm{diam}(G) + 11.2\log(n) + 11.2}{p}.$$

For all non-empty graphs this is better than the general upper bound

$$\frac{8\,\mathrm{diam}(G) + 8\log n}{p(1-e^{-1})} \approx \frac{12.7\,\mathrm{diam}(G) + 12.7\log n}{p}$$

from Rowe, Mitavskiy, and Cannings [36]. However, the asymptotic behavior of both bounds is the same (since $\max\{x, y\} = \Theta(x + y)$ for all $x, y \in \mathbb{R}_0^+$).

Now we are prepared to analyze parallel EAs with concrete topologies.

4 Parallel EAs with Ring Structures

We start with ring graphs as they are often used as topologies [46]. Rings can either be unidirectional, in which case there is exactly one directed cycle, or bidirectional, when all edges are undirected. The following theorem holds for both kinds of graphs, and in fact for all strongly connected graphs. Recall that a directed graph is called *strongly connected* if for every two vertices u, v there is a directed path from u to v (implying that there is also a path from v to u).

Theorem 8. *Consider an island model running elitists EAs on a function f with a fitness-level partition $A_1 <_f \cdots <_f A_m$ and success probabilities $s_1, \ldots, s_{m-1}$. Let p be (a lower bound on) the probability that a specific island on fitness level i informs a specific neighbor in the topology in one generation. The expected parallel running time on f with an unidirectional or* *bidirectional ring and μ islands—or in fact any strongly connected topology—is bounded from above by*

$$\frac{2}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}} + \frac{1}{\mu} \cdot \sum_{i=1}^{m-1} \frac{1}{s_i}\ .$$

The expected communication effort for ring graphs is by a factor of at most $2p\mu$ larger than the expected parallel time.

The shape of this formula deserves some explanation. The second term $\frac{1}{\mu} \cdot \sum_{i=1}^{m-1} \frac{1}{s_i}$ is by a factor of μ smaller than the upper bound for a single island by Theorem 3. If the latter is asymptotically tight, the second term in Theorem 8, regarded in isolation, would give a perfect linear speedup. The first term is related to the speed at which information is propagated; it reflects the time needed to bring a reasonably large number of islands to the current best fitness level. Unlike for the second term, it is independent of μ, but it depends on the transmission probability p. We do have a linear speedup if the first term $\frac{2}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}}$ asymptotically does not grow faster than the second term, again assuming that the bound for a single island is tight.

As μ grows, the second term becomes smaller, while the first term remains fixed. So if we have a linear speedup for small μ, there is a point where with growing μ the linear speedup disappears. This threshold can be easily computed by checking which value of μ gives rise to the first and second terms being of equal asymptotic order. As will be seen in the next sections, the same also holds for other migration topologies.

Proof of Theorem 8. For the unidirectional ring we have $\mathrm{E}\left(\xi\left(k\right)\right) \leq (k-1)/p$ since a new island is informed with probability at least p. As this happens independently in each generation, the expected waiting time until this happens is at most $1/p$. In fact, this argument holds for all strongly connected topologies and in particular for the bidirectional ring.

Now, if $1 \leq k := p^{1/2}/s_i^{1/2} \leq \mu$ (ignoring rounding issues), by Lemma 6 the expected number of generations on fitness level i is bounded from above by

$$\frac{k-1}{p} + 1 + \frac{1}{k} \cdot \frac{1}{s_i} \leq \frac{1}{p^{1/2} s_i^{1/2}} + \frac{1}{p^{1/2} s_i^{1/2}} = \frac{2}{p^{1/2} s_i^{1/2}}.$$

In case $p^{1/2}/s_i^{1/2} < 1$ we trivially get an upper bound of

$$\frac{1}{s_i} \leq \frac{1}{p^{1/2} s_i^{1/2}}.$$

If $p^{1/2}/s_i^{1/2} > \mu$, Lemma 6 for $k := \mu$ gives an upper bound of

$$\frac{\mu-1}{p} + 1 + \frac{1}{\mu} \cdot \frac{1}{s_i} < \frac{1}{p^{1/2} s_i^{1/2}} + \frac{1}{\mu} \cdot \frac{1}{s_i}.$$

Taking the maximum of the above upper bounds gives

$$\max\left(\frac{2}{p^{1/2} s_i^{1/2}}, \frac{1}{p^{1/2} s_i^{1/2}} + \frac{1}{\mu} \cdot \frac{1}{s_i}\right) \leq \frac{2}{p^{1/2} s_i^{1/2}} + \frac{1}{\mu} \cdot \frac{1}{s_i}\ .$$

Summing over all fitness levels proves the claim.

The claim on the expected communication effort follows from Theorem 2. □

As remarked in the proof, the bound from Theorem 8 holds for arbitrary strongly connected topologies as the unidirectional ring is a worst case for the μ_t-values. Along with Theorem 2, this also gives a general upper bound on the expected communication effort for any strongly connected topology.

For bidirectional rings we have $\mathrm{E}\left(\xi\left(k\right)\right) \leq \frac{k}{2p}$. This can be seen from applying Johannsen's drift theorem, stated in the appendix as Theorem 20, applied to the difference between k and the current number of informed vertices. If there is more than one uninformed vertex, there are always at least two vertices neighboring to informed ones. The number of informed vertices then increases by $2p$ in expectation. This means that we can use $h(1) = p$ and $h(x) = 2p$ for $x > 1$ as drift function. This decreases the constant 2 in the first term towards $\sqrt{2}$, at the expense of an additional term $m - 1$. In some settings this upper bound may be better than the upper bound from Theorem 8 for unidirectional rings; where it isn't we may still use the latter for bidirectional rings as Theorem 8 applies to all strongly connected topologies.

Also note that if $p < s_i$ then the trivial bound $1/s_i$ gives a better estimate for the time until this fitness level is left. If this holds for all fitness levels, information is propagated too slowly and our method does not give any provable speedups for the parallel model.

Contrarily, if, say, $p = \Omega(1)$, compared to a single island in a ring the expected waiting time for every fitness level can be replaced by its square root. This can yield significant speedups. We make this precise for concrete functions in the following theorem. For comparing these times with runtime bounds for the (1+1) EA we refer to Table 1.

Theorem 9. *The following holds for the parallel (1+1) EA with transmission probability at least p on a unidirectional or bidirectional ring (or any other strongly connected topology):*

- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{n}{p^{1/2}} + \frac{n \log n}{\mu}\right)$ *for* OneMax,
- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{dn^{1/2}}{p^{1/2}} + \frac{dn}{\mu}\right)$ *for every unimodal function with $d+1$ function values,*
- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{n^{k/2}}{p^{1/2}} + \frac{n^k}{\mu}\right)$ *for* Jump_k *with $k \geq 2$.*

Proof. For OneMax we choose the canonical partition $A_i := \{x \mid \text{OneMax}(x) = i\}$. The probability of increasing the current fitness from fitness level i is at least $s_i \geq (n-i) \cdot 1/(en)$ since there are $n - i$ Hamming neighbors of larger fitness and a specific Hamming neighbor is created with probability at least $1/n \cdot (1-1/n)^{n-1} \geq 1/(en)$. The second sum in Theorem 8 is

$$\frac{1}{\mu} \cdot \sum_{i=0}^{n-1} \frac{en}{n-i} = \frac{en}{\mu} \sum_{i=1}^{n} \frac{1}{i} = O\left(\frac{n \log n}{\mu}\right).$$

The first sum in Theorem 8 is

$$\begin{aligned} 2\sum_{i=0}^{n-1} \left(\frac{en}{n-i} \cdot \frac{1}{p}\right)^{1/2} &= 2\left(\frac{en}{p}\right)^{1/2} \sum_{i=1}^{n} \frac{1}{\sqrt{i}} \\ &\leq 2\left(\frac{en}{p}\right)^{1/2} \int_0^n \frac{1}{\sqrt{i}}\, \mathrm{d}i \leq 2\left(\frac{en}{p}\right)^{1/2} \cdot \sqrt{n} = O(n/p^{1/2}). \end{aligned}$$

For unimodal functions we choose a partition $A_1, \dots, A_{d+1}$ where A_i contains all search points with the i-th smallest function value. The probability of improving the fitness from level i is at least $s_i \geq 1/(en)$ because there is at least one search point in the next fitness level which is at Hamming distance one. Theorem 8 gives an upper bound of

$$2\sum_{i=1}^{d} \left(\frac{en}{p}\right)^{1/2} + \sum_{i=1}^{d} \frac{en}{\mu} \leq 2d \cdot \left(\frac{en}{p}\right)^{1/2} + \frac{den}{\mu} = O\!\left(\frac{dn^{1/2}}{p^{1/2}} + \frac{dn}{\mu}\right).$$

For Jump_k we again choose A_i to contain all search points with fitness i, leading to sets $A_1, \dots, A_n$ and A_{n+k}. The levels $A_1, \dots, A_{k-1}$ resemble those for OneMax (modulo swapping the meaning of zeros and ones), and the same estimations for the s_i apply. The same holds for levels $A_k, \dots, A_{n-1}$, where A_i corresponds to a success probability of s_{i-k} in the context of OneMax. So the expected time for getting to either of the two best fitness levels, the set $A_n \cup A_{n+k}$, is bounded by twice the above upper bound for OneMax. In case we have reached A_n, to reach the highest level A_{n+k}, a specific bit string with Hamming distance k has to be created. This has probability at least

$$s_n \geq \left(\frac{1}{n}\right)^{k} \cdot \left(1 - \frac{1}{n}\right)^{n-k} \geq \left(\frac{1}{n}\right)^{k} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{en^k} .$$

Theorem 8 and the above bound for OneMax give

$$O\!\left(\frac{n}{p^{1/2}} + \frac{n \log n}{\mu}\right) + 2\left(\frac{en^k}{p}\right)^{1/2} + \frac{en^k}{\mu} = O\!\left(\frac{n^{k/2}}{p^{1/2}} + \frac{n^k}{\mu}\right). \qquad \square$$

The speedups obtained are indeed significant, particularly for those functions where improvements are hard to find.

The proof of Theorem 9 uses well-known fitness-level partitions [8,47], and hence it simply consists of plugging in known values s_i and simplifying. This shows how easy it is to obtain results for parallel EAs based on analyses of panmictic EAs.

5 Parallel EAs with Two-Dimensional Grids and Tori

For two-dimensional grids and tori we adapt Theorem 4 in a similar manner, making an effort to get the best possible leading constant in the first term of the running time bound. We also consider applications of the resulting theorem similar to the applications for ring graphs.

Theorem 10. *Consider an island model running elitists EAs on a function f with a fitness-level partition $A_1 <_f \cdots <_f A_m$ and success probabilities $s_1, \dots, s_{m-1}$. Let p be (a lower bound on) the probability that a specific island on fitness level i informs a specific neighbor in the topology in one generation. The expected parallel running time of the island model on a grid or torus topology whose side lengths are at least $\sqrt{\mu}$ in both directions is bounded from above by*

$$\frac{3^{5/3}}{p^{2/3}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}} + \frac{1}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i} .$$

The expected communication effort for μ-vertex topologies is by a factor of at most $4p\mu$ larger than the expected parallel time.

Proof. Note that within a square area of $\sqrt{k} \times \sqrt{k}$ vertices in the graph all shortest paths between any two vertices have at most $2\sqrt{k}-2$ edges. So for every vertex in this area, the number of vertices that can be reached via up to $k' := 2\sqrt{k}-2$ edges is $n'_k \geq k$. We pessimistically assume $n'_k = k$; if not, we consider a slower propagation process where we remove edges of the graph to ensure $n'_k = k$. Applying Lemma 7 with respect to the primed variables $k' := 2\sqrt{k}-2$, $n'_k = k$ and $c = 4$ we have that for every $k \leq \mu$ the expected time until k islands are informed is bounded from above by

$$\max\left\{\frac{16/3 \cdot \sqrt{k} - 16/3}{p}, \frac{32/3 \cdot \ln(4k)}{p}\right\}.$$

We also get an upper bound of $k/(2p)$ using Johannsen's variable drift theorem [21], Theorem 20 in the appendix, as before. If there is more than one uninformed vertex, there are always at least two vertices neighboring to informed ones. So the expected number of informed vertices increases by $2p$ in expectation. Applying Johannsen's drift theorem as for the bidirectional ring gives an upper bound of $k/(2p)$. It is easy to check that the best upper bound is as follows: for all $k \in \mathbb{N}$

$$\min\left\{\frac{k}{2p}, \max\left\{\frac{16/3 \cdot \sqrt{k} - 16/3}{p}, \frac{32/3 \cdot \ln(4k)}{p}\right\}\right\} \leq \frac{6\sqrt{k}-1}{p}.$$

Now, if $1 \leq k := 3^{-2/3} \cdot (p/s_i)^{2/3} \leq \mu$ (ignoring rounding issues) by Lemma 6 the expected number of generations on fitness level i is bounded from above by

$$\frac{6\sqrt{k}-1}{p} + 1 + \frac{1}{k} \cdot \frac{1}{s_i} \leq \frac{6 \cdot 3^{-1/3}}{p^{2/3} s_i^{1/3}} + \frac{3^{2/3}}{p^{2/3} s_i^{1/3}} = \frac{3^{5/3}}{p^{2/3} s_i^{1/3}}.$$

If $3^{-2/3} \cdot (p/s_i)^{2/3} < 1$ we trivially get an upper bound of

$$\frac{1}{s_i} \leq \frac{3^{2/3}}{p^{2/3} s_i^{1/3}}.$$

If $3^{-2/3} \cdot (p/s_i)^{2/3} > \mu$, we get for $k := \mu$ an upper bound of

$$\frac{6\sqrt{\mu}-1}{p} + 1 + \frac{1}{\mu} \cdot \frac{1}{s_i} < \frac{2 \cdot 3^{2/3}}{p^{2/3} s_i^{1/3}} + \frac{1}{\mu} \cdot \frac{1}{s_i}.$$

Taking the maximum of the above upper bounds gives

$$\max\left(\frac{3^{5/3}}{p^{2/3} s_i^{1/3}}, \frac{2 \cdot 3^{2/3}}{p^{2/3} s_i^{1/3}} + \frac{1}{\mu} \cdot \frac{1}{s_i}\right) \leq \frac{3^{5/3}}{p^{2/3} s_i^{1/3}} + \frac{1}{\mu} \cdot \frac{1}{s_i}.$$

Summing over all fitness levels yields the claim.

The claim on the expected communication effort follows from Theorem 2. □

Note that the communication effort in one generation is asymptotically as large as for ring graphs, but for large p the upper bound on the parallel running time is generally smaller (or asymptotically equal, in case upper bounds are dominated by the term $1/\mu \cdot \sum_{i=1}^{m-1} 1/s_i$). If $p < 3s_i$ then again the trivial upper bound $1/s_i$ is better as then the spread of information is too slow.

Compared to a single island, in a torus the expected waiting time for every fitness level can be replaced by its third root. This leads to improved upper bounds for unimodal functions and Jump_k.

Theorem 11. *The following holds for the parallel (1+1) EA with transmission probability* p *on a grid or torus topology whose side lengths are at least* $\sqrt{\mu}$ *in both directions:*

- $E\left(T^{\text{par}}\right) = O\Big(\frac{n}{p^{2/3}} + \frac{n \log n}{\mu}\Big)$ *for* OneMax,
- $E\left(T^{\text{par}}\right) = O\Big(\frac{dn^{1/3}}{p^{2/3}} + \frac{dn}{\mu}\Big)$ *for every unimodal function with* $d+1$ *function values,*
- $E\left(T^{\text{par}}\right) = O\Big(\frac{n+n^{k/3}}{p^{2/3}} + \frac{n^k}{\mu}\Big)$ *for* Jump_k *with* $k \geq 2$.

Proof. We choose the same partitions as in the proof of Theorem 9. Note that the second terms in Theorem 8 and 10 are identical, so we only estimate the first terms and refer to Theorem 9 for the second terms.

For OneMax the first sum in Theorem 10 is

$$\begin{aligned}\frac{3^{5/3}}{p^{2/3}} \sum_{i=0}^{n-1} \left(\frac{en}{n-i}\right)^{1/3} &= \frac{3^{5/3} e^{1/3} n^{1/3}}{p^{2/3}} \sum_{i=1}^{n} \left(\frac{1}{i}\right)^{1/3} \\ &\leq \frac{3^{5/3} e^{1/3} n^{1/3}}{p^{2/3}} \int_{i=0}^{n} \left(\frac{1}{i}\right)^{1/3} \mathrm{d}i \\ &= \frac{3^{5/3} e^{1/3} n^{1/3}}{p^{2/3}} \cdot \frac{3}{2} \cdot n^{2/3} = \frac{3^{8/3}/2 \cdot e^{1/3} n}{p^{2/3}}.\end{aligned}$$

This gives an upper bound of

$$O\bigg(\frac{n}{p^{2/3}} + \frac{n \log n}{\mu}\bigg).$$

For unimodal functions Theorem 10 gives

$$\frac{3^{5/3}}{p^{2/3}} \sum_{i=1}^{d} (en)^{1/3} + \sum_{i=1}^{d} \frac{en}{\mu} \leq \frac{3^{5/3} d \cdot e^{1/3} n^{1/3}}{p^{2/3}} + \frac{den}{\mu} = O\bigg(\frac{dn^{1/3}}{p^{2/3}} + \frac{dn}{\mu}\bigg).$$

For Jump_k we get

$$O\bigg(\frac{n}{p^{2/3}} + \frac{n \log n}{\mu}\bigg) + 3^{5/3} \cdot \frac{(en^k)^{1/3}}{p^{2/3}} + \frac{en^k}{\mu} = O\bigg(\frac{n + n^{k/3}}{p^{2/3}} + \frac{n^k}{\mu}\bigg). \qquad \square$$

6 Parallel EAs with Hypercube Graphs

Hypercube graphs are popular topologies in parallel computation. In a d-dimensional hypercube each vertex has a label of d bits. Two vertices are neighboring if and only if their labels differ in exactly one bit. The number of vertices is then 2^d, and each vertex has d neighbors. The diameter of a d-dimensional hypercube is d, hence only logarithmic in the size of the graph. The small diameter implies that in many communication models information is spread rapidly, even though the degree of vertices is quite small. With regard to the propagation process investigated here, we get a small first term in the following running time bound, and still have a very moderate communication effort.

Theorem 12. *Consider an island model running elitists EAs on a function f with a fitness-level partition $A_1 <_f \cdots <_f A_m$ and success probabilities $s_1, \dots, s_{m-1}$. Let p be (a lower bound on) the probability that a specific island on fitness level i informs a specific neighbor in the topology in one generation. The expected parallel running time of the island model on a $(\log \mu)$-dimensional hypercube graph with μ islands is bounded from above by*

$$\frac{25m + 12\sum_{i=1}^{m-1} \log(\frac{1}{s_i})}{p} + \frac{1}{\mu} \cdot \sum_{i=1}^{m-1} \frac{1}{s_i}.$$

The expected communication effort is by a factor of at most $p\mu \log \mu$ larger than the expected parallel time.

Proof. In the notation of Lemma 7 we have for the hypercube and $1 \leq k \leq \log \mu$ that the number of vertices reachable with at most k edges is bounded from below by

$$n_k = \sum_{i=1}^{k} \binom{\mu}{i} \geq 2^k.$$

As in the proof of Theorem 8 we consider $n'_k = 2^k$ instead of n_k, justified by the fact that we can remove edges in the graph to slow down propagation. Invoking Lemma 7 with k, $n'_k = 2^k$, and $c = 2$, the expected time until 2^k vertices are informed is therefore at most

$$\frac{16 \ln(2 \cdot 2^k)}{p} = \frac{16 \cdot (\ln 2) \cdot (k+1)}{p} < \frac{12(k+1)}{p}.$$

By Lemma 6 the expected time on fitness level i is hence bounded, for any integer $0 \leq k \leq \log \mu$, by

$$\frac{12(k+1)}{p} + 1 + \frac{1}{2^k} \cdot \frac{1}{s_i} \leq \frac{13}{p} + \frac{12k}{p} + \frac{1}{2^k} \cdot \frac{1}{s_i}. \tag{4}$$

If $p/(12s_i) < 1$, we get a trivial upper bound of $1/s_i \leq 12/p$. If $p/(12s_i) > \mu$, which implies $d < \log(p/(12s_i)) \leq \log(1/s_i)$, we get an upper bound of

$$\frac{13 + 12d}{p} + \frac{1}{\mu} \cdot \frac{1}{s_i} < \frac{13 + 12\log(\frac{1}{s_i})}{p} + \frac{1}{\mu} \cdot \frac{1}{s_i}.$$

Otherwise, (4) is minimized for $2^k = p/(12s_i)$, leading to

$$\frac{13}{p} + \frac{12\log(\frac{p}{12s_i})}{p} + \frac{12}{p} \leq \frac{25}{p} + \frac{12\log(\frac{1}{s_i})}{p}.$$

The maximum over all these bounds is at most

$$\frac{25 + 12\log(\frac{1}{s_i})}{p} + \frac{1}{\mu} \cdot \frac{1}{s_i}.$$

Summing over all fitness levels yields the claim.

The claim on the expected communication effort follows from Theorem 2. □

Results for our example applications are as follows.

Theorem 13. *The following holds for the parallel (1+1) EA with transmission probability p on a $(\log \mu)$-dimensional hypercube:*

- $E(T^{\mathrm{par}}) = O\Big(\frac{n}{p} + \frac{n \log n}{\mu}\Big)$ *for* OneMax*,*
- $E(T^{\mathrm{par}}) = O\Big(\frac{d \log n}{p} + \frac{dn}{\mu}\Big)$ *for every unimodal function with $d+1$ function values,*
- $E(T^{\mathrm{par}}) = O\Big(\frac{n + k \log n}{p} + \frac{n^k}{\mu}\Big)$ *for* Jump_k *with $k \geq 2$.*

Proof. For OneMax we have

$$\begin{aligned}\sum_{i=0}^{n-1} \log\left(\frac{1}{s_i}\right) = \sum_{i=1}^{n} \log\left(\frac{en}{i}\right) &= \log\left(\prod_{i=1}^{n} \frac{en}{i}\right)\\ &= \log\left(\frac{e^n n^n}{n!}\right) \leq \log\left(\frac{e^n n^n}{(n/e)^n}\right) = \log\left(e^{2n}\right) = 2n \log(e).\end{aligned}$$

Theorem 12 gives an upper bound of

$$\frac{25n + 24n \log e}{p} + O\bigg(\frac{n \log n}{\mu}\bigg) = O\bigg(\frac{n}{p} + \frac{n \log n}{\mu}\bigg).$$

For unimodal functions Theorem 12 gives

$$\frac{25d + 12d \log(en)}{p} + O\bigg(\frac{dn}{\mu}\bigg) = O\bigg(\frac{d \log n}{p} + \frac{dn}{\mu}\bigg).$$

For Jump_k we get, using $\log(en^k) \leq \log(e^k n^k) = O(k \log n)$,

$$\begin{aligned}&O\bigg(\frac{n}{p} + \frac{n \log n}{\mu}\bigg) + \frac{25 + 12 \log(en^k)}{p} + O\bigg(\frac{n^k}{\mu}\bigg)\\ &= O\bigg(\frac{n + k \log n}{p} + \frac{n^k}{\mu}\bigg).\end{aligned}$$

□

If $p = \Omega(1)$, we get linear speedups for OneMax if $\mu = O(\log n)$, and linear speedups for unimodal functions where the bound $O(dn)$ for a single island is tight, if $\mu = O(n/\log n)$. For Jump_k, if $k = O(n/\log n)$ we can choose $\mu = O(n^{k-1})$ to get a linear speedup. As can be seen from Table 1 the upper bounds on the expected parallel times for LO and Jump_k are much better for the hypercube than for rings and torus graphs, if p is large.

7 Parallel EAs with Complete Topologies

Finally, we consider the densest topology, the complete graph K_μ, where every island is neighboring to every other island. The complete graph is interesting because it represents an extreme case: the largest possible communication effort with regard to one generation, but also the fastest possible spread of information.

For the special case of $p = 1$ a parallel (1+1) EA is basically equivalent to a (1+μ) EA, which creates μ offspring independently and then compares a best offspring against the current search point. The only difference is that the parallel (1+1) EA can store different individuals of the same fitness. But this issue is irrelevant when using the fitness-level method. Hence our results for a parallel (1+1) EA with a complete topology and $p = 1$ also apply to the (1+μ) EA. For $p < 1$ the two models are generally different.

We start with a simple argument. Clearly, if there is at least one informed island, each other island will become informed with probability at least p.

Theorem 14. *Consider an island model running elitists EAs on a function f with a fitness-level partition $A_1 <_f \cdots <_f A_m$ and success probabilities $s_1, \dots, s_{m-1}$. Let p be (a lower bound on) the probability that a specific island on fitness level i informs a specific neighbor in the topology in one generation. The expected parallel running time of the island model on a complete topology is*

$$E(T^{\mathrm{par}}) \leq m + \frac{2m}{p} + \frac{2}{\mu} \sum_{i=1}^{m-1} \frac{1}{s_i} .$$

The expected communication effort is by a factor of at most $p\mu(\mu - 1) < p\mu^2$ larger than the expected parallel time.

Proof. We estimate the expected time until at least $\mu/2$ islands are informed after an improvement. If more than $\mu/2$ islands are uninformed, the expected number of islands that become informed in one generation is at least $p\mu/2$. By standard drift analysis arguments [16] the desired expectation is bounded from above by $2/p$.

By Lemma 6 we then get that the expected time on fitness level i is at most

$$1 + \frac{2}{p} + \frac{2}{\mu} \cdot \frac{1}{s_i}.$$

Adding these times for all fitness levels proves the claim.

The claim on the expected communication effort follows from Theorem 2. □

As mentioned, the complete graph leads to a maximal spread of information. In comparison to the previous sections, we obtain the best upper bounds for the considered function classes. However, a maximum amount of migration takes place in each generation, so the expected total communication effort is also highest (cf. Tables 1 and 3).

Theorem 15. *Let $\mu \in \mathbb{N}$. The following holds for the expected parallel running time of the parallel (1+1) EA with topology K_μ. In the case $p = 1$, the same holds for the (1+μ) EA:*

- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{n}{p} + \frac{n \log n}{\mu}\right)$ *for* OneMax,
- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{d}{p} + \frac{dn}{\mu}\right)$ *for every unimodal function with $d + 1$ function values, and*
- $E\left(T^{\mathrm{par}}\right) = O\left(\frac{n}{p} + \frac{n^k}{\mu}\right)$ *for* Jump_k *with $k \geq 2$.*

The proof is obvious by now.

The term $2/p$ for the time until at least $\mu/2$ islands are informed is a reasonable estimate if p is large (e. g., $p = \Omega(1)$). But for small p this estimation is quite loose as we have completely neglected that *all* informed vertices have a chance to inform other islands.

We therefore also present a more detailed analysis for small p. The motivation for studying complete graphs and small p is that it captures random migration policies. Assume that each island decides randomly with probability p for each other island whether to migrate individuals to that island. Then this can be regarded as a complete topology with transmission probability p.

Values around $p = 1/\mu$ seem particularly interesting as then in each generation one migration takes place for each island in expectation. In fact, we get different results for $p > 1/\mu$ and $p < 1/\mu$.

Lemma 16. *Consider propagation with transmission probability p on the complete topology with μ vertices. Let $\xi(k)$ be as in Lemma 6, then*

$$\xi(\mu) \le \frac{8\log(\mu)}{\min(p\mu, 1)}.$$

Proof. The claim is obvious for $\mu \le 2$, so we assume $\mu \ge 3$ in the following. Let X_t denote the random number of informed vertices after t iterations. We first estimate the expected time until at least $\mu/2$ vertices become informed, and then estimate how long it takes to get from $\mu/2$ informed vertices to μ informed ones.

If $X_t = i$ each presently uninformed vertex is being informed in one iteration with probability (using Lemma 19 in the appendix)

$$1 - (1-p)^i \ge 1 - \frac{1}{1+ip} = \frac{ip}{1+ip}.$$

This holds independently from other presently uninformed vertices. In fact, the number of newly informed vertices follows a binomial distribution with parameters $\mu - i$ and $\frac{ip}{1+ip}$. The median of this binomial distribution is $i(\mu - i) \cdot \frac{p}{1+ip}$ (assuming that this is an integer), hence with probability at least $1/2$ we have at least $i(\mu - i) \cdot \frac{p}{1+ip}$ newly informed vertices in one iteration. Hence, it takes an expected number of at most 2 iterations to increase the number of informed vertices by $i(\mu - i) \cdot \frac{p}{1+ip}$, which for $i \le \mu/2$ is at least $i \cdot \frac{p\mu}{2+p\mu}$.

For every $0 \le j \le \log(\mu) - 2$ the following holds. If $i \ge 2^j$ and $i \le \mu/2$ then in an expected number of 2 generations at least $2^j \cdot \frac{p\mu}{2+p\mu}$ new vertices are informed. The expected number of iterations for informing a total of 2^j new vertices is therefore at most $2 \cdot \frac{2+p\mu}{p\mu}$. Then we have gone from at least 2^j informed vertices to at least 2^{j+1} informed vertices. Summing up all times across all j, the expected time until at least $2^{\log(\mu)-1} = \mu/2$ vertices are informed is at most

$$2(\log(\mu) - 1) \cdot \frac{2 + p\mu}{p\mu}.$$

For $p\mu \le 1$ we have $\frac{2+p\mu}{p\mu} \le 3/(p\mu)$, yielding an upper time bound of $6(\log(\mu) - 1)/(p\mu)$. Otherwise, we use $\frac{2+p\mu}{p\mu} \le 3$ to get a bound of $6(\log(\mu) - 1)$.

For the time to get from $\mu/2$ to μ informed vertices, observe that the expected number of newly informed vertices is still $\frac{i(\mu-i)p}{1+ip}$, if currently i vertices are informed. Equivalently, if i vertices are *uninformed*, the expected decrease of the number of uninformed vertices is $\frac{i(\mu-i)p}{1+(\mu-i)p}$. This function is monotone increasing if $i \le \mu/2$. Applying Johannsen's drift theorem, Theorem 20 in the appendix, for the number of uninformed nodes, using $h(i) := \frac{i(\mu-i)p}{1+(\mu-i)p}$ as drift function, gives an upper bound of

$$\begin{aligned}
&\frac{1+(\mu-1)p}{(\mu-1)p} + \int_1^{\mu/2} \frac{1+(\mu-i)p}{i(\mu-i)p}\,\mathrm{d}i \\
\le\ &\frac{1+(\mu-1)p}{(\mu-1)p} + \frac{\ln(\mu-1)(1+p\mu)}{p\mu} \\
\le\ &\frac{1+p\mu}{p\mu} + \frac{1}{\mu(\mu-1)p} + \frac{\ln(\mu-1)(1+p\mu)}{p\mu} \\
\le\ &\frac{(\ln(\mu)+1)(1+p\mu)+\frac{1}{\mu-1}}{p\mu}.
\end{aligned}$$

For $p\mu \le 1$ this is at most

$$\frac{2\ln(\mu)+2+\frac{1}{\mu-1}}{p\mu} \le \frac{2\ln(\mu)+5/2}{p\mu}.$$

Otherwise, this is at most

$$\frac{(\ln(\mu)+1)\cdot 2p\mu + \frac{p\mu}{\mu-1}}{p\mu} \le 2\ln(\mu)+5/2.$$

Together, along with $\ln(\mu) \le \log(\mu)$ this proves the claim. □

Combining Lemma 16 with Lemma 6 gives the following. Apart from an additive term m, the case of $p \le 1/\mu$ yields a bound where the first term is smaller by a factor of order $\log(\mu)/\mu$. For fairly large transmission probabilities, $p \ge 1/\mu$, in the first term we have replaced the factor $1/p$ by $\log(\mu)$. These improvements reflect that the complete graph can spread information much more quickly than previously estimated in the proof of Theorem 14.

Theorem 17. *Consider an island model running elitists EAs on a function f with a fitness-level partition $A_1 <_f \cdots <_f A_m$ and success probabilities $s_1, \dots, s_{m-1}$. Let p be (a lower bound on) the probability that a specific island on fitness level i informs a specific neighbor in the topology in one generation. The expected parallel running time of the island model on a complete topology is bounded as follows. If $p \ge 1/\mu$ we have*

$$E(T^{\mathrm{par}}) \ \le\ m + 8m\log\mu + \frac{1}{\mu}\sum_{i=1}^{m-1}\frac{1}{s_i}$$

and if $p \le 1/\mu$ we have

$$E(T^{\mathrm{par}}) \ \le\ m + \frac{8m\log\mu}{p\mu} + \frac{1}{\mu}\sum_{i=1}^{m-1}\frac{1}{s_i}\ .$$

For our example applications, the refinements in Theorem 17 result in the following refined bounds. As we only get improved upper bounds for $p = O(1/\mu)$, we do not mention the special case of the (1+μ) EA with $p = 1$.

Theorem 18. *Let $\mu \in \mathbb{N}$. The following holds for the expected parallel running time of the parallel (1+1) EA with topology K_μ:*

- $E\left(T^{\mathrm{par}}\right) = O\Big(n\log(\mu) + \frac{n\log n}{\mu}\Big)$ *for* OneMax *if* $p \geq 1/\mu$ *and*
  $E\left(T^{\mathrm{par}}\right) = O\Big(n + \frac{n\log\mu}{p\mu} + \frac{n\log n}{\mu}\Big)$ *otherwise,*

- $E\left(T^{\mathrm{par}}\right) = O\Big(d\log(\mu) + \frac{dn}{\mu}\Big)$ *for unimodal functions with* $d+1$ *values, if* $p \geq 1/\mu$*, and*
  $E\left(T^{\mathrm{par}}\right) = O\Big(d + \frac{d\log(\mu)}{p\mu} + \frac{dn}{\mu}\Big)$ *otherwise, and*

- $E\left(T^{\mathrm{par}}\right) = O\Big(n\log(\mu) + \frac{n^k}{\mu}\Big)$ *for* Jump_k *with* $k \geq 2$*, if* $p \geq 1/\mu$ *and*
  $E\left(T^{\mathrm{par}}\right) = O\Big(n + \frac{n\log(\mu)}{p\mu} + \frac{n^k}{\mu}\Big)$ *otherwise.*

8 Experiments

In order to complement the analytical results above, we also give experimental results on the behavior of island models for different topologies. As a detailed experimental evaluation is beyond the scope of this paper, we only present illustrative results for the two functions OneMax and LO.

First we investigate the parallel running time T^{par} and the communication effort for different transmission probabilities. The experiments were repeated 100 times per data point for the parallel (1+1) EA with $\mu = 64$ islands and an instance size of $n = 256$ for both example functions, varying the transmission probability p in steps of 0.01. Figure 1 shows the behavior for the topologies K_{64}, a bidirectional ring graph, an 8×8 torus graph, and a 6-dimensional hypercube.

Looking at the influence of the transmission probability on the running time, a higher transmission probability improves the running time behavior of the algorithm, also according to the expectations from our theoretical analysis. In particular, all not too small values p lead to much smaller running times compared to the pathological setting $p = 0$, where we have no communication, but μ independent runs of the (1+1) EA. This demonstrates for our functions that parallelization and migration can lead to drastic speedups. For larger or intermediate values for p the parallel running time does not vary much, as then for all topologies the running time is dominated by the second terms from our bounds: $1/\mu \cdot O(n\log n)$ and $1/\mu \cdot O(n^2)$ for OneMax and LO, respectively.

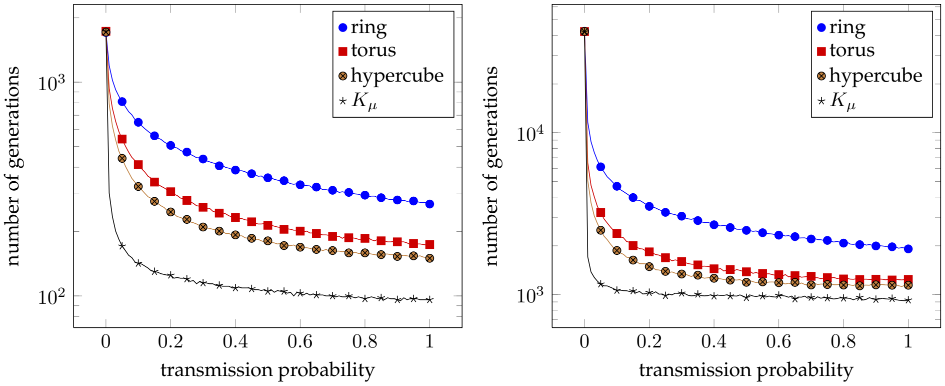
Comparing the behavior of those topologies, we see that the parallel running time indeed depends on the density of the topology, i. e., more dense topologies spread information more efficiently, which results in a faster convergence. As expected, the topology K_μ performs best, the ring graph performs worst.

We have used two-sided Mann-Whitney U tests on the data from Figure 1(a) and 1(b), and a comparison of mean ranks, to make pairwise comparisons between the topologies concerning the parallel running time. We performed separate tests for each individual data point (e. g. each tested transition probability) as this illuminates in which settings one topology is better than another[3]. For both OneMax and LO and all transmission probabilities at least 0.01, the outcome is that $K_\mu <$ hypercube $<$ torus $<$ ring on a significance level of 0.001.

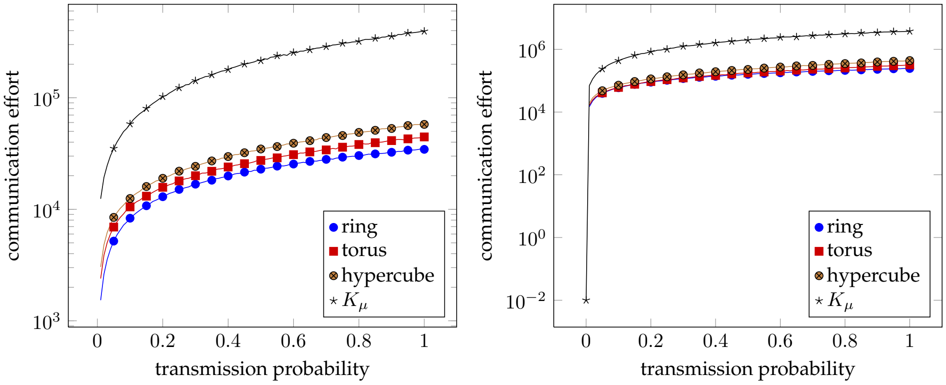
Looking at the communication effort, Figure 1(c) and 1(d), it seems that it is larger for more dense topologies, as expected. Hence, although the topology K_μ shows the

---

[3] This is particularly relevant as for the upcoming experiments in Figure 2 some of the curves cross, and then we cannot say that one topology is always better than another. The large number of tests means that we cannot exclude the possibility that some of the results may be false positives. To minimize this issue, we report results for a very low significance level of 0.001 wherever possible.



(a) OneMax (b) LO



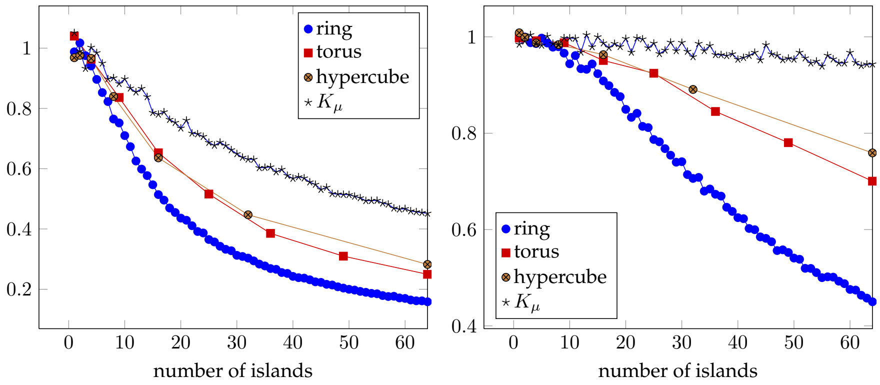
(c) OneMax (d) LO

Figure 1: Average parallel running time and communication effort for the parallel (1+1) EA using $\mu = 64$ islands and different transmission probabilities, both for OneMax and LO on $n = 256$ bits. The lines connect all data points, i. e. varying transmission probabilities in increments of 0.01; plot marks are put in increments of 0.05.

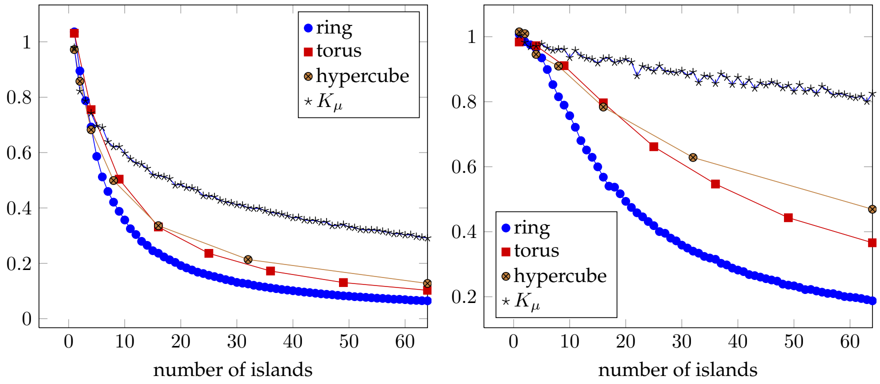
best runtime behavior, the communication effort is highest for all transmission probabilities. Interestingly, the communication effort is about the same for the other three topologies. This is in particular the case for LO, i. e., although e. g. the ring graph is more sparse, its parallel running time is higher, so that the communication effort remains similar to the hypercube and the torus graph.

Applying the Mann-Whitney U test for the communication effort and OneMax we have ring $<$ torus $<$ hypercube $< K_\mu$ for all transmission probabilities at least 0.01 on a significance level of 0.001. Looking at LO, the level of significance is slightly less when comparing ring and torus. For the communication effort and LO, the relation ring $<$ torus for a transmission probability of at least 0.01 only holds on a significance level of 0.05. Additionally, there are three exceptions: for transmission probabilities 0.13, 0.14, and 0.17 results were not significant.

Next we investigate the impact of the number of islands on the performance, with regard to different topologies and transmission probabilities, see Figures 2(a) and 2(b) for a transmission probability $p = 1.0$ and Figures 2(c) and 2(d) for a transmission



(a) Efficiency for OneMax with $p = 1.0$ (b) Efficiency for LO with $p = 1.0$



(c) Efficiency for OneMax with $p = 0.1$ (d) Efficiency for LO with $p = 0.1$

Figure 2: Efficiency for the parallel (1+1) EA with transmission probabilities $p \in \{0.1, 1\}$ for $\mu \in \{1, \dots, 64\}$ numbers of islands.

probability $p = 0.1$. As the parallel running time shows a steep decrease, we plot the *efficiency* instead, defined as

$$\frac{T^{\text{seq}}}{T^{\text{par}} \cdot \mu} \,.$$

It can be regarded as a normalized version of speedup, normalized by the number of islands. Small efficiencies indicate small speedups, large efficiencies indicate good speedups. An efficiency of 1 corresponds to a perfect linear speedup.

Again, the instance size of the benchmark functions was set to $n = 256$ and the number of islands μ was chosen from 1 to 64. Only square torus graphs were used. So our torus graphs and hypercubes are only defined for square numbers and powers of 2, respectively, leading to fewer data points.

For lower numbers of islands the efficiency of the algorithm is better than for larger numbers of islands. This is expected as a single (1+1) EA, i. e., our setting with $\mu = 1$ minimizes the number of function evaluations for both OneMax and LO [43], among all EAs that only use standard bit mutation. This excludes superlinear speedups on OneMax and LO, for such EAs, from a theoretical perspective.

It can be seen that more dense topologies are more efficient than sparse topologies. In accordance with our theoretical analyses, the efficiency decreases more rapidly for OneMax. For OneMax, and $p = \Omega(1)$, only values $\mu = O(\log n)$ were guaranteed to give a linear speedup. And indeed the efficiency in Figure 2(c) degrades quite quickly for OneMax and $p = 1.0$.

Higher numbers of islands are still efficient for LO. For the ring, the range of good μ-values is up to $\mu = O(\sqrt{n})$. This is reflected in Figure 2(d) as the efficiency degrades as μ increases beyond $\sqrt{n} = 16$. For denser topologies the efficiency only degrades for large μ. The complete graph remains effective throughout the whole scale—even stronger, for values up to $\mu = 256$ (not shown in Figure 2) the efficiency was always above 0.75. This was also expected as $\mu = \Theta(n)$ still guarantees a linear speedup for LO.

Comparing the running time behavior for different transmission probabilities, the plots confirm again that in our examples a higher transmission probability for individuals allows for a better overall performance.

Also for the results in Figure 2 pairwise two-sided Mann-Whitney U tests were performed as before. For the efficiency the mean ranks in each setup indicate the ordering ring < torus < hypercube < K_μ, but with different levels of significance. In Table 2 we list those numbers of islands for which the pairwise comparisons are statistically significant on a significance level of 0.001. Note that torus and hypercube by definition only share very few data points (squares of powers of two): $\mu \in \{1, 4, 16, 64\}$. For very small values of μ results are not significant, as the topologies are too similar and hence show indistinguishable performance. But for larger topologies, that is, $\mu > 16$, all comparisons are indeed significant on a very low level of 0.001.

		Torus	Hypercube	Complete
OneMax, 0.1	Ring	$\mu > 1$	$\mu > 4$	$\mu > 4$
OneMax, 0.1	Torus	-	$\mu > 16$	$\mu > 4$
OneMax, 0.1	Hypercube	-	-	$\mu > 1$
LO, 0.1	Ring	$\mu > 4$	$\mu > 4$	$\mu > 5$
LO, 0.1	Torus	-	$\mu > 16$	$\mu > 4$
LO, 0.1	Hypercube	-	-	$\mu > 1$
OneMax, 1.0	Ring	$\mu > 4$	$\mu > 4$	$\mu > 7$
OneMax, 1.0	Torus	-	$\mu > 16$	$\mu > 9$
OneMax, 1.0	Hypercube	-	-	$\mu > 8$
LO, 1.0	Ring	$\mu > 16$	$\mu > 16$	$\mu > 8$
LO, 1.0	Torus	-	$\mu > 16$	$\mu > 16$
LO, 1.0	Hypercube	-	-	$\mu > 16$

Table 2: A summary of test results regarding pairwise comparisons between the different migration topologies with Mann-Whitney U tests, according to the efficiency shown in Figure 2. Here the lines of the table are for different fitness functions and migration probabilities (0.1 or 1.0) and a line-column combination describes the number of islands of the model, where the Mann-Whitney U test was significant on a significance level of 0.001.

9 Conclusions

We have provided a general method for the running time analysis of parallel evolutionary algorithms, including applications to a set of well-known and illustrative example functions. Our method provides a way of automatically transforming running time bounds obtained for panmictic EAs to parallel EAs with spatial structures. In addition to a general result, we have provided methods tailored towards specific topologies: ring graphs, torus graphs, hypercubes and complete graphs. The latter also covers offspring populations and random migration topologies as special cases. Our results can estimate the expected parallel running time from above, thus lower-bounding the speedup obtained through parallelization with regard to the number of generations. They also bound the expected total communication effort in terms of the total number of individuals migrated as an indicator of the bandwidth used.

Our example applications revealed insights which are remarkable in their own right, see Table 1 and a more general version in Table 3. Compared to upper bounds obtained for a single panmictic island by the fitness-level method, for ring graphs the expected waiting time for an improvement can be replaced by its square root in the parallel running time, provided the number of islands is large enough and improvements are transmitted efficiently, i. e., $p = \Omega(1)$. This leads to a speedup of order $\log n$ for OneMax and of order $\sqrt{n}$ for some unimodal functions, such as LO. On Jump_k the speedup is even of order at least $n^{k/2}$. A similar effect is observed for torus graphs where the expected waiting time can be replaced by its third root. The hypercube reduces the (upper bound on the) expected waiting time on each level to its logarithm, and on the complete graph it is reduced to a constant, again provided there are sufficiently many islands. This way, even on functions like LO and Jump_k ($3 \leq k = O(n/\log n)$) the expected parallel time can be reduced to $O(n)$. In all these results the population size can be chosen in such a way that the total number of function evaluations does not increase, in an asymptotic sense. The population sizes leading to best possible upper bounds on the parallel running time have been stated explicitly (cf. Tables 1 and 3), therefore giving hints on how to parametrize parallel EAs.

The tables also reveal that in certain situations there is a tradeoff between the expected parallel time and the communication effort—at least with regard to the upper bounds shown here. For instance, on LO the torus graph has the smallest communication effort of $O(n^2)$ at the expense of a higher parallel time bound of $O(n^{4/3})$. The complete graph has the smallest bound for the parallel time, $O(n)$, but the largest bound for the communication effort: $O(n^3)$. The hypercube provides a good compromise, combining the smallest bounds up to polylogarithmic factors. A similar observation can be made for Jump_k, but there the hypercube is the better choice than the complete graph (strictly better in terms of communication effort and equally good in the parallel time bound). In all our examples the ring never gave better upper bounds than torus graphs in both objectives.

We also gave experimental results for the parallel (1+1) EA on OneMax and LO with the four topologies ring, torus, hypercube, and complete graph (in order of increasing density). These results and statistical tests confirm that, also in a non-asymptotic sense, in almost all parameters settings the following holds. Going from sparse to dense topologies, the parallel running time of the parallel (1+1) EA decreases, but the communication effort increases.

Future work should deal with lower bounds on the running time of parallel evolutionary algorithms to establish in which cases our upper bounds are asymptotically tight. It might be possible to extend drift analysis methods [16, 21] in such a way that both the current best fitness and the number of islands on the current best fitness level are considered; a similar strategy was used for the analysis of population-based evolutionary algorithms by Lehre and Yao [28]. Also in our example functions no diversity was needed. Further studies are needed in order to better understand how the topology and the parameters of migration affect diversity, and how diversity helps for optimizing more difficult, multimodal problems.

Acknowledgments

The authors were supported by postdoctoral fellowships from the German Academic Exchange Service while visiting the International Computer Science Institute, Berkeley, CA, USA. Dirk Sudholt was also supported by EPSRC grant EP/D052785/1. We would like to thank the anonymous reviewers for their comments that helped to improve the paper.

References

[1] E. Alba. Parallel evolutionary algorithms can achieve super-linear performance. *Information Processing Letters*, 82(1):7–13, 2002.

[2] E. Alba. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley-Interscience, 2005.

[3] E. Alba and G. Luque. Growth curves and takeover time in distributed evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 3102 of *LNCS*, pages 864–876. Springer, 2004.

[4] A. Auger and B. Doerr, editors. *Theory of Randomized Search Heuristics – Foundations and Recent Developments*. Number 1 in Series on Theoretical Computer Science. World Scientific, 2011.

[5] E. Cantú Paz. A survey of parallel genetic algorithms. Technical report, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign, Urbana, IL, 1997.

[6] B. Doerr, E. Happ, and C. Klein. A tight analysis of the (1+1)-EA for the single source shortest path problem. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pages 1890–1895. IEEE Press, 2007.

[7] B. Doerr, E. Happ, and C. Klein. Tight analysis of the (1+1)-EA for the single source shortest path problem. *Evolutionary Computation*, 19(4):673–691, 2011.

[8] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[9] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4):617–633, 2010.

[10] M. Giacobini, E. Alba, A. Tettamanzi, and M. Tomassini. Selection intensity in cellular evolutionary algorithms for regular lattices. *IEEE Transactions on Evolutionary Computation*, 9:489–505, 2005.

[11] M. Giacobini, E. Alba, and M. Tomassini. Selection intensity in asynchronous cellular evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '03)*, pages 955–966. Springer, 2003.

[12] M. Giacobini, M. Tomassini, and A. Tettamanzi. Modelling selection intensity for linear cellular evolutionary algorithms. In *Proceedings of the Sixth International Conference on Artificial Evolution, Evolution Artificielle*, pages 345–356. Springer, 2003.

[13] M. Giacobini, M. Tomassini, and A. Tettamanzi. Takeover time curves in random and small-world structured populations. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pages 1333–1340. ACM Press, 2005.

[14] W. J. Gutjahr and G. Sebastiani. Runtime analysis of ant colony optimization with best-so-far reinforcement. *Methodology and Computing in Applied Probability*, 10:409–433, 2008.

[15] J. He and X. Yao. Towards an analytic framework for analysing the computation time of evolutionary algorithms. *Artificial Intelligence*, 145(1–2):59–97, 2003.

[16] J. He and X. Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35, 2004.

[17] C. Horoba. Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem. *Evolutionary Computation*, 18(3):357–381, 2010.

[18] T. Jansen. *Analyzing Evolutionary Algorithms – The Computer Science Perspective.* Springer, 2013.

[19] T. Jansen and I. Wegener. On the analysis of evolutionary algorithms—a proof that crossover really can help. *Algorithmica*, 34(1):47–66, 2002.

[20] J. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30:175–193, 1906.

[21] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik, 2010.

[22] T. Kötzing, D. Sudholt, and M. Theile. How crossover helps in pseudo-Boolean optimization. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 989–996. ACM Press, 2011.

[23] J. Lässig and D. Sudholt. The benefit of migration in parallel evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 1105–1112. ACM Press, 2010.

[24] J. Lässig and D. Sudholt. General scheme for analyzing running times of parallel evolutionary algorithms. In *11th International Conference on Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238 of *LNCS*, pages 234–243. Springer, 2010.

[25] J. Lässig and D. Sudholt. Adaptive population models for offspring populations and parallel evolutionary algorithms. In *Proceedings of the 11th Workshop on Foundations of Genetic Algorithms (FOGA 2011)*, pages 181–192. ACM Press, 2011.

[26] J. Lässig and D. Sudholt. Analysis of speedups in parallel evolutionary algorithms for combinatorial optimization. In *22nd International Symposium on Algorithms and Computation (ISAAC 2011)*, volume 7074 of *LNCS*, pages 405–414. Springer, 2011.

[27] P. K. Lehre. Fitness-levels for non-elitist populations. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO '11)*, pages 2075–2082. ACM Press, 2011.

[28] P. K. Lehre and X. Yao. On the impact of mutation-selection balance on the runtime of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 16(2):225–241, 2012.

[29] G. Luque and E. Alba. *Parallel Genetic Algorithms–Theory and Real World Applications*, volume 367 of *Studies in Computational Intelligence*. Springer, 2011.

[30] A. Mambrini, D. Sudholt, and X. Yao. Homogeneous and heterogeneous island models for the set cover problem. In *Parallel Problem Solving from Nature (PPSN 2012)*, volume 7491 of *LNCS*, pages 11–20. Springer, 2012.

[31] N. Nedjah, L. de Macedo Mourelle, and E. Alba. *Parallel Evolutionary Computations*. Springer, May 2006.

[32] F. Neumann, P. S. Oliveto, G. Rudolph, and D. Sudholt. On the effectiveness of crossover for migration in parallel evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 1587–1594. ACM Press, 2011.

[33] F. Neumann, D. Sudholt, and C. Witt. Analysis of different MMAS ACO algorithms on unimodal functions and plateaus. *Swarm Intelligence*, 3(1):35–68, 2009.

[34] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Springer, 2010.

[35] P. S. Oliveto, J. He, and X. Yao. Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3):281–293, 2007.

[36] J. Rowe, B. Mitavskiy, and C. Cannings. Propagation time in stochastic communication networks. In *Second IEEE International Conference on Digital Ecosystems and Technologies*, pages 426–431, 2008.

[37] J. Rowe and D. Sudholt. The choice of the offspring population size in the (1,λ) EA. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 1349–1356, 2012.

[38] G. Rudolph. On takeover times in spatially structured populations: Array and ring. In *Proceedings of the 2nd Asia-Pacific Conference on Genetic Algorithms and Applications*, pages 144–151. Global-Link Publishing Company, 2000.

[39] G. Rudolph. Takeover time in parallel populations with migration. In *BIOMA 2006*, pages 63–72, 2006.

[40] J. Sarma and K. De Jong. An analysis of local selection algorithms in a spatially structured evolutionary algorithm. In *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 181–186. Morgan Kaufmann, 1997.

[41] Z. Skolicki and K. De Jong. The influence of migration sizes and intervals on island models. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 1295–1302. ACM, 2005.

[42] D. Sudholt. Parallel evolutionary algorithms. In *Handbook of Computational Intelligence*. Springer. To appear.

[43] D. Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 17(3):418–435, 2013.

[44] D. Sudholt and C. Thyssen. Running time analysis of ant colony optimization for shortest path problems. *Journal of Discrete Algorithms*, 10:165–180, 2012.

[45] D. Sudholt and C. Witt. Runtime analysis of a binary particle swarm optimizer. *Theoretical Computer Science*, 411(21):2084–2100, 2010.

[46] M. Tomassini. *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time*. Springer, 2005.

[47] I. Wegener. Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In R. Sarker, X. Yao, and M. Mohammadian, editors, *Evolutionary Optimization*, pages 349–369. Kluwer, 2002.

[48] C. Witt. Worst-case and average-case approximations by simple randomized search heuristics. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science (STACS '05)*, volume 3404 of *LNCS*, pages 44–56. Springer, 2005.

[49] C. Witt. Runtime analysis of the (μ+1) EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.

[50] Y. Yu, X. Yao, and Z.-H. Zhou. On the approximation ability of evolutionary optimization with application to minimum set cover. *Artificial Intelligence*, 180–181(0):20–33, 2012.

[51] D. Zhou, D. Luo, R. Lu, and Z. Han. The use of tail inequalities on the probable computational time of randomized search heuristics. *Theoretical Computer Science*, 436(0):106 – 117, 2012.

A Appendix

The following inequality was brought to our attention by Jon Rowe. A proof is found in [37, Lemma 3].

Lemma 19. *For any $0 \le x \le 1$, and any $n > 0$*

$$(1-x)^n \le \frac{1}{1+nx}.$$

We also state Johannsen's variable drift theorem [21], in a version with slightly improved conditions [37].

Theorem 20 (Johannsen's Variable Drift Theorem [21, 37])**.** *Consider a stochastic process $\{X\}_{t \ge 0}$ on $\{0, 1, \ldots, m\}$, with $m \in \mathbb{N}$. Suppose there is a monotonic increasing function $h : \mathbb{R}^+ \to \mathbb{R}^+$ such that the function $1/h(x)$ is integrable on $\{1, \ldots, m\}$, and with*

$$E\left(X_t - X_{t+1} \mid X_t = k\right) \ge h(k)$$

for all $k \in \{1, \ldots, m\}$. Then the expected first hitting time of state 0 is at most

$$\frac{1}{h(1)} + \int_1^m \frac{1}{h(x)}\,\mathrm{d}x.$$

	(1+1) EA	Ring	Grid/Torus	Hypercube	Complete/K_μ	K_μ with $p = O(1/\mu)$
OneMax						
E (T^{par})	$\Theta(n \log n)$	$O\left(\frac{n}{p^{1/2}} + \frac{n \log n}{\mu}\right)$	$O\left(\frac{n}{p^{2/3}} + \frac{n \log n}{\mu}\right)$	$O\left(\frac{n}{p} + \frac{n \log n}{\mu}\right)$	$O\left(\frac{n}{p} + \frac{n \log n}{\mu}\right)$	$O\left(\frac{n \log \mu}{p\mu} + \frac{n \log n}{\mu}\right)$
E (T^{seq})	$\Theta(n \log n)$	$O\left(\frac{\mu n}{p^{1/2}} + n \log n\right)$	$O\left(\frac{\mu n}{p^{2/3}} + n \log n\right)$	$O\left(\frac{\mu n}{p} + n \log n\right)$	$O\left(\frac{\mu n}{p} + n \log n\right)$	$O\left(\frac{\mu (\log \mu) n}{p\mu} + n \log n\right)$
E (T^{com})	0	$O\left(p^{1/2}\mu n + pn \log n\right)$	$O\left(p^{1/3}\mu n + pn \log n\right)$	$O(\mu n (\log \mu) + p(\log \mu) n \log n)$	$O\left(\mu^2 n + p\mu n \log n\right)$	$O(\mu(\log \mu)n + p\mu n \log n)$
LO						
E (T^{par})	$\Theta(n^2)$	$O\left(\frac{n^{3/2}}{p^{1/2}} + \frac{n^2}{\mu}\right)$	$O\left(\frac{n^{4/3}}{p^{2/3}} + \frac{n^2}{\mu}\right)$	$O\left(\frac{n \log n}{p} + \frac{n^2}{\mu}\right)$	$O\left(\frac{n}{p} + \frac{n^2}{\mu}\right)$	$O\left(\frac{n \log \mu}{p\mu} + \frac{n^2}{\mu}\right)$
E (T^{seq})	$\Theta(n^2)$	$O\left(\frac{\mu n^{3/2}}{p^{1/2}} + n^2\right)$	$O\left(\frac{\mu n^{4/3}}{p^{2/3}} + n^2\right)$	$O\left(\frac{\mu n \log n}{p} + n^2\right)$	$O\left(\frac{\mu n}{p} + n^2\right)$	$O\left(\frac{\mu (\log \mu) n}{p\mu} + n^2\right)$
E (T^{com})	0	$O\left(p^{1/2}\mu n^{3/2} + pn^2\right)$	$O\left(p^{1/3}\mu n^{4/3} + pn^2\right)$	$O\left(\mu n (\log n)(\log \mu) + p(\log \mu) n^2\right)$	$O\left(\mu^2 n + p\mu n^2\right)$	$O\left(\mu(\log \mu)n + p\mu n^2\right)$
unimodal						
E (T^{par})	$O(dn)$	$O\left(\frac{dn^{1/2}}{p^{1/2}} + \frac{dn}{\mu}\right)$	$O\left(\frac{dn^{1/3}}{p^{2/3}} + \frac{dn}{\mu}\right)$	$O\left(\frac{d \log n}{p} + \frac{dn}{\mu}\right)$	$O\left(\frac{d}{p} + \frac{dn}{\mu}\right)$	$O\left(\frac{d \log \mu}{p\mu} + \frac{dn}{\mu}\right)$
E (T^{seq})	$O(dn)$	$O\left(\frac{d\mu n^{1/2}}{p^{1/2}} + dn\right)$	$O\left(\frac{d\mu n^{1/3}}{p^{2/3}} + dn\right)$	$O\left(\frac{d\mu \log n}{p} + dn\right)$	$O\left(\frac{d\mu}{p} + dn\right)$	$O\left(\frac{d \log \mu}{p\mu} + dn\right)$
E (T^{com})	0	$O\left(p^{1/2}d\mu n^{1/2} + pdn\right)$	$O\left(p^{1/3}d\mu n^{1/3} + pdn\right)$	$O(d\mu (\log n)(\log \mu) + p(\log \mu) dn)$	$O\left(d\mu^2 + pd\mu n\right)$	$O(d\mu \log \mu + p\mu dn)$
Jump_k						
E (T^{par})	$\Theta(n^k)$	$O\left(\frac{n^{k/2}}{p^{1/2}} + \frac{n^k}{\mu}\right)$	$O\left(\frac{n+n^{k/3}}{p^{2/3}} + \frac{n^k}{\mu}\right)$	$O\left(\frac{n}{p} + \frac{n^k}{\mu}\right)$	$O\left(\frac{n}{p} + \frac{n^k}{\mu}\right)$	$O\left(\frac{n \log \mu}{p\mu} + \frac{n^k}{\mu}\right)$
E (T^{seq})	$\Theta(n^k)$	$O\left(\frac{\mu n^{k/2}}{p^{1/2}} + n^k\right)$	$O\left(\frac{\mu(n+n^{k/3})}{p^{2/3}} + n^k\right)$	$O\left(\frac{\mu n}{p} + n^k\right)$	$O\left(\frac{\mu n}{p} + n^k\right)$	$O\left(\frac{n \log \mu}{p} + n^k\right)$
E (T^{com})	0	$O\left(p^{1/2}\mu n^{k/2} + pn^k\right)$	$O\left(p^{1/3}\mu n^{k/3} + pn^k\right)$	$O\left(\mu n (\log \mu) + p(\log \mu) n^k\right)$	$O\left(\mu^2 n + p\mu n^k\right)$	$O\left(\mu(\log \mu)n + p\mu n^k\right)$

Table 3: Asymptotic bounds on expected parallel (T^{par}, number of generations) and sequential (T^{seq}, number of function evaluations) running times and expected communication efforts (T^{com}, total number of migrated individuals) for various n-bit functions and island models with μ islands running the (1+1) EA and using migration probability p. The number of islands μ was always chosen to give the best possible upper bound on the parallel running time, while not increasing the upper bound on the sequential running time by more than a constant factor. For unimodal functions $d+1$ denotes the number of function values. See [8] for bounds for the (1+1) EA. Results for Jump_k were restricted to $3 \leq k = O(n/\log n)$ for simplicity.