

On the Utility of Island Models in Dynamic Optimization

Andrei Lissovoi

DTU Compute, Technical University of Denmark
Kongens Lyngby, Denmark

Carsten Witt

DTU Compute, Technical University of Denmark
Kongens Lyngby, Denmark

ABSTRACT

A simple island model with λ islands and migration occurring after every τ iterations is studied on the dynamic fitness function MAZE. This model is equivalent to a $(1+\lambda)$ EA if $\tau = 1$, i.e., migration occurs during every iteration. It is proved that even for an increased offspring population size up to $\lambda = O(n^{1-\epsilon})$, the $(1+\lambda)$ EA is still not able to track the optimum of MAZE. If the migration interval is increased, the algorithm is able to track the optimum even for logarithmic λ . Finally, the relationship of τ , λ , and the ability of the island model to track the optimum is investigated more closely.

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity

Keywords

Evolutionary Algorithms; Island Models; Dynamic Problems; Populations; Runtime Analysis

1. INTRODUCTION

Evolutionary algorithms (EAs) are a class of nature-inspired algorithms which can be applied to solve a wide variety of optimization problems. Rigorous runtime analysis of nature-inspired algorithms, building on mathematical methods from the analysis of classical algorithms, has advanced considerably in recent years [3, 15]. While most of these studies focus on so-called static optimization problems, whose set of optima is fixed, there has been increasing interest in the analysis of evolutionary and other nature-inspired algorithms on so-called dynamic problems. Many real-world optimization problems are subject to dynamics in that the optimal solution may change as the problem conditions change over time, and the algorithms therefore need to be able to not only find or approximate the optimum at

some point of time, but also to *track* the optimal solution over time as the problem changes.

Application of EAs to dynamic optimization problems is the subject of study in the Evolutionary Dynamic Optimization field, which in recent years has attracted much activity. Many applications of evolutionary algorithms on dynamic problems are considered in the literature [2, 16], and there are already a number of runtime analyses of evolutionary algorithms for dynamic problems [6–8, 10, 17].

Despite the increasing interest in the area, it has not been well understood what mechanisms allow EAs or related nature-inspired algorithms to efficiently track the optimum of a dynamic problem. In [11], Kötzing and Molter introduced a dynamic pseudo-Boolean function called MAZE that separates simple evolutionary algorithms and ant colony optimization. More precisely, the paper shows that while a Max-Min Ant System (MMAS) is able to track the changes occurring in the MAZE fitness function and finds the optimum within polynomial time, a $(1+1)$ EA loses track of the optimum and requires with high probability an exponential amount of time to find the optimum. Very recently, Lissovoi and Witt [13] have built upon this study and shown that introducing a parent population makes the evolutionary algorithm efficient in tracking the MAZE again. Moreover, they present a generalization of the MAZE function that allows for a hierarchy result. More precisely, a family of functions parameterized by μ is defined such that a parent population size of μ allows efficient tracking of the optimum, while population size $\mu - 1$ makes the algorithm lose track of the optimum with high probability. Moreover, a simple MMAS is proved insensitive with respect to the parameter as it is able to track the optimum without any modifications for a wide range of values for μ .

In this work, we consider a different mechanism and analyze its benefit in tracking the optimum of the MAZE benchmark function. More precisely, we focus on parallel nature-inspired algorithms, which are heavily employed in practice due the rapid development of parallel computer architectures. The survey by Alba, Luque and Nesmachnow [1] describes important applications and theoretical studies in this area. In particular, it refers to experiments with parallel nature-inspired algorithms in dynamic optimization, including a study of a parallel swarm algorithm for dynamic vehicle routing problems [9]. It is therefore interesting to determine the theoretical properties of parallel nature-inspired algorithms that allow them to track the optimum of a dynamic problem. Both the number of so-called islands (independent subpopulations) and the communication between them

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '15 July 11–15, 2015, Madrid, Spain

© 2015 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-3472-3/15/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2739480.2754734>

seem influential. From a more general perspective, [16] emphasizes the usefulness of memory and diversity-maintaining operators in EAs for dynamic optimization.

Our contribution is represented by a runtime analysis of a parallel EA for the dynamic MAZE problem. We define a simple parallel EA using an island model with communication occurring within regular intervals, the so-called migration intervals, in the style of Lässig and Sudholt [12], who pioneered the runtime analysis of parallel EAs. The impact of two parameters is studied, namely the number of islands λ and length of the migration intervals τ . In the extreme case that $\tau = 1$, i.e., migration occurs in every generation, the model boils down to a $(1+\lambda)$ EA. It is shown that offspring population sizes, i.e., number of islands, of up to $\lambda = O(n^{1-\epsilon})$, where n is the problem size and ϵ an arbitrarily small positive constant, do not allow this algorithm to track the MAZE efficiently. In contrast, if τ is chosen appropriately, ensuring that migration does not occur too frequently, already $\lambda = \Omega(\log n)$ islands allow efficient tracking of the optimum of the MAZE. Moreover, more general conditions on the choice of τ are worked out, resulting in either efficient tracking or losing track of the optimum. To the best of our knowledge, our contribution represents the first runtime analysis of parallel EAs in dynamic optimization. The results indicate that carefully choosing the migration policy and thereby the communication strategy of an island model can be more advantageous than a mere increase of offspring population size.

This paper is structured as follows. In Section 2, we introduce the parallel EA and the dynamic optimization problem MAZE studied throughout the paper. Moreover, important tools for the analysis are defined. Section 3 is concerned with the negative result for the parallel EA with $\tau = 1$, i.e., the $(1+\lambda)$ EA. The case of appropriately chosen τ , leading to efficient tracking with a small number of islands, is analyzed in Section 4. Moreover, the section elaborates on the impact of the choice of τ on efficient tracking in a more general sense. We finish with some conclusions.

2. PRELIMINARIES

The MAZE fitness function, proposed in [11], and defined formally below, consists of $n + 1$ phases of $t_0 = kn^3 \log n$ iterations each. This phase length was used in [11] to allow the Max-Min Ant System (MMAS) algorithm time to adjust the pheromone values during each phase, and is preserved here for mostly historical reasons. For convenience, we will assume that k is chosen such that t_0 is a multiple of 3.

During the first phase of the MAZE, which we will for convenience refer to as phase 0, the function is equivalent to ONEMAX: the fitness of an n -bit string is equal to the number of 1-bits in the string. During phase p , where $1 \leq p \leq n$, higher fitness values $n + 1$ and $n + 2$ are assigned to two individuals determined by the phase in an oscillating pattern: every two iterations out of three, the OPT_p individual is assigned the fitness value $n + 2$ while the ALT_p individual is assigned the fitness value $n + 1$, and during every third iteration, these assignments are reversed; all other individuals retain their ONEMAX values. Past the last oscillating phase (“phase n ”), MAZE behaves in a fashion similar to TRAP: all individuals except 0^n are assigned ONEMAX values, while 0^n is the global optimum, being assigned the highest fitness value. [11] proves that a $(1+1)$ EA loses track of the optimum of this MAZE function, reverting to optimizing

ONEMAX, and is therefore not able to construct the final $\text{OPT}_n = 0^n$ optimum in a polynomial number of iterations.

$$\begin{aligned} \text{MAZE}(x, t) &= \begin{cases} n + 2 & \text{if } t > (n + 1) \cdot t_0 \wedge x = 0^n \\ n + 2 & \text{if } t > t_0 \wedge x = \text{OPT}(t) \\ n + 1 & \text{if } t > t_0 \wedge x = \text{ALT}(t) \\ \text{ONEMAX}(x) & \text{otherwise} \end{cases} \\ \text{OPT}(t) &= \begin{cases} \text{OPT}_{\lfloor t/t_0 \rfloor} & \text{if } t \not\equiv 0 \pmod{3} \\ \text{ALT}_{\lfloor t/t_0 \rfloor} & \text{otherwise} \end{cases} \\ \text{ALT}(t) &= \begin{cases} \text{ALT}_{\lfloor t/t_0 \rfloor} & \text{if } t \not\equiv 0 \pmod{3} \\ \text{OPT}_{\lfloor t/t_0 \rfloor} & \text{otherwise} \end{cases} \\ \text{OPT}_p &= 0^p 1^{n-p} & \text{for } p \leq n \\ \text{ALT}_p &= 0^{p-1} 1^{n-p+1} & \text{for } p \leq n \\ \text{ALL}_p &= \{\text{ALT}_p, \text{OPT}_p\} \end{aligned}$$

We note that the clock t is considered external to the MAZE function, allowing the fitness value of multiple solutions to be evaluated in each clock value t . For the $(1+\lambda)$ EA, and the λ island model, this corresponds to having hardware available to evaluate many solutions in parallel, or having the problem changes occur at fixed intervals regardless of the number of parallel evaluations.

We consider the behavior of the $(1+\lambda)$ EA, shown as Algorithm 1, and that of a simple island model running λ $(1+1)$ EAs in parallel, using a complete migration topology (i.e. during migration, all islands communicate with all other islands), with various choices for the frequency of migration, shown as Algorithm 2, on the MAZE function. Both algorithms use the standard bit mutation operator, formalized in Definition 1.

Notably, the $(1+\lambda)$ EA is a special case of the island model where migration occurs during every iteration.

Algorithm 1 $(1+\lambda)$ EA

```
Select  $x^*$  uniformly at random from  $\{0, 1\}^n$ .
for  $t \leftarrow 1, 2, \dots$  do
  for  $i \leftarrow 1, \dots, \lambda$  do
     $x_i \leftarrow \text{mutate}(x^*)$ 
   $x_m \leftarrow \arg \max_{x_i} f(x_i, t)$ 
  if  $f(x_m, t) \geq f(x^*, t)$  then
     $x^* \leftarrow x_m$ 
```

Algorithm 2 Island model with λ islands running $(1+1)$ EAs in parallel.

```
for  $i \leftarrow 1, \dots, \lambda$  do
  Select  $x_i^*$  uniformly at random from  $\{0, 1\}^n$ .
for  $t \leftarrow 1, 2, \dots$  do
  if migration occurs during iteration  $t$  then
    Let  $x_m = \arg \max_{x_i^*} f(x_i^*, t)$ 
     $x_i^* \leftarrow x_m$  for all  $i \in \{1, \dots, \lambda\}$ 
  for  $i \leftarrow 1, \dots, \lambda$  in parallel do
     $x_i \leftarrow \text{mutate}(x_i^*)$ 
    if  $f(x_i, t) \geq f(x_i^*, t)$  then
       $x_i^* \leftarrow x_i$ 
```

DEFINITION 1 (STANDARD BIT MUTATION). *The mutation operator $\text{mutate}(x)$ creates an image $y \in \{0, 1\}^n$ from*

$x \in \{0, 1\}^n$ by independently replacing each bit x_i of x ($1 \leq i \leq n$) with $1 - x_i$ with probability $1/n$.

In the analysis of the $(1+\lambda)$ EA and the simple island model, we make use of Markov chain mixing times to bound the probability that the algorithm is in a particular state (i. e. has a particular best-so-far individual) after a certain number of iterations. This has been applied to ant colony optimization in [18] and to analyze the $(\mu + 1)$ EA on MAZE in [13]; for completeness, we repeat the definitions of mixing and coupling times below.

DEFINITION 2 (MIXING TIME). Consider an ergodic Markov chain over a state space Ω with stationary distribution π . Let $p_x^{(t)}$ denote the distribution of the Markov chain t iterations after starting in state x , and let

$$t(\varepsilon) := \max_{x \in \Omega} \min \left\{ t : \frac{1}{2} \sum_{y \in \Omega} p_x^{(t)}(y) - \pi(y) \leq \varepsilon \right\}.$$

The mixing time t_{mix} of the Markov chain is then defined as $t_{\text{mix}} = t(1/(2\varepsilon))$.

DEFINITION 3 (COUPLING TIME). Consider a pair process $(X^{(t)}, Y^{(t)})$, where both $X^{(t)}$ and $Y^{(t)}$, viewed in isolation, are instances of the same Markov chain. Coupling time T_{xy} is the random time until the two processes, initialized in different states x and y , are in the same state for the first time:

$$T_{xy} = \min\{t : X^{(t)} = Y^{(t)} \mid X^{(0)} = x, Y^{(0)} = y\}.$$

As shown in [18], coupling time can be used as an upper-bound on mixing time:

LEMMA 4. The worst-case coupling time of a Markov chain over a state space Ω is an upper bound on its mixing time:

$$t(\varepsilon) \leq \min \left\{ t : \max_{x, y \in \Omega} P(T_{xy} > t) \leq \varepsilon \right\}.$$

Additionally, the following drift theorem, adapted from [4, 5], is useful when considering longer migration intervals.

THEOREM 5 (MULTIPLICATIVE DRIFT). Let $S \subseteq \mathbb{R}$ be a finite set of positive numbers with minimum $s_{\min} > 0$. Let $\{X^{(t)}\}_{t \geq 0}$ be a sequence of random variables over $S \cup \{0\}$. Let T be the random first point in time $t \geq 0$ for which $X^{(t)} = 0$.

Suppose there exists a $\delta > 0$ such that

$$E(X^{(t)} - X^{(t+1)} \mid X^{(t)} = s) \geq \delta s$$

for all $s \in S$ with $P(X^{(t)} = s) > 0$. Then for all $s_0 \in S$ with $P(X^{(0)} = s_0) > 0$,

$$E(T \mid X^{(0)} = s_0) \leq \frac{\ln(s_0/s_{\min}) + 1}{\delta}$$

Moreover, it holds that $P(T > (\ln(s_0/s_{\min}) + r)/\delta) \leq e^{-r}$ for any $r > 0$.

Several lemmas throughout this paper state that “a specific event occurs with high probability.” Definition 6 provides a more formal definition of this concept.

DEFINITION 6. An event E is said to occur with high probability if, for every constant $c > 0$, $P(E) = 1 - O(n^{-c})$.

In general, we say that an algorithm is able to track the optimum of the MAZE when it is able to construct the OPT_n individual in polynomial time (with high probability). Typically, this would correspond to maintaining at most a constant Hamming distance to the OPT_p intermediate optima during the oscillating phases.

3. $(1+\lambda)$ EA ON MAZE

In this section, we analyze the behavior of the $(1+\lambda)$ EA on MAZE. As long as λ is not too large (trivializing the problem by exploring the entire neighborhood of hamming distance 1 during every iteration), the EA is not able to track the optimum of the MAZE, and reverts to optimizing ONE-MAX. This is formalized in the following theorem, whose proof is inspired by the strategy taken in [13].

THEOREM 7. The $(1+\lambda)$ EA with $\lambda \in O(n^{1-\epsilon})$, for any constant $\epsilon > 0$, will with high probability lose track of the optimum of MAZE, i. e. with high probability it will require an exponential number of iterations to construct OPT_n .

We first note that the EA is able to find the $\text{OPT}_0 = 1^n$ optimum easily during the initial phase. For the subsequent n oscillating phases, we will show that the EA has at most a constant probability of ending each phase with $x^* = \text{OPT}_p$. Additionally, if an oscillating phase $p > n/2 + 3$ begins with $x^* \notin \text{ALL}_p$ (i. e. $x^* \neq \text{OPT}_{p-1}$), there is at least a constant probability that it will end with $x^* = 1^n$, and if this occurs, the $(1+\lambda)$ EA will with high probability not be able to construct the 0^n optimum at the end of the MAZE.

LEMMA 8. The $(1+\lambda)$ EA constructs $\text{OPT}_0 = 1^n$ with high probability after $O(n \log n)$ iterations during the initial phase.

PROOF. The initial phase consists of optimizing ONE-MAX, which a $(1+1)$ EA accomplishes in $O(n \log n)$ iterations with high probability. Increasing the size of the offspring population λ can only decrease time required; and we note that t_0 is asymptotically greater than $O(n \log n)$.

To prove this for both $(1+1)$ EA and $(1+\lambda)$ EA, Theorem 5 can be applied. Let $X^{(t)}$ be the number of zero-bits in the bit string x^* during iteration t :

$$E(X^{(t)} - X^{(t+1)} \mid X^{(t)} = x) \geq \frac{x(1 - 1/n)^{n-1}}{n} \geq \frac{1}{ne}x,$$

noting that for the $(1+\lambda)$ EA, this is a coarse lower bound stemming from the assumption that only the first offspring in each iteration may have a higher ONEMAX value than x^* . As $X^{(0)} \leq n$,

$$E(T) \leq \frac{\ln(n) + 1}{1/(ne)} = ne \ln(n) + ne.$$

Applying the tail-bound with $r = c_1 \ln n$ yields an upper bound on the probability of exceeding the expected number of iterations to reach $X^{(t)} = 0$ (i. e. to find the 1^n optimum) by more than $c_2 n \ln n$ additional iterations, where both c_1, c_2 are constants:

$$e^{-c_1 \ln n} = n^{-c_1}$$

i. e. 1^n is constructed by the EA in $ne \ln(n) + ne + c_2 n \ln n = O(n \log n)$ iterations with probability $1 - n^{-c_1}$. \square

LEMMA 9. *The probability that the $(1+\lambda)$ EA with $\lambda = O(n^{1-\epsilon})$ ends phase p with OPT_p as the best-so-far individual, given that $x^* \in ALL_p$ at least $c\lambda/n$ iterations before the end of phase p , where c is a sufficiently large constant, is at least $\Omega(1)$ and at most $1 - \Omega(1)$.*

PROOF. Once an ALL_p individual is constructed, all future best-so-far individuals within a phase will either be OPT_p or ALT_p , with single-bit mutations at specific iterations of the oscillation allowing the EA to switch between the two.

Let p_1 be the probability that a mutation flipping only one particular bit of x^* occurs in a single iteration of the $(1+\lambda)$ EA, i.e. the probability OPT_p is constructed from ALT_p (or vice versa) as at least one of the λ offspring individuals:

$$p_1 = 1 - (1 - (1 - 1/n)^{n-1} / n)^\lambda$$

which can be bounded by assuming $n \geq 2$, and using $1 - x \leq e^{-x}$ and $e^{-x} \leq 1 - x/2$ (for $x \leq 1$) for the upper and lower bounds respectively:

$$1 - (1 - 1/(en))^\lambda \leq p_1 \leq 1 - (1 - 1/(2n))^\lambda$$

$$\lambda/(2en) \leq 1 - e^{-\lambda/(en)} \leq p_1 \leq \lambda/(2n).$$

Consider the phase as a series of oscillations, each consisting of three iterations. During the first two iterations of an oscillation, OPT_n has a higher fitness value than ALT_n (and vice versa during the third and final iteration). We then consider the probabilities p_A and p_O of the best-so-far individual switching to ALT_p from OPT_p , and to OPT_p from ALT_p , respectively, over a single oscillation. For p_A , a specific one-bit mutation has to occur during the final iteration of the oscillation, while for p_O , a specific one-bit mutation has to occur during either the first or the second iteration of the oscillation, and a specific one-bit mutation must not occur during the third iteration:

$$p_A = p_1$$

$$p_O = (p_1 + (1 - p_1)p_1)(1 - p_1)$$

$$= 2p_1 - 3p_1^2 + p_1^3$$

$$= p_1(2 - 3p_1 + p_1^2)$$

The identity of the best-so-far individual x^* of the $(1+\lambda)$ EA, observed at the end of each OPT - OPT - ALT oscillation can be modeled using a two-state Markov chain, with one state corresponding to $x^* = OPT_p$, the other to $x^* = ALT_p$, and transition probabilities between the states as above.

Let π_O and $\pi_A = 1 - \pi_O$ be the steady-state probabilities of $x^* = OPT_p$ and $x^* = ALT_p$ respectively; as the Markov chain only contains two states, the following holds for the steady-state:

$$\pi_O p_A = \pi_A p_O$$

$$\pi_O p_1 = (1 - \pi_O)p_1(2 - 3p_1 + p_1^2)$$

$$\pi_O = \frac{2 - 3p_1 + p_1^2}{3 - 3p_1 + p_1^2}$$

i.e. π_O approaches a constant; we note that $\pi_O \leq 2/3$, and as $\lambda \in o(n)$ and hence $p_1 \leq \lambda/(2n) \leq 0.5$, $\pi_O \geq 3/7$.

Over time, the probability of OPT_p being the best-so-far individual at the end of an oscillation will approach the

steady-state probability π_O . By applying Lemma 4, coupling time can be used as a bound on the Markov chain mixing time, and hence on the number of oscillations required until this probability is within a sufficiently small ϵ of π_O .

For our two-state Markov chain, the pair processes used to determine the worst-case coupling time are initialized in opposite states (i.e. one with $x^* = OPT_p$ and the other with $x^* = ALT_p$). During each step, the probability that coupling does not occur is the sum of probabilities that neither, or both, of the processes changed state:

$$\max P(T_{xy} > t) = (p_A p_O + (1 - p_A)(1 - p_O))^t$$

$$= (1 + 2p_A p_O - p_A - p_O)^t$$

$$= (1 - p_1(3 - 7p_1 + 7p_1^2 - 2p_1^3))^t$$

$$\leq \left(1 - \frac{\lambda}{2en} \left(3 - \frac{7\lambda}{2n} + \frac{7\lambda^2}{4e^2 n^2} - \frac{2\lambda^3}{8n^3}\right)\right)^t$$

$$< \left(1 - \frac{\lambda}{2en}\right)^t$$

by recalling that $\lambda = o(n)$, and observing that the expression in the inner parentheses is greater than 1 when $\lambda/n \leq 0.5$. Then, an upper bound on the coupling time is:

$$t(\epsilon) \leq \min \{t : (1 - \lambda/(2en))^t \leq \epsilon\}$$

After at most $t(0.01) < 9.22en/\lambda$ steps of the Markov chain, or at most $76n/\lambda$ iterations of the $(1+\lambda)$ EA, the probability that $x^* = OPT_p$ is therefore within $[\pi_O - 0.01, \pi_O + 0.01]$, which, as $3/7 \leq \pi_O \leq 2/3$, proves the lemma. \square

LEMMA 10. *If a phase $p > n/2 + 3$ begins with $x^* \notin ALL_p$ satisfying $f(x^*) > n - p + 1$, the $(1+\lambda)$ EA with offspring population size $\lambda = O(n^{1-\epsilon})$, for any constant $\epsilon > 0$, ends the phase with $x^* = 1^n$ with at least constant probability.*

PROOF. At the start of phase p , x^* contains strictly more 1-bits than any individual in ALL_p , and the Hamming distance between x^* and the closest ALL_p individual is at least 1. Let $p_R \leq p_1$ be the probability that an ALL_p individual is constructed during an iteration (where p_1 is the probability of at least one of the λ offspring being constructed via a specific single-bit mutation, as used in the proof of Lemma 9).

We want to consider the probability that the number of 1-bits in x^* exceeds that in any ALL_p individual by at least 3 (i.e. is at least $n + 4 - p \leq n/2$) before an ALL_p individual is constructed. An individual with a greater ONEMAX value is constructed via a single-bit mutation with probability at least p_L :

$$p_L \geq 1 - (1 - n/2 \cdot (1 - 1/n)^{n-1} / n)^\lambda$$

$$\geq 1 - 0.75^\lambda \geq 1/4$$

as there are at least $n/2$ 0-bits that can be flipped to increase ONEMAX value. We note that after at most 2 ONEMAX-improvements, constructing the closest ALL_p individual requires at least 3 1-bits to be flipped simultaneously.

Consider the probability that two ONEMAX improvements occur before an ALL_p individual is constructed. Let L be the event that a ONEMAX-improving single-bit mutation occurs,

and R be the event that an ALL_p individual is constructed:

$$\begin{aligned} P(R \mid R \vee L) &\leq \frac{p_R}{p_R + p_L} \leq \frac{\lambda}{2n \left(\frac{\lambda}{2en} + 1/4 \right)} \\ &= \frac{2e\lambda}{en + 2\lambda} \in O(\lambda/n) \end{aligned}$$

assuming, as before, $\lambda \in O(n^{1-\epsilon})$, where $\epsilon > 0$.

Let D be the event that $f(x^*) \geq n + 4 - p$, i.e. either $x \in \text{ALL}_p$, or x^* has at least three more 1-bits than any individual in ALL_p :

$$P(x \notin \text{ALL}_p \mid D) \geq (1 - P(R \mid R \vee L))^2 \in \Omega(1),$$

by pessimistically requiring that at least two separate improvements to the ONEMAX value occurred (L) without an ALL_p individual being constructed (R).

Let T_D be the number of iterations between the start of the phase and the first time D occurs. We can upper-bound $E(T_D)$ considering the time required for two one-bit mutations that increase ONEMAX value of x^* occur; noting that there are $\Omega(n)$ 0-bits in x^* at the start of the phase, the probability that a one-bit mutation improving the ONEMAX value occurs during an iteration is at least a constant, and hence $E(T_D) \leq c$ for some constant c . Applying Markov's inequality, $P(T_D > 2c) \leq 1/2$, and hence the probability that D occurs within $c' \cdot cn = O(n)$ iterations is at least $1 - 2^{-c'n} = 1 - 2^{-\Omega(n)}$.

Once D occurs and $x^* \notin \text{ALL}_p$, and if no ALL_p individuals are constructed in subsequent iterations, we can consider the fitness function equivalent to ONEMAX , and per Lemma 8, 1^n is found with high probability in $O(n \log n)$ iterations. As constructing an ALL_p individual after D has occurred requires at least 3 specific bits to be mutated simultaneously (which occurs with probability at most $1/n^3$ during each mutation), the probability that an ALL_p individual is constructed within the $O(n \log n)$ iterations before $x^* = 1^n$ is at most $O(\lambda n \log n / n^3) = o(1)$, i.e. with at least constant probability, this will not occur, and $x^* = 1^n$ will be found as in Lemma 8. Once $x^* = 1^n$, constructing an ALL_p individual requires at least $n/2$ specific bits to mutate simultaneously, which with high probability will not occur in a polynomial number of iterations.

Thus, the $(1+\lambda)$ EA has at least a constant probability of ending the phase with $x^* = 1^n$. \square

These lemmas can then be combined to prove Theorem 7.

PROOF OF THEOREM 7. With high probability, $\text{OPT}_0 = 1^n$ is found during phase 0 per Lemma 8. At the start of each subsequent phase p , $f(x^*) > n - p$, as only individuals in the ALL sets of the preceding phases can be accepted while decreasing the number of 1-bits in x^* , and the minimum ONEMAX value of any individual in sets $\text{ALL}_0, \dots, \text{ALL}_{p-1}$ is $n - p + 1$. Furthermore, if $x^* \notin \text{ALL}_p$, $f(x^*) > n - p + 1$, as this excludes $x^* = \text{OPT}_{p-1}$, which had the lowest fitness value of all individuals in the union of the previous ALL sets.

If $x^* \neq 1^n$ at the start of phase $p \geq n/2 + 3$, the phase has at least a constant probability of ending with $x^* \neq \text{OPT}_p$ per Lemma 9, and hence $x^* \notin \text{ALL}_{p+1}$.

If phase $p + 1$ begins with $x^* \notin \text{ALL}_{p+1}$, it has at least a constant probability of ending with $x^* = 1^n$ per Lemma 10.

Thus, at least a constant fraction of $\Omega(n)$ phases beyond $n/2 + 3$ have at least a constant probability of ending with

$x^* = 1^n$; i.e. with high probability, at least one of those phases will end with $x^* = 1^n$. Constructing an ALL_p individual from 1^n in future phases requires at least $\Omega(n)$ bits to be flipped simultaneously, which with high probability does not occur in polynomial time. \square

We note that the proof of Theorem 7 relies on $\lambda = o(n)$ primarily in Lemma 9's bounds on p_1 , although, if λ is increased a little further to $\Omega(n \log n)$, the behavior described by Lemma 10 would also no longer occur, allowing the $(1+\lambda)$ EA to recover from any phase which ends with an ALT_p with high probability.

THEOREM 11. *The $(1+\lambda)$ EA with offspring population size $\lambda \geq c_1 n \log n$, where $c_1 > 0$ is a sufficiently large constant, is able to track the optimum of the MAZE function, constructing OPT_n at the end of the MAZE with high probability.*

PROOF. We pessimistically assume that all of the n oscillating MAZE phases which start with, or discover an ALL_p individual, end with $x^* = \text{ALT}_p$ rather than $x^* = \text{OPT}_p$. Notably, the initial phase ends with $x^* = 1^n$ with high probability per Lemma 8, so the first oscillating phase starts with an ALL_1 individual.

Consider the probability p_r that an oscillating phase p which begins with $x^* = \text{ALT}_{p-1}$ recovers immediately – i.e. constructs (and hence also accepts) an ALL_p individual in its first iteration. This probability can be lower-bounded by the probability that ALT_p is constructed from ALT_{p-1} , i.e. a specific one-bit mutation occurs:

$$\begin{aligned} p_r &= 1 - (1 - (1 - 1/n)^{n-1}/n)^{c_1 n \log n} \\ &\geq 1 - (1 - 1/(ne))^{c_1 n \log n} \\ &\geq 1 - e^{-c_2 \log n} = 1 - n^{-c_2}, \end{aligned}$$

where $c_2 > 0$ is a constant absorbing c_1 and e .

As there are only n oscillating phases, at most n such recoveries are necessary to allow the last phase to end with $x^* \in \text{ALL}_p$. We can then lower-bound the probability p_R that all phases that begin with $x^* = \text{ALT}_{p-1}$ construct an ALL_p individual in their first iteration, and hence also that all phases end with an ALL_p individual conditioned on the initial phase finding $x^* = 1^n$:

$$p_R \geq 1 - n \cdot n^{-c_2} \geq 1 - n^{1-c_2}.$$

Thus, by choosing a sufficiently large constant c_1 in $\lambda = c_1 n \log n$, we can ensure that the $(1+\lambda)$ EA ends the first iteration following a phase transition with $x^* \in \text{ALL}_p$ with high probability, and is therefore able to track the optimum of the MAZE. \square

4. A SIMPLE ISLAND MODEL

Splitting the λ offspring onto λ islands, which only compare the fitness values of their x^* individuals periodically (for instance, every τ iterations, where $\tau > 0$ is the migration interval), allows the resulting island model to track MAZE even with a modest λ . In this section, we consider the effect of various migration schedules on how the island model is able to track the MAZE.

To begin with, consider an island model where migration occurs on the first iteration of every phase, i.e. every $\tau = t_0$ iterations of the MAZE. This ensures that an ALL_p individual migrates to all islands if any of the islands end the preceding phase with $x^* = \text{OPT}_{p-1}$.

THEOREM 12. *An island model with $\lambda = c \log n$ islands, where c is a sufficiently large constant, each island running a (1+1) EA, and migration occurring during the first iteration of every phase (i. e. with migration interval $\tau = t_0$) is able to find the OPT_n optimum of the MAZE with phase length $t_0 = kn^3 \log n$ in polynomial time with high probability.*

PROOF. We note that individually, the islands behave exactly like a (1+1) EA on MAZE, and the effects of synchronization are limited to selecting the best-so-far individual at the start of each phase, and propagating it to all islands. Thus, as long as any island ends phase p with $x^* = \text{OPT}_p$, all islands will begin phase $p+1$ with $x^* \in \text{ALL}_{p+1}$.

The initial ONEMAX optimum, OPT_0 , is found during phase 0 on each island with high probability. Lemma 9, applied with $\lambda = 1$, states that the probability that an island that begins phase p with $x^* \in \text{ALL}_p$ ends the phase with $x^* = \text{OPT}_p$ with at least constant probability; let $p_s = \Omega(1)$ be a lower bound on this probability, and p_f an upper bound on the probability that all λ islands end the phase with $x^* \neq \text{OPT}_p$. As long as the latter event does not occur, all islands will begin the next phase with an ALL_{p+1} individual due to migration, allowing the argument to be repeated inductively. A union bound can then be used to upper-bound the probability of failing in any of the n phases:

$$p_f \leq (1 - p_s)^\lambda$$

$$np_f \leq n(1 - p_s)^\lambda \leq nc_1^{c_2 \log n} \leq n^{1+c_2 \log c_1}$$

noting that for any constant $c > 0$, choosing $c_2 \geq -(1+c)/\log c_1$ (recall that $c_1 \leq p_f < 1$, so $\log(c_1)$ is negative) results in $p_f \leq n^{-c}$.

Thus, with $\lambda = c_2 \log n$ islands, where c_2 is a sufficiently large constant, at least one island ends each phase with $x^* = \text{OPT}_p$ with high probability; this individual is propagated to all other islands at the start of the next phase, allowing OPT_n to be constructed and propagated to all islands at the end of the last phase. \square

4.1 Shorter migration intervals

It is also possible to track MAZE with shorter migration intervals $\tau < t_0$. We first consider the case where migration may occur multiple times during a phase, as long as the final migration occurs at least $\Omega(n)$ iterations before the end of the phase, allowing the islands time to reconstruct OPT_p if migration propagated ALT_p to all islands.

THEOREM 13. *With the migration interval $\tau \leq t_0$, $\lambda = c_1 \log n$ islands are sufficient to track the optimum of the MAZE as long as no migration occurs during $c_2 n$ iterations preceding any phase transition, where c_1 and c_2 are sufficiently large constants, and migration occurs at least once during each phase.*

PROOF. We follow the proof of Theorem 12, and pessimistically assume that the last synchronization during each phase occurs $c_2 n$ iterations before the end of the phase, where $c_2 > 0$ is a sufficiently large constant, during an ALT-favoring iteration, and propagates the ALT individual to all islands.

Lemma 9 can then be applied: on each island, the probability of ending phase p with $x^* = \text{OPT}_p$ is then at least a constant greater than 0, as $x^* = \text{ALT}_p \in \text{ALL}_p$ has been propagated to the island at least $c_2 n$ iterations before the end of the phase. Thus, the situation at the end of the

phase returns to that considered in Theorem 12: each of $\lambda = \Omega(\log n)$ islands has at least a constant probability of ending phase p with $x^* = \text{OPT}_p$.

With a large-enough λ , we can with high probability conclude that during each phase p , migration ensures that all islands have an ALL_p individual as the best-so-far solution at least $c_2 n$ iterations before the end of the phase, and therefore each phase will with at least one island having $x^* = \text{OPT}_p$. That island will therefore have an ALL_{p+1} individual during the next phase, which will migrate to the any islands which lose track of the MAZE during the next phase, allowing the argument to be repeated inductively for each of n phases. \square

Additionally, if migration is only allowed to occur during iterations assigning the OPT individual the greatest fitness value, ALT_p individuals can never migrate to islands where $x^* = \text{OPT}_p$; in this case, migration can occur close to the end of each phase without negative consequences.

THEOREM 14. *When migration occurs only during iterations when $f(\text{ALT}_p) < f(\text{OPT}_p)$, and occurs for the first time at least $c_1 n$ iterations before the end of each phase, $\lambda = c_2 \log n$ islands, where c_1 and c_2 are sufficiently large constants, are sufficient to track the optimum of the MAZE.*

PROOF. The initial ONEMAX phase still succeeds with high probability, as synchronization cannot decrease the fitness of x^* on each island, and will instead only increase the fitness of the islands lagging behind the current global best-so-far.

During the subsequent MAZE phases, synchronization only occurs when OPT individuals have the highest fitness value, and therefore cannot replace an OPT best-so-far individual on an island with an ALT individual; thus, it can only increase the probability that $x^* = \text{OPT}_p$ on an island at any given time in comparison with the situation considered in Theorem 12.

Thus, the argument from the proof of Theorem 12 applies: by selecting a sufficiently-large constant for $\lambda = c \log n$, the probability that all islands end phase p with $x^* \neq \text{OPT}_p$ can be made small enough to ensure that this event does not occur with high probability during the n phases of MAZE: then, as long as at least one island ends each phase with $x^* = \text{OPT}_p$, that island's $x^* \in \text{ALL}_{p+1}$ will be propagated to the other islands, and per Lemma 9, $c_1 n$ iterations later, every island will once again have at least a constant probability of having $x^* = \text{OPT}_{p+1}$. \square

Thus, we have shown that $\lambda = c_2 \log n$ islands running a (1+1) EA are sufficient to track the optimum of the MAZE with varying migration intervals, as long as migration occurs at least $c_1 n$ iterations before a phase transition, and any migration occurring within $c_1 n$ iterations prior to a phase transition occurs only while the OPT individual has a higher fitness value than the ALT individual. Both theorems rely on a mixing time argument to ensure that at least a constant fraction of the $\Omega(\log n)$ islands end the phase with $x^* = \text{OPT}_p$, and migration is used to repopulate any islands that lose track of the MAZE prior to the next phase transition.

4.2 Longer migration intervals

A straightforward consequence of Theorem 12 is that migration phases longer than the MAZE phase length are also

viable: there is no need to repopulate all islands after *every* phase transition, and choosing e.g. $\tau = ct_0$ where $c > 1$ is a constant is also viable with $\Omega(\log n)$ islands. However, the migration interval cannot be increased much further than that: we will show that if $\tau \geq ct_0 \log \lambda$, $\lambda = O(\log n)$ islands are no longer sufficient to track the MAZE.

THEOREM 15. *For $\tau = c_1 t_0 \log \lambda$, where $c_1 > 0$ is a sufficiently large constant, $\lambda = O(\log n)$ islands are not sufficient to track the optimum of the MAZE.*

PROOF. Consider an interval of $ct_0 \log \lambda$ iterations during which no migration will occur, where $c_1 \geq c > 0$ is a sufficiently large constant, starting at the beginning of some phase p , such that $n/2 + 3 < p < n - \log \lambda$. Pessimistically assume that before the start of phase p , none of the $\lambda = c \log n$ islands lost track of the MAZE, and thus all islands begin phase p with at least a constant probability of having an ALL_p individual per Lemma 9.

Considering each island individually, each of the $\log \lambda$ phases in the interval has at least a constant probability of ending with $x^* \neq \text{OPT}_p$, causing the next phase to have at least a constant probability of ending with $x^* = 1^n$; let $p_L > 0$ be a constant lower bound on the probability of each phase ending with $x^* = 1^n$. Let $X^{(t)}$ be the number of islands with $x^* \neq 1^n$ at the start of phase $p+t$ (and $X^{(0)} = \lambda$); it then holds that:

$$E(X^{(t)} - X^{(t+1)} \mid X^{(t)}) \geq p_L X^{(t)}.$$

Applying Theorem 5, the expected number of phase transitions $T = \min_t \{X^{(t)} = 0\}$ until all islands have lost track of the MAZE (i.e. have $x^* = 1^n$) is:

$$E(T \mid X^{(0)}) \leq \frac{\ln(X^{(0)}) + 1}{1 - p_L} = O(\ln \lambda).$$

By applying Markov's inequality, we can bound $P(T \geq 2E(T \mid X^{(0)})) \leq 1/2$, and hence, by choosing a sufficiently large constant c such that $\tau = c \log \lambda \geq 2E(T \mid X^{(0)})$, lower-bound the probability of all islands ending the interval with $x^* = 1^n$ by at least $1/2$. As there are $\Omega(n/(\log \lambda)) = \Omega(n/(\log \log n))$ such intervals following phase $p > n/2 + 3$, the probability that the island model does lose track of the MAZE in at least one of these intervals is at least $1 - 2^{-\Omega(n/(\log \log n))}$. \square

5. CONCLUSIONS

We have presented a first runtime analysis of parallel EAs in dynamic optimization. A simple island model with λ islands and length of the migration interval τ was studied on the dynamic benchmark function MAZE. In the case of extreme communication, i.e., $\tau = 1$, even a large number of islands does not allow efficient tracking of the optimum. However, with a carefully chosen value for τ , already a logarithmic number of islands was proven to be sufficient for efficient tracking. Finally, the relationship of τ , λ , and the ability of the island model to track the optimum was investigated more closely. Our results indicate that the careful choice of the migration policy, and more generally, the communication in parallel EAs, can be significantly more advantageous than a large population.

In future work, we would like to study parallel EAs on different dynamic optimization problems in order to understand the interplay of migration intervals and number of islands more thoroughly. As our positive results are crucially

dependent on a proper choice of τ , it may also be worth studying adaptive or even self-adaptive choices of the migration interval in order to automatically determine a good value for τ . Here the adaptive model suggested in [14] could be interesting.

6. REFERENCES

- [1] Enrique Alba, Gabriel Luque, and Sergio Nesmachnow. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.
- [2] Enrique Alba, Amir Nakib, and Patrick Siarry. *Metaheuristics for Dynamic Optimization*. Studies in Computational Intelligence. Springer, 2013.
- [3] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing, 2011.
- [4] Benjamin Doerr and Leslie Ann Goldberg. Adaptive drift analysis. *Algorithmica*, 65(1):224–250, 2013.
- [5] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.
- [6] Stefan Droste. Analysis of the (1+1) EA for a dynamically bitwise changing OneMax. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, pages 909–921. Springer, 2003.
- [7] Thomas Jansen and Ulf Schellbach. Theoretical analysis of a mutation-based evolutionary algorithm for a tracking problem in the lattice. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 841–848. ACM Press, 2005.
- [8] Thomas Jansen and Christine Zarges. Evolutionary algorithms and artificial immune systems on a bi-stable dynamic optimisation problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 975–982. ACM Press, 2014.
- [9] Mostepha Redouane Khoudja, Briseida Sarasola, Enrique Alba, Laetitia Jourdan, and El-Ghazali Talbi. Multi-environmental cooperative parallel metaheuristics for solving dynamic optimization problems. In *Proceedings of 2011 IEEE International Parallel & Distributed Processing Symposium*, pages 395–403. IEEE Press, 2011.
- [10] Timo Kötzing, Andrei Lissovoi, and Carsten Witt. (1+1) EA on generalized dynamic OneMax. In *Proceedings of Foundations of Genetic Algorithms Workshop (FOGA 2015)*. ACM Press, 2015. To appear.
- [11] Timo Kötzing and Hendrik Molter. ACO beats EA on a dynamic pseudo-boolean function. In *Proceedings of Parallel Problem Solving from Nature XII (PPSN '12)*, pages 113–122. Springer, 2012.
- [12] Jörg Lässig and Dirk Sudholt. General upper bounds on the runtime of parallel evolutionary algorithms. *Evolutionary Computation*, 22(3):405–437, 2014.
- [13] Andrei Lissovoi and Carsten Witt. MMAS versus population-based EA on a family of dynamic fitness functions. *Algorithmica*, 2015. In press, DOI

<http://dx.doi.org/10.1007/s00453-015-9975-z>,
preliminary version in GECCO 2014.

- [14] Andrea Mambrini and Dirk Sudholt. Design and analysis of adaptive migration intervals in parallel evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 1047–1054. ACM Press, 2014.
- [15] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Natural Computing Series. Springer, 2010.
- [16] Trung Thanh Nguyen, Shengxiang Yang, and Jürgen Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 2012.
- [17] Philipp Rohlfshagen, Per Kristian Lehre, and Xin Yao. Dynamic evolutionary optimisation: An analysis of frequency and magnitude of change. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 1713–1720. ACM Press, 2009.
- [18] Dirk Sudholt. Using markov-chain mixing time estimates for the analysis of ant colony optimization. In *Proceedings of Foundations of Genetic Algorithms Workshop (FOGA 2011)*, pages 139–150. ACM Press, 2011.