Design and Analysis of Adaptive Migration Intervals in Parallel Evolutionary Algorithms

Andrea Mambrini
School of Computer Science
University of Birmingham
Birmingham B15 2TT, United Kingdom

Dirk Sudholt
Department of Computer Science
University of Sheffield
Sheffield S1 4DP, United Kingdom

ABSTRACT

The migration interval is one of the fundamental parameters governing the dynamic behaviour of island models. Yet, there is little understanding on how this parameter affects performance, and how to optimally set it given a problem in hand. We propose schemes for adapting the migration interval according to whether fitness improvements have been found. As long as no improvement is found, the migration interval is increased to minimise communication. Once the best fitness has improved, the migration interval is decreased to spread new best solutions more quickly. We provide a method for analysing the expected running time and the communication effort, defined as the expected number of migrants sent. Example applications of this method to common example functions show that our adaptive schemes are able to compete with, or even outperform, the optimal fixed choice of the migration interval, with regard to running time and communication effort.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

Keywords

Parallel evolutionary algorithms, island model, migration interval, runtime analysis, theory

1. INTRODUCTION

Evolutionary algorithms (EAs) have given rise to many parallel variants [13, 20] fuelled by the rapidly increasing number of CPU cores and the ready availability of computation power through GPUs and cloud computing. Parallelization provides a cost-effective approach to solving problems in real time and for tackling large-scale problems.

There are many variants of parallel evolutionary algorithms, from parallelising function evaluations on multiple processors to fine-grained models such as cellular EAs and coarse-grained EAs such as island models [20, 13]. In the latter approach, multiple populations evolve independently for a certain period of time. Every τ generations, for a parameter τ called *migration interval*, individuals migrate between islands to coordinate their searches. Communication takes place according to a spatial structure, a topology connecting populations. Common topologies include rings, two-dimensional grids/toroids, hypercubes, or complete graphs.

Island models are popular optimisers for several reasons:

- Multiple communicating populations can make the same progress as a single population in a fraction of the time, speeding up computation.
- Small populations can be simulated faster than large populations, reducing execution times [1].
- Periodic communication only requires small bandwidth, leading to low communication costs.
- Solution quality is improved as different populations can explore different regions of the search space.

The usefulness of parallel populations has been demonstrated in thousands of successful applications ranging from language tagging, circuit design, scheduling and planning to bioinformatics [13, 3].

However, designing an effective parallel evolutionary algorithm can be challenging as the method and amount of communication needs to be tuned carefully. Too frequent communication leads to high communication costs, and it can compromise exploration. Too little communication means that the populations become too isolated and unable to coordinate their searches effectively. There is agreement that even the effect of the most fundamental parameters on performance is not well understood [13, 3].

We make a contribution towards finding good values for the migration interval, the parameter describing the frequency of migration. We propose adaptive schemes that adjust the migration interval, depending on whether islands have managed to find improvements during the last migration interval or not. The goal is to reduce communication, while not compromising the exploitation of good solutions. The main idea is: if an island has improved its current best fitness, migration is intensified to spread this solution to other islands. Otherwise, islands decrease the frequency of migration in order to avoid large communication costs.

Two different adaptive schemes are proposed. In both of them islands have individual migration intervals which are adapted throughout the run. In Scheme A if an island has not improved its current best fitness during the last migration interval, its migration interval is doubled. Once an

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'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598347.

improvement is found, the migration interval is set to 1 to communicate this new solution quickly. In Scheme B an island also doubles the migration interval when no improvement was found, while when an improvement is found, it halves τ at the end of the current migration interval.

We show that doubling the migration interval guarantees for elitist EAs that the number of migrations from an island is logarithmic in the time this island spends on a certain fitness level, i. e. for any value of the current best fitness.

We contribute a rigorous analytical framework that yields upper bounds on the expected optimisation time and the expected communication effort, defined as the total number of migrants sent. This is done for fixed migration intervals in Section 3, Scheme A in Section 4, and Scheme B in Section 5. Our adaptive schemes are then compared in Section 6 against the best fixed values of the migration interval for classical test problems. The results reveal that our adaptive schemes are able to match or even outperform the best fixed migration intervals with regard to the expected parallel time and the expected communication effort.

1.1 Related Work

This paper is in line with recent theoretical research on the running time of parallel EAs. Lässig and Sudholt [8] presented a method for analysing speedups in island models, and applications to combinatorial problems [10]. Neumann, Oliveto, Rudolph, and Sudholt [15] considered the benefit of crossover during migration for artificial problems and instances of the Vertex Cover problem. Mambrini, Sudholt, and Yao [14] studied the running time and communication effort of homogeneous and heterogeneous island models for finding good solutions for the NP-hard Set Cover problem.

Skolicki and De Jong [19] investigated the impact of the migration interval and the number of migrants on performance, while different migration policies were compared by Araujo and Merelo [4]. Bravo, Luque and Alba [5] studied the effect of the migration interval when tackling dynamic optimization problems.

Alba and Luque [2] analysed growth curves and takeover times based on migration intervals and migration topologies, showing how quickly good solutions spread in an island model. Theoretical analyses were presented by Rudolph [18].

Lässig and Sudholt [11] presented a theoretical analysis and a problem where island models excel over both panmictic populations as well as independent runs. This requires a delicate choice of the migration interval, and performance degrades drastically when suboptimal parameter values are being used. This again emphasises the importance of this parameter for the performance of island models.

Hong, Lin, Liu, and Lin [6] and Lin, Hong, and Lin [12] presented a fitness-based adaptive migration scheme: each island compares its increase of its best fitness over the last migration interval with the same quantity from the migration interval before. If the new fitness increase is larger than the old one, the migration interval is increased by some constant value. Otherwise, it is decreased by the same constant value. A preliminary experimental study on random 0/1 knapsack problem instances showed that the adaptive scheme can lead to a better final fitness [6].

The difference in our scheme is that we adapt the migration interval in opposite directions: we decrease the migration interval in case of good fitness gains and increase it otherwise. Furthermore, our changes to the migration interval are most drastic than their additive changes.

Osorio, Luque, and Alba [17, 16] presented adaptive schemes for the migration interval, which aim for convergence at the end of the run (for runs of fixed length). The migration interval is set according to growth curves of good individuals and the remaining number of generations; migration intensifies towards the end of a run. They obtained competitive performance results, compared to optimal fixed parameters, for MAX-SAT instances [16]. Our perspective is different as we do not optimise for fixed-length runs.

Finally, Lässig and Sudholt [9] presented schemes for adapting the number of islands during a run of an island model (and offspring populations in a $(1+\lambda)$ EA). Scheme A doubles the number of islands if no improvement was found in a generation. Otherwise, the number of islands drops to one island. Scheme B also doubles the number of islands when no improvement is found, and halves it otherwise. Both schemes achieve optimal or near-optimal parallel running times, while not increasing the total number of function evaluations by more than a constant factor. Our schemes for adapting the migration interval are inspired by this work.

2. PRELIMINARIES

We define the parallel EAs considered in this work, which contain our adaptive schemes. Our analytical framework is applicable to all kinds of elitist EAs: EAs that do not lose their current best solution. We define our schemes for maximisation problems.

Scheme A (Algorithm 1) maintains a migration interval τ_i for each island. As soon as the current best fitness on an island has improved, the migration interval for that island drops to 1, so that the island will communicate this solution to its neighbouring islands at the end of the generation. If no improvement of the current best fitness is found after τ_i generations, the migration interval τ_i doubles.

Algorithm 1 Elitist parallel EA with adaptive Scheme A

1: Initialize λ islands $P_1^1, \dots, P_1^\lambda$ uniformly at random and let $\tau_i := 1$ for each island.
2: Let f_i^* be the best fitness value for each island $P_1^1, \dots, P_1^\lambda$.
3: **for** $t := 1$ to ∞ **do**
4: **for all** $1 \le i \le \lambda$ in parallel **do**
5: Simulate one generation of the EA and create a new population P_{t+1}^i.
6: Update P_{t+1}^i by selecting a new population from P_{t+1}^i and all immigrants, keeping the best individuals.
7: Let $f_i^{*\prime}$ be the best fitness value for P_{t+1}^i.
8: **if** $f_i^{*\prime} > f_i^*$ **then**
9: Let $\tau_i := 1$
10: Let $f_i^* := f_i^{*\prime}$
11: **else**
12: Let $\tau_i := \tau_i \cdot 2$
13: **end if**
14: **if** $t \bmod \tau_i = 0$ **then**
15: Send a copy of a fittest offspring in P_{t+1}^i to all neighbouring islands.
16: **end if**
17: **end for**
18: **end for**

For the purpose of a theoretical analysis, we assume that all islands run in synchronicity: the t-th generation is executed on all islands at the same time. However, this is not a restriction of our adaptive scheme as it can be applied in asynchronous parallel architectures using message passing for implementing migration.

Inspired by [9], we also consider a Scheme B where the migration interval is being halved once an improvement has been detected. Thus line 9 in Algorithm 1 is replaced with "Let $\tau_i := \tau_i/2$". The advantage of Scheme B is that it uses less communication than Scheme A, and if there is a good region in the parameter space of τ, our hope is that it will maintain a good parameter value in that region over time.

We provide general methods for analysing the expected parallel time and the expected communication effort for arbitrary elitist EAs that migrate copies of selected individuals (the original individuals remain on their island). The parallel time is defined as the number of generations until a global optimum is found and denoted T^{par}. The communication effort T^{com} is defined as the total number of individuals migrated until a global optimum is found. For simplicity and ease of presentation, we assume that each migration only transfers one individual; if $\nu > 1$ individuals migrate, the communication effort has to be multiplied by ν.

In order to demonstrate and illustrate this approach, we consider one simple algorithm in more detail: following [9], the parallel (1+1) EA is a special case where each island runs a (1+1) EA.

In terms of communication topologies, for Scheme A we consider general graphs on λ vertices and the following common graphs. A unidirectional ring is a ring with edges going in one direction. A grid graph contains undirected edges with vertices arranged on a 2-dimensional grid. A torus can be regarded a grid where edges wrap around horizontally and vertically. A hypercube graph of dimension d contains 2^d vertices. Each vertex has a d-bit label, and vertices are neighboured if and only if their labels differ in exactly one bit. The complete graph contains all possible edges. For Scheme B we only consider complete topologies as on sparse topologies migration intervals may adapt too slowly (see Section 5). Then all islands use the same migration interval.

The diameter $\text{diam}(G)$ of a graph G with λ vertices is defined as the largest number of edges on any shortest path between two vertices. The unidirectional ring has the largest diameter of λ. The diameter of any grid or torus graph with side lengths $\sqrt{\lambda} \times \sqrt{\lambda}$ is at most $2\sqrt{\lambda}$. The diameter of a $(\log \lambda)$-dimensional hypercube is $\log \lambda$, and that of a complete topology is 1.

3. FIXED MIGRATION INTERVALS

In order to compare our adaptive schemes against fixed migration intervals, we first need to investigate the latter. Lässig and Sudholt [8, 11] presented general upper bounds for the parallel optimisation time of island models with different topologies. Their method is based on the so-called fitness-level method, also known as fitness-based partitions [21].

Throughout this work, we use a special case of this method: without loss of generality consider a problem with fitness values $1, \ldots, m$. Consider fitness-level sets $A_1, \ldots, A_m$ such that A_i contains all points with fitness i. In particular, A_m contains all global optima. We further assume that, if the current best individual of a population is in A_i, there is a lower bound s_i for the probability of finding a higher fitness level $A_{i+1} \cup \cdots \cup A_m$ in one generation. It is easy to show that then $\sum_{i=1}^{m-1} 1/s_i$ is an upper bound for the expected running time of an elitist EA.

Lässig and Sudholt [8, 10] showed how upper bounds on the parallel optimisation time can be derived through general functions on these success probabilities $s_1, \ldots, s_{m-1}$. They considered migration in every generation ($\tau = 1$) [10] as well as probabilistic migration, where every island independently decides for each neighbouring island whether migration occurs, and the probability for a migration is a fixed parameter p [11]. The following theorem is an adaptation of their results for periodic migration with migration interval τ [11]. The results for the expected communication effort on a topology with edge set E follow from multiplying the expected parallel time by $|E|/\tau$, as this term reflects the average number of migrated individuals across the topology in one generation. The upper bounds on the expected parallel time can be derived as in [11]. We do not give a formal proof due to space restrictions.

THEOREM 1. *Consider an island model with λ islands where each island runs an elitist EA. Every τ iterations each island sends a copy of its best individual to all neighbouring islands. Each island incorporates the best out of its own individuals and its immigrants. For fitness-level sets $A_1, \ldots, A_m$, A_i containing all points of the i-th fitness value, let s_i be a lower bound for the probability that in one generation an island in A_i finds a search point in $A_{i+1} \cup \cdots \cup A_m$. Then the expected parallel optimization time $\mathrm{E}(T^{\text{par}})$ and the expected communication effort $\mathrm{E}(T^{\text{com}})$ are at most*

$$\mathrm{E}(T^{\text{par}}) \leq 2\tau^{1/2} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}} + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

$$\mathrm{E}(T^{\text{com}}) \leq 2\lambda\tau^{-1/2} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}} + \frac{1}{\tau} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

for every unidirectional ring,

$$\mathrm{E}(T^{\text{par}}) \leq 3\tau^{2/3} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}} + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

$$\mathrm{E}(T^{\text{com}}) \leq 12\lambda\tau^{-1/3} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}} + \frac{4}{\tau} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

for an undirected grid or torus with side lengths $\sqrt{\lambda} \times \sqrt{\lambda}$,

$$\mathrm{E}(T^{\text{par}}) \leq \tau m + \tau \sum_{i=1}^{m-1} \log\left(\frac{1}{\tau s_i}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

$$\mathrm{E}(T^{\text{com}}) \leq \lambda(\log\lambda)m + \lambda(\log\lambda) \sum_{i=1}^{m-1} \log\left(\frac{1}{\tau s_i}\right) + \frac{\log\lambda}{\tau} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

for the $(\log\lambda)$-dimensional hypercube, and

$$\mathrm{E}(T^{\text{par}}) \leq m\tau + m + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

$$\mathrm{E}(T^{\text{com}}) \leq \lambda(\lambda-1)m + \frac{\lambda(\lambda-1)m}{\tau} + \frac{\lambda-1}{\tau} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

for the complete topology K_λ.

These bounds match the ones from [7] for the case of probabilistic migration, if we compare τ against a migration probability of $p = 1/\tau$; the constant factors here are even better. The constants for probabilistic migration are higher to account for the variation in the spread of information. Periodic migration is more reliable in this respect since information is guaranteed to be spread every τ generations.

4. ADAPTIVE SCHEME A

In this section we analyse Scheme A on different topologies, including those from Theorem 1. Note that whenever an island improves its current best solution, a copy of this solution is being spread to all neighbouring islands immediately. Thus, good fitness levels spread in the same way as a migrating in every generation would do, i. e., using a global parameter $\tau = 1$. This means that the upper bounds from Theorem 1 apply for $\tau = 1$.

THEOREM 2. *For Scheme A on topologies from Theorem 1, the expected parallel optimisation time is bounded from above as in Theorem 1 with $\tau = 1$.*

Note that the bounds on the expected parallel time from Theorem 1 are minimised for $\tau = 1$. This implies that we get upper bounds on the expected parallel time equal to the best upper bounds for any fixed choice of the migration interval. In case these bounds are asymptotically tight, this means that our adaptive Scheme A never increases the expected parallel running time asymptotically.

The intended benefit of Scheme A comes from a reduced communication effort as all islands decrease communication while no improvement is encountered through either variation or immigration. The expected communication effort is bounded from above in the following theorem. The main observation is that for each fitness level, the number of migrations from an island is logarithmic in the time it remains on that fitness level. For an upper bound we consider the expected worst-case time spent on a fitness level A_i, where the worst case is taken over all populations with its best individual in A_i.

THEOREM 3. *Consider Scheme A on an arbitrary communication topology $G = (V, E)$ with diameter* $\operatorname{diam}(G)$. *Let $\mathrm{E}(T_i^{\mathrm{par}})$ be (an upper bound on) the worst-case expected number of generations during which the current best search point in the island model is on fitness level i. Then the expected communication effort is at most*

$$\mathrm{E}(T^{\mathrm{com}}) \leq |E| \sum_{i=1}^{m-1} \log\left(\mathrm{E}(T_i^{\mathrm{par}}) + \operatorname{diam}(G)\right).$$

Proof. Initially, and after improving its current best fitness, an island will double its migration interval until its current best fitness improves again. If the current best fitness does not improve for t generations, the island will perform at most $\log t$ migrations.

Consider an island v after reaching fitness level i for the first time, either through variation or immigration. If no other island has found a better fitness level, the random parallel time for some island finding such an improvement is given (or bounded) by T_i^{par}. Then this solution (or another solution of fitness better than i) will be propagated through the topology, advancing to neighbouring islands in every generation. Hence, some solution on a better fitness level than i will eventually reach v within the following $\operatorname{diam}(G)$ generations. The latter also holds if some island has already found a better fitness level than i at the time v reaches fitness level i. In any case, the total time v will spend on fitness level i is at most $T_i^{\mathrm{par}} + \operatorname{diam}(G)$.

An island v in the topology with outdegree $\deg^+(v)$ will send $\deg^+(v)$ individuals during each migration. Hence, the total number of migrated solutions in t generations on fitness level i is at most

$$\sum_{v \in V} \deg^+(v) \cdot \log t = |E| \log t.$$

The expected communication effort, therefore, is at most

$$\begin{aligned}
\mathrm{E}(T^{\mathrm{com}}) &= \sum_t \Pr\left(T_i^{\mathrm{par}} + \operatorname{diam}(G) = t\right) \cdot |E| \log t \\
&= |E| \cdot \mathrm{E}\left(\log(T_i^{\mathrm{par}} + \operatorname{diam}(G))\right) \\
&\leq |E| \cdot \log\left(\mathrm{E}\left(T_i^{\mathrm{par}} + \operatorname{diam}(G)\right)\right) \\
&= |E| \log\left(\mathrm{E}\left(T_i^{\mathrm{par}}\right) + \operatorname{diam}(G)\right),
\end{aligned}$$

where the inequality follows from Jensen's inequality and the fact that log is a concave function. □

The communication effort is proportional to the logarithm of the expected time spend on each fitness level. For functions that can be optimised by the island model in expected polynomial time, and for polynomial λ (note $\operatorname{diam}(G) \leq \lambda$), this logarithm is always at most $O(\log n)$. Then Theorem 3 gives the following upper bound.

COROLLARY 1. *Consider a fitness function f with m fitness values, such that f is being optimised by the island model in an expected polynomial number of generations, for every initial population. If also $\lambda = n^{O(1)}$, we have*

$$\mathrm{E}(T^{\mathrm{com}}) \leq O(|E| m \cdot \log n).$$

The expected parallel time on a fitness level can be smaller than a polynomial: if sufficiently many islands are being used, and the topology spreads information quickly enough, it can be logarithmic or even constant. For specific topologies we get the following results by combining Theorem 3 with bounds on the parallel time from Theorem 1.

THEOREM 4. *Given success probabilities $s_1, \ldots, s_{m-1}$ as in Theorem 1, the expected communication effort for Scheme A is bounded from above for certain topologies with λ islands as follows:*

1. $\lambda \sum_{i=1}^{m-1} \log\left(\frac{1}{s_i^{1/2}} + \frac{1}{\lambda s_i} + \lambda\right)$ *for a unidirectional ring,*
2. $4\lambda \sum_{i=1}^{m-1} \log\left(\frac{1}{s_i^{1/3}} + \frac{1}{\lambda s_i} + 2\sqrt{\lambda}\right)$ *for every undirected grid or torus with side lengths $\sqrt{\lambda} \times \sqrt{\lambda}$,*
3. $\lambda(\log\lambda) \sum_{i=1}^{m-1} \log\left(\log\left(\frac{1}{s_i}\right) + \frac{1}{\lambda s_i} + \log\lambda\right)$ *for the $(\log\lambda)$-dimensional hypercube,*
4. $\lambda(\lambda-1) \sum_{i=1}^{m-1} \log\left(2 + \frac{1}{\lambda s_i}\right)$ *for the complete graph.*

We demonstrate the application of this theorem in Section 6.

5. ADAPTIVE SCHEME B

Scheme A resets the migration interval of an island to 1 every time an improvement is found. We propose Scheme B which halves this value instead. This may be advantageous if there is a "Goldilocks region" of good values for the migration interval. In contrast to Scheme A, Scheme B should be able to maintain a value in that region over several fitness levels.

When improvements are being found with probability p, good parameter values are close to $\tau \approx 1/(\lambda p)$, as then we find one improvement in expectation. If the current migration interval is much smaller, chances of finding an improvement are small, and τ is likely to increase. Likewise, if τ is large, the island will find an improvement with high probability and halve τ. Thus, the migration interval will reach an equilibrium state close to $1/(\lambda p)$.

Scheme B might have a smaller communication effort as it does not reset the migration interval to 1, where communication is most frequent. This, however, only holds if Scheme B does not lead to an increase in the parallel time. In fact, for sparse topologies G such as the unidirectional ring there is a risk that improvements may take a very long time to be communicated. If, say, all islands had the same migration interval τ, and one island found a new best solution, it may take up to $\mathrm{diam}(G) \cdot \tau$ generations for this information to arrive at the last island.

Scheme B therefore makes more sense for dense topologies such as the complete graph, where $\mathrm{diam}(G) = 1$ and decreases of τ quickly propagate to all islands.

We follow the analysis of Scheme B for adapting the number of islands in [9]. In both approaches over some time span a resource is being doubled and halved, depending on whether an improvement of the best fitness has been found. In [9] this resource is the number of islands, and hence the number of function evaluations executed in one generation. Here it is the number of generations within one migration period. The time span in [9] is just one generation, leading to the parallel time in [9] as performance measure. In our case the time span is the current migration interval, leading to the communication effort as performance measure.

For $\lambda = 1$ the parallel time in our work equals the sequential time in [9], and the communication effort in this work equals the parallel time in [9]. However, a difference emerges for $\lambda > 1$ as in our scenario an island has $\lambda\tau$ trials to find an improvement, so the resources for finding improvements are by a factor of λ higher, compared to [9].

We adapt the analysis from [9] to accommodate this difference. The following lemma is a straightforward adaptation of parts of [9, Lemma 1], hence a proof is omitted. It estimates the expected communication effort for finding a single improvement, based on a given initial migration interval τ_0. We abbreviate $\max\{x, 0\}$ by $(x)^+$.

LEMMA 1. *Assume an island model with a complete topology starts with a migration interval of τ_0 and that in each generation each island finds an improvement over the current best individual with probability at least p. Let T^{com} be the random number of individuals migrated between islands. Let τ^* be the migration interval at this time (before it is halved). Then for every $\alpha \in \mathbb{N}_0$*

1. $\Pr\left(\log(\tau^*) - \log(\tau_0) > (\log(1/(\lambda p)) - \log(\tau_0))^+ + \alpha\right) \leq \exp(-2^\alpha)$,
2. $\mathrm{E}\left(T^{\mathrm{com}}\right) \leq \lambda(\lambda - 1) \cdot \left((\log(1/(\lambda p)) - \log(\tau_0))^+ + 2\right)$.

The expected number of migrations is expressed as the difference between logarithms of the ideal value $1/(\lambda p)$ and the initial value τ_0. If the initial migration interval is larger, $\tau_0 \geq 1/(\lambda p)$, the expected number of migrations is just 2.

This fact is reflected in the following Theorem. The upper bound on the communication effort only contains values $\log(1/(\lambda s_j))$ when the migration interval needs to be increased, i. e. $s_j < s_{j-1}$. For the special case where fitness levels get progressively harder, $s_1 \geq s_2 \geq \cdots \geq s_{m-1}$, the bound simplifies significantly.

THEOREM 5. *Given success probabilities $s_1, \ldots, s_{m-1}$ as in Theorem 1, for Scheme B we have*

$$\mathrm{E}\left(T_B^{\mathrm{par}}\right) \leq 3m + \frac{3}{2\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i} \quad \textit{and}$$

$$\mathrm{E}\left(T_B^{\mathrm{com}}\right) \leq \lambda(\lambda - 1) \cdot \left(3m + \log\left(\frac{1}{\lambda s_1}\right) + \sum_{j=2}^{m-1} \left(\log\left(\frac{1}{\lambda s_j}\right) - \log\left(\frac{1}{\lambda s_{j-1}}\right)\right)^+\right).$$

For $s_1 \geq s_2 \geq \cdots \geq s_{m-1}$ the latter simplifies to

$$\mathrm{E}\left(T_B^{\mathrm{com}}\right) \leq \lambda(\lambda - 1) \cdot \left(3m + \log\left(\frac{1}{\lambda s_{m-1}}\right)\right).$$

The upper bound for the expected parallel time is only by a factor of 3/2 larger than the upper bound for Scheme A. Hence, both upper bounds are asymptotically equal. In other words, the reduced communication in Scheme B does not worsen the running time, for problems where the upper bounds for Schemes A and B are asymptotically tight.

We give an informal argument to convey the intuition for this result. Assume that Scheme B has raised the migration interval from 1 to some large value $\tau^* = 2^i$ before an improvement is found. Then Scheme B has spent $1 + 2 + 4 + \cdots + 2^i = 2^{i+1} - 1$ generations leading up to this value. In the worst case, the new migration interval $\tau^*/2 = 2^{i-1}$ is too large: improvements are found easily, and then the algorithm has to idle for nearly $\tau^*/2$ generations before being able to communicate the current best fitness. The total time spent is around $3/2 \cdot (2^{i+1} - 1)$, whereas Scheme A in the same situation would spend $2^{i+1} - 1$ generations. So, in this setting Scheme B needs at most 3/2 as many generations as Scheme A.

A formal argument was given in [9] to derive an upper bound for the parallel time of Scheme B, which is 3/2 that for Scheme A. The bound on the expected communication effort follows from similar arguments and applying our refined Lemma 1. We refrain from giving a formal proof here as it can be obtained with straightforward modifications from the proof of Theorem 3 in [9].

6. PERFORMANCE ON COMMON EXAMPLE FUNCTIONS

The analytical frameworks for analysing fixed migration intervals and our two adaptive schemes can be applied by simply using lower bounds on success probabilities for improving the current fitness. We demonstrate this approach, and how to use the results from the previous sections to analyse the parallel time and the communication effort on common test problems.

We provide an analysis for the maximisation of the same pseudo-Boolean test functions investigated in [7]. For

a search point $x \in \{0,1\}^n$, $x = x_1 \dots x_n$, we define $\mathrm{OneMax}(x) := \sum_{i=1}^{n} x_i$ as the number of ones in x, and $\mathrm{LO}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_i$ as the number of leading ones in x. We also consider the class of unimodal functions taking d fitness values $f_1, \dots, f_d$. A function is called unimodal if every non-optimal search point has an Hamming neighbour with strictly larger fitness. Finally for $1 \le k \le n$ we consider

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

Where a "jump", a mutation flipping k specific bits at the same time, has to be made. The parameter k tunes the difficulty of this function.

6.1 Fitness partition and success probabilites

In order to apply Theorems 1, 4, 5, it is just necessary to define probability s_i to move from the fitness level A_i to a better one. Recall that s_i is a lower bound on the probability of one island finding a search point of strictly higher fitness, given that the population has current best fitness i.

For the simple (1+1) EA, these values are easy to derive:

- For OneMax, a search point with i ones has $n-i$ zeros. The probability of a mutation flipping only one of these zeros is $s_i \ge (n-i)/n \cdot (1-1/n)^{n-1} \ge (n-i)/(en)$.
- For LO, it is sufficient to flip the first 0-bit, which has probability $s_i \ge 1/n \cdot (1-1/n)^{n-1} \ge 1/(en)$.
- For unimodal functions the success probability on each fitness level is at least $s_i \ge 1/n \cdot (1-1/n)^{n-1} \ge 1/(en)$ as for any non-optimal point there is always a better Hamming neighbour.
- For Jump_k with $k \ge 2$, it is possible to find an improvement to an individual having up to $n-k$ 1-bits just increasing the number of ones, thus the s_i with $i < n-k$ are equal to the ones for OneMax. A similar argument applies to levels $n-k < i < n$. From an individual with $n-k$ 1-bits an improvement is found by generating as offspring a specific bitstring having Hamming distance k from the parent, which has probability $s_{n-k} \ge \left(\frac{1}{n}\right)^k \cdot \left(1-\frac{1}{n}\right)^{n-k} \ge \left(\frac{1}{n}\right)^k \cdot \left(1-\frac{1}{n}\right)^{n-1} \ge \frac{1}{en^k}$.

6.2 Fixed scheme

Given Theorem 1 it is possible to bound the parallel time and the communication effort for fixed migration intervals similarly to [7]. For example, for OneMax and the complete topology we get an expected parallel time of $\mathrm{E}(T^{\mathrm{par}}) = O\left(n\tau + \frac{n \log n}{\lambda}\right)$ and an expected communication effort of $\mathrm{E}(T^{\mathrm{com}}) = O\left(\lambda^2 n + \frac{\lambda n \log n}{\tau}\right)$.

For fixed λ, the value of τ yields a trade-off between upper bounds for the parallel time and that of the communication effort. In our example, for say $\lambda = O(1)$, we get $\mathrm{E}(T^{\mathrm{par}}) = O(n\tau + n \log n)$ and $\mathrm{E}(T^{\mathrm{com}}) = O\left(n + \frac{n \log n}{\tau}\right)$. We can notice how a large τ always minimises the bound for the communication effort, while a small one (i. e. $\tau = 1$) minimises the bound for the parallel time.

Given a fixed number of islands λ, we define the best τ as the largest τ that does not asymptotically increase the bound for the parallel time (compared to $\tau = 1$). This assures good scalability while minimising the communication effort. For the example proposed the best τ is $\tau = \Theta(\frac{\log n}{\lambda})$. This leads to $\mathrm{E}(T^{\mathrm{par}}) = O(\frac{n \log n}{\lambda})$ and $\mathrm{E}(T^{\mathrm{com}}) = O(\frac{|E|}{\tau} \cdot \mathrm{E}(T^{\mathrm{par}})) = O(\frac{\lambda^2}{\tau} \cdot \mathrm{E}(T^{\mathrm{par}})) = O(\lambda^2 n)$. The results for other topologies and problems are summarised in Table 1. Notice that fixing τ to its best value is possible, provided that the number of island is small enough. Particularly for the example proposed the number of islands must be $\lambda = O(\log n)$ for the best τ to be defined ($\tau \ge 1$).

6.3 Adaptive scheme

In order to calculate the parallel time for the adaptive Scheme A we can refer to the results for the fixed scheme when $\tau = 1$, as shown in Theorem 2. For example, our Scheme A running on a complete topology solves OneMax in time $O(\frac{n \log n}{\lambda} + n)$. Scheme B has asymptotically the same parallel time of Scheme A, as shown in Theorem 5.

We only consider values of λ that lead to a linear speedup as defined in [11]: the expected parallel time for λ islands is by a factor of λ smaller than the expected optimisation time of a single island, in an asymptotic sense. In this setting an island model thus makes the same expected number of function evaluations, compared to a single island. In the example proposed linear speedup is achieved for a number of islands up to $\lambda = O(\log n)$, in fact for a larger number of islands the upper bound on the parallel time would be $O(n)$ regardless of λ. The bounds on λ limit the best upper bound on the parallel time achievable with our analytical framework. Table 1 shows the bound on λ for different problems and topologies and the best achievable parallel time bound.

In order to calculate the communication effort we use Theorem 4 and 5 for Schemes A and B, respectively. We first get a general bound for every λ included in the linear speedup bound, and then we calculate it for the maximum value of λ, thus providing the communication effort for the value of λ leading to the best achievable parallel time. Table 1 shows all results derived in this manner.

6.4 Evaluation of Results

Recall that Table 1 only shows results for linear speedups, hence all parallel times are equal, but the range of λ values varies between topologies. Comparing the communication efforts for the best fixed value of τ against our adaptive schemes, for OneMax on all topologies the communication effort is by a small $O(\log \log n)$ term larger for the adaptive schemes, compared to the best fixed τ. The latter varies according to the topology: it is $\tau = ((\log n)/\lambda)^2$ for the ring, $\tau = ((\log n)/\lambda)^{3/2}$ for the grid, and $\tau = (\log n)/\lambda$ for the hypercube and the complete graph. So, the additional $O(\log \log n)$ factor is a small price to pay for the convenience of adapting τ automatically.

For LO and the general bounds for unimodal functions, Scheme A on the ring has a communication effort of $O(n^{3/2} \log n)$ compared to $O(n^2)$, and for the grid it is $O(n^{5/3} \log n)$ versus $O(n^2)$. These significant improvements show that decreasing the migration interval is an effective strategy in lowering the communication costs, without harming the parallel running time. For the hypercube the communication effort is lower by a factor of $O(\log(n)/\log \log n)$, whereas for the complete graph no differences are visible in the upper bounds.

For Jump there are no differences for the complete graph, while on the hypercube Scheme A is by a $O(\log \log(n^{k-1}))$ factor worse than the best fixed value. For rings and grids the adaptive scheme is better; the performance gap even grows with k and hence the difficulty of the function.

Comparing Schemes A and B, both achieve similar results. For LO we see an advantage of Scheme B over Scheme A: the general bound for the communication effort of Scheme A is $O(\lambda^2 n \log(n/\lambda))$, whereas that for Scheme B is $O(\lambda^2 n)$. This makes sense as the probability for finding an improvement in one generation is of order $\Theta(\lambda/n)$ for the considered λ, and the ideal value for the migration interval is in the region of $\Theta(n/\lambda)$. Scheme A needs to increase the migration interval around $\log(n/\lambda)$ times to get into this range, which is precisely the performance difference visible in our upper bounds. The difference disappears for $\lambda = \Theta(n)$.

The same argument also applies to the more general function class of unimodal functions.

7. CONCLUSIONS

We have presented two adaptive schemes for adapting the migration interval in island models. Both schemes have been accompanied by analytical frameworks that yield upper bounds on the expected parallel time and the expected communication effort, based on probabilities of fitness improvements in single islands running elitist EAs. The results show that our schemes are able to decrease the communication effort without significantly compromising exploitation. For arbitrary topologies, we got upper bounds on the expected parallel time no larger than those for maximum exploitation, that is, migration in every generation.

Example applications to common example functions revealed that, in the realm of linear speedups and comparing against the best fixed choice of the migration interval, the expected communication effort was larger by a tiny $O(\log \log n)$ term for OneMax and similarly for the hypercube on Jump, but significantly smaller on LO, for a general analysis of unimodal functions, and for rings and grids on Jump. Both schemes are able to match the performance of the best fixed migration interval up to doubly logarithmic factors, or they can perform significantly better by polynomial factors.

Acknowledgments: The authors would like to thank Joseph Kempka and the participants of Dagstuhl seminar 13271 "Theory of Evolutionary Algorithms" for fruitful discussions. Andrea Mambrini was partially funded by EPSRC (Grant No. EP/I010297/1). The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 618091 (SAGE).

8. REFERENCES

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

[2] E. Alba and G. Luque. Theoretical models of selection pressure for dEAs: topology influence. In *Proc. of IEEE CEC 2005*, pages 214–221, 2005.

[3] E. Alba, G. Luque, and S. Nesmachnow. Parallel metaheuristics: Recent advances and new trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.

[4] L. Araujo and J. J. M. Guervós. Diversity through multiculturality: Assessing migrant choice policies in an island model. pages 456–469, 2011.

[5] Y. Bravo, G. Luque, and E. Alba. Influence of the migration period in parallel distributed GAs for dynamic optimization. In *Proc. of LION*, pages 343–348, 2012.

[6] T.-P. Hong, W.-Y. Lin, S.-M. Liu, and J.-H. Lin. Experimental analysis of dynamic migration intervals on 0/1 knapsack problems. In *Proc. of IEEE CEC 2007*, pages 1163–1167, 2007.

[7] J. Lässig and D. Sudholt. General upper bounds on the running time of parallel evolutionary algorithms. *Evolutionary Computation*. In press. Available from `http://www.mitpressjournals.org/doi/pdf/10.1162/EVCO_a_00114`.

[8] J. Lässig and D. Sudholt. General scheme for analyzing running times of parallel evolutionary algorithms. In *Proc. of PPSN 2010*, pages 234–243. Springer, 2010.

[9] J. Lässig and D. Sudholt. Adaptive population models for offspring populations and parallel evolutionary algorithms. In *Proc. of FOGA 2011*, pages 181–192. ACM Press, 2011.

[10] J. Lässig and D. Sudholt. Analysis of speedups in parallel evolutionary algorithms for combinatorial optimization. In *Proc. of ISAAC 2011*, pages 405–414. Springer, 2011.

[11] J. Lässig and D. Sudholt. Design and analysis of migration in parallel evolutionary algorithms. *Soft Computing*, 17(7):1121–1144, 2013.

[12] W.-Y. Lin, T.-P. Hong, S.-M. Liu, and J.-H. Lin. Revisiting the design of adaptive migration schemes for multipopulation genetic algorithms. In *Proc. of TAAI 2012*, pages 338–343, 2012.

[13] G. Luque and E. Alba. *Parallel Genetic Algorithms–Theory and Real World Applications*. Springer, 2011.

[14] A. Mambrini, D. Sudholt, and X. Yao. Homogeneous and heterogeneous island models for the set cover problem. In *Proc. of PPSN 2012*, pages 11–20. Springer, 2012.

[15] F. Neumann, P. S. Oliveto, G. Rudolph, and D. Sudholt. On the effectiveness of crossover for migration in parallel evolutionary algorithms. In *Proc. of GECCO 2011*, pages 1587–1594. ACM Press, 2011.

[16] K. Osorio, E. Alba, and G. Luque. Using theory to self-tune migration periods in distributed genetic algorithms. In *Proc. of CEC 2013*, pages 2595–2601, 2013.

[17] K. Osorio, G. Luque, and E. Alba. Distributed evolutionary algorithms with adaptive migration period. In *Proc. of ISDA*, pages 259–264, 2011.

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

[19] Z. Skolicki and K. De Jong. The influence of migration sizes and intervals on island models. In *Proc. of GECCO 2005*, pages 1295–1302. ACM, 2005.

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

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

Table 1: Overview of expected parallel optimisation times and expected communication efforts for island models with λ islands running a simple (1+1) EA on common example functions, derived using Theorems 1, 2, 3, and 5. As explained in Sections 6.2 and 6.3, the table shows restrictions on λ to yield linear speedups and fixed values for τ leading to the best upper bounds for the communication effort, while not increasing the parallel running time. For both parallel time and communication effort we show bounds for general λ in the realm of linear speedups, and the best parallel time $\mathrm{E}(T^{\mathrm{par}}_{\mathrm{best}})$ achieved by using the largest such λ, along with the corresponding communication effort $\mathrm{E}(T^{\mathrm{par}}_{\mathrm{best}})$ for the same λ. For Jump_k we assume $3 \leq k = O(n/(\log n))$.

Problem	Topology and Scheme	λ	$\mathrm{E}(T^{\mathrm{par}}) \rightsquigarrow \mathrm{E}(T^{\mathrm{par}}_{\mathrm{best}})$	$\mathrm{E}(T^{\mathrm{com}}) \rightsquigarrow \mathrm{E}(T^{\mathrm{com}}_{\mathrm{best}})$
OneMax	Complete, Scheme A	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n\log^2(n))$
	Complete, Scheme B	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n\log^2(n))$
	Complete, $\tau = \Theta\left(\frac{\log n}{\lambda}\right)$	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n\log^2(n))$
	Ring, Scheme A	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(n\lambda\log\lambda) \rightsquigarrow O(n\log(n)\log\log(n))$
	Ring, $\tau = \Theta\left(\left(\frac{\log n}{\lambda}\right)^2\right)$	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O\left(\frac{\lambda^2 n}{\log n}\right) \rightsquigarrow O(n\log(n))$
	Grid, Scheme A	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(n\lambda\log\lambda) \rightsquigarrow O(n\log(n)\log\log(n))$
	Grid, $\tau = \Theta\left(\left(\frac{\log n}{\lambda}\right)^{3/2}\right)$	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O\left(\frac{\lambda^{3/2} n}{\log^{1/2}(n)}\right) \rightsquigarrow O(n\log(n))$
	Hypercube, Scheme A	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(n\lambda\log(\lambda)\log\log(\lambda)) \rightsquigarrow O(n\log(n)\log\log(n)\log\log\log(n))$
	Hypercube, $\tau = \Theta\left(\frac{\log n}{\lambda}\right)$	$O(\log n)$	$O\left(\frac{n}{\lambda}\log n\right) \rightsquigarrow O(n)$	$O(n\lambda\log\lambda) \rightsquigarrow O(n\log(n)\log\log(n))$
LO	Complete, Scheme A	$O(n)$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n)$	$O\left(\lambda^2 n\log\left(\frac{n}{\lambda}\right)\right) \rightsquigarrow O(n^3)$
	Complete, Scheme B	$O(n)$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n)$	$O\left(\lambda^2 n\right) \rightsquigarrow O(n^3)$
	Complete, $\tau = \Theta\left(\frac{n}{\lambda}\right)$	$O(n)$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n)$	$O\left(\lambda^2 n\right) \rightsquigarrow O(n^3)$
	Ring, Scheme A	$O(n^{1/2})$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n^{3/2})$	$O(\lambda n\log\left(\frac{n}{\lambda}\right)) \rightsquigarrow O(n^{3/2}\log n)$
	Ring, $\tau = \Theta\left(\frac{n}{\lambda^2}\right)$	$O(n^{1/2})$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n^{3/2})$	$O(\lambda^2 n) \rightsquigarrow O(n^2)$
	Grid, Scheme A	$O(n^{2/3})$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n^{4/3})$	$O(\lambda n\log\left(\frac{n}{\lambda}\right)) \rightsquigarrow O(n^{5/3}\log n)$
	Grid, $\tau = \Theta\left(\frac{n}{\lambda^{3/2}}\right)$	$O(n^{2/3})$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n^{4/3})$	$O(n\lambda^{3/2}) \rightsquigarrow O(n^2)$
	Hypercube, Scheme A	$O\left(\frac{n}{\log n}\right)$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n\log n)$	$O\left(\lambda\log(\lambda)n\log\left(\frac{n}{\lambda}\right)\right) \rightsquigarrow O(n^2\log\log n)$
	Hypercube, $\tau = \Theta\left(\frac{n}{\lambda\log n}\right)$	$O\left(\frac{n}{\log n}\right)$	$O\left(\frac{n^2}{\lambda}\right) \rightsquigarrow O(n\log n)$	$O(\lambda\log(\lambda)n\log(n)) \rightsquigarrow O(n^2\log n)$
unimodal f	Complete, Scheme A	$O(n)$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(d)$	$O(\lambda^2 d\log\left(\frac{n}{\lambda}\right)) \rightsquigarrow O(n^2 d)$
with d f-values	Complete, Scheme B	$O(n)$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(d)$	$O(\lambda^2 d) \rightsquigarrow O(n^2 d)$
	Complete, $\tau = \Theta\left(\frac{n}{\lambda}\right)$	$O(n)$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(d)$	$O(\lambda^2 d) \rightsquigarrow O(n^2 d)$
	Ring, Scheme A	$O(n^{1/2})$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(dn^{1/2})$	$O\left(\lambda d\log\left(\frac{n}{\lambda}\right)\right) \rightsquigarrow O(dn^{1/2}\log n)$
	Ring, $\tau = \Theta\left(\frac{n}{\lambda^2}\right)$	$O(n^{1/2})$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(dn^{1/2})$	$O(\lambda^2 d) \rightsquigarrow O(dn)$
	Grid, Scheme A	$O(n^{2/3}))$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(dn^{1/3})$	$O\left(\lambda d\log\left(\frac{n}{\lambda}\right)\right) \rightsquigarrow O(dn^{2/3}\log n)$
	Grid, $\tau = \Theta\left(\frac{n}{\lambda^{3/2}}\right)$	$O(n^{2/3}))$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(dn^{1/3})$	$O(\lambda^{3/2} d) \rightsquigarrow O(dn)$
	Hypercube, Scheme A	$O\left(\frac{n}{\log n}\right)$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(d\log n)$	$O\left(\lambda\log(\lambda)d\log\left(\frac{n}{\lambda}\right)\right) \rightsquigarrow O(dn\log\log n)$
	Hypercube, $\tau = \Theta\left(\frac{n}{\lambda\log n}\right)$	$O\left(\frac{n}{\log n}\right)$	$O\left(\frac{dn}{\lambda}\right) \rightsquigarrow O(d\log n)$	$O(\lambda\log(\lambda)d\log(n)) \rightsquigarrow O(dn\log n)$
Jump_k	Complete, Scheme A	$O(n^{k-1})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n^{2k-1})$
	Complete, Scheme B	$O(n^{k-1})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n^{2k-1})$
	Complete, $\tau = \Theta\left(\frac{n^{k-1}}{\lambda}\right)$	$O(n^{k-1})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n)$	$O(\lambda^2 n) \rightsquigarrow O(n^{2k-1})$
	Ring, Scheme A	$O(n^{k/2})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n^{k/2})$	$O(n\lambda\log\lambda) \rightsquigarrow O(kn^{k/2+1}\log n)$
	Ring, $\tau = \Theta\left(\frac{n^k}{\lambda^2}\right)$	$O(n^{k/2})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n^{k/2})$	$O(\lambda^2) \rightsquigarrow O(n^k)$
	Grid, Scheme A	$O\left(n^{2k/3}\right)$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n^{k/3})$	$O(n\lambda\log\lambda) \rightsquigarrow O(kn^{2k/3+1}\log n)$
	Grid, $\tau = \Theta\left(\frac{n^k}{\lambda^{3/2}})\right)$	$O\left(n^{2k/3}\right)$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n^{k/3})$	$O(\lambda^{3/2}) \rightsquigarrow O(n^k)$
	Hypercube, Scheme A	$O(n^{k-1})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n)$	$O(n\lambda\log(\lambda)\log\log(\lambda)) \rightsquigarrow O(kn^k\log(n)\log\log(n^{k-1}))$
	Hypercube, $\tau = \Theta\left(\frac{n^{k-1}}{\lambda}\right)$	$O(n^{k-1})$	$O\left(\frac{n^k}{\lambda}\right) \rightsquigarrow O(n)$	$O(n\lambda\log\lambda) \rightsquigarrow O(kn^k\log n)$