Soft Comput (2013) 17:1121–1144
DOI 10.1007/s00500-013-0991-0

FOCUS

# Design and analysis of migration in parallel evolutionary algorithms

Jörg Lässig · Dirk Sudholt

Published online: 1 May 2013

**Abstract** Parallelization is becoming a more important issue for solving difficult optimization problems. Island models combine phases of independent evolution with migration where genetic information is spread out to neighbored islands. This can lead to an increased diversity within the population. Despite many successful applications, the reasons behind the success of island models are not well understood. We perform a first rigorous runtime analysis for island models and construct a function where phases of independent evolution as well as communication among the islands are essential. A simple island model with migration finds a global optimum in polynomial time, while panmictic populations as well as island models without migration need exponential time, with very high probability. Our results lead to new insights into the usefulness of migration, how information is propagated in island models, and how to set parameters such as the migration interval. This is a novel contribution to the theoretical foundation of parallel EAs. Further, we provide empirical results that complement the theoretical results, investigate the robustness with respect to the choice of the migration interval and compare various migration topologies using statistical tests.

**Keywords** Parallel evolutionary algorithms · Island model · Migration · Distributed evolutionary algorithms · Spatial structures · Multi-deme model · Runtime analysis

## 1 Introduction

Evolutionary algorithms (EAs) are one of the most powerful tools to heuristically solve optimization problems for which no efficient methods are known. There is an ample number of successful applications in many different fields. EAs are generally able to find good solutions in a reasonable amount of time, but as they are applied to harder and bigger problems, there is an increase in the time required to find adequate solutions. As a consequence, there have been multiple efforts to make EAs faster, and one of the most promising choices is to use the parallel implementations (Cantú Paz 1997; Cantú-Paz and Goldberg 1999; Tomassini 2005). Due to multi-core architectures in the current CPUs and GPUs, these efforts are highly relevant today (Alba 2005; Luque and Alba 2011). Parallel implementations can tap on to these resources to exploit the full computational power of current machines. Parallelization also plays an important role in computational grids, including cloud computing. Using many machines in parallel is the method of choice when faced with a problem where solutions need to be obtained in real time.

Parallelization can be used in various ways. The most obvious way is to perform operations like function evaluations in parallel. This makes sense as function evaluations are often the computationally most expensive operations.

Communicated by E. Alba.

This manuscript is partly based on preliminary results from Lässig and Sudholt (2010a, b).

J. Lässig
Department of Electrical Engineering and Computer Science,
University of Applied Sciences Zittau/Görlitz, Görlitz, Germany
e-mail: jlaessig@hszg.de

D. Sudholt (✉)
Department of Computer Science, University of Sheffield,
Sheffield, UK
e-mail: d.sudholt@sheffield.ac.uk

This kind of parallelization does not alter the behavior of the underlying sequential algorithm.

Another popular approach is to parallelize evolution itself. Instead of evolving one large population, the population is split into smaller subpopulations. These subpopulations evolve independently for certain periods of time and they periodically exchange solutions through a process called *migration*. This model is known as *island model*, the subpopulations are called *islands*. Other common names are *multi-deme* models, the subpopulations being referred to as *demes*, *distributed EAs*, or *coarse-grained* parallel EAs.

Using such a coarse-grained parallelization can have several advantages. First of all, parallelization in this approach requires very little overhead, compared to parallelizing function evaluations, because the amount of communication between different machines is very low. Furthermore, the effort of managing a small population can be much lower than the effort of managing a large, panmictic population. Some operations require time that grows larger than linear with the population size. It is also more likely that a small population fits into a cache than a big one. In fact, for evolutionary algorithms speed-ups, in the size of the population or even superlinear speed-ups have been reported (Alba 2002; He and Yao 2006). The latter means that the total execution time across all machines is lower for the parallel algorithm than that for its sequential counterpart.

Besides this obvious advantage to speed-up computation, also other effects motivate parallelization. One of them is diversity. In coarse-grained approaches, the subpopulations evolve independently for certain periods of time. While specific diversity preserving mechanisms such as deterministic crowding or fitness sharing can be provably effective for panmictic populations (Friedrich et al. 2009), this mechanism is directly built-in for these parallel EAs.

Despite the wide-spread use of parallel EAs, the theoretical understanding of the dynamics in a parallel EA is very limited. Even the role of basic parameters is not well understood (Skolicki and De Jong 2005). Leading researchers in the field have, therefore, called out for more fundamental research on parallel EAs (Luque and Alba 2011; Skolicki and De Jong 2005).

In the most recent text book by Luque and Alba (2011, p. 25), developing theoretical issues for parallel GAs (pGAs) is mentioned as a promising research line:

> Improving the formal explanations on the influence of parameters on the convergence and search of pGAs will endow the research community with tools allowing to analyze, understand, and customize a GA family for a given problem.

One theoretical research area is the consideration of takeover times (Rudolph 2000, 2006; Giacobini et al. 2003, 2005a, b; Luque and Alba 2010). In this setting, one considers evolutionary algorithms that only perform selection and migration, without genetic variation operators such as mutation or crossover. The population is initialized with a single best individual and several worse individuals. The takeover time is then defined as the time until all individuals represent copies of the best individual. A good survey is given in Luque and Alba (2011, Chapter 4). While takeover times are a useful indicator for the spread of information in a parallel EA, this line of research does not answer how real evolutionary algorithms behave when all genetic components—selection, migration and variation—come together.

Astonishingly, though parallel EAs have been known for decades, there is no rigorous theoretical analysis that considers the runtime of a complete coarse-grained parallel evolutionary algorithm with a non-trivial migration topology and, hence, goes beyond takeover times. In the paper at hand, we provide such an analysis. We demonstrate the usefulness of parallel models with migration from a theoretical perspective and investigate the role of parameters such as the migration topology and the migration interval.

To this end, we construct a problem where parallel EAs provably perform better than the panmictic populations. The problem provides a structure such that phases of independent evolution as well as communication between islands are necessary to find a global optimum. We also want the performance difference to be as large as possible to demonstrate how drastic the results can be. We consider a simple island model with a mild amount of migration and compare it against a panmictic population of the same size. Furthermore, we compare the island model against a parallel algorithm with separated subpopulations without communication. This is done to exclude that the good performance of the island model is simply due to multiple independent runs of the same algorithm.

For the considered problem, the following holds:

- Panmictic populations need exponential time to find an optimal solution with overwhelming probability.
- Separate subpopulations need exponential time to find an optimal solution with overwhelming probability.
- An island model with properly configured migration needs only polynomial time to find an optimal solution with overwhelming probability.

The negative results hold with such a high probability that also restart strategies cannot help to find a global optimum in less than exponential time. Further, the island model is only effective if migration is parametrized correctly. We also show that inappropriate parameter settings lead to exponential running times. If the migration interval is too small, the island model behaves like a panmictic population. It is then prone to genetic drift and quick loss of diversity. If the migration interval is too large, many islands get stuck in local optima.

Our results and analyses give insight into the dynamic behavior of island models. This allows for immediate conclusions about a proper choice of the migration interval. These conclusions do not only apply to the considered problem. Many arguments generalize to other problems. For example, in Sect. 5.1 we examine how long the period of independent evolution must be so that islands can gain diversity in between migrations and evolve in different directions. We also find how small the migration interval can be before an island model behaves like a panmictic EA (see Sect. 6).

The methods developed in this work will prove useful in further studies of parallel EAs and pave the way for a theoretical foundation with direct implications for practice.

Our theoretical results are complemented by detailed experimental studies that help to find the optimal migration interval and the best topology for the problem at hand.

### 1.1 Related work

We review theoretical work on parallelization. Offspring populations can be considered a basic form of parallelization. Jansen et al. (2005) were the first to analyze the expected running time of a $(1+\lambda)$ EA, i.e., a variant of the (1+1) EA comparing the parent against the best out of $\lambda$ offspring created independently. They analyzed the expected number of generations needed to find a global optimum for well-known test functions as OneMax and LO. They also constructed example functions with exponential performance gaps between the (1+1) EA and the $(1+\lambda)$ EA. This shows that offspring populations can be either helpful or detrimental, depending on the problem at hand. In terms of the number of function evaluations, they proved that on OneMax and LO offspring populations can only increase the expected number of function evaluations. Furthermore, they presented an adaptive scheme for choosing the offspring population size and accompanied this by experimental results.

Offspring populations were also considered by Jägersküpper and Storch (2007), with a focus on comma strategies. They considered the $(1,\lambda)$ EA which takes over the best offspring as new current search point, even if it is worse than its parent. Their results show that a minimum offspring population size of $\lambda = \Omega(\log n)$ is needed to optimize any function with a unique global optimum in less than exponential time.

Parallelization, in the form of offspring populations, has also been considered for evolution strategies in continuous spaces by Teytaud and Teytaud (2010).

Watson and Jansen (2007) defined a “royal road function” for crossover, i.e., a function where crossover is essential. Their function contains a clear building-block structure. To get the diversity needed to evolve and recombine all building blocks, they used a simple island model with a topology and dubbed it *single-receiver model*. There is a single island that receives immigrants from all other islands. The other islands run independently as they do not receive any immigrants. Due to this independence, they were able to prove that crossover leads to polynomial expected optimization times. Note, however, that their topology is rather uncommon for island models as all islands but the receiver do not receive any immigrants themselves.

This work (Lässig and Sudholt 2010a) has initiated a series of publications by the authors and others. One strand of research deals with speed-ups in parallel evolutionary algorithms. This includes island models as well as simple evolutionary algorithms with offspring populations. The speed-up is defined as the quotient of the sequential time (number of function evaluations across all islands/offspring) and the parallel time (number of generations of the parallel system). In Alba’s (2002) taxonomy, this corresponds to an orthodox weak speed-up.

In Lässig and Sudholt (2010c), the authors presented a general method for analyzing such speed-ups. It is based on the fitness-level method, also known as method of $f$-based partitions (Wegener 2002). The idea is to partition the search space into disjoint sets (fitness levels) $A_1, \ldots, A_m$ that are strictly ordered with respect to their fitness: all individuals in a lower fitness level are strictly worse than all individuals in a higher fitness level. Thereby, an elitist EA is said to be on fitness level $i$ if the best individual in the system is in $A_i$. An upper bound on the expected running time is obtained by estimating the expected time until a current elitist has been improved by variation.

Compared to a single EA, an island model running several such EAs can increase the probability of finding a better fitness level. This can significantly reduce the expected time until a better fitness level is found. There are several factors which are the key in this argument. The topology of the network is important as well as the probability of transmitting an elitist to a neighbored island. The quicker the spread of information, the quicker the number of elitists in the whole system increases. This number has a direct impact on the upper bound gained by the fitness-level method. So, a dense topology with a high probability of transmitting elitists leads to the best upper bounds on the expected parallel time—and hence the speedup (Lässig and Sudholt 2010c). This reasoning also applies to offspring populations as they can be regarded as a complete topology.

In Lässig and Sudholt (2010c), a general upper bound as well as bounds tailored to different topologies have been established: an unidirectional ring, a two-dimensional grid or torus graph, and the complete topology. Results in Lässig and Sudholt (2010c), applied to pseudo-Boolean example functions, show that drastic speedups are possible. The magnitude of the speedup depends on the topology as indicated above: the bounds for the complete topology are better than those for the torus, and those for the torus are better than those for rings. Similar results have recently been shown for problems from combinatorial optimization: sorting (as maximizing sortedness), shortest paths, and Eulerian cycles (Lässig and Sudholt 2011).

In Lässig and Sudholt (2011), we presented a simple and powerful scheme for adapting the number of islands in an island model. The idea is to double or halve this number, depending on whether the current generation has led to an improvement of the best fitness value or not. This is similar to the scheme presented in Jansen et al. (2005). Our scheme was proven to lead to optimal speedups in many cases: decreasing the expected parallel running time, without (asymptotically) increasing the expected sequential running time. The same scheme also works for the choice of the offspring population size.

Neumann et al. (2011) considered the impact of migration when crossover is used to recombine immigrants with individuals present on the target island. They gave an example from pseudo-Boolean optimization where such an island model with crossover drastically outperforms panmictic algorithms, independent islands, as well as island models without crossover.

Similar results were shown for instances of the Vertex Cover problem. With Watson and Jansen's single-receiver model as topology, an island model with crossover finds global optima for the considered instance class with high probability (Neumann et al. 2011).

### 1.2 Outline

The structure of the paper is as follows: in Sect. 2, the overall scenario under investigation is described and the example problem LOLZ is introduced in Sect. 3. In Sect. 4, we prove that panmictic EAs are not able to optimize the function LOLZ in reasonable time. The same holds for separate subpopulations. This can be reasoned directly from the results for panmictic populations. A more elaborate analysis of parallel EAs with properly configured migration directly follows in Sect. 5, which contains our main result, along with side results that are of independent interest as well as a discussion about possible extensions of the main result. The positive result for the island model requires a particular choice of the migration interval. In Sect. 6, we consider the performance of the island model if the migration interval is set incorrectly. We prove that both too small and too large values can render the island model as ineffective as panmictic populations or independent runs, respectively.

We then turn from theory to experimental studies. Section 7 deals with reproducing the theoretical results. In Sect. 8, we compare several classical topologies as well as random topologies, across a broad range of migration intervals. Statistical tests in Sect. 9 help to rank topologies according to their performance. We conclude in Sect. 10.

## 2 Definitions

First we restrict ourselves to specific types of algorithms under investigation. One restriction is that we do not consider crossover in our algorithms. The usefulness of crossover for panmictic populations has been discussed elsewhere (see, e.g., Jansen and Wegener 2005; Sudholt 2005; Kötzing et al. 2011). In particular, Watson and Jansen (2007) already presented an example in which an island model with crossover succeeds in optimizing a function with a clear building-block structure. They coined their algorithm *single-receiver multi-deme* model as there is only one island that receives migrants from all other islands. This migration topology is rather trivial as there is no mutual interaction between the islands. The focus in our work is different: we want to get insight into the exchange of information in more realistic migration topologies. Moreover, we do not want the good performance of the algorithm to arise from the use of recombination, but rather from a slow dissemination of information and phases of independent evolution.

As a baseline for our comparison of algorithms, we consider the ($\mu$+1) EA (Algorithm 1) that has already been considered in similar studies (Friedrich et al. 2009; Witt 2006). It is a simple steady-state algorithm that in each generation selects a parent uniformly at random and generates an offspring by mutation. The offspring replaces one of the worst individuals in the population, unless it is inferior to all individuals in the population. Throughout this work, $n$ denotes the number of bits.

**Algorithm 1** Panmictic ($\mu$+1) EA

```
Let $t := 0$ and initialize $P_0$ with $\mu$ individuals chosen
    uniformly at random.
repeat
    Choose $x \in P_t$ uniformly at random.
    Create $y$ by flipping each bit in $x$ independently
        with probability $1/n$.
    Choose $z \in P_t$ with worst fitness u.a.r.
    if $f(y) \geq f(z)$ then $P_{t+1} = P_t \setminus \{z\} \cup \{y\}$
                        else $P_{t+1} = P_t$.
    Let $t = t + 1$.
```

Let $P = P^1 \dot\cup P^2 \dot\cup \cdots \dot\cup P^k$ be a partition of the whole population in multiple subpopulations, also referred to as *islands*. Further assume that there is a so-called *migration* *topology*, given by a directed graph. Islands represent vertices of the topology and directed edges indicate neighborhoods between the islands. Algorithm 2 represents a *parallel EA*, where $k$ subpopulations $P_i, i = 1, 2, \ldots, k$ evolve independently as in the ($\mu$+1) EA from Algorithm 1, except for special migration steps.

Every $\tau$ steps migrants from each island, in this case copies of the island's best individual, are sent to all islands that are connected in the migration topology via a directed edge. The incoming migrants are included into the island using the same selection as in the panmictic ($\mu$+1) EA: for each subpopulation, the best immigrant replaces a worst individual on the island, unless the former is inferior to all individuals on the island. The value $\tau$ is called *migration interval*. The special case of $\tau = \infty$, i.e., no migration, is called the *parallel EA with independent subpopulations*. In case all subpopulations have size 1, we call it the *parallel (1+1) EA with migration*.

**Algorithm 2** Parallel EA with migration

```
Let $t := 0$.
For all $1 \leq i \leq k$ initialize $P_0^i$ uniformly at random.
repeat
    For all $1 \leq i \leq k$ do in parallel
        if $t$ mod $\tau = 0$ and $t > 0$ then
            Send copies of an individual with maximum
                fitness in $P^i$ to all neighbored populations.
            Let $y^i$ be an individual with maximum
                fitness among all immigrants.
        else
            Choose $x^i \in P_t^i$ uniformly at random.
            Create $y^i$ by flipping each bit in $x^i$
                independently with probability $1/n$.
        Choose $z^i \in P_t$ with worst fitness u.a.r.
        if $f(y^i) \geq f(z^i)$ then $P_{t+1}^i = P_t^i \setminus \{z^i\} \cup \{y^i\}$
                          else $P_{t+1}^i = P_t^i$.
    Let $t = t + 1$.
```

In the following, we derive upper and lower bounds on the number of generations for each algorithm. Note that the number of operations in one generation of the panmictic ($\mu$+1) EA is constant, while in a parallel model, the total number of operations in one generation is usually much larger. Comparing a sequential algorithm like the panmictic ($\mu$+1) EA with a parallel one in terms of the number of generations is, hence, unfair. We keep this in mind, but nevertheless stick to the number of generations as performance measure. In our upcoming results, the impact of the computational effort in one generation is negligible, since the comparison with respect to the number of generations will be between polynomial and exponential values.

The running time bounds are asymptotic (with respect to the problem size $n$) and they are stated using common notations for asymptotic growth as described in Cormen et al. (2001). In addition, our bounds hold with a very high probability. We say that an event happens *with overwhelming probability* (w.o.p.) if its probability is at least $1 - \exp(-\Omega(n^{\varepsilon}))$ for some constant $\varepsilon > 0$. In other words, the probability of the event not happening decreases exponentially with growing $n$. An important observation that will be used often implicitly in our proofs is the following. Assume we have a polynomial number (denoted by $\mathrm{poly}(n)$) of "typical" events in a run of an algorithm that all occur w.o.p. The events need not be independent. Taking the union bound for the complementary events, the probability that all typical events happen is still at least $1 - \mathrm{poly}(n) \cdot \exp(-\Omega(n^{\varepsilon})) = 1 - \exp(-\Omega(n^{\varepsilon}) + \log n) = 1 - \exp(-\Omega(n^{\varepsilon}))$ and, hence, overwhelming.

## 3 The function LOLZ

We now formally define the problem investigated in the following. The construction is based on the well-known leading ones function LO that simply counts the number of leading ones in the bit string. We give a formal definition for some bit string $y = y_1 y_2 \ldots y_{|y|}$ of length $|y|$ and also define a symmetric counterpart LZ that counts the number of leading zeros:

$$\mathrm{LO}(y) = \sum_{i=1}^{|y|} \prod_{j=1}^{i} y_j \text{ and } \mathrm{LZ}(y) = \sum_{i=1}^{|y|} \prod_{j=1}^{i} (1 - y_j).$$

If we take the sum of these functions, $\mathrm{LO}(x) + \mathrm{LZ}(x)$, as fitness function, this corresponds to the length of the largest prefix where all bits have the same value. An evolutionary algorithm is encouraged to either establish a prefix of ones or a prefix of zeros. At some point, this decision becomes irreversible: once every individual in the population has a prefix that is long enough, many bits have to be flipped in one generation to switch from one prefix-value to the other. By symmetry, the probability of gathering leading ones equals the probability of gathering leading zeros. We now make the function asymmetric by capping the maximum fitness values in case of leading zeros: $\mathrm{LZ}(x)$ is replaced by $\min\{z, \mathrm{LZ}(x)\}$ where $z$ indicates the maximum number of zeros. This means that once the number of leading zeros exceeds $z$, the fitness cannot be increased by adding more leading zeros and the solution is a local optimum. From a global perspective, gathering leading zeros is, hence, a bad choice. To achieve drastic effects on performance, we use a composition of several blocks of this type to define the function LOLZ (leading ones, leading zeros) as described below. The number of blocks is denoted by $b$, each block contains $\ell$ bits and the variables $n, z, b$, and $\ell$ are parameters of the function. Figure 1 shows a sketch of the block structure.

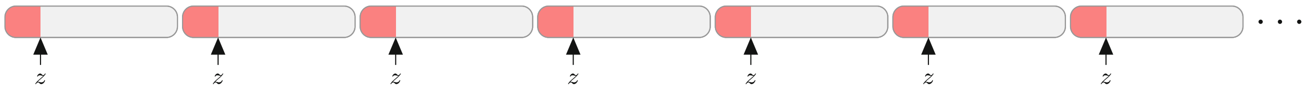


**Fig. 1** Sketch of the function LOLZ. The *red area* indicates the region where leading ones and leading zeros are treated symmetrically

**Definition 1** Let $z, b, \ell \in \mathbb{N}$ such that $b\ell \leq n$ and $z < \ell$. For a bit string $x = x_1 \ldots x_n$ we abbreviate $x_{i\ell+1} \ldots x_{(i+1)\ell}$ by $x^{(i)}$ and let

$$\mathrm{LOLZ}_{n,z,b,\ell}(x) \\ = \sum_{i=1}^{b} \prod_{j=1}^{(i-1)\ell} x_j \cdot \left[\mathrm{LO}(x^{(i)}) + \min\left(z, \mathrm{LZ}(x^{(i)})\right)\right].$$

The block structure is implemented in a way that the blocks have to be optimized one by one, from left to right; only if the $i$th block has been turned to the all-ones substring, the product for summand $i + 1$ can be positive. Also note that each bit always contributes a value of 0 or 1 to the fitness. Unless the algorithm gets stuck with a block of leading zeros, the fitness corresponds to the number of leading bits that have been set correctly so far. In other words, unless stuck in a local optimum, the fitness can always be increased by fixing bits from left to right. All individuals with a prefix of $b\ell$ leading ones represent global optima.

The intuition behind the function is as follows: in a panmictic population, the whole population tends to move towards one specific type of prefix. Hence, for each block, there is a probability of about $1/2$ that the whole population starts gathering leading zeros in the block, thus getting stuck in a local optimum that is hard to overcome. The probability of always making the correct decision is close to $2^{-b}$, so it decreases exponentially with the number of blocks. Independent subpopulations without communication also tend to get stuck as even multiple independent populations cannot make up for the very small success probability of $2^{-b}$ if $b$ is not too small.

Contrarily, an island model with a well-chosen migration interval and a suitable migration topology can help. First, the islands make independent decisions whether to gather leading ones or leading zeros. After some time, the islands that have chosen leading zeros will get stuck and the other islands will exhibit a larger fitness. If a migration happens at this point, the islands that have made the right decision are able to take over islands which are stuck in worse local optima. This way, information about the "good" decision can be spread throughout the islands, so that the islands that are stuck in local optima can be reactivated to participate in the search on new blocks. If the parameters and the migration topology are set right, we have a good chance of finding the global optimum in the end.

Before starting the analysis of the mentioned algorithms, we elaborate on the randomness of bits. In our function, the bits need to be fixed from left to right. Let $i^*$ be the largest fitness in the population. This means all fitness evaluations so far only depended on the bits $1, \ldots, i^* + 1$. The bits at positions $i^* + 2, \ldots, n$ are *unbiased* in a sense that there has been no preference for bit values. Instead, these bits have been initialized uniformly at random and have been subject to random mutations. As a result, these bits are still random, i.e., the probability that a bit $i^* + j$, $2 \leq j \leq n - i^*$ has value 1 in any specific individual is exactly $1/2$. A formal proof for the function LO is given in Droste et al. (2002). This observation will prove to be useful for showing that subpopulations make independent decisions when starting a new block.

## 4 Panmictic and independent populations fail

We first consider panmictic populations and prove that the panmictic ($\mu$+1) EA cannot optimize LOLZ efficiently, w.o.p. As mentioned before, a panmictic population tends to gather leading bits of the same value. With respect to an individual $x$ we call a block $i$ the *current block* if it is the first block that does not contain only ones in $x$. We say that $x$ is of type 1 if its current block starts with a one; otherwise, $x$ is of type 0.

The next lemma makes the following statement: if the population size is not too large and some individuals of equal type have reached a new (better) fitness level, there is a constant probability ("$\Omega(1)$" in asymptotic notation) that these individuals take over the population, so that the population only contains individuals of this type.

**Lemma 1** *Assume the best fitness in the population is $k$ and all individuals of fitness $k$ are of type $s$, for some $s \in \{0, 1\}$. Further assume that $\mu \leq cn/(\log n)$ for an arbitrary constant $c > 0$ and $n \geq 2$. Then with probability $\Omega(1)$ the panmictic* ($\mu$+1) EA *reaches a population where all individuals are of type $s$ and have fitness exactly $k$.*

*Proof* Assuming there are $i$ individuals of type $s$ and fitness $k$, a "good event" $G_i$ is given if one of the $i$ type-$s$ individuals is chosen and none of the bits are mutated. Using $(1 - 1/n)^n \geq 1/4$ for $n \geq 2$ and recalling that every individual is selected with probability $1/\mu$,

$$P(G_i) \geq \frac{i}{\mu} \cdot \left(1 - \frac{1}{n}\right)^n \geq \frac{i}{4\mu}.$$

For every individual with fitness $j$ the following holds. If mutation does not flip the bits at positions $j$ and $j+1$, the offspring cannot have a larger fitness than its parent. The reason is that the bit at position $j+1$ does not match the prefix in the corresponding block in the parent. This then also holds for the offspring. Note that bits $j$ and $j+1$ must be in the same block as we assume fitness $j$ and the first bit of each block contributes to the fitness.

We therefore define a "bad event" $B_i$ as the event that mutation flips at least one of the two mentioned bits. Clearly,

$$P(B_i) \leq \frac{2}{n}.$$

The probability that a good event occurs next is

$$\begin{aligned} &P(G_i \text{ before } B_i) \\ &\quad = P(G_i \mid B_i \cup G_i) = \frac{P(G_i)}{P(G_i) + P(B_i)} \\ &\quad \geq \frac{i/(4\mu)}{i/(4\mu) + 2/n} = 1 - \frac{2/n}{i/(4\mu) + 2/n} \\ &\quad \geq 1 - \frac{2/n}{i/(4\mu)} = 1 - \frac{8\mu}{in}. \end{aligned}$$

A desired population is created if $\mu - 1$ successive good events occur, before any bad event happens. The probability for this is at least

$$P(G) = \prod_{i=1}^{\mu-1} P(G_i \mid G_i \cup B_i) \geq \prod_{i=1}^{\mu-1} \left(1 - \frac{8\mu}{in}\right).$$

For $\mu \leq cn/\log n$, using $1 - x \geq \mathrm{e}^{-2x}$ for $x \leq 1/2$ (see, e.g., Abramowitz and Stegun 1964, 4.2.37), the product is at least

$$\begin{aligned} &\prod_{i=1}^{\mu-1} \left(1 - \frac{8c}{i \log n}\right) \geq \prod_{i=1}^{\mu-1} \exp\left(-\frac{16c}{i \log n}\right) \\ &\quad = \exp\left(\frac{16c}{\log n} \cdot \sum_{i=1}^{\mu-1} \frac{1}{i}\right) \geq \exp\left(\frac{16c}{\log n} \cdot (\ln \mu + 1)\right) \\ &\quad = \Omega(1), \end{aligned}$$

where in the last inequality, we have used $\sum_{i=1}^{\mu-1} \frac{1}{i} \leq \ln(\mu - 1) + 1$ (Mitzenmacher and Upfal 2005, p. 34).

Now, we know that each time the best fitness is improved, there is a constant probability that one type of individual gets extinct. To prove that w.o.p., this happens at least once in each block, we need a lower bound on the number of improvements for each block. Note that although the lemma is formulated for LO, it also holds for LZ and it will be straightforward to apply the lemma to LOLZ.

**Lemma 2** *Consider the panmictic* ($\mu$+1) EA *at an arbitrary stage while optimizing LO. Let $\alpha$ be an arbitrary function of n and consider the time until the fitness of the best individual has increased from its current value by at least $\alpha$. With probability $1 - \mathrm{e}^{-\Omega(\alpha)}$ the fitness of the best individual increases at least $\alpha/3$ times before the end of this time period.*

*Proof* Let $k$ be the current best fitness in the population. Note that when the algorithm increases the fitness of the best individual, the new best individual must have the first $k+1$ bits set to 1. However, it may happen that further bits are also set to one. These bits are then called *free-riders* in Droste et al. (2002). By our previous arguments from Sect. 3, the bits $k+2, k+3, \ldots, n$ are uniform. The probability of having $\ell$ free-riders is, hence, $2^{-(\ell+1)}$. Following Droste et al. (2002), we model the free-riders by considering a random 0–1 bit string of arbitrary length with the following interpretation. The random number of bits between the $(i-1)$th and the $i$th 0-bit corresponds to the random number of free-riders during the $i$th improvement. During $\alpha/3$ improvements, we have a total number of at least $2\alpha/3$ free-riders if and only if there are at most $\alpha/3$ zeros among the first $\alpha$ positions. We can apply Chernoff bounds (see, e.g., Mitzenmacher and Upfal 2005) to the random 0–1 bit string as it consists of independent 0–1-variables. This yields that the described event happens with probability at most $\mathrm{e}^{-\alpha/36}$. Hence, with probability $1 - \mathrm{e}^{-\Omega(\alpha)}$ after $\alpha/3$ improvements the total fitness gain is less than $\alpha$.

The following lower bound on the running time increases exponentially with $z$, the maximum number of zeros, that determines how far the local optima are away from all better solutions. The probability bound depends on $z$ and $b$, the number of blocks. In case, say, $b = \ell = \sqrt{n}$ and $z = \ell/2$, the result gives a lower bound of $n^{\Omega(\sqrt{n})}$ that holds w.o.p.

**Theorem 1** *Consider the panmictic* ($\mu$+1) EA *with $\mu \leq cn/(\log n)$ for an arbitrary constant $c > 0$ on* LOLZ$_{n,z,b,\ell}$ *with $z = \omega(\log b)$, $b\ell \leq n$, and $z < \ell$. With probability at least $1 - \exp(-\Omega(z)) - 2^{-b}$ the* ($\mu$+1) EA *does not find a global optimum within $n^{z/3}$ generations.*

*Proof* To prove the theorem, we argue that—apart from a very small error probability—each time a new block is optimized, only one type of individual survives before the threshold of $z$ fixed bits is reached. By symmetry, this implies that type 1 only survives with probability $1/2$. As the decisions for each block are independent, the probability that type 1 survives in all $b$ blocks is $2^{-b}$. In the case type 1 gets extinct, we argue that then at least $z/2$ bits have to be flipped in one generation. This yields the claimed bound on the running time.

Now we only need to argue that extinction happens for each block with high probability. We consider the ($\mu$+1) EA at the first point of time where at least $z/2$ bits in the current block have been fixed. By Lemma 2 the

($\mu$+1) EA makes at least $z/6$ improvements before the first $z$ bits of the current block have been fixed. Lemma 1 states that after all these improvements there is a constant probability that one type of individuals gets extinct. Hence, the probability that the extinction does not happen during $z/6$ trials is $\exp(-\Omega(z))$.

Adding up all above-mentioned error probabilities yields that extinction happens in all $b$ blocks with probability at least

$$\begin{aligned}1 - b \cdot \exp(-\Omega(z)) &= 1 - \exp(-\Omega(z) + \ln b)\\ &= 1 - \exp(-\Omega(z))\end{aligned}$$

since by assumption, $z = \omega(\ln b)$. We conclude that with probability $1 - \exp(-\Omega(z)) - 2^{-b}$ the algorithm reaches a population where every individual is of type 0 with at least $z/2$ leading zeros. A necessary condition for creating any accepted type-1 individual is to flip at least $z/2$ leading zeros in the current block. The probability for this event in one generation is $n^{-z/2}$. The probability that this happens within $n^{z/3}$ generations is still at most $n^{z/3} \cdot n^{-z/2} = n^{-z/6}$. So, with probability at least $1 - \exp(-\Omega(z)) - n^{-z/6} - 2^{-b} = 1 - \exp(-\Omega(z)) -2^{-b}$ the ($\mu$+1) EA needs at least $n^{z/3}$ generations.

Now we justify that migration is necessary. If the subpopulations do not communicate at all, each of them behaves like a panmictic population. As the negative result for Theorem 1 holds w.o.p., it is easy to show that also multiple independent panmictic populations fail, w.o.p.

Now Theorem 2 follows directly from Theorem 1 and taking the union bound for all subpopulations.

**Theorem 2** *Consider the parallel EA with $s \in \mathbb{N}$ independent subpopulations of size $\mu \le cn/(\log n)$ each, $c > 0$ an arbitrary constant, on* $\text{LOLZ}_{n,z,b,\ell}$ *with $z = \omega(\log b)$, $b\ell \le n$, and $z < \ell$. With probability at least $1 - s\exp(-\Omega(z)) - s2^{-b}$ the* ($\mu$+1) EA *does not find a global optimum within $n^{z/3}$ generations.*

If $z, b \ge n^{\Omega(1)}$ and the number of subpopulations is polynomial, this gives an exponential lower bound holding w.o.p.

## 5 Island model with migration succeeds

In contrast to the two algorithms considered so far, the island model with a suitable parametrization can optimize the function LOLZ efficiently. This even holds for the parallel (1+1) EA with migration in which each island behaves like the simple (1+1) EA between migrations. We use $\mu$ to denote the number of subpopulations to compare the algorithm to the panmictic ($\mu$+1) EA.

We believe that island models using larger populations on each island, instead of population size 1, show a similar behavior as long as the population size is not too large. However, proving this formally is difficult as already the analysis for the parallel (1+1) EA is long and quite technical. Considering population dynamics on each island adds to this complexity, and estimating the probability and magnitude of fitness improvements becomes much harder. We therefore only consider the parallel (1+1) EA in the following and leave the study of larger island populations for future work.

### 5.1 How islands communicate and evolve independently

The main proof idea is that each island makes an independent decision for the first block. With high probability at least one island starts gathering leading ones. The first migration typically happens after more than $z$ leading ones have been collected. As all type-1 individuals are strictly better than all type-0 individuals, the former islands take over the latter islands. If the migration topology is not too sparse, we still have many islands that have made the right decision and carry on toward the next block. This process is repeated for the following blocks with the information about the right decision in a block being communicated to other islands. This way, with high probability, some island makes it to the end of the last block and finds a global optimum.

We first define migration topologies that allow a sufficiently large spread of information. The idea is as follows. Whenever there is a small set of islands that have made correct decisions so far, we want the topology to spread this "good" information to sufficiently many neighbored islands. This is ensured if each small set of vertices in the topology has a sufficiently large set of neighbors. If this holds, we can later show that the "good" information will not die out, with high probability.

**Definition 2** Consider a migration topology $G = (V, E)$. For a vertex set $V' \subseteq V$ let $N(V') = \{v \mid \exists u \in V' : (u, v) \in E\}$ denote the neighborhood of $V'$. We call $G$ *well-expanding* if there is a constant $\varepsilon > 0$ such that the following holds. For every subset $V' \subseteq V$ with $|V'| \le |V|^{\varepsilon}$ we have $|N(V')| \ge (2 + \varepsilon)|V'|$.

We remark without giving a formal proof that the $d$-dimensional hypercube graph is well-expanding if $d \ge 3$. In addition, note that a graph is well-expanding if it has minimum degree at least $(2 + \epsilon)|V|^{\varepsilon}$. On the other hand, sparse graphs such as the ring or torus graphs are not well-expanding. However, we will see in Sect. 5.4 how our results can be adapted to hold for any (strongly) connected migration topology.

One aspect of our proof idea deserves a more detailed explanation. To find the global optimum with high probability, it is essential that in each block, all islands make independent decisions. After the "right" decision for one block has been communicated to the surrounding islands, the islands should make independent decisions for the next block, so that again some islands make a good decision. However, these decisions are not independent. Observe that during the migration, all bits of a bit string are transmitted, including information about "random" bits that have not yet had an impact on the fitness in any individual. These bits are still unbiased in a sense that the probability of one particular bit having value 0 equals the probability that it has value 1. However, it is very likely that these bits are correlated among different individuals. In the extreme case of a complete topology, a unique best individual might take over the whole population such that all bits in all individuals are the same.

The following observation provides a solution to this dilemma. While there might be strong dependencies immediately after migration, when the islands continue to evolve independently they regain independence among "random" bits. This is made precise for one bit by Lemma 3. A similar argument has been derived independently in Doerr et al. (2007).

**Lemma 3** *Let $x^0, x^1, \ldots, x^t$ be a sequence of bit strings where $x^{j+1}$ results from $x^j$ by flipping each bit in $x^j$ independently with probability $1/n$. Then for $t \geq n \ln n$, every $x^* \in \{0,1\}^n$, and every bit $i$*

$$\left| P\left(x_i^t = x_i^*\right) - \frac{1}{2} \right| = \frac{1}{2}\left(1 - \frac{2}{n}\right)^t.$$

*Proof* Let $p^k := \binom{t}{k}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{t-k}$ be the probability that bit $i$ is flipped exactly $k$ times during $t$ mutations. Let $p^{\text{even}} := \sum_{k \text{ even}} p^k$ and $p^{\text{odd}} := \sum_{k \text{ odd}} p^k$, then $p^{\text{even}} + p^{\text{odd}} = 1$ and

$$\begin{aligned} p^{\text{even}} - p^{\text{odd}} &= \sum_{k=0}^{t} (-1)^k p^k \\ &= \sum_{k=0}^{t} (-1)^k \binom{t}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{t-k} \\ &= \sum_{k=0}^{t} \binom{t}{k} \left(-\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{t-k} \\ &= \left(1 - \frac{2}{n}\right)^t, \end{aligned}$$

where the last equality follows from the binomial theorem. We have $P\left(x_i^t = x_i^*\right) \in \{p^{\text{even}}, p^{\text{odd}}\}$, depending on the value of $x_i^*$. As $p^{\text{even}} - 1/2 = 1/2 - p^{\text{odd}}$,

$$\left| P\left(x_i^t = x_i^*\right) - \frac{1}{2} \right| = \frac{p^{\text{even}} - p^{\text{odd}}}{2} = \frac{1}{2}\left(1 - \frac{2}{n}\right)^t.$$

## 5.2 Progress estimates

Another crucial component in our proof is a good estimation of the bit positions where migrations take place. If a migration happens too early, an individual that has gathered many leading zeros might take over neighboring islands and, hence, spread "wrong" information. We also want one migration to happen in every block. We have based LOLZ on the function LO (and of course LZ) since the progress on LO is highly concentrated around the expectation. Droste et al. (2002) prove that with probability $1 - \exp(-\Omega(n))$ the optimization time of the (1+1) EA on LO is within an interval $[c_1 n^2, c_2 n^2]$ for two constants $c_1 < c_2$. We need a much stronger bound, though migrations must happen in a rather small area of the bit string. Moreover, the variances accumulate for every block and the cumulated variance for all blocks must still be small.

Note that the progress of the (1+1) EA on LO (and LZ) is fairly uniform. To increase the current fitness, the first bit with the wrong bit value has to be flipped. This probability is always $1/n$. The progress over a longer period of time is hence determined by a sum of independent and almost identical random variables. We say "almost identical" for a good reason. If the current fitness is $i$, the first $i$ bits must not be flipped to yield an improvement. Hence, the probability for improving the current fitness mildly depends on the current fitness. It varies from $1/n$ for $i = 0$ to $1/n \cdot (1 - 1/n)^{n-1} \approx 1/(en)$ for $i = n - 1$. The expected optimization time of the (1+1) EA on LZ is very close to $(e - 1)/2 \cdot n^2$ (Böttcher et al. 2011; Sudholt 2010).

Having to keep leading bits implies that the (1+1) EA experiences a slowdown as more and more leading bits are gathered. In between two migrations, the progress tends to decrease over time. This effect is sketched in Fig. 2. In contrast to an ideal setting where the progress in each block is the same, the slowdown complicates the analysis.

We next present a time bound for the time until a certain progress has been made. This time bound holds with high probability and it depends on the initial number $i$ of leading bits. The lemma significantly improves upon the tail bounds in Droste et al. (2002) and is of independent interest.

**Lemma 4** *For $\alpha \in \mathbb{N}$ and $0 \leq i + \alpha \leq n$ let $T_{i,\alpha}$ denote the elapsed random time until the* (1+1) EA *on* LO *has*

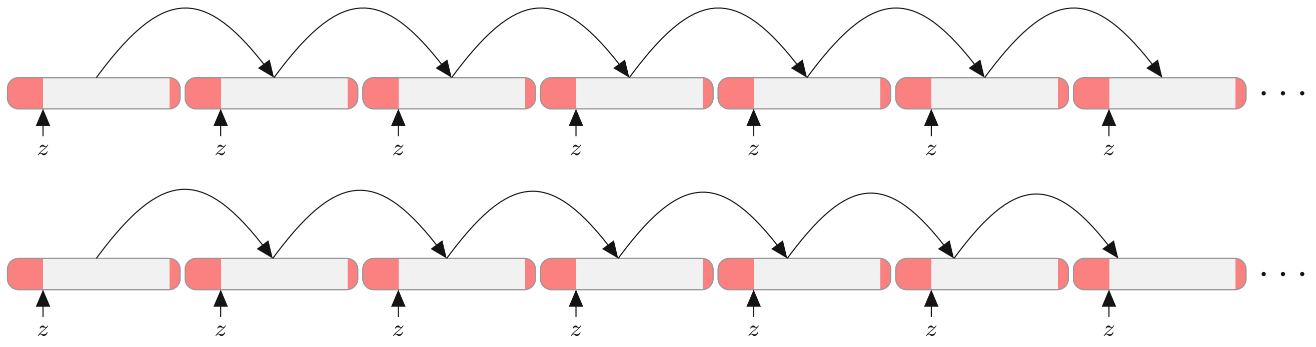


**Fig. 2** Sketch of the progress of the island model between migrations. The *arrows* depict the progress every island typically makes, unless it gets stuck in a local optimum. The *shaded areas* depict locations where migration can be problematic as it can decrease diversity. The *top figure* shows an idealized setting where the progress is the same for each block. The *bottom figure* accounts for the slowdown experienced as the bits in the prefix must be maintained. The progress gets smaller with every block

*increased its fitness from $i$ to at least $i+\alpha$. With probability $1-\exp(-\Omega(\alpha^{2\varepsilon}))$ we have*

$$T_{i,\alpha} \le n(1-1/n)^{-i-\alpha}(\alpha/2+\alpha^{1/2+\varepsilon}) \text{ and}$$
$$T_{i,\alpha} \ge n(1-1/n)^{-i}(\alpha/2-\alpha^{1/2+\varepsilon}).$$

*Proof* The claim is obvious for $\alpha = O(1)$, hence, we assume in the following that $\alpha$ grows with $n$. We only prove the first bound; the second bound can be proven in exactly the same fashion.

The event $T_{i,\alpha} \le n(1-1/n)^{-i-\alpha}(\alpha/2+\alpha^{1/2+\varepsilon})$ is equivalent to the event that the fitness has not increased by $\alpha$ after $n(1-1/n)^{-i-\alpha}(\alpha/2+\alpha^{1/2+\varepsilon})$ generations. This implies that one of the following events has occurred:

1. the (1+1) EA has made less than $\alpha/2+\alpha^{1/2+\varepsilon}/2$ improvements in time $n(1-1/n)^{-i-\alpha}(\alpha/2+\alpha^{1/2+\varepsilon})$ or
2. the (1+1) EA has encountered less than $\alpha/2-\alpha^{1/2+\varepsilon}/2$ free riders during at least $\alpha/2$ improvements.

The probability of the second event is maximal if we consider *exactly* $\alpha/2$ improvements. Modeling the number of free riders by a random 0–1-string as in Lemma 2, Chernoff bounds yield that the probability for the second event is at most $\exp(-\Omega(\alpha^{2\varepsilon}))$.

As the probability of an improvement from a LO-value at most $i+\alpha$ is at least $1/n\cdot(1-1/n)^{i+\alpha}$, the expected number of improvements made within $n(1-1/n)^{-i-\alpha}$ $(\alpha/2+\alpha^{1/2+\varepsilon})$ generations is at least $\alpha/2+\alpha^{1/2+\varepsilon}$. For $\delta := \alpha^{-1/2+\varepsilon}/2$,

$$(1-\Delta)\left(\frac{\alpha}{2}+\alpha^{1/2+\varepsilon}\right) = \frac{\alpha}{2}+\alpha^{1/2+\varepsilon}-\frac{\alpha^{1/2+\varepsilon}}{4}-\frac{\alpha^{2\varepsilon}}{2} \ge \frac{\alpha}{2}+\frac{\alpha^{1/2+\varepsilon}}{2}. \tag{1}$$

Hence, the probability that this random number is less than $\alpha/2+\alpha^{1/2+\varepsilon}/2$ is, by an application of Chernoff bounds, at most $\exp(-\Omega(\alpha\Delta^2)) = \exp(-\Omega(\alpha^{2\varepsilon}))$.

The above time bounds to obtain a specific progress can be turned into bounds on the progress in a specific amount of time. Using Lemma 4 and standard calculations, the corollary below can be shown. The proof simply verifies that if the progress bounds are violated, this implies that one of the time bounds from Lemma 4 is violated. The condition $t = O(n^{5/3})$ is necessary to make up for the difference between the factors $(1-1/n)^{-i}$ and $(1-1/n)^{-i-\alpha}$ in the bounds from Lemma 4.

**Corollary 1** *Let $R_i(t)$ be the progress (i.e. fitness increase) of the* (1+1) EA *on* LO *in $t$ generations when starting with a fitness value of $i$. Let $0<\varepsilon<1/2$ and $t \ge cn$ for a sufficiently large constant $c>0$ and $t = O(n^{5/3})$. With probability $1-\exp(-\Omega((t/n)^{2\varepsilon}))$ we have*

$$R_i(t) \le \frac{2t}{n}\cdot\left(1-\frac{1}{n}\right)^i + 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon} \text{ and}$$
$$R_i(t) \ge \frac{2t}{n}\cdot\left(1-\frac{1}{n}\right)^i - 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon}.$$

*Proof* We start with the upper bound on $R_i(t)$. Define $\alpha := \frac{2t}{n}\cdot\left(1-\frac{1}{n}\right)^i + 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon}$. We argue that if $R_i(t) > \alpha$ then this implies

$$T_{i,\alpha} < t \le n\left(1-\frac{1}{n}\right)^{-i}(\alpha/2-\alpha^{1/2+\varepsilon}).$$

By Lemma 4, the probability for the latter event is $\exp(-\Omega(\alpha^{2\varepsilon})) = \exp(-\Omega((t/n)^{2\varepsilon}))$. As $P(A) \le P(B)$ if $A \Rightarrow B$ this is also an upper bound for the former event and the claim follows.

We only need to verify that the inequality $t \le n(1-1/n)^{-i}(\alpha/2 - \alpha^{1/2+\varepsilon})$ holds. Plugging in $\alpha$ for the first term, we have

$$\begin{aligned} & n\left(1-\frac{1}{n}\right)^{-i}\left(\frac{\alpha}{2}-\alpha^{1/2+\varepsilon}\right) \\ & \quad = t + n\left(1-\frac{1}{n}\right)^{-i}\left[\frac{3}{2}\left(\frac{2t}{n}\right)^{1/2+\varepsilon} - \alpha^{1/2+\varepsilon}\right] \end{aligned}$$

and the desired inequality follows if we can show $\alpha^{1/2+\varepsilon} \le \frac{3}{2}\left(\frac{2t}{n}\right)^{1/2+\varepsilon}$. Using $(x+y)^{\gamma} \le x^{\gamma} + y^{\gamma}$ for $x, y \ge 0$ and $0 \le \gamma \le 1$,

$$\begin{aligned} \alpha^{1/2+\varepsilon} &= \left(\frac{2t}{n}\cdot\left(1-\frac{1}{n}\right)^{i} + 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon}\right)^{1/2+\varepsilon} \\ &\le \left(\frac{2t}{n} + 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon}\right)^{1/2+\varepsilon} \\ &\le \left(\frac{2t}{n}\right)^{1/2+\varepsilon} + 3\left(\frac{2t}{n}\right)^{(1/2+\varepsilon)^2} \\ &\le \frac{3}{2}\left(\frac{2t}{n}\right)^{1/2+\varepsilon} \end{aligned}$$

where the last inequality holds, since $(1/2+\varepsilon)^2 < 1/2+\varepsilon$ and $t/n \ge c$ for a large enough constant $c > 0$.

The lower bound on $R_i(t)$ is proved in a similar fashion. Define $\alpha := \frac{2t}{n}\cdot\left(1-\frac{1}{n}\right)^{i} - 3\left(\frac{2t}{n}\right)^{1/2+\varepsilon}$. We argue that if $R_i(t) < \alpha$ then this implies

$$T_{i,\alpha} > t \ge n\left(1-\frac{1}{n}\right)^{-i-\alpha}(\alpha/2 + \alpha^{1/2+\varepsilon}).$$

By Lemma 4 the probability for the latter event is $\exp(-\Omega(\alpha^{2\varepsilon})) = \exp(-\Omega((t/n)^{2\varepsilon}))$. As $P(A) \le P(B)$ if $A \Rightarrow B$ this is also an upper bound for the former event and the claim follows.

We only need to verify that the inequality $t \ge n(1-1/n)^{-i-\alpha}(\alpha/2 + \alpha^{1/2+\varepsilon})$ holds. Plugging in $\alpha$ for the first term, we have

$$\begin{aligned} & n\left(1-\frac{1}{n}\right)^{-i-\alpha}\left(\frac{\alpha}{2}+\alpha^{1/2+\varepsilon}\right) \\ & \quad = t\left(1-\frac{1}{n}\right)^{-\alpha} + n\left(1-\frac{1}{n}\right)^{-i-\alpha}\left[-\frac{3}{2}\left(\frac{2t}{n}\right)^{1/2+\varepsilon} + \alpha^{1/2+\varepsilon}\right]. \end{aligned}$$

Bernoulli's inequality yields $\left(1-\frac{1}{n}\right)^{\alpha} \ge 1 - \frac{\alpha}{n}$, hence,

$$\left(1-\frac{1}{n}\right)^{-\alpha} \le \frac{1}{1-\alpha/n} = 1 + \frac{\alpha/n}{1-\alpha/n} \le 1 + \frac{2\alpha}{n}.$$

In addition, reusing the previous calculations from the upper bound the expression in square brackets is bounded from above by $-1/4\cdot\alpha^{1/2+\varepsilon}$. Together, along with $2t/n \le \alpha/3$ and $t = O(n^{5/3})$ implying $n = \Omega(n^{3/2})$,

$$\begin{aligned} & n\left(1-\frac{1}{n}\right)^{-i-\alpha}\left(\frac{\alpha}{2}+\alpha^{1/2+\varepsilon}\right) \\ & \quad \le t\cdot\left(1+\frac{2\alpha}{n}\right) - \frac{n}{4}\left(1-\frac{1}{n}\right)^{-i-\alpha}\alpha^{1/2+\varepsilon} \\ & \quad \le t + \frac{2t}{n}\cdot\alpha - \frac{n}{4}\left(1-\frac{1}{n}\right)^{-i-\alpha}\alpha^{1/2+\varepsilon} \\ & \quad \le t + \frac{\alpha^2}{3} - \Omega(\alpha^{2+\varepsilon}) \\ & \quad \le t \end{aligned}$$

since $t \ge cn$ for a sufficiently large constant $c > 0$.

These progress estimates are now used to locate the bit positions in the population at the time a migration happens. For simplicity, we consider the island model on the function LO and argue later how this transfers to the function LOLZ.

**Lemma 5** *Consider the parallel* (1+1) EA *with migration on the function* LO *with* $\tau = \Omega(n^{1+\Omega(1)})$, $\tau = O(n^{5/3})$, *and* $\mu \le \mathrm{poly}(n)$ *subpopulations. At the time of the kth migration,* $k \in \mathbb{N}$, *for every individual x in the population we have with overwhelming probability*

$$\begin{aligned} & \frac{2k\tau}{n} - \frac{6k^2\tau^2}{n^3} - 3k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} \\ & \quad \le \mathrm{LO}(x) \le \frac{2k\tau}{n} + 4k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} \end{aligned}$$

*provided* $k\cdot\frac{2\tau}{n} + 4k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} \le n$.

*Proof* For the upper bound, we observe $\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} \ge n^{\Omega(1)}$. The probability that the initial population contains an individual with fitness larger than $\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}$ is bounded by $\mu\cdot 2^{-n^{\Omega(1)}}$. Hence, w.o.p., all initial individuals have fitness less than $\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}$.

According to Corollary 1 the probability that during $\tau$ generations following a migration, one individual with fitness $i$ makes progress larger than $\frac{2\tau}{n}\cdot\left(1-\frac{1}{n}\right)^{i} + 3\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}$ is $\exp(-2\tau/n)$ and, hence, exponentially small. The same holds for the weaker progress bound $\frac{2\tau}{n} + 3\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}$ which is independent of $i$. The probability that during $k$ migrations, there is one individual making a larger progress is $k\mu\exp(-2\tau/n)$ and still exponentially small. Hence, after $k$ migrations, all individuals will have fitness at most

$$\begin{aligned} & \left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} + k\cdot\frac{2\tau}{n} + 3k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon} \\ & \quad \le k\cdot\frac{2\tau}{n} + 4k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}. \end{aligned}$$

For the lower bound, we pessimistically assume an initial fitness of 0. By Corollary 1 the probability that the progress is less than $\frac{2\tau}{n}\cdot\left(1-\frac{1}{n}\right)^{i}-3\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}$ is exponentially small if $i$ is the fitness after the previous migration. We use a pessimistic estimate of $i$ for all individuals in the population and all considered migrations. Due to the upper bound of this lemma, the fitness is always bounded above by $k\cdot\frac{2\tau}{n}+4k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}\le 3k\tau/n$. Hence, w.o.p., all individuals in all $k$ migrations make progress at least

$$k\cdot\frac{2\tau}{n}\cdot\left(1-\frac{1}{n}\right)^{3k\tau/n}-3k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}.$$

By Bernoulli's inequality, $\left(1-\frac{1}{n}\right)^{3k\tau/n}\ge 1-3k\tau/n^2$, which yields the lower bound

$$k\cdot\frac{2\tau}{n}-\frac{6k^2\tau^2}{n^3}-3k\left(\frac{2\tau}{n}\right)^{1/2+\varepsilon}.$$

### 5.3 Proof of the main result

Now we are prepared to prove the main result of this section. To obtain the following upper bound, we assume that the migration counter $t$ in Algorithm 2 is initialized with a value of $\tau/2$ instead of 0. This restricts the migrations to take place in the middle of each block. Alternatively, we could achieve the same effect by enlarging the first block in our function LOLZ by $\ell/2$ bits. This modification would not affect the negative results for the other algorithms. However, we would like to keep the fitness function as simple as possible and, therefore, decide to modify the migration counter instead.

Due to the slowdown effect described earlier, we also need to assume that the number of blocks is rather small and all blocks are not too long. Otherwise, at some point, migration is likely to happen too early in one of the blocks, as shown in Fig. 2. Restricting the number and length of the blocks, we only consider the first bits of the bit string, where the slowdown is not significant.

**Theorem 3** *Consider the parallel* (1+1) EA *with migration on a well-expanding migration topology with* $\tau=n^{5/3}$ *and* $\mu$ *subpopulations for* $\mu\le\text{poly}(n)$ *and* $\mu\ge n^{\Omega(1)}$. *Let the function* $\text{LOLZ}_{n,z,b,\ell}$ *be parametrized according to* $\ell=2\tau/n=2n^{2/3}$, $z=\ell/4=n^{2/3}/2$, *and* $b\le n^{1/6}/16$. *If the migration counter starts at* $\tau/2=n^{5/3}/2$ *then with overwhelming probability the algorithm finds a global optimum within* $O(b\ell n)=O(n^2)$ *generations.*

*Proof* Call an individual $x$ *stuck* in some block $i$ if $i$ is the current block for $x$ and the first $z$ bits in block $i$ are set to 0. Call an individual *good* if it is not stuck in any block. We first prove that each time a migration happens the following conditions hold w.o.p.:

1. all individuals are either stuck or have more than $z$ leading ones in their current block and
2. for all individuals, the last $z$ bits of the current block are not part of the leading ones.

Moreover, we will prove that in every block at least one migration happens. We will use Lemma 5 to prove this, but first we have to argue in how far the progress bounds proven for LO transfer to the function LOLZ.

The progress on LO equals the progress on LZ by symmetry of bit values. Recall that LOLZ consists of blocks of the function LO + LZ, loosely speaking. A lower bound on the progress on LO is clearly also a lower bound for the progress on the function LO + LZ. In a sense, the first bit is guaranteed to be a free rider. Hence, the progress on LO + LZ is dominated by the progress on LO increased by 1. Considering blocks of LO + LZ, all lower bounds on the progress still hold and our upper bounds for LO are only off by the number of blocks, provided that the individual does not get stuck.

Now the conditions follow if for each good individual, the $i$th migration happens when the number of leading ones in its current block is larger than $z=\ell/4$ and smaller than $\ell-z=3/4\cdot\ell$. Due to the modified migration counter and the previous lemmas, the real value of leading ones is by a term $\tau/n$ smaller than that stated in Lemma 5. At the $i$th migration, choosing $\varepsilon:=1/4$, this value is at least (recalling $\ell=2\tau/n$ and using $i\le k$)

$$\frac{2i\tau}{n}+\frac{\tau}{n}-\frac{6i^2\tau^2}{n^3}-3i\left(\frac{2\tau}{n}\right)^{3/4}$$
$$\ge i\ell+\frac{\ell}{2}-6k^2n^{1/3}-3k\left(2n^{2/3}\right)^{3/4}$$

and the desired lower bound follows if the sum of the negative terms is less than $-\ell/4=-n^{2/3}/2$. Recalling $k\le n^{1/6}/16$, we verify that

$$6k^2n^{1/3}+3\cdot 2^{3/4}\cdot kn^{1/2}$$
$$\le\frac{6}{16^2}n^{2/3}+\frac{3\cdot 2^{3/4}}{16}n^{2/3}<\frac{n^{2/3}}{2}.$$

Similarly, using Lemma 5 and adding a value of 1 for each block, w.o.p., the position $i\ell+\ell/2$ is exceeded by at most

$$i+4i\left(\frac{2\tau}{n}\right)^{3/4}\le k+4\cdot 2^{3/4}kn^{1/2}$$
$$\le\frac{n^{2/3}}{16}+\frac{4\cdot 2^{3/4}}{16}n^{2/3}<\frac{n^{2/3}}{2}.$$

Now we use these conditions to prove that with high probability, one good individual finds a global optimum.

We first argue that the decisions for every block are made nearly independently. To apply Lemma 3 with a large enough $t$-value for the number of mutations, we make use of condition 2: after migration in each individual at least $z$ bits have to be fixed to advance to the next block. When considering a single island, these last $z$ bits are still unbiased, and hence uniform. A single island then behaves like the (1+1) EA, which allows us to apply Lemma 4. By Lemma 4, at least $\Omega(nz)$ generations are needed to fix the last $z$ bits of the block, w.o.p. In one generation without migration, an individual on an island is replaced by an offspring that results from a mutation if all bits that currently contribute to the fitness are not flipped. This happens with probability at least $(1 - 1/n)^{n-1} \geq 1/e$. Hence, during $\Omega(nz)$ generations, the bits in the following blocks are mutated at least $\Omega(nz)$ times in expectation. By Chernoff bounds, the number of mutations is w.o.p., at least $cnz$ for some small constant $c > 0$. Invoking Lemma 3 with $t = cnz = cn^{5/3}/2$, the difference between the distribution of bits in the following blocks under the real and the uniform distribution is exponentially small. Assuming that these bits are exactly uniform only introduces an exponentially small error. Hence, keeping this in mind, we continue our considerations and assume that the bits in the following blocks are uniform.

The first condition implies that at the time of a migration all good individuals have a strictly larger fitness than all individuals that are stuck in some block. Hence, after migration, all good individuals will propagate "goodness" to their neighbors, i.e., all neighbors of good individuals in the topology will also contain a good individual after migration.

By assumption the topology is well-expanding, so there is a constant $\gamma > 0$ such that every subset of size $s \leq \mu^{\gamma}$ vertices has at least $(2 + \gamma)s$ neighbors. W. l. o. g. $\gamma \leq 1/2$ since making $\gamma$ smaller only relaxes the condition. We claim that w.o.p. after each migration there are at least $\mu^{\gamma}$ good individuals in the population. In the first block, evolution is independent for all islands. By symmetry, we have a probability of at least $1/2$ that during the first migration, an individual will be good. (We say "at least" $1/2$ since there is a tiny chance of a stuck individual becoming good again before the first migration.) The probability that fewer than $\mu^{\gamma}$ good individuals emerge is $\exp(-\Omega(\mu^{\gamma}))$ by Chernoff bounds.

The complementary probability is overwhelming, since $\mu \geq n^{\Omega(1)}$ and, hence, $\mu^{\gamma} \geq n^{\Omega(1)}$. Assume that we have $\mu_i \geq \mu^{\gamma}$ good individuals after migration in the $i$th block. During phases of independent evolution each individual remains good with probability at least $1/2$. Again, by Chernoff bounds at least $\mu_{i+1} \geq \mu_i/(2 + \gamma)$ individuals remain good when optimizing the $(i + 1)$-st block, w.o.p. If $\mu_{i+1} \geq \mu^{\gamma}$, we are fine. Otherwise, during migration in the $(i + 1)$-st block all neighbors of good individuals are taken over by good individuals. Due to well-expansion, the number of good individuals increases to at least $\mu_i$ again, hence $\mu_{i+1} \geq \mu^{\gamma}$ in any case.

Repeating these arguments for all blocks and summing up all error probabilities, one good individual reaches the end of the last block and thus finds a global optimum w.o.p.

Note that although Theorem 3 is formulated for the fixed value $\tau = n^{5/3}$, it can be adapted towards other values for the migration interval as long as the conditions from Lemma 5 are met and the relation $\tau = 2\ell/n$ is maintained.

### 5.4 Extensions: crossover and sparse topologies

We believe that the positive result for the island model can also be extended toward crossover in the following way. If 1-point or 2-point crossover is applied to migrants when entering a new island, there is a good chance that the block structure will be preserved. So, the probability of crossover being disruptive is rather small and good individuals still have a good chance of taking over neighbored islands. A rigorous proof, however, remains a topic for future work.

Theorem 3 only holds for well-expanding topologies. Intuitively, we would expect sparse topologies like the ring graph to work even better for LOLZ as communication is slowed down and diversity is increased, in comparison to dense topologies. The reason why the previous theorem is restricted is because we rely on good information being spread quickly. If we only have one migration per block, this can only be guaranteed if the migration topology is well-expanding. However, our arguments can be transferred quite easily if we consider LOLZ with larger blocks. The blocks are chosen so large that several migrations take place in each block. If $d$ migrations happen within each block, a good type-1 individual can take over all islands reachable via at most $d$-directed edges of the topology. If $d \geq n^{\Omega(1)}$, this implies that a sufficient number of islands is taken over by good individuals in each block. By these arguments, we can show the following theorem, analogously to Theorem 3. Recall that a directed graph is strongly connected if every two vertices are connected via a directed path.

**Theorem 4** *Consider the parallel* (1+1) EA *with migration on an arbitrary strongly connected migration topology with* $\tau = n^{5/3}$ *and* $\mu$ *subpopulations for* $\mu \leq \text{poly}(n)$ *and* $\mu \geq n^{\Omega(1)}$. *Let* $\text{LOLZ}_{n,z,b/d,d\ell}$ *be parametrized according to* $\ell = 2\tau/n = 2n^{2/3}$, $z \leq \ell/2 = n^{2/3}$, $d \geq n^{\Omega(1)}$, *and* $bd \leq n^{1/6}/16$. *If the migration counter starts at* $\tau/2 = n^{5/3}/2$ *then with overwhelming probability, the* *algorithm finds a global optimum within* $O(b\ell n) = O(n^2)$ *generations.*

We only remark that instead of scaling up the block size by $d$ the same effect can be achieved by dividing the migration interval and the value of $z$ by $d$, each, and adapting the technical constraints as well as the value of $\varepsilon$ used in the calculations in the proof of Theorem 3.

## 6 Wrong parameter settings

We have seen that the island model works well if the migration interval is set to a very precise value. An obvious question is whether this delicate choice is really necessary—or whether the island model is robust with respect to the parameter choice. To see this, we will perform a detailed experimental study in Sect. 7 and give some theoretical arguments in the following.

There are several possible reasons why a wrong migration interval can prove harmful, see the sketch in Fig. 3. If the migration interval is too small, migration happens frequently. This can be harmful in case migration happens in some block before the symmetry of leading ones vs. leading zeros is broken. Then there is a high chance that the wrong information is propagated to other islands and genetic drift rapidly decreases diversity. Another reason for failure is that the island model does not get enough time for islands to regain their independence, if migration happens just before a new block is reached.

Theorem 3 not only required the migration interval to be in the correct order but also required to be synchronized with the block structure. If this is not the case, it is likely that migration will happen at a "wrong" point of time. This can have the same consequences as for too small migration intervals, see Fig. 3. Finally, the migration interval might be too large, hence leaving the island model in a situation similar to independent islands.

We analyze the two cases of too small and too large migration intervals. The case that the migration interval is not synchronized with the block structure will only be investigated experimentally in Sect. 7. The following theorem shows that if the migration interval is too small, the performance of the island model is similar to that of a panmictic population. The diameter diam(G) of a graph $G$ denotes the length of the longest shortest path between any two vertices in a graph.

**Theorem 5** *Consider the parallel* (1+1) EA *with* $\mu = \mathrm{poly}(n)$ *islands on a topology* $T$ *with diameter* diam(T). *If* $\tau = O(n/(\mu \cdot \mathrm{diam(T)}))$ *then the parallel* (1+1) EA *needs at least* $n^{z/3}$ *generations with probability* $1 - b \cdot 2^{-\Omega(z/(\log \mu))} - 2^{-b}$.

*Proof* We reuse arguments from the proof of Theorem 1. First, we prove that with probability $2^{-\Omega(z/\log \mu)}$ we have at least $z/(16 \log \mu)$ improvements of the best fitness in each block, when the $\max\{\mathrm{LO}, \mathrm{LZ}\}$-value in the block is between $z/2$ and $z$.

Whenever the best fitness is improved, the amount of increase is determined by potential free riders. The probability of having more than $(\log \mu) + 1$ free riders is at most $2^{(\log \mu)+1} = 1/(2\mu)$. Even if all $\mu$ islands were to find improvements of the best fitness in the same generation, the probability that there is one island improving the best fitness with $(\log \mu) + 1$ free riders is at most $\mu \cdot 1/(2\mu) = 1/2$. The maximum number of free riders is thus stochastically dominated by a random variable $((\log \mu) + 1) \cdot X$ where $X$ is a geometrically distributed random variable with parameter $1/2$. Arguing with "blocks" of $(\log \mu) + 1$ free riders instead of single bits, we apply the same reasoning as in the proof of Lemma 4. With probability $2^{-\Omega(\mu/\log \mu)}$ we have that there are at most $3/8 \cdot z/((\log \mu) + 1)$ blocks of free riders in $1/8 \cdot z/((\log \mu) + 1)$ generations where the best fitness increases. Then the total fitness increase is at most $3/8 \cdot z + 1/8 \cdot z/((\log \mu) + 1) \le z/2$. As $1/8 \cdot z/((\log \mu) + 1) \ge z/(16 \log \mu)$, this proves the claim that at least $z/(16 \log \mu)$ fitness increases are necessary to increase the $\max\{\mathrm{LO}, \mathrm{LZ}\}$-value from $z/2$ to $z$.

Assume that this "typical" event occurs for each block. Consider the following event, which we call a *landslide takeover*. The idea is that the event spreads a unique best solution throughout the whole topology. Assuming a generation $t^*$ where at least one island improves upon the current best fitness, a landslide takeover is given when

- no other island improves upon its fitness within the following $\tau \cdot \mathrm{diam(T)}$ generations and
- in generation $t^*$ only one island finds an improvement.

Note that a landslide takeover implies that the single improvement will take over all islands as diam(T) migrations suffice to propagate this solution to all islands of the topology.

Recall from the proof of Lemma 1 and the estimation of "bad events" that the probability of making an improvement by mutation is at most $2/n$. The probability of a landslide takeover is at least

$$\left(1 - \frac{2}{n}\right)^{(\tau \mu\, \mathrm{diam(T)}) + \mu} \ge 1 - \frac{4\tau\, \mu\, \mathrm{diam(T)}}{n}.$$

The probability that in $z/(16 \log \mu)$ generations no landslide takeover occurs is at most

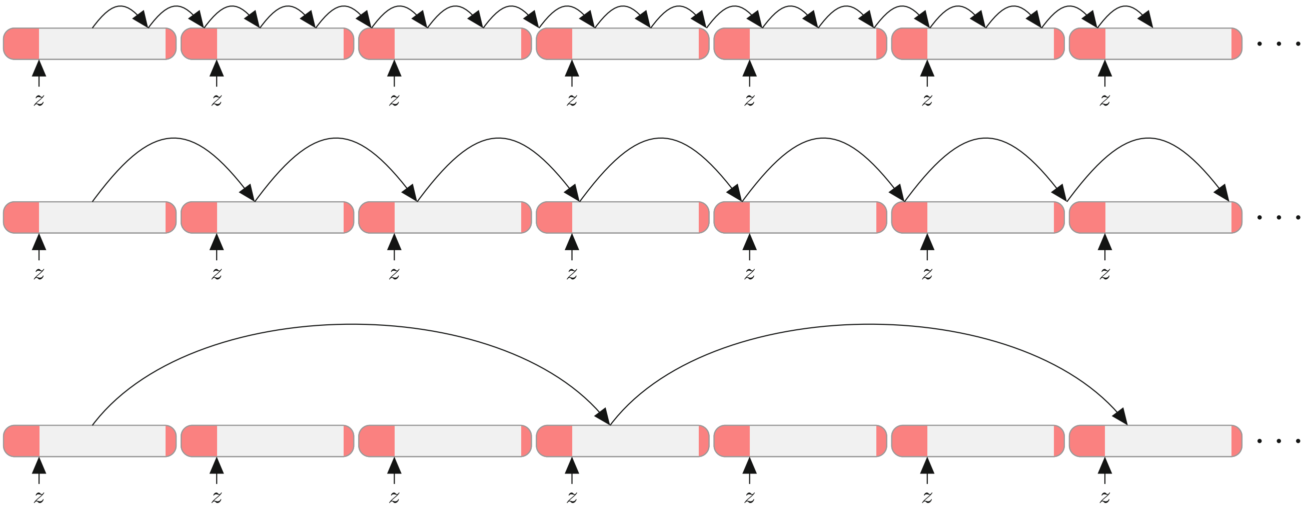


**Fig. 3** A sketch of possible wrong choices of the migration interval. The *top figure* shows a too small migration interval, with migration happening at problematic locations and possibly decreasing diversity. The migration interval in the center is in the right order of magnitude, but it is not synchronized with the block structure. The migration interval at the bottom is too large, so that islands have to make the correct decision for several blocks

$$\begin{aligned}\left(1-\frac{4\tau\,\mu\,\mathrm{diam(T)}}{n}\right)^{z/(16\log\mu)} &\le \exp\left(-\frac{\tau\,\mu\,\mathrm{diam(T)z}}{4n\log\mu}\right)\\ &\le \exp\left(-\frac{z}{4\log\mu}\right)\\ &\le 2^{-\Omega(z/\log\mu)}.\end{aligned}$$

With the converse probability, a landslide takeover takes place. Summing up all error probabilities for each block, with probability at least $1-b\cdot 2^{-\Omega(z/\log\mu)}$ in each block, a landslide takeover takes place before the length of the current prefix on the current block exceeds $z$. Under this condition, all islands will have the same prefix. With probability $1-2^{-b}$ a 0-prefix will be propagated at least once. Repeating the lower-bound arguments from the proof of Theorem 1 proves the claim.

We now consider the situation where the migration interval is too large, i.e., by some factor larger than the "ideal" value. This means that islands are left alone while making decisions for several blocks one after another. This decreases the probability that an island makes a correct decision on all encountered blocks, before the next migration happens. If, additionally, the expansion of the topology is limited, then the islands that do manage to make all these decisions correctly cannot transmit this information to sufficiently many islands. Then the "good information" slowly dies out over time. This reasoning is made precise in the following theorem. The limited expansion is expressed by means of the maximum degree of the topology, i.e., the maximum number of islands connected to any island.

**Theorem 6** *Consider the parallel* (1+1) EA *with migration on a migration topology with* $\mu$ *subpopulations for* $\mu \le \mathrm{poly}(n)$ *and* $\mu \ge n^{\Omega(1)}$ *and maximum degree* $\Delta \ge 1$. *Let the function* $\mathrm{LOLZ}_{n,z,b,\ell}$ *be parametrized according to* $b\ell \le n$, $\ell \ge n^{\Omega(1)}$, $1 \le z < \ell$, *and* $b \ge 1$. *If* $\tau \ge e\ell n\cdot\log(8\Delta+8)$ *then with probability at least* $1-2^{-\Omega(n^{\varepsilon})}-2^{-\Omega(b/\log(\Delta))}$ *for some constant* $\varepsilon > 0$ *the algorithm reaches a situation where every island is stuck in a local optimum.*

*Proof* Assume each island still has to optimize at least $\log(8\Delta+8)$ further blocks. Note that $\mu\cdot 2^{-\Omega(n^{\varepsilon})} = 2^{-\Omega(n^{\varepsilon})}$ by our condition on $\mu$. The expected progress made on a single island in $\tau$ generations between migrations is close to a value in between $2\ell\cdot\log(8\Delta+8)$ and $2e\ell\cdot\log(8\Delta+8)$, depending on the initial length of the prefix.

Applying Corollary 1 with $t=\tau$ yields the following, for an appropriate choice of $0<\varepsilon<1/2$ and regardless of the initial prefix length $i$. With probability $1-2^{-\Omega(n^{\varepsilon})}$ in the $\tau$ generations between migrations each island will have made progress by at least $\log(8\Delta+8)=3+\log(\Delta+1)$ blocks, or be stuck in a local optimum. This implies that at least $2+\log(\Delta+1)$ complete blocks have been covered. By the same corollary, again regardless of the initial prefix length $i$ the number of optimized blocks in $\tau$ generations is at most $6\log(8\Delta+8)$ with probability $1-2^{-\Omega(n^{\varepsilon})}$.

Using Lemma 3 and taking into account an error probability of $2^{-\Omega(n)}$ for each island, we can assume that the decisions made on at least $1+\log(\Delta+1)=\log(2\Delta+2)$ blocks are independent for all islands. Each island makes all these decisions correctly with probability at most $1/(2\Delta+2)$.

We focus on the expected number of good islands, i.e., islands that are not stuck. Let $X_t$ denote this number just before the $t$th migration and $Y_t$ denote this number right after the $t$th migration. We trivially have $Y_0 \leq \mu$. By the above observations, we have

$$\mathrm{E}\ (X_{t+1} \mid Y_t) \leq Y_t/(2\Delta + 2).$$

As each island can transmit to at most $\Delta$ further islands, we have $Y_{t+1} \leq (\Delta + 1)X_{t+1}$ and hence

$$\mathrm{E}\ (Y_{t+1} \mid Y_t) \leq Y_t/2.$$

Iterating this argument, $\mathrm{E}\ (Y_t \mid Y_0) \leq Y_0 \cdot 2^{-t} \leq \mu \cdot 2^{-t}$. As this holds for every $Y_0$, for the unconditional expectation $\mathrm{E}\ (Y_t) \leq \mu \cdot 2^{-t}$. This implies $P(Y_t \geq 1) \leq 2^{-t}$, i.e., the probability of having at least one good island decreases exponentially with $t$.

We have already shown that during $\tau$ generations at most $6 \log(8\Delta + 8)$ blocks are optimized. Putting $t := \lfloor b/(6 \log(8\Delta + 8)) \rfloor = \Omega(b/\log(\Delta))$ and noting that all error probabilities sum up to $2^{-\Omega(n^{\varepsilon})}$ proves the theorem.

## 7 Experimental reproduction

We complement our theoretical studies with the results of simulations. So far, the consideration of topologies was limited to certain characteristics such as the diameter or the maximum degree of the topology. Our experiments also consider the structure of the topologies. In particular, we compare common topologies such as rings, torus graphs, a hypercube and a complete graph with random topologies with various degrees for all vertices.

Our basic setup for all experiments is the optimization of $\mathrm{LOLZ}_{n,z,b,\ell}$ for dimension $n = 1{,}000$ and $z = 10$. Recall that our theoretical result from Theorem 3 was limited to a small number of blocks, hence not making use of the complete bit string. For experiments, we think that it makes more sense to use all bits. This does not strictly comply with the theoretical part, but it serves as a good test on how robust the island model is. We therefore use $b = 10$ blocks of length $\ell = 100$ each. The population size for the panmictic ($\mu$+1) EA was chosen as $\mu = 32$. All parallel EAs use $\mu = 32$ islands with subpopulations of size 1, each.

In our first experiment we considered four common topologies for the island model: a ring, a torus (i. e. a two-dimensional grid with edges wrapping around on all sides) with side lengths 4 × 8, a five-dimensional hypercube, and the complete graph $K_\mu$. All edges are bidirectional. The migration interval was fixed to $\tau = 50{,}000$, which meets the condition $\ell = 2\tau/n$ from Theorem 3. In accordance with Theorem 3 we initialize the migration counter such that the first migration takes place in generation $\tau/2 = 25{,}000$. This is when roughly the first half of the first block has been optimized.

All algorithms were stopped when either the global optimum had been found or each individual had at least $z$ leading zeros in the block currently to be optimized, which means that at least $z$ bits would have to be flipped in one mutation to get out of this local optimum. This event has a negligibly small probability of less than $n^{-z} = 10^{-30}$. The expected time until a local optimum is left is at least $10^{30}$, so it makes sense to stop beforehand.

In our first experiments, we simulate 1,000 runs for each algorithm and record the success rate, i.e., the fraction of runs that were stopped in a global optimum. We also record the mean number of generations until an algorithm has been stopped as well as the mean best fitness value in the final population. The latter corresponds to the mean number of leading bits that have been fixed in the best individual. The results are shown in Table 1.

For the panmictic ($\mu$+1) EA no run was successful. The mean final fitness value was 114, hence, on average, the algorithm got stuck already in the second block, after only 92,367 generations. Independent subpopulations led to a higher success rate of 0.038 and the algorithm stopped with a higher mean best final fitness of 550. Note that the performance of the parallel EA with independent subpopulations is determined by the best out of $\mu = 32$ independent runs of a (1+1) EA. Here independent runs clearly outperform a panmictic population.

The island model performed far better than the two previous algorithms for all topologies. Surprisingly, the most sparse topologies, the ring and the torus, performed best. With a torus every run was successful and so was almost every run on the ring. For the hypercube the success rate decreased to 0.651 and for the complete graph $K_\mu$ it was even only 0.327.

To get a more detailed impression of the dynamics within a run, we repeated 100 runs for each algorithm and observed the number of "good" individuals over time. An

**Table 1** Success rate, mean best fitness value after stopping and mean number of generations until runs were stopped for the considered algorithms

| Algorithm | Success rate | Final fitness | # Generations |
|---|---|---|---|
| Panmictic ($\mu$ + 1) EA | 0.0 | 114 | 92,367 |
| Independent subpopulations | 0.038 | 550 | 377,472 |
| Island model on ring | 0.995 | 999 | 709,704 |
| Island model on torus | 1.0 | 1000 | 655,858 |
| Island model on hypercube | 0.651 | 907 | 647,339 |
| Island model on $K_\mu$ | 0.327 | 651 | 344,759 |

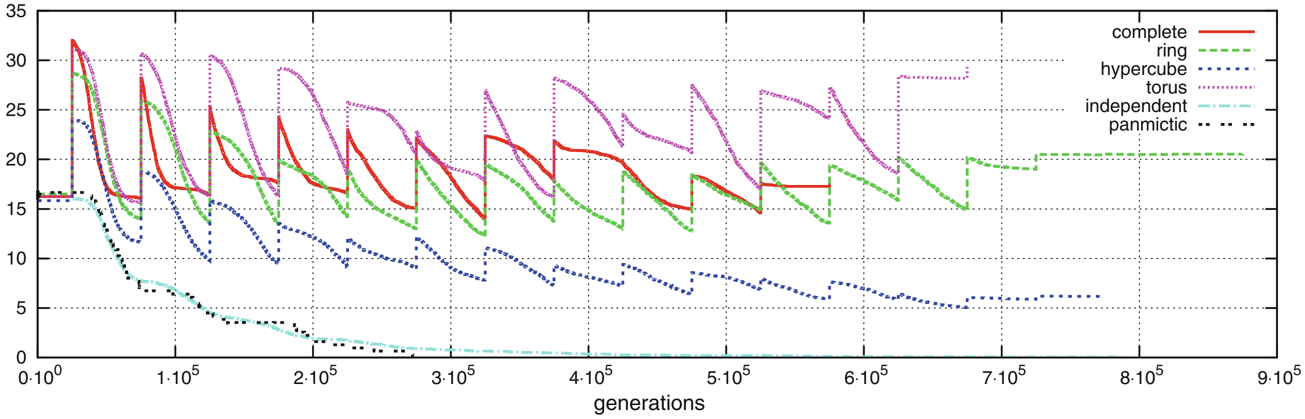


**Fig. 4** Average number of good individuals in 100 runs for the panmictic ($\mu$+1) EA, separate subpopulations, and the island model with $\tau = 50{,}000$

individual is called good if it has leading ones in its current block (i.e., the first block not completely filled with 1-bit). Note that goodness may change when a new block is reached. Unless we have to deal with correlated bit values, the probability of a good individual being good in the next block is $1/2$. Figure 4 shows the number of good individuals over time, averaged over 100 runs for all algorithms. For runs that were stopped with a global optimum, the number of good individuals in the final generation was used to compute the average. We stopped recording once all runs were stopped.

The number of good individuals decreases quickly for the panmictic ($\mu$+1) EA as well as for independent subpopulations. For all island models during migration, the number of good individuals increases as good individuals take over neighbored islands without good individuals. This effect is particularly pronounced for the torus, while the fluctuations are smallest for the hypercube. The number of good individuals is, in general, higher for the torus than for the hypercube, with the ring and $K_\mu$ in between.

Note that the dense topologies hypercube and complete graph performed worse than expected from Theorem 3. Recall that in our setting, we have violated the preconditions for this theorem, so strictly speaking its statement does not apply here. What we can learn is that the slowdown of the progress on LOand similar functions seems to make a difference for dense topologies (recall Fig. 2). The proof of Theorem 3 required migration to happen roughly in the middle of each block. In the light of the slowdown, it is unlikely that this property can be maintained if the whole bit string is used. It is not clear any more that $\tau = \ell n/2$ is a best choice for the migration interval—or whether slightly larger values are better. For every migration interval, there is a chance that migration might happen at the "wrong" point in time, so that the wrong information is communicated to other islands. Sparse topologies are less prone to this potential loss of diversity than dense topologies.

## 8 Comparison of topologies

Now we investigate whether other migration intervals lead to better results, particularly for dense topologies, and how the choice of the migration interval is related to the choice of the migration topology. We start with the classical topologies from Sect. 7 and later consider random graphs.

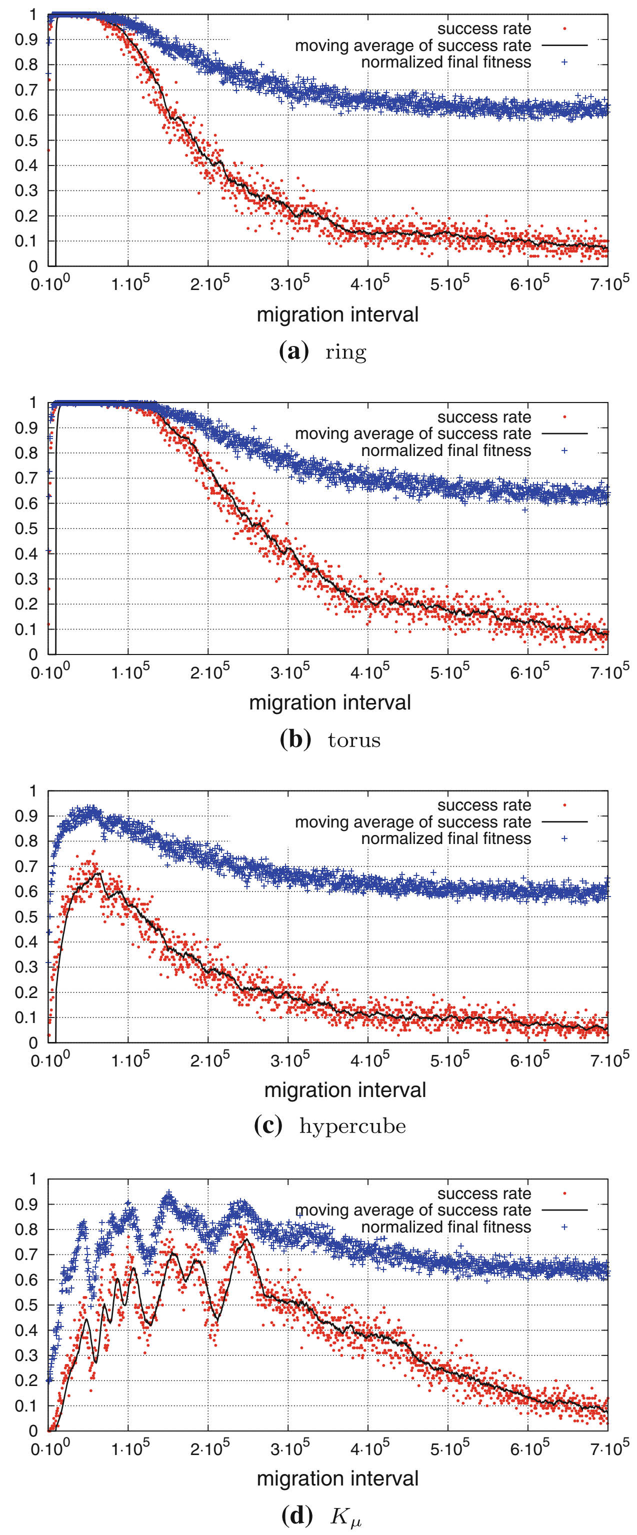
### 8.1 Classical topologies

In a series of computationally expensive experiments, we recorded the success rate in 100 runs for a broad range of migration intervals, increasing $\tau$ in steps of 500 from 500 to 700,000 (CPU time: 319.6 h ring, 351.7 h torus, 298.3 h hypercube, 295.2 h $K_\mu$ on dual-core Opteron 270 processors with 2.0 GHz, 8GB RAM DDR 400). All other parameters were chosen as in Sect. 7; in particular, the migration counter was always started at 25,000.[1] The result is shown in Fig. 5.

For all topologies the success rate is relatively high for the values of $\tau$ around $\tau = 50,000$ (albeit it is not maximal for $K_\mu$). For larger migration intervals $\tau \geq 250{,}000$ the success rate starts to decrease continuously. This can be explained with the fact that between two migrations, the number of good individuals decreases roughly by factors of $1/2$ with each new block. Such a decrease is in accordance with the result from Theorem 6.

Sparse topologies like ring and torus seem to be robust with respect to the migration interval as for $\tau \leq 100{,}000$ the success rates are close to 1. This is in accordance with Theorem 5 as small migration intervals were shown to be most harmful for topologies with small diameter.

[1] Recall that this is a provably good value for $\tau = 50{,}000$ by Theorem 3. For other migration intervals, there generally is no provably good value, so for $\tau \neq 50{,}000$ the choice of 25,000 is somewhat arbitrary.



**(a)** ring

**(b)** torus

**(c)** hypercube

**(d)** $K_\mu$

**Fig. 5** Success rates, moving averages for 20 data points, and final fitness values for different topologies and migration intervals. The number of runs was 100 for each setting. The final fitness was normalized with a factor 0.001 to fit the interval [0, 1]

The curve for $K_\mu$ is particularly interesting due to its fluctuations. These fluctuations appear to be random at first sight, but due to the large number of 100 runs and strong correlations between neighbored $\tau$-values the empirical data is, in fact, reliable. What we observe seems to be the result of a number theoretic question. Let us temporarily disregard the slowdown and assume that the typical progress between migrations is always a fixed value, $\Delta$. Of course $\Delta$ strongly depends on the migration interval. The success rate is low if $\Delta$ is such that $i\Delta \bmod \ell \leq z$ for some $1 \leq i \leq b$. Whenever this happens, there is a chance of losing diversity during migration due to genetic drift. The more often this happens, the lower the success rate. Hence, those migration intervals seem to be best where $i\Delta \bmod \ell \leq z$ rarely happens. If we consider the slowdown and the randomness of the progress between migrations, we get conditions that are messier, but in similar spirit.

For large migration intervals the success rate for all algorithms is best for $K_\mu$. This is again in accordance with our theoretical results as for large migration intervals topologies with a large degree perform best, see Theorem 6. In contrast, sparser topologies do not utilize the rare migration events that efficiently. The "good" information dies out quickly for sparse topologies, if the migration interval is large.

For $K_\mu$ we would have expected that the best $\tau$, i.e., the $\tau$-value with the largest success rate is around $\tau = 50{,}000$. Figure 5 shows that this is not the case—values around multiples of 50,000 and especially 250,000 seem to be better. This makes sense as with $\tau = k \cdot 50{,}000$, $k \in \mathbb{N}$, migration tends to take place in the middle of every $k$th block. The reason why large migration intervals are good for $K_\mu$ might be that the number of good individuals is very high after each migration. The risk that all these individuals get stuck in one or a few blocks is hence very low, even if no migration takes place in the meantime. If only one individual manages to remain good, the next migration can turn all individuals into good individuals again.

On the other hand, low migration intervals are a disadvantage for $K_\mu$. Each migration has some risk for dense topologies because if it is conducted at the wrong place, all individuals can get stuck immediately. The reason is that if the best individual during such a migration step has leading zeros in its current block—which should be the case with probability about 1/2 unless the threshold of $z$ fixed leading bits has been reached—then the complete population is dominated by individuals of this kind afterwards and the progress of the optimization stops. Hence, up to some point, the success rate is increasing, if the number of these risky migrations is reduced.

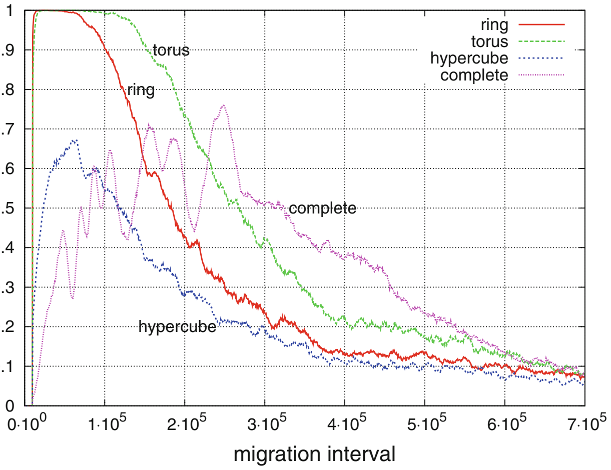


**Fig. 6** Moving averages of the success rate depending on the migration interval for the classic migration topologies

Figure 6 shows the moving averages of the success probability from Fig. 5 in one plot. Here one can see well that the hypercube, ring and torus show a similar behavior regarding the success probability, where the torus performs best except for very small migration intervals. The behavior of the algorithm with the migration structure $K_\mu$ is very different and the best choice for very high migration intervals.

## 8.2 Random topologies

After investigating some classical graph structures as migration topologies we also apply random structures. The computational setup for the investigations is exactly the same as for the classical graph structures. We now investigate random graphs where each node $v$ has directed or undirected edges to $x$ different neighboring nodes for $x \in \{1, 3, 8\}$. These neighbors were chosen uniformly at random, without replacement. In one setup, the same migration topology is used during the complete run of the algorithm (static). In another setup, a new topology was generated for each migration. As the running time of the experiments shows, the additional work for generating a new topology for each migration event is negligible concerning CPU time. For all experiments, the running time was around 300 h, comparable to the classical topologies.

As for the classical migration structures, we first look at the success rate, the mean best fitness value after stopping and the mean number of generations until the algorithms are stopped, Table 2. One can see that the results for all random topologies are much better than for panmictic populations, independent subpopulations, and the island model on the hypercube and $K_\mu$. Only in the setting of static directed graphs with outdegree 1 we get a rather poor success rate of 0.73.

Figure 7 shows the development of the number of good individuals for the migration structures as described above. From the table and the figure one can already clearly see that the dynamic setup seems to be better than the static one. Dense topologies are better than sparse topologies—looking at the number of generations to the optimum and also looking at the final number of good individuals.

The results in Fig. 8 have been computed analogously to Fig. 5, but we omitted to plot again the success probability and mean final fitness for each migration interval. Instead, we focus only on the moving averages of the directed migration topologies, combining them in one plot. The figure shows two trends well. First, dense topologies result in higher success probabilities for a larger bandwidth of migration intervals. Second, the dynamic strategy of using newly generated migration topologies for each migration has a similar effect. As already seen in Fig. 7, the second effect gets more and more negligible for more dense topologies. This can be explained by the fact that in a dense random graph, the distance between any two specific nodes is typically quite small.

Looking at the same scenario for undirected graphs, Fig. 9, the same trends can be detected. Here, the same algorithms for the generation of the topology was used as for directed graphs. But then in a second step all edges of the resulting graph have been made undirected (by mirroring the adjacency matrix at the main diagonal—in both directions). The investigated undirected graphs result in higher success probabilities and mean final objective values. This was expected because making a directed graph undirected by just removing the direction of the edges is basically the same as choosing a higher degree for each node in a connected setup.

Finally, in Fig. 10, we compare the two worlds—the classical migration topologies and the random graphs—by

**Table 2** Success rate, mean best fitness value after stopping and mean number of generations until runs were stopped for different directed random topologies

| Algorithm | Success rate | Final fitness | # Generations |
|---|---|---|---|
| Dynamic directed graph with outdegree 1 | 0.98 | 994 | 742,409 |
| Dynamic directed graph with outdegree 3 | 1.0 | 1000 | 675,777 |
| Dynamic directed graph with outdegree 8 | 1.0 | 1000 | 620,872 |
| Static directed graph with outdegree 1 | 0.73 | 935 | 715,287 |
| Static directed graph with outdegree 3 | 1.0 | 1000 | 674,675 |
| Static directed graph with outdegree 8 | 1.0 | 1000 | 625,579 |

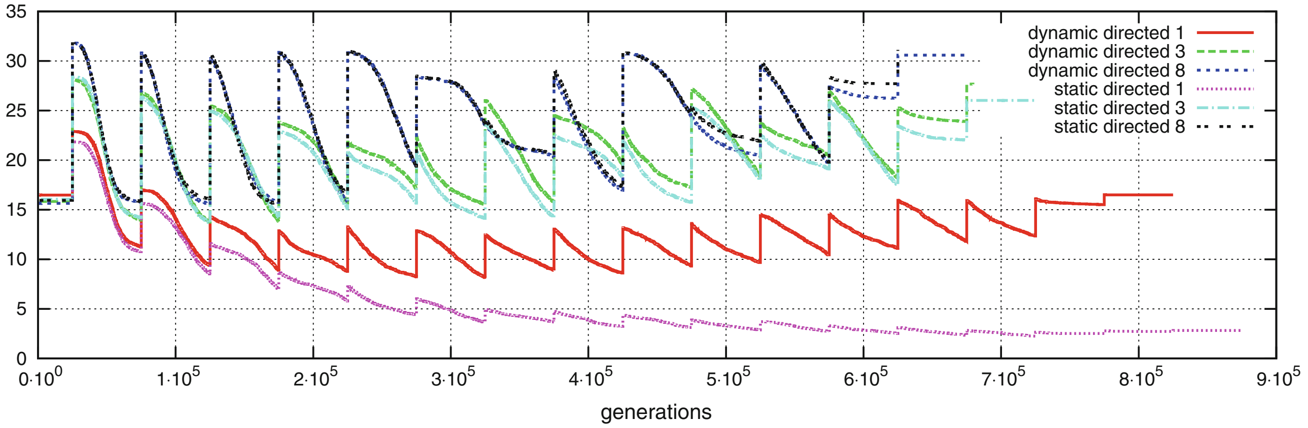


**Fig. 7** Average number of good individuals in 100 runs for the island model with $\tau = 50{,}000$ and different directed random topologies

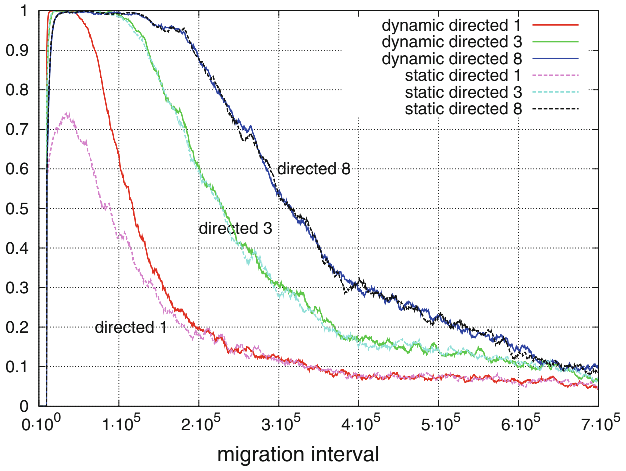


**Fig. 8** Moving averages of the success rate depending on the migration interval for the directed random migration topologies

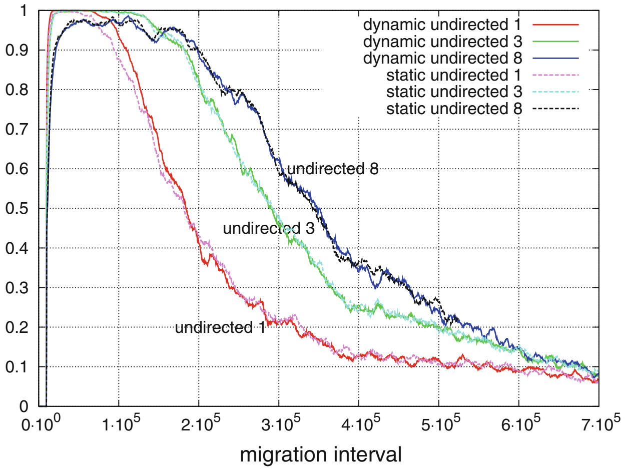


**Fig. 9** Moving averages of the success rate depending on the migration interval for the undirected random migration topologies

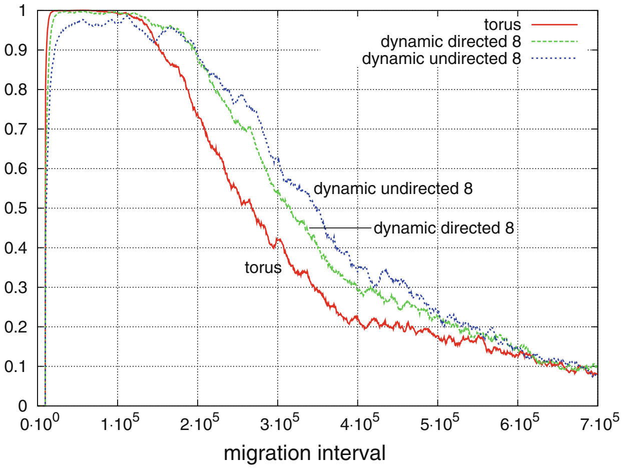
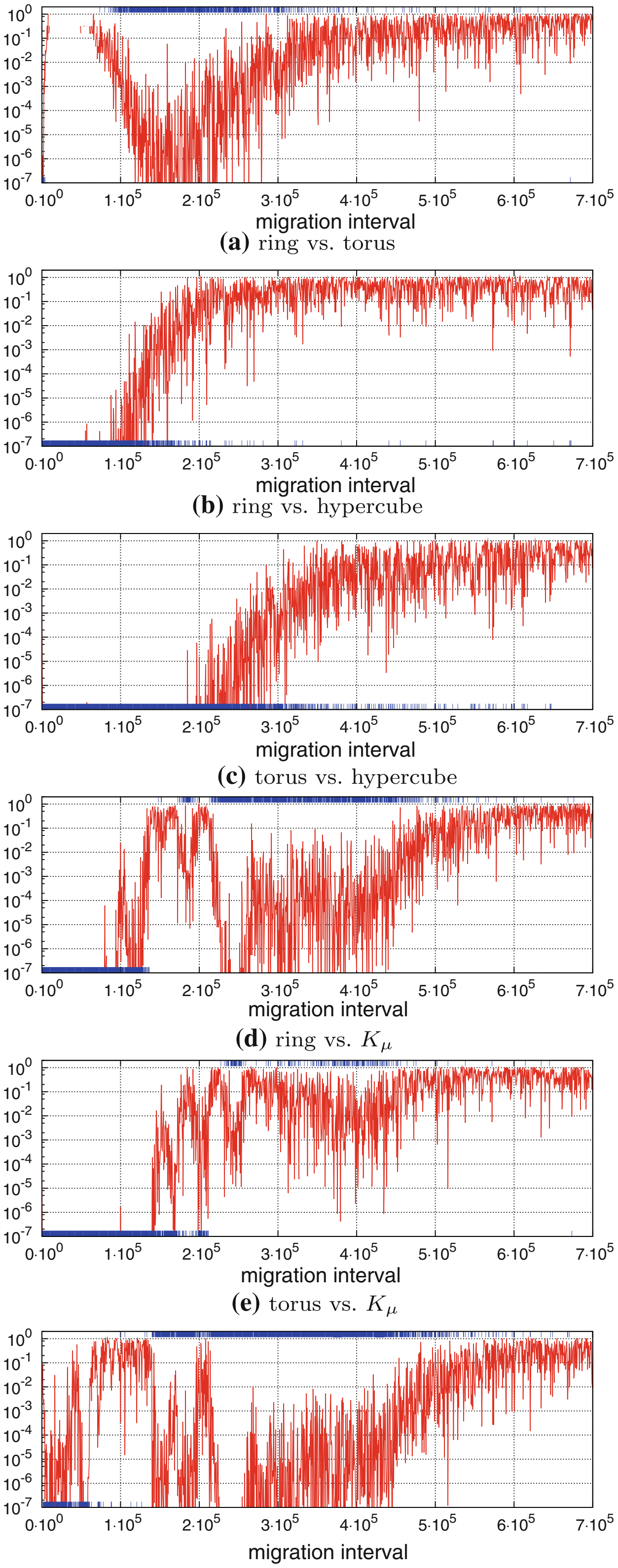


**Fig. 10** Moving averages of the success rate depending on the migration interval for the torus graph and the dynamic variants of the directed and undirected topologies with eight connections

comparing the best performing examples. The torus graphs had the best success probabilities for migration intervals which are of practical interest. For the randomized migration topologies, this was the case for the directed and undirected dynamic topologies with eight connections. Figure 10 shows that the random graphs seem to provide a better performance.

## 9 Statistical validation

Finally, we aim at ranking the different migration topologies for different ranges of the migration interval. This is done using statistical tests for success rates. As the underlying probability distributions are binomial distributions, we use *t*-tests for the comparison of two such distributions as described in Wineberg and Christensen



**(a)** ring vs. torus

**(b)** ring vs. hypercube

**(c)** torus vs. hypercube

**(d)** ring vs. $K_\mu$

**(e)** torus vs. $K_\mu$

**(f)** hypercube vs. $K_\mu$

◀ **Fig. 11** Plot of the $p$ values of two-sided $t$ tests comparing success rates of two algorithms. Values below 0.01 indicate that the underlying success probabilities are different on a significance level of 0.01. These values are marked with *bars* outside the $p$ scale. A *bar at the bottom* indicates that the first algorithm has a higher success probability than the second (judging from which success rate was higher); a *bar at the top* indicates the opposite

(2009). Separate tests are made for each choice of the migration interval and for each pair of migration topologies. Figure 11 shows the resulting $p$-values of two-sided tests. Low $p$-values indicate that the two algorithms have different underlying success probabilities. In case $p$ is large, no conclusion can be made.

## 9.1 Classical topologies

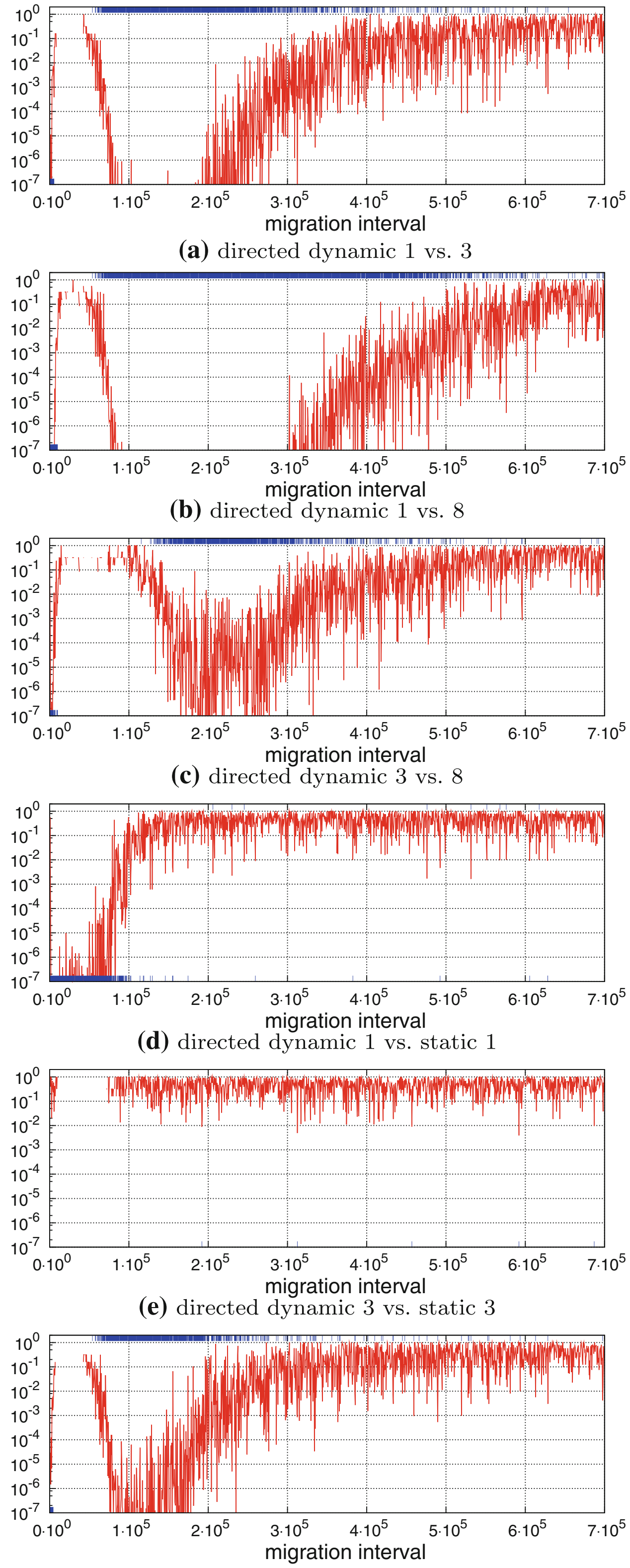
First we only look at the classical topologies, Fig. 11. For very large migration intervals $\tau > 450{,}000$ all topologies show a similar behavior as migration hardly ever happens. For almost all other values the torus is significantly better than the hypercube. The same holds for the ring up to $\tau = 200{,}000$. Except for very small values $\tau \leq 4{,}000$, in which the ring is better, the torus is better than the ring if $\tau$ is roughly in between 100,000 and 300,000.

Comparing $K_\mu$ to all other classical topologies, the ring works better for small migration intervals $\tau$ up to about 130,000 generations. For about 220,000 to 450,000 generations the opposite is the case. The torus shows a similar behavior, but outperforms $K_\mu$ for a larger range of small $\tau$-values, and for larger values $K_\mu$ is not always better than the torus. The performance of the hypercube (compared to $K_\mu$) is slightly worse—this is consistent with the previous comparisons of ring, torus, and hypercube.

We conclude that the ring is the best classical choice for very small migration intervals $\tau \leq 4,000$. In this range, migration tends to be harmful as in situations where less than $z$ leading bits have been fixed, the "wrong" decision might be communicated to neighbored islands. Here the sparsest topology performs best. Contrarily, for large migration intervals, roughly $\tau \geq 200{,}000$, the most expanding topology $K_\mu$ performs best as here a good decision can spread to many islands that are stuck in local optima. The torus seems to be the best compromise for $\tau$-values in between, which are most interesting from a practical point of view. The hypercube was never found to be the best topology in our setting.

## 9.2 Random topologies

Next, we consider the random topologies as introduced in Sect. 8 in our significance analysis. Considering the comparison of each combination of the random topologies (directed vs. undirected, 1 to 8 connections, dynamic vs. static topology generation) would end up in 66



**(a)** directed dynamic 1 vs. 3

**(b)** directed dynamic 1 vs. 8

**(c)** directed dynamic 3 vs. 8

**(d)** directed dynamic 1 vs. static 1

**(e)** directed dynamic 3 vs. static 3

**(f)** directed dynamic 1 vs. undirected 1

**Fig. 12** Plot of the $p$ values of two-sided $t$ tests comparing success rates of two algorithms on random topologies. The test setup is the same as for Fig. 11

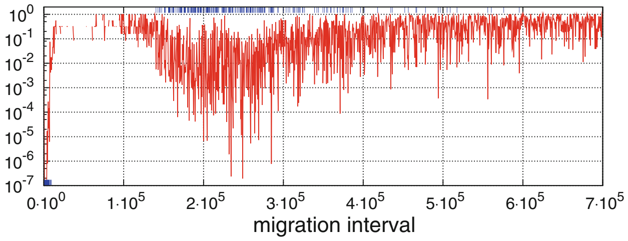


**Fig. 13** Comparison of the torus graph with the random topology directed dynamic with outdegree 8; same setup as for Fig. 11

combinations. Hence, we focus on a few interesting cases, which are visualized in Fig. 12.

Figure 12a–c compares the directed dynamic case (generation of new directed topologies for each migration event) for different densities of the topology. The plots show that except for very small migration intervals the dense topology performs better.

As Fig. 12d shows, generating new topologies for each migration is for sparse topologies significantly better than using the same topology for all migration events at least for migration intervals up to about $10^5$. This advantage of the dynamic paradigm gets lost for more dense topologies. Figure 12e shows that this is already the case if each node has three directed neighbors.

A general observation in the significance tests between directed and undirected topologies of the same density level and without diversifying the dynamic/static parameter is that undirected topologies perform significantly better if the topology is not too dense. Figure 12f shows the case with one directed or undirected neighbor. This observation is what one would expect because an undirected edge yields the same behavior as having two directed edges between two nodes.

Finally we compare the best performing classical topology with the best performing randomized setup. We focus on the most reasonable migration intervals that yield the best performance in both classes of topologies. The best classical topology for this range is the torus graph and the best randomized topology is the directed dynamic topology with outdegree 8 (we could also consider the undirected dynamic topology). As Fig. 13 shows, specifically in the interesting range for the migration interval, the random topology performs better than the torus graph. The torus graph only has an advantage for very small migration intervals, because for very small migration intervals, the sparsest topologies are the best choice.

## 10 Conclusions

This work has initiated a novel approach toward understanding the parallel evolutionary algorithms. Our rigorous analyses help to understand how island models with migration work and when they excel over panmictic or independent populations.

We have demonstrated for the LOLZ function that parallelism and just the right amount of migration between subpopulations can be essential. Comparing Theorems 1, 2, and 3, for the same function $\text{LOLZ}_{n,z,b,\ell}$ with $b = n^{1/6}/16$ blocks, block length $\ell = 2n^{2/3}$, maximum number of zeros $z = n^{2/3}/2$ and $\mu = n/\log n$ we obtain exponential lower bounds for the panmictic ($\mu$+1) EA and the parallel EA with polynomially many independent subpopulations of size $\mu$ each as well as a polynomial upper bound for the parallel (1+1) EA with migration on $\mu$ subpopulations. All bounds hold with overwhelming probability. The positive result for the parallel (1+1) EA with migration can easily be extended to all strongly connected migration topologies.

Our analyses have revealed insight into how information is spread throughout spatially structured parallel EAs. We have learned how populations can gain information by communication and how information can be lost during periods of independent evolution. This can lead to an increased diversity, which is essential for multimodal problems.

The study of wrong parameter settings has exemplified how a too rapid takeover can be harmful. We have proved that if migration happens too frequently, the island model is subjected to genetic drift in the same way as a panmictic population. If migration happens too rarely, "good" information can slowly die out, depending on the maximum degree of the topology.

Complementing the theoretical results on the function LOLZ where migration was proven to be essential, our empirical results show that island models with migration every $50{,}000$ generations clearly perform better than panmictic populations and independent subpopulations without migration on LOLZ, in terms of success rates and final fitness values. Sparse migration topologies lead to a better performance than dense topologies. This result is remarkable since Theorem 3 only makes a statement about dense topologies. Our empirical results suggest that a similar or even a stronger statement might hold for sparse topologies. Interestingly, most random topologies performed better than the classical, structured topologies hypercube and complete graph.

An extensive study of different migration intervals, along with statistical tests, revealed that sparse migration topologies are better for small migration intervals, while dense topologies are better for larger migration intervals. This holds for classical topologies as well as for random topologies. The number of connections per vertex is essential for performance. Undirected edges lead to more dense graphs than directed edges, when the same number of connections is used.

For random topologies we have also seen that it does not make a difference whether the same random topology is used through the whole run (static model) or whether the topology is generated anew in each migration (dynamic model). Only for very sparse graphs such as those with just one connection per vertex, the static model performed worse.

With our theoretical and empirical results we have made progress in understanding how island models work, when and why they excel over other models, and how design choices affect diversity. Our formal proofs only hold for LOLZ, but its main characteristic—decisions about input variables need to be fixed sequentially—can also be found in more realistic problems. Examples are shortest path problems (Baswana et al. 2009) and test input generation for the branch coverage of C programs with nested structures (McMinn 2012). Along the way we have also gathered insights that apply generally, to broader classes of problems. These insights allow for making informed decisions about algorithm design such as choosing the right migration interval and a proper migration topology.

**Acknowledgments** We thank Thomas Jansen, Nicola Beume, Mike Preuß, Günter Rudolph, and Thomas Sauerwald for fruitful discussions on preliminary results. The authors were partially supported by postdoctoral fellowships from the German Academic Exchange Service. Dirk Sudholt was partially supported by EPSRC grant EP/D052785/1.

## References

Abramowitz M, Stegun IA (1964) Handbook of mathematical functions with formulas, graphs, and mathematical tables. 9th Dover printing, 10th GPO printing edition. Dover, New York

Alba E (2002) Parallel evolutionary algorithms can achieve super-linear performance. Inf Process Lett 82(1):7–13

Alba E (2005) Parallel metaheuristics: a new class of algorithms. Wiley-Interscience, New York

Baswana S, Biswas S, Doerr B, Friedrich T, Kurur PP, Neumann F (2009) Computing single source shortest paths using single-objective fitness functions. In: Proceedings of the 10th international workshop on foundations of genetic algorithms (FOGA '09). ACM Press, New York, pp 59–66

Böttcher S, Doerr B, Neumann F (2011) Optimal fixed and adaptive mutation rates for the leadingones problem. In: 11th international conference on parallel problem solving from nature (PPSN 2010). LNCS, vol 6238. Springer, Berlin, pp 1–10

Cantú-Paz E (1997) A survey of parallel genetic algorithms. Technical report, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign, Urbana

Cantú-Paz E, Goldberg DE (1999) On the scalability of parallel genetic algorithms. Evol Comput 7(4):429–449

Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms, 2nd edn. The MIT Press, Massachusetts

Doerr B, Gnewuch M, Hebbinghaus N, Neumann F (2007) A rigorous view on neutrality. In: Proceedings of the IEEE congress on evolutionary computation (CEC 2007). IEEE Press, New Jersey, pp 2591–2597

Droste S, Jansen T, Wegener I (2002) On the analysis of the (1+1) evolutionary algorithm. Theoret Comput Sci 276:51–81

Friedrich T, Oliveto PS, Sudholt D, Witt C (2009) Analysis of diversity-preserving mechanisms for global exploration. Evol Comput 17(4):455–476

Giacobini M, Tomassini M, Tettamanzi A (2003) Modelling selection intensity for linear cellular evolutionary algorithms. In: Proceedings of the sixth international conference on artificial evolution, evolution artificielle. Springer, Berlin, pp 345–356

Giacobini M, Alba E, Tettamanzi A, Tomassini M (2005a) Selection intensity in cellular evolutionary algorithms for regular lattices. IEEE Trans Evol Comput 9:489–505

Giacobini M, Tomassini M, Tettamanzi A (2005b) Takeover time curves in random and small-world structured populations. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2005). ACM Press, New York, pp 1333–1340

He J, Yao X (2006) Analysis of scalable parallel evolutionary algorithms. In: Proceedings of the IEEE congress on evolutionary computation (CEC 2006), July 2006, pp 120–127

Jägersküpper J, Storch T (2007) When the plus strategy outperforms the comma strategy and when not. In: Proceedings of the IEEE symposium on foundations of computational intelligence, FOCI 2007. IEEE, New Jersey, pp 25–32

Jansen T, Wegener I (2005) Real royal road functions: where crossover provably is essential. Discrete Appl Math 149(1–3): 111–125

Jansen T, De Jong KA, Wegener I (2005) On the choice of the offspring population size in evolutionary algorithms. Evol Comput 13:413–440

Kötzing T, Sudholt D, Theile M (2011) How crossover helps in pseudo-Boolean optimization. In: Proceedings of the 13th annual genetic and evolutionary computation conference (GECCO 2011). ACM Press, New York, pp 989–996

Lässig J, Sudholt D (2010a) The benefit of migration in parallel evolutionary algorithms. In: Proceedings of the annual genetic and evolutionary computation conference (GECCO 2010), pp 1105–1112

Lässig J, Sudholt D (2010b) Experimental supplements to the theoretical analysis of migration in the island model. In: 11th international conference on parallel problem solving from nature (PPSN 2010). LNCS, vol 6238. Springer, Berlin, pp 224–233

Lässig J, Sudholt D (2010c) General scheme for analyzing running times of parallel evolutionary algorithms. In: 11th international conference on parallel problem solving from nature (PPSN 2010). Springer, Berlin, pp 234–243

Lässig J, Sudholt D (2011) Adaptive population models for offspring populations and parallel evolutionary algorithms. In: Proceedings of the 11th workshop on foundations of genetic algorithms (FOGA 2011). ACM Press, New York, pp 181–192

Luque G, Alba E (2010) Selection pressure and takeover time of distributed evolutionary algorithms. In: Proceedings of the annual genetic and evolutionary computation conference (GECCO 2010). ACM, New York, pp 1083–1088

Luque G, Alba E (2011) Parallel genetic algorithms theory and real world applications. In: Studies in computational intelligence, vol 367. Springer, Berlin

McMinn P (2013) An identification of program factors that impact crossover performance in evolutionary test input generation for the branch coverage of C programs. Inf Softw Technol (to appear). http://philmcminn.staff.shef.ac.uk/papers/2012-ist.pdf.

Mitzenmacher M, Upfal E (2005) Probability and computing. Cambridge University Press, Oxford

Neumann F, Oliveto PS, Rudolph G, Sudholt D (2011) On the effectiveness of crossover for migration in parallel evolutionary algorithms. In: Proceedings of the 13th annual genetic and evolutionary computation conference (GECCO 2011). ACM Press, New York, pp 1587–1594

Rudolph G (2000) On takeover times in spatially structured populations: array and ring. In: Proceedings of the 2nd Asia-Pacific conference on genetic algorithms and applications. Global-Link Publishing Company, Hong Kong, pp 144–151

Rudolph G (2006) Takeover time in parallel populations with migration. In: Proceedings of the second international conference on bioinspired optimization methods and their applications (BIOMA 2006), pp 63–72

Skolicki Z, De Jong K (2005) The influence of migration sizes and intervals on island models. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2005). ACM Press, New York, pp 1295–1302

Sudholt D (2005) Crossover is provably essential for the Ising model on trees. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2005). ACM Press, New York, pp 1161–1167

Sudholt D (2010) General lower bounds for the running time of evolutionary algorithms. In: 11th international conference on parallel problem solving from nature (PPSN 2010). LNCS, vol 6238. Springer, Berlin, pp 124–133

Teytaud F, Teytaud O (2010) Log(lambda) modifications for optimal parallelism. In: 11th international conference on parallel problem solving from nature (PPSN 2010). LNCS, vol 6238, pp 254–263

Tomassini M (2005) Spatially structured evolutionary algorithms: artificial evolution in space and time. Springer, Berlin

Watson RA, Jansen T (2007) A building-block royal road where crossover is provably essential. In: Proceedings of the genetic and evolutionary computation conference (GECCO 2007). ACM Press, New York, pp 1452–1459

Wegener I (2002) Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In: Sarker R, Yao X, Mohammadian M (eds) Evolutionary optimization. Kluwer, Dordrecht, pp 349–369

Wineberg M, Christensen S (2009) Statistical analysis for evolutionary computation: introduction. In: Companion of GECCO 2009. ACM Press, New York, pp 2949–2976

Witt C (2006) Runtime analysis of the ($\mu$+1) EA on simple pseudo-Boolean functions. Evol Comput 14(1):65–86