Theory of Parallel Evolutionary Algorithms

Dirk Sudholt

University of Sheffield, UK

Based on joint work with Jörg Lässig, Andrea Mambrini, Frank Neumann, Pietro Oliveto, Günter Rudolph, and Xin Yao
See chapter in the upcoming *Handbook of Computational Intelligence*, Springer 2015
http://staffwww.dcs.shef.ac.uk/~dirk/parallel-eas.pdf

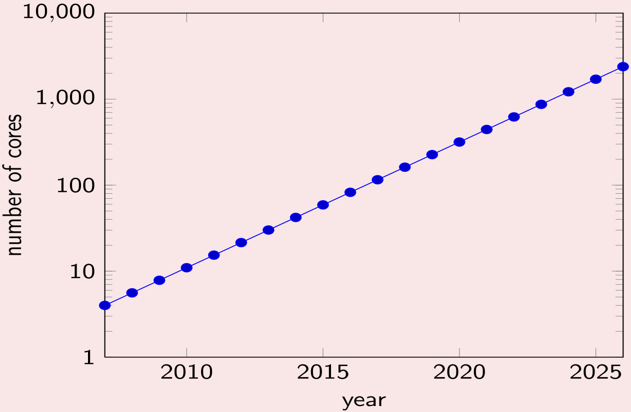
Parallel Problem Solving from Nature – PPSN 2014

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618091 (SAGE).

Overview

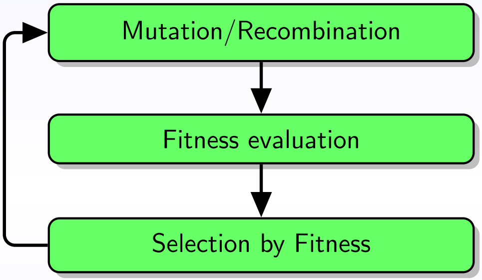
Why Parallelisation is Important

International Technology Roadmap for Semiconductors 2011



How to best make use of parallel computing power?

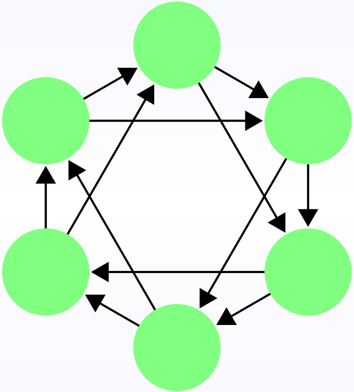
Evolutionary Algorithms



Parallelization

- low-level parallelization: parallelize execution of EA
- high-level parallelization: parallelize evolution $\rightarrow$ different EA

Island Models



λ islands,
migration every τ generations.

Advantages

- Multiple communicating populations speed up optimization
- Small populations can be executed faster than large populations
- Periodic communication only requires small bandwidth
- Better solution quality through better exploration

Challenge

Little understanding of how fundamental parameters affect performance.

Runtime Analysis of Parallel EAs

How long does a parallel EA need to optimise a given problem?

Goals

- Understanding effects of parallelisation
- How the runtime scales with the problem size n
- When and why are parallel EAs "better" than standard EAs?
- Better answers to design questions
- How to use parallelisation most effectively?

Challenge: Analyze interacting complex dynamic systems.

Skolicki's two-level view [Skolicki 2000]

- intra-island dynamics: evolution within islands
- inter-island dynamics: evolution between islands

Content

What this tutorial is about

- Runtime analysis of parallel EAs
- Insight into their working principles
- Impact of parameters and design choices on performance
- Consider parallel versions of simple EAs
- Overview of interesting results (bibliography at end)
- Teach basic methods and proof **ideas**

What this tutorial is not about

- Continuous optimisation (e. g. [Fabien and Olivier Teytaud, PPSN '10])
- Parallel implementations not changing the algorithm
- No intent to be exhaustive

(1+1) EA: a Bare-Bones EA

Study effect of parallelisation while keeping EAs simple.

(1+1) EA

Start with uniform random solution x^* and repeat:

- Create x by flipping each bit in x^* independently with prob. $1/n$.
- Replace x^* by x if $f(x) \geq f(x^*)$.

Offspring populations: $(1+\lambda)$ EA creates λ offspring in parallel.

Parallel (1+1) EA: island model running λ communicating (1+1) EAs.

Runtime in Parallel EAs

Notions of time for parallel EAs

$$\begin{aligned} T^{\mathrm{par}} &= \text{parallel runtime} \\ &= \text{number of generations till solution found} \end{aligned}$$

$$\begin{aligned} T^{\mathrm{seq}} &= \text{sequential time, total effort} \\ &= \text{number of function evaluations till solution found} \end{aligned}$$

"solution found": global optimum found/approximation/you name it

If every generation evaluates a fixed number λ of search points,

$$T^{\mathrm{seq}} = \lambda \cdot T^{\mathrm{par}}$$

and we only need to estimate one quantity.

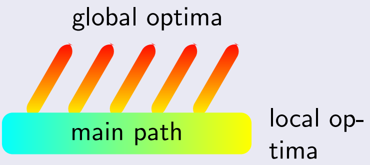
A Cautionary Tale

Claim: the more the merrier

Using more parallel resources can only decrease the parallel time.

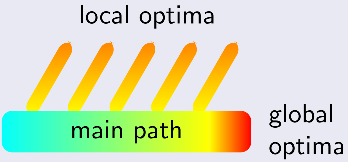
Two examples by [Jansen, De Jong, Wegener, 2005]:

SufSamp



$(1+\lambda)$ EA outperforms $(1+1)$ EA

SufSamp'



$(1+1)$ EA outperforms $(1+\lambda)$ EA

disproves the claim!

Parallelisation changes EAs' dynamic behaviour.

Effects on performance can be unforeseen and depend on the problem.

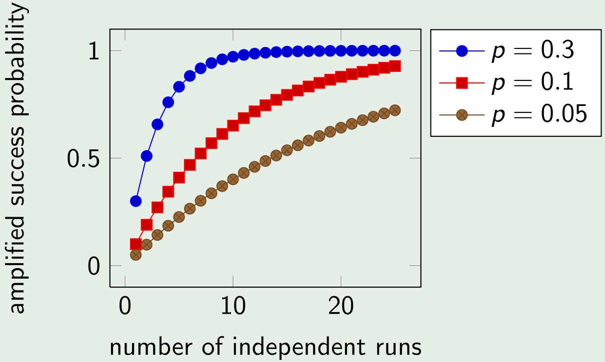
Overview

Independent Runs

Consider λ identical algorithms, each solving a problem with probability p.

Theorem

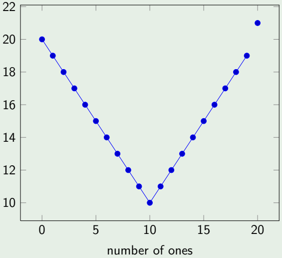
The probability that at least one run solves the problem is $1-(1-p)^{\lambda}$.



λ independent (1+1) EAs on TwoMax

TwoMax

$$\textsc{TwoMax}(x) := \max\left\{\sum_{i=1}^{n} x_i, \sum_{i=1}^{n}(1-x_i)\right\} + \prod_{i=1}^{n} x_i$$



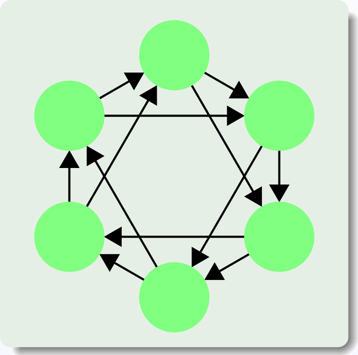
Success probability for single (1+1) EA is 1/2.

λ independent (1+1) EAs find a global optimum in $O(n \log n)$ generations with probability $1 - 2^{-\lambda}$ [Friedrich, Oliveto, Sudholt, Witt'09].

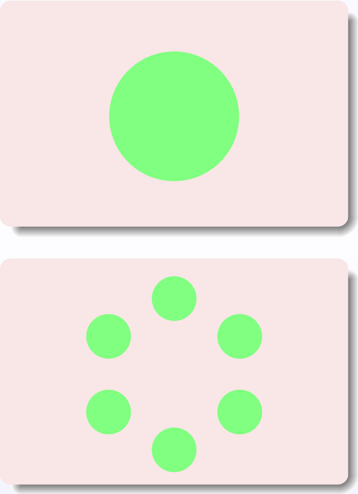
Overview

A Royal Road Function for Island Models

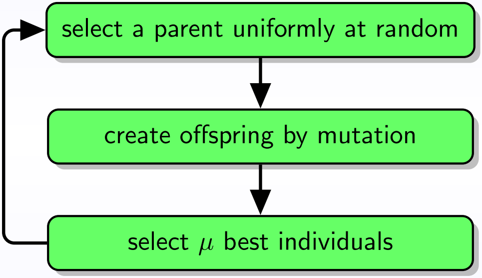
[Lässig and Sudholt, GECCO 2010 & Soft Computing, 2013]



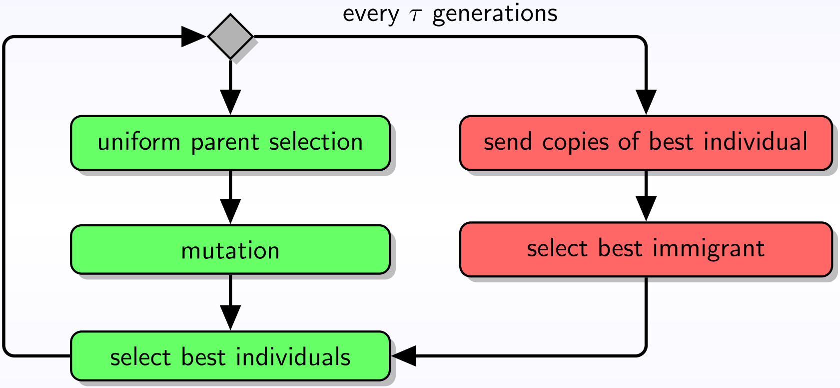
vs.



Panmictic (μ+1) EA



Island Model



Special cases

$\tau = \infty \longrightarrow$ independent subpopulations
all islands run (1+1) EAs $\longrightarrow$ parallel (1+1) EA

$\mathrm{LO}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_i$ 11110110...

$\mathrm{LZ}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} (1 - x_i)$ 00011010...

$\mathrm{LO}(x) + \mathrm{LZ}(x)$ 11110110...
00011010...

$\mathrm{LO}(x) + \min\{\mathrm{LZ}(x), z\}$ 11111101...
00000011...

Definition

Let $z, b, \ell \in \mathbb{N}$ such that $b\ell \le n$ and $z < \ell$. Let $x^{(i)} := x_{i(\ell-1)+1} \dots x_{i\ell}$.

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

LOLZ 11111111 11111111 00000011 01011110...

Why Panmictic Populations Fail

Chance of extinction of prefix in every improvement of best fitness.

111111110 . . .		111111110 . . .
11111110 . . .		111111110 . . .
00000001 . . .	→	111111110 . . .
1111110 . . .	constant prob.	111111110 . . .
0000001 . . .		111111110 . . .
11111110 . . .		111111110 . . .

Probability of extinction before completing block is $1 - \exp(-\Omega(z))$.

The probability that in all blocks 1s survive is 2^{-b}.

Otherwise, many bits have to flip simultaneously to escape.

Theorem

If $\mu \le n/(\log n)$ then with probability at least $1 - \exp(-\Omega(z)) - 2^{-b}$ the panmictic $(\mu{+}1)$ EA does not find a global optimum within $n^{z/3}$ generations.

Independent Subpopulations Fail

Amplified success prob.: $1-(1-p)^\lambda \leq p\lambda$ with $p = \exp(-\Omega(z)) + 2^{-b}$.

Probability of failure is still at least $1 - p\lambda$:

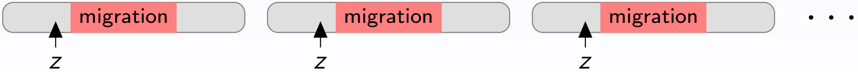
Theorem

Consider $\lambda \in \mathbb{N}$ independent subpopulations of size $\mu \leq n/(\log n)$ each. With probability at least $1 - \lambda \exp(-\Omega(z)) - \lambda 2^{-b}$ the EA does not find a global optimum within $n^{z/3}$ generations.

Why the Island Model Succeeds

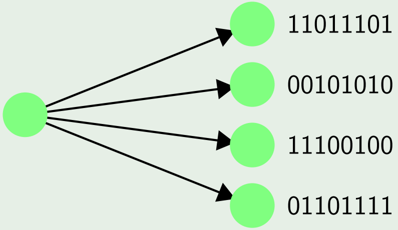
Key for success

- communication
- phases of independent evolution



At migration all 1-type islands are better than 0-type islands.
⇒ takeover can reactivate islands that got stuck.

Independent evolution creates diversity



Why the Island Model Succeeds

For topologies with a good "expansion" (e. g. hypercube) the island model maintains a sufficient number of islands on track to the optimum.

Theorem

For proper choices of τ, z, b, ℓ, $\mu = n^{\Theta(1)}$ *islands, and a proper topology the parallel* (1+1) EA *finds an optimum in* $O(b\ell n) = O(n^2)$ *generations, with overwhelming probability.*

Overview

Speedups

Classic notion of speedup from Alba's taxonomy [Alba, 2002]

- **Strong speedup:** parallel execution time vs. execution time of best known sequential algorithm
- **Weak speedup:** parallel execution time vs. its own sequential execution time
  - **Single machine/panmixia:** parallel EA vs. panmictic version of it
  - **Orthodox:** parallel EA on λ machines vs. parallel EA on one machine

Notion of "speedup" in runtime analysis

- Execution times depend on hardware – infeasible for theory
- Using speedup with regard to the number of generations: if T_λ^{par} is the parallel runtime for λ islands,

$$\text{speedup } s_\lambda = \frac{\mathrm{E}(T_1)}{\mathrm{E}(T_\lambda)}.$$

- Abstraction of weak orthodox speedup, ignoring overhead.

Linear Speedups

Speedups

sublinear speedups: $s_\lambda < \lambda$, total effort of parallel EA increases.

linear speedup: $s_\lambda = \lambda$, total effort remains constant.

superlinear speedup: $s_\lambda > \lambda$, total effort of parallel EA decreases.

Linear speedup means perfect use of parallel resources: the parallel time decreases with λ, at no increase of the total effort.

"Asymptotic" definition of linear speedups [Lässig and Sudholt, 2010]:

$$s_\lambda = \Omega(\lambda)$$

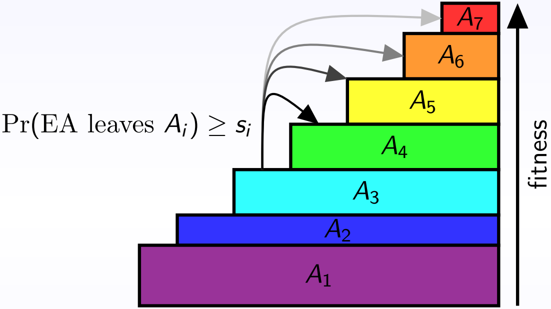
the total effort does not increase by more than a constant factor.

Coming up: a simple method for estimating parallel times and speedups in parallel EAs.

Assumption: all islands run elitist EAs.

Fitness-level Method for Elitist EAs

EA is "on level i" if best point is in A_i.



Expected optimization time of EA at most $\sum_{i=1}^{m-1} \frac{1}{s_i}$.

Tuesday, Poster Session 4, Proc. p. 912 [Cörüs, Dang, Eremeev, Lehre]

Most advanced fitness-level method: "Level-Based Analysis of Genetic Algorithms and Other Search Processes"

Bounds with Fitness Levels

$\textsc{OneMax}(x) = \sum_{i=1}^{n} x_i$: sufficient to flip a single 0-bit.

$$s_i \geq (n-i) \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{n-i}{en}$$

Theorem

(1+1) EA *on* $\textsc{OneMax}$: $en \sum_{i=0}^{n-1} \frac{1}{n-i} = en \cdot H_n = O(n \log n)$

Jump_k: like $\textsc{OneMax}$, but "jump" of $k \geq 2$ specific bits needed:

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

Theorem

(1+1) EA *on* Jump_k: $en \sum_{i=0}^{n-k} \frac{1}{n-i} + en^k = O(n \log n + n^k) = O(n^k)$

Bounds with Fitness Levels (2)

LO 11110010

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

Theorem

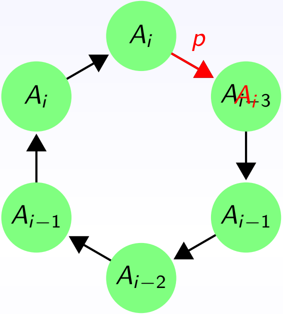
$$\text{(1+1) EA } \textit{on } \mathrm{LO}: \quad \sum_{i=0}^{n-1} en = en^2$$

Unimodal functions with d function values:

Theorem

$$\text{(1+1) EA } \textit{on } d\textit{-unimodal function}: \quad \sum_{i=0}^{d-1} en = end$$

Fitness-level Method for Parallel EAs



Transmission probability p

Each edge independently transmits a better fitness level with probability at least p.

Transmission probability p can model...

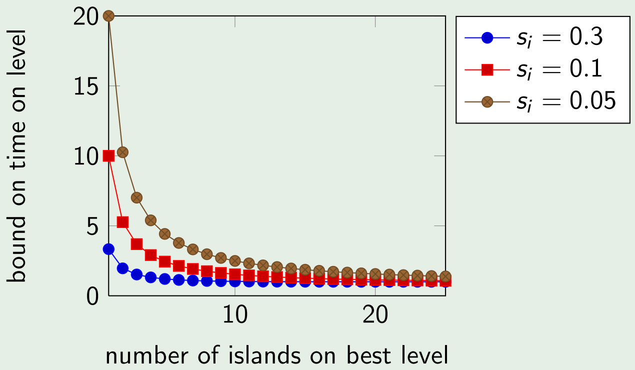
- probabilistic migration schemes
- probabilistic selection of emigrants
- probability of accepting immigrants
- probability of a crossover between islands being non-disruptive
- probability of not having a fault in the network

Spread of Best Solutions

More islands on best level yield better upper time bounds: $\frac{1}{1-(1-s_i)^{\text{number}}}$

Upper bound on expected time on level (following [Witt, 2006])

1. Wait for migration to raise a "critical mass" of islands to best level.
2. Wait for critical mass to find a better fitness level.



Time on Fitness Level i

Lemma

Let $\xi(k)$ be the number of generations for increasing the number of islands on level i from 1 to k.
For every integer $k \le \lambda$ the expected time on level i is at most

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

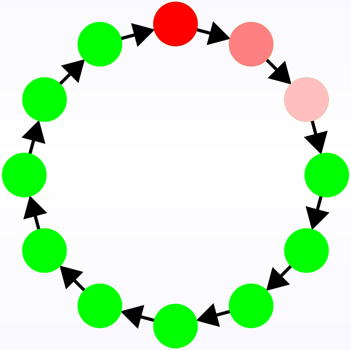
Proof:

- After $\xi(k)$ generations there are k islands on level i.
- From here on, the probability of leaving A_i is at least $1 - (1 - s_i)^k$.
- The expected time for this improvement is at most

$$\frac{1}{1 - (1 - s_i)^k} \le 1 + \frac{1}{k} \cdot \frac{1}{s_i}.$$

Upper Bounds for Rings

$\xi(k)$ depends on transmission probability p and topology.



On a unidirectional ring we have $\xi(k) = (k-1)/p$.

Critical Mass on Rings

$$\mathrm{E}(\text{generations on level } i) \leq \frac{k-1}{p} + 1 + \frac{1}{k} \cdot \frac{1}{s_i}$$
$$\leq \frac{k}{p} + \frac{1}{k} \cdot \frac{1}{s_i}$$

Best choice for critical mass: $k := p^{1/2}/s^{1/2}$ (if $\leq \lambda$), yields upper bound

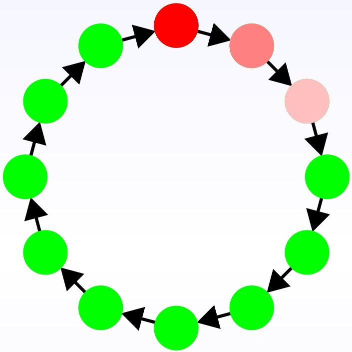
$$\frac{1}{p^{1/2}s_i^{1/2}} + \frac{1}{p^{1/2}s_i^{1/2}} = \frac{2}{p^{1/2}s_i^{1/2}}.$$

If $p^{1/2}/s^{1/2} > \lambda$, the best critical mass is $k := \lambda$, yielding upper bound

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

Maximum of these bounds is upper bound whether or not $p^{1/2}/s^{1/2} > \lambda$.

Upper Bounds for Rings

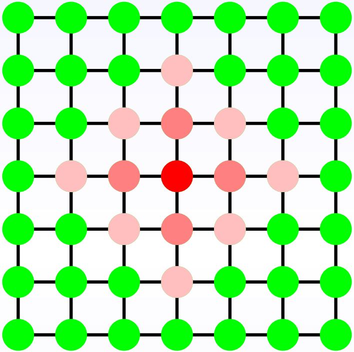


Theorem

On a unidirectional or bidirectional ring with λ islands

$$\mathrm{E}(T^{\mathrm{par}}) \le O\left(\frac{1}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$$

Upper Bounds for Torus Graphs

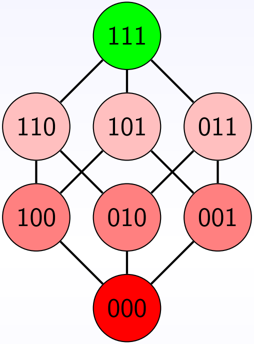


Theorem

On a two-dimensional $\sqrt{\lambda} \times \sqrt{\lambda}$ grid or toroid

$$\mathrm{E}(T^{\mathrm{par}}) \leq O\left(\frac{1}{p^{2/3}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i} .$$

Upper Bounds for Hypercubes

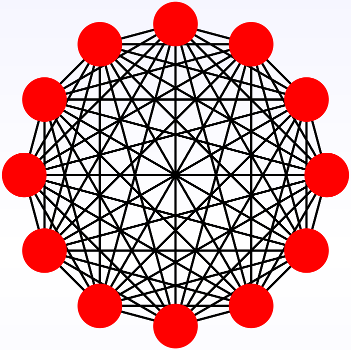


Theorem

On the $(\log \lambda)$*-dimensional hypercube*

$$E(T^{\mathrm{par}}) \le O\left(\frac{m + \sum_{i=1}^{m-1} \log(1/s_i)}{p}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}.$$

Upper Bounds for Complete Graphs/Offspring Populations



Theorem

On the λ-vertex complete graph K_λ (or the $(1+\lambda)$ EA, if $p = 1$)

$$E(T^{\mathrm{par}}) \;\leq\; O(m/p) + \frac{1}{\lambda}\sum_{i=1}^{m-1}\frac{1}{s_i}\,.$$

Big Hammer

Upper bounds on expected parallel time

Ring: $O\left(\frac{1}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Grid: $O\left(\frac{1}{p^{2/3}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Hypercube: $O\left(\frac{m + \sum_{i=1}^{m-1} \log(1/s_i)}{p}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Complete: $O(m/p) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$.

Remarks

- "O" used for convenience, constant factors available and small
- Refined bound for complete graph with small p (small probability of migrating to any island) [Lässig and Sudholt, ECJ 2014].
- Similar upper bounds hold for periodic migration with migration interval $\tau = 1/p$ [Mambrini and Sudholt, GECCO 2014].

Big Hammer Applied to Parallel (1+1) EA on LeadingOnes

Recall: $s_i \geq 1/(en)$ for all $0 \leq i < n$.

Upper bounds on expected parallel time

Ring: $$O\left(\frac{1}{p^{1/2}} \sum_{i=0}^{n-1} e^{1/2} n^{1/2}\right) + \frac{1}{\lambda} \sum_{i=0}^{n-1} en = O\left(\frac{n^{3/2}}{p^{1/2}} + \frac{n^2}{\lambda}\right)$$

Grid: $$O\left(\frac{1}{p^{2/3}} \sum_{i=0}^{n-1} e^{1/3} n^{1/3}\right) + \frac{1}{\lambda} \sum_{i=0}^{n-1} en = O\left(\frac{n^{4/3}}{p^{2/3}} + \frac{n^2}{\lambda}\right)$$

Hypercube: $$O\left(\frac{n + \sum_{i=0}^{n-1} \log(en)}{p}\right) + \frac{1}{\lambda} \sum_{i=0}^{n-1} en = O\left(\frac{n \log n}{p} + \frac{n^2}{\lambda}\right)$$

Complete: $$O(m/p) + \frac{1}{\lambda} \sum_{i=0}^{n-1} en = O\left(\frac{n}{p} + \frac{n^2}{\lambda}\right)$$

So What?

Asymptotic linear speedup for λ such that red term $= O\left(\frac{1}{\lambda} \cdot \sum_{i=1}^{m-1} \frac{1}{s_i}\right)$

if $\sum_{i=1}^{m-1} \frac{1}{s_i}$ is asymptotically tight for a single island.

Parallel (1+1) EA with $p = 1$ on LeadingOnes

	parallel time	linear speedup if	best time bound
Ring:	$O\left(n^{3/2} + \frac{n^2}{\lambda}\right)$	$\lambda = O\left(n^{1/2}\right)$	$O\left(n^{3/2}\right)$
Grid:	$O\left(n^{4/3} + \frac{n^2}{\lambda}\right)$	$\lambda = O\left(n^{2/3}\right)$	$O\left(n^{4/3}\right)$
Hypercube:	$O\left(n \log n + \frac{n^2}{\lambda}\right)$	$\lambda = O(n/\log n)$	$O(n \log n)$
Complete:	$O\left(n + \frac{n^2}{\lambda}\right)$	$\lambda = O(n)$	$O(n)$

Upper bounds and realms for linear speedups improve with density.

Caution

Upper bounds and speedup conditions may not be tight.

Further Applications [Lässig and Sudholt, ECJ 2014]

		(1+1) EA	Ring	Grid/Torus	Hypercube	Complete
OneMax	best λ		$\lambda = \Theta(\log n)$	$\lambda = \Theta(\log n)$	$\lambda = \Theta(\log n)$	$\lambda = \Theta(\log n)^{\dagger}$
	$\mathrm{E}(T^{\mathrm{par}})$	$\Theta(n \log n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
	$\mathrm{E}(T^{\mathrm{seq}})$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$
LO	best λ		$\lambda = \Theta(n^{1/2})$	$\lambda = \Theta(n^{2/3})$	$\lambda = \Theta\left(\frac{n}{\log n}\right)$	$\lambda = \Theta(n)$
	$\mathrm{E}(T^{\mathrm{par}})$	$\Theta(n^2)$	$\Theta(n^{3/2})$	$\Theta(n^{4/3})$	$\Theta(n \log n)$	$\Theta(n)$
	$\mathrm{E}(T^{\mathrm{seq}})$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$
unimodal	best λ		$\lambda = \Theta(n^{1/2})$	$\lambda = \Theta(n^{2/3})$	$\lambda = \Theta\left(\frac{n}{\log n}\right)$	$\lambda = \Theta(n)$
	$\mathrm{E}(T^{\mathrm{par}})$	$O(dn)$	$O\left(dn^{1/2}\right)$	$O\left(dn^{1/3}\right)$	$O(d \log n)$	$O(d)$
	$\mathrm{E}(T^{\mathrm{seq}})$	$O(dn)$	$O(dn)$	$O(dn)$	$O(dn)$	$O(dn)$
Jump_k	best λ		$\lambda = \Theta(n^{k/2})$	$\lambda = \Theta(n^{2k/3})$	$\lambda = \Theta(n^{k-1})$	$\lambda = \Theta(n^{k-1})$
	$\mathrm{E}(T^{\mathrm{par}})$	$\Theta(n^k)$	$O\left(n^{k/2}\right)$	$O\left(n^{k/3}\right)$	$O(n)$	$O(n)$
	$\mathrm{E}(T^{\mathrm{seq}})$	$\Theta(n^k)$	$O\left(n^k\right)$	$O\left(n^k\right)$	$O\left(n^k\right)$	$O\left(n^k\right)$

† Refined analysis for (1+λ) EA on OneMax: linear speedups for $\lambda = O\left(\frac{(\ln n)(\ln \ln n)}{\ln \ln \ln n}\right)$ [Jansen, De Jong, Wegener, 2005]

Conclusions for Fitness-Levels for Parallel EAs

Upper bounds on expected parallel time

Ring: $O\left(\frac{1}{p^{1/2}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/2}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Grid: $O\left(\frac{1}{p^{2/3}} \sum_{i=1}^{m-1} \frac{1}{s_i^{1/3}}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Hypercube: $O\left(\frac{m + \sum_{i=1}^{m-1} \log(1/s_i)}{p}\right) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$

Complete: $O(m/p) + \frac{1}{\lambda} \sum_{i=1}^{m-1} \frac{1}{s_i}$.

Applicable to island models running any elitist EA.

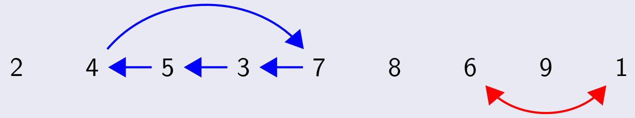
Transfer bounds for panmictic EAs to parallel EAs: plug in s_i's, simplify.

Can find range of λ that guarantees linear speedups.

Overview

Sorting [Scharnow, Tinnefeld, Wegener 2004]

Task: maximize sortedness of n different elements.



Measures of sortedness

- INV(π): number of pairs $(i, j), 1 \leq i < j \leq n$, such that $\pi(i) < \pi(j)$
- HAM(π): number of indices i such that $\pi(i) = i$
- LAS(π): largest k such that $\pi(i_1) < \cdots < \pi(i_k)$
- EXC(π): minimal number of exchanges to sort the sequence

Speedups for Sorting [Lässig and Sudholt, ISAAC 2011 & TCS 2014]

Setting: islands run (1+1) EA, deterministic migration ($p = 1$).

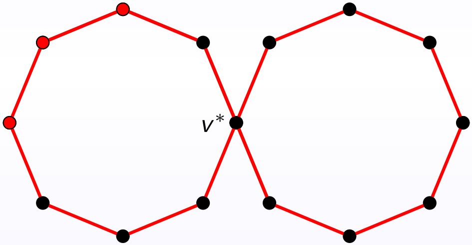
Expected parallel optimization times

Algorithm	INV	HAM, LAS, EXC
single island	$O(n^2 \log n)$ [STW04]	$O(n^2 \log n)$ [STW04]
island model on ring	$O\left(n^2 + \frac{n^2 \log n}{\lambda}\right)$	$O\left(n^{3/2} + \frac{n^2 \log n}{\lambda}\right)$
island model on torus	$O\left(n^2 + \frac{n^2 \log n}{\lambda}\right)$	$O\left(n^{4/3} + \frac{n^2 \log n}{\lambda}\right)$
island model on K_λ	$O\left(n^2 + \frac{n^2 \log n}{\lambda}\right)$	$O\left(n + \frac{n^2 \log n}{\lambda}\right)$

Eulerian Cycles [Neumann 2008]

An illustrative example where diversity in island models helps.

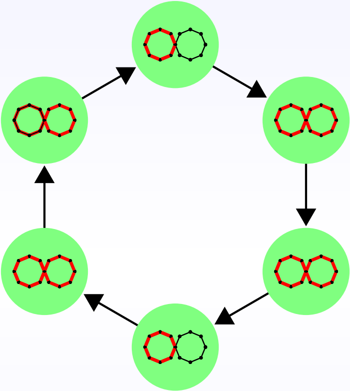
Representation: edge sequence encodes walk.



Expected time for rotation: $\Theta(m^4)$.

Expected time without rotation: $\Theta(m^3)$.

Speedups for Eulerian Cycles on G'



Frequent migrations

$\tau = O(m^2/(\text{diam}(T) \cdot \lambda))$ implies expected time $\Omega(m^4/(\log \lambda))$.

Rare migrations

$\tau \geq m^3$ implies expected time $O(m^3 + 3^{-\lambda} \cdot m^4)$.

Migration interval τ decides between logarithmic vs. exponential speedup!

Eulerian Cycles: More Clever Designs

More efficient operators

Using tailored mutation operators [Doerr, Hebbinghaus, Neumann, ECJ'07] removes the random-walk behaviour and the performance gap disappears.

More efficient representations

The best known representation, adjacency list matchings [Doerr, Johannsen, GECCO 2007], can be parallelised efficiently for all instances (fitness-level method applies).

Parallelisability depends on operators and representation!

Island Models with Crossover

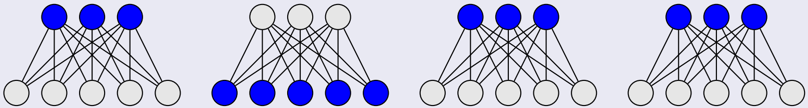
[Neumann, Oliveto, Rudolph, Sudholt, GECCO 2011]

Crossover requires good diversity between parents.

Solutions on different islands might have good diversity.

How efficient are island models when crossing immigrants with residents?
(Common practice in cellular EAs.)

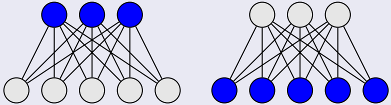
Vertex Cover instance



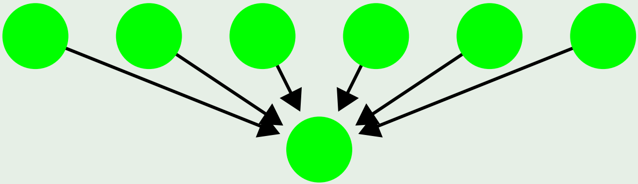
Difficult for (μ+1) EAs [Oliveto, He, Yao, IEEE TEVC 2009].

Island Models with Crossover

Vertex Cover instance



Single-receiver model [Watson and Jansen, GECCO 07]



Each globally optimal configuration is found on some island.

Receiver island uses crossover to assemble all of these.

Island model succeeds in polynomial time with high probability.

Communication Effort

Infrequent migration and sparse topologies increase diversity.

Further benefit: fewer individuals being transmitted between islands.

- migration takes time (adding to number of generations)
- traffic might incur additional costs

Communication effort

Total number of individuals transmitted between islands during a run of an island model.

Can be bounded using our "big hammer":

$$\mathrm{E}(\text{communication effort}) = p\nu \cdot |E| \cdot \mathrm{E}(T^{\mathrm{par}}).$$

Set Cover Problem

- $S = \{s_1, \cdots, s_m\}$ a set containing m elements
- $C = \{C_1, \cdots, C_n\}$ a collection of n subsets of S
- Each set C_i has a cost $c_i > 0$.
- Goal: find selection $x_1 \cdots x_n$ of sets such that $\bigcup_{i:x_i=1} C_i = S$ and $\sum_{i:x_i=1} c_i$ is minimised.

SETCOVER is NP-hard, so look for poly-time approximation algorithms.

Greedy algorithm with approximation ratio H_m

Starting from an empty selection, the greedy algorithm in each step adds the most cost-effective set (highest ratio between the number of newly covered elements and the cost of the set).

SetCover as a multi-objective optimization problem

Minimize $f(x) = (u(x), \text{cost}(x))$ [Friedrich *et al.*, ECJ 2010]

- $\text{cost}(x)$ is the cost of the selection
- $u(x)$ is the number of uncovered elements

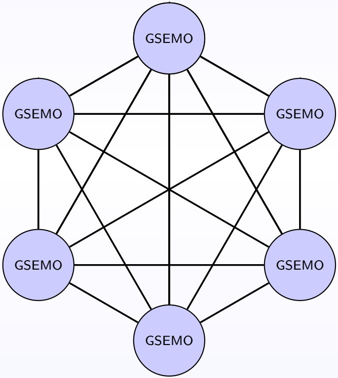
Global SEMO (GSEMO)

1. Initialize $P := \{s\}$ uniformly at random
2. Repeat
   a) Choose $s \in P$ randomly
   b) Define s' by flipping each bit of s independently with probability $1/n$
   c) Add s' to P
   d) Remove the dominated individuals from P

Theorem ([Friedrich *et al.*, 2010])

For an empty initialisation and every SetCover *instance GSEMO finds an H_m-approximate solution in $O(m^2 n)$ generations.*

Homogeneous island model for SetCover



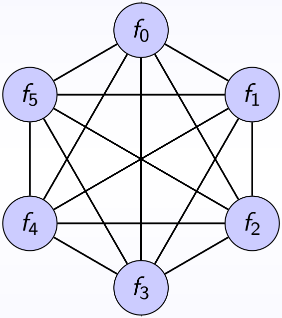
[Mambrini, Sudholt, Yao, PPSN 2012]

The algorithm

For each island i:

- Simulate one generation of GSEMO
- Send a copy of the population $P_i^{(t)}$ to each neighbouring island with probability p.
- Unify $P_i^{(t)}$ with all populations received from other islands.
- Remove all dominated points from $P_i^{(t)}$.

Heterogeneous island model



- $m + 1$ islands
- each island runs a (1+1) EA on a different fitness function

$$f_i(X) = \begin{cases} nc_{\max} - \mathrm{cost}(X) & \text{if } c(X) = i \\ -|c(X) - i| & \text{if } c(X) \neq i \end{cases}$$

Migration policies

- Complete: send copy to all islands.
- "Smart": send copy to island $c(X)$.

Results comparison

Algorithm	parallel time bounds general b. $\rightsquigarrow$ best bound		comm. effort
Non-parallel GSEMO	$O(nm^2)$	$\rightsquigarrow O(nm^2)$	0
GSEMO based homogeneous island models with topology...			
– complete ($\lambda \leq pnm$)	$O\left(\frac{nm^2}{\lambda}\right)$	$\rightsquigarrow O\left(\frac{m}{p}\right)$	$O(p^2 n^2 m^4)$
– grid ($\lambda \leq (pnm)^{2/3}$)	$O\left(\frac{nm^2}{\lambda}\right)$	$\rightsquigarrow O\left(\frac{n^{1/3}m^{4/3}}{p^{2/3}}\right)$	$O(pnm^3)$
– ring ($\lambda \leq \sqrt{pnm}$)	$O\left(\frac{nm^2}{\lambda}\right)$	$\rightsquigarrow O\left(\frac{n^{1/2}m^{3/2}}{p^{1/2}}\right)$	$O(pnm^3)$
(1+1) EA based heterogeneous island models with migration policy...			
– complete ($\lambda \leq m$)	$O\left(\frac{nm^2}{\lambda}\right)$	$\rightsquigarrow O(nm)$	$O(nm^3)$
– smart migration ($\lambda \leq m$)	$O\left(\frac{nm^2}{\lambda}\right)$	$\rightsquigarrow O(nm)$	$O(nm^2)$

- Homogeneous: trade-offs between parallel time and comm. effort
- Heterogeneous: A simpler model providing the linear speed-up with less communication effort (with smart policy)

Monday, Poster Session 2, Proc. p. 243 [Joshi, Rowe, Zarges]

An Immune-Inspired Algorithm for the Set Cover Problem

Overview

Adaptive Schemes for Choice of λ

How to find a proper number of islands/offspring? [Lässig and Sudholt, FOGA 2011]

Here: only consider K_μ.

Scheme A

- double population size if no improvement
- if improvement reset population size to 1

Scheme B

- double population size if no improvement
- if improvement halve population size

Offspring population size in (1+λ) EA [Jansen, De Jong, Wegener, 2005]

- double population size if no improvement
- if $s \geq 1$ improvements then divide population size by s

Schema A

Theorem

Given a fitness-level partition $A_1, \dots, A_m$,

$$\mathrm{E}(T_{\mathrm{A}}^{\mathrm{seq}}) \le 2 \sum_{i=1}^{m-1} \frac{1}{s_i}\ .$$

If each A_i contains a single fitness value, then also

$$\mathrm{E}(T_{\mathrm{A}}^{\mathrm{par}}) \le 2 \sum_{i=1}^{m-1} \log\left(\frac{2}{s_i}\right)\ .$$

Population size reaches "critical mass" $1/s_i$ after doubling $\log(1/s_i)$ times.

Schema B

Theorem

Given a fitness-level partition $A_1, \dots, A_m$,

$$\mathrm{E}(T_{\mathrm{B}}^{\mathrm{seq}}) \leq 3 \sum_{i=1}^{m-1} \frac{1}{s_i} .$$

If each A_i contains a single fitness value, then also

$$\mathrm{E}(T_{\mathrm{B}}^{\mathrm{par}}) \leq 4 \sum_{i=1}^{m-1} \log\left(\frac{2}{s_j}\right) .$$

Stronger bound: if additionally $s_1 \geq s_2 \geq \dots \geq s_{m-1}$ then

$$\mathrm{E}(T_{\mathrm{B}}^{\mathrm{par}}) \leq 3m + \log\left(\frac{1}{s_{m-1}}\right) .$$

Scheme B is able to track good parameters over time.

Example Applications

Parallel (1+1) EA/(1+λ) EA with Adaptive λ

		$\mathrm{E}(T^{\mathrm{seq}})$	$\mathrm{E}(T^{\mathrm{par}})$	best fixed λ
OneMax	A	$\Theta(n \log n)$	$O(n)$	$O\left(\frac{n}{\ln \ln n}\right)$
	B	$\Theta(n \log n)$	$O(n)$	$O\left(\frac{n}{\ln \ln n}\right)$
LO	A	$\Theta(n^2)$	$\Theta(n \log n)$	$O(n)$
	B	$\Theta(n^2)$	$O(n)$	$O(n)$
unimodal f	A	$O(dn)$	$O(d \log n)$	$O(d)$
with d f-values	B	$O(dn)$	$O(d)$	$O(d)$
Jump_k	A	$O(n^k)$	$O(n)$	$O(n)$
$2 \le k \le n/\log n$	B	$O(n^k)$	$O(n)$	$O(n)$

Scheme B performance matches best fixed choice of λ in almost all cases.

Adaptive Migration Intervals [Mambrini and Sudholt, GECCO 2014]

Can we use the same idea to adapt the migration interval τ?

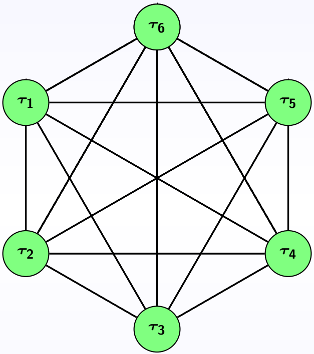
- Goal: minimize communication without compromising exploitation
- Idea: reduce migration if no improvement was found.

Scheme A: double τ if no improvement was found, otherwise set to 1.
Scheme B: double τ if no improvement was found, otherwise halve it.

Related work

Not the first theory-inspired adaptive scheme: Adaptation for fixed-length runs [Osorio, Luque, Alba, ISDA'11 & CEC'13].

Adaptive Scheme A



- Each island has migration interval τ_i

Adapting τ_i

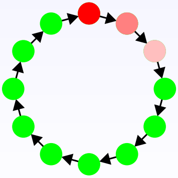
For each island i:

- When an improvement is found set $\boldsymbol{\tau_i = 1}$
- If τ_i generations have passed, migrate and double the migration interval $\boldsymbol{\tau_i' = 2 \cdot \tau_i}$

Effect: improvements are communicated fast, otherwise communication cost is kept low.

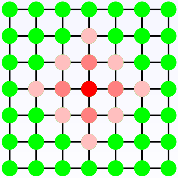
- **Runtime**: same bounds as for fixed scheme with $\tau = 1$
- **Comm. effort**: $\mathrm{E}(T^{\text{com}}) \leq |E| \sum_{i=1}^{m-1} \log\left(\mathrm{E}(T_i^{\text{par}}) + \text{diam}(G)\right)$

Adaptive Scheme A: Bounds for Common Topologies



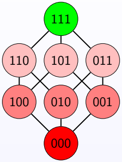
$$\mathrm{E}(T^{\mathrm{par}}) \leq \text{as for } \tau = 1$$

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



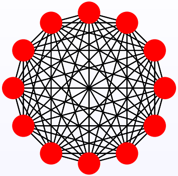
$$\mathrm{E}(T^{\mathrm{par}}) \leq \text{as for } \tau = 1$$

$$\mathrm{E}(T^{\mathrm{com}}) \leq 4\lambda \sum_{i=1}^{m-1} \log\left(\frac{1}{s_i^{1/3}} + \frac{1}{\lambda s_i} + 2\sqrt{\lambda}\right)$$



$$\mathrm{E}(T^{\mathrm{par}}) \leq \text{as for } \tau = 1$$

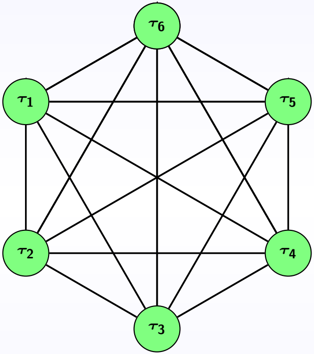
$$\mathrm{E}(T^{\mathrm{com}}) \leq \lambda(\log \lambda) \sum_{i=1}^{m-1} \log\left(\log\left(\frac{1}{s_i}\right) + \frac{1}{\lambda s_i} + \log \lambda\right)$$



$$\mathrm{E}(T^{\mathrm{par}}) \leq \text{as for } \tau = 1$$

$$\mathrm{E}(T^{\mathrm{com}}) \leq \lambda(\lambda - 1) \sum_{i=1}^{m-1} \log\left(2 + \frac{1}{\lambda s_i}\right)$$

Adaptive scheme B



Adapting τ_i

For each island i:

- When an improvement is found, halve the migration interval $\tau_i' = \tau_i/2$
- If τ_i generation have passed, migrate and double the migration interval $\tau_i' = 2 \cdot \tau_i$

Upper bounds for complete topology

- similar results as for adaptive λ

Communication Efforts: Summary of Results

- All schemes have the same parallel runtime bound.
- Comparison of communication effort: adaptive vs. fixed scheme with best τ and maximum number of islands:

	OneMax	LeadingOnes	Unimodal	Jump_k
Complete	—	—	—	—
Ring	$\log\log n$	$\sqrt{n}/\log n$	$\sqrt{n}/\log n$	$n^{\frac{k}{2}-1}/(k\log n)$
Grid/Torus	$\log\log n$	$\sqrt[3]{n}/\log n$	$\sqrt[3]{n}/\log n$	$n^{\frac{k}{3}-1}/(k\log n)$
Hypercube	$\log\log\log n$	$\log n/\log\log n$	$\log n/\log\log n$	$\log\log n^{k-1}$

- — same performance
- $f(\cdot)$ Adaptive Scheme is better by $f(\cdot)$
- $f(\cdot)$ Fixed Scheme is better by $f(\cdot)$

Overview

Black-Box Complexity for Parallel EAs

Black-Box Complexity of function class $\mathcal{F}_n$ [Droste, Jansen, Wegener 2006]

- Black-box algorithms: query x_t based on $f(x_0), f(x_1), \dots, f(x_{t-1})$.
- Minimum number of queries to the black box needed by *every black-box algorithm* to find optimum on hardest instance in $\mathcal{F}_n$.
- General limits on performance across all search heuristics.

All black-box models query one search point at a time.
$(\mu+\lambda)$ EAs, (μ,λ) EAs, island models, cellular EAs query λ search points.

How about a black-box complexity for λ parallel queries?

- Universal lower bounds considering the degree of parallelism λ.
- Identify for which λ (strong) linear speedups are impossible.

Want to know more? Come see our poster! :-)

Monday, Poster Session 1, Proc. p. 892 [Badkobeh, Lehre, Sudholt]

Unbiased Black-Box Complexity of Parallel Search

Structured Populations in Population Genetics

Speed of Adaptation in Evolutionary Computation and Population Genetics (SAGE, 2014–2016)

Interdisciplinary EU FET project between Evolutionary Computation (Nottingham, Jena, Sheffield) and Population Genetics (IST Austria)

Goals

- Bring together Evolutionary Computation and Population Genetics
- Transfer and development of methods and results
- A unified theory of evolutionary processes in natural and artificial evolution.

Can Population Genetics help us understand how structured populations evolve?

Conclusions

Insight into how parallel evolutionary algorithms work.

- Examples where parallel EAs excel
- Methods and ideas for the analysis of parallel EAs
- How to transfer fitness-level bounds from panmictic to parallel EAs
- How to determine good parameters
- Inspiration for new EA designs

Speedup/parallelizability determined by

- migration topology
- fitness function
- mutation operators
- representation
- migration interval τ
- use of crossover

Rich, fruitful and exciting research area!

Selected Literature I

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

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

E. Alba, M. Giacobini, M. Tomassini, and S. Romero.
Comparing synchronous and asynchronous cellular genetic algorithms.
In *Parallel Problem Solving from Nature VII*, volume 2439 of *LNCS*, pages 601–610. Springer, 2002.

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

E. Alba and M. Tomassini.
Parallelism and evolutionary algorithms.
IEEE Transactions on Evolutionary Computation, 6:443–462, 2002.

E. Alba and J. M. Troya.
A survey of parallel distributed genetic algorithms.
Complexity, 4:31–52, 1999.

G. Badkobeh, P. K. Lehre, and D. Sudholt.
Unbiased black-box complexity of parallel search.
In *13th International Conference on Parallel Problem Solving from Nature (PPSN 2014)*, pages 892–901. Springer, 2014.

Selected Literature II

R. S. Barr and B. L. Hickman.
Reporting computational experiments with parallel algorithms: Issues, measures, and experts' opinion.
ORSA Journal on Computing, 5(1):2–18, 1993.

E. Cantú Paz.
A survey of parallel genetic algorithms, technical report, illinois genetic algorithms laboratory, university of illinois at urbana champaign, urbana, il, 1997.
Technical report, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana Champaign, Urbana, IL.

D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre.
Level-based analysis of genetic algorithms and other search processes.
In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, number 8672 in Lecture Notes in Computer Science, pages 912–921. Springer, 2014.

T. G. Crainic and N. Hail.
Parallel Metaheuristics Applications.
Wiley-Interscience, 2005.

D.-C. Dang and Lehre, Per Kristian.
Refined upper bounds on the expected runtime of non-elitist populations from fitness-levels.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 1367–1374, 2014.

M. De Felice, S. Meloni, and S. Panzieri.
Effect of topology on diversity of spatially-structured evolutionary algorithms.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '11)*, pages 1579–1586. ACM, 2011.

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

Selected Literature III

B. Doerr, N. Hebbinghaus, and F. Neumann.
Speeding up evolutionary algorithms through asymmetric mutation operators.
Evolutionary Computation, 15:401–410, 2007.

B. Doerr and D. Johannsen.
Adjacency list matchings—an ideal genotype for cycle covers.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '07)*, pages 1203–1210. ACM Press, 2007.

B. Doerr, C. Klein, and T. Storch.
Faster evolutionary algorithms by superior graph representation.
In *First IEEE Symposium on Foundations of Computational Intelligence (FOCI '07)*, pages 245–250. IEEE, 2007.

S. Droste, T. Jansen, and I. Wegener.
On the analysis of the (1+1) evolutionary algorithm.
Theoretical Computer Science, 276(1–2):51–81, 2002.

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

T. Friedrich, P. S. Oliveto, D. Sudholt, and C. Witt.
Analysis of diversity-preserving mechanisms for global exploration.
Evolutionary Computation, 17(4):455–476, 2009.

M. Giacobini, E. Alba, A. Tettamanzi, and M. Tomassini.
Modeling selection intensity for toroidal cellular evolutionary algorithms.
In *Proceedings of the Genetic and Evolutionary Computation conference (GECCO '04)*, pages 1138–1149. Springer, 2004.

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

Selected Literature IV

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

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

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

T. Jansen, K. A. De Jong, and I. Wegener.
On the choice of the offspring population size in evolutionary algorithms.
Evolutionary Computation, 13:413–440, 2005.

T. Jansen, P. S. Oliveto, and C. Zarges.
On the analysis of the immune-inspired b-cell algorithm for the vertex cover problem.
In *Proc. of the 10th International Conference on Artificial Immune Systems (ICARIS 2011)*, volume 6825 of *LNCS*, pages 117–131. Springer, 2011.

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

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`.

Selected Literature V

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

J. Lässig and D. Sudholt.
Experimental supplements to the theoretical analysis of migration in the island model.
In *11th International Conference on Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238 of *LNCS*, pages 224–233. Springer, 2010.

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

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

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

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

J. Lässig and D. Sudholt.
Analysis of speedups in parallel evolutionary algorithms and $(1+\lambda)$ EAs for combinatorial optimization.
Theoretical Computer Science, 551:66–83, Sept. 2014.

Selected Literature VI

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

A. Mambrini and D. 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.

N. Nedjah, L. de Macedo Mourelle, and E. Alba.
Parallel Evolutionary Computations.
Springer, 2006.

F. Neumann.
Expected runtimes of evolutionary algorithms for the Eulerian cycle problem.
Computers & Operations Research, 35(9):2750–2759, 2008.

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

F. Neumann and I. Wegener.
Randomized local search, evolutionary algorithms, and the minimum spanning tree problem.
Theoretical Computer Science, 378(1):32–40, 2007.

P. S. Oliveto, J. He, and X. Yao.
Analysis of the (1+1)-EA for finding approximate solutions to vertex cover problems.
IEEE Transactions on Evolutionary Computation, 13(5):1006–1029, 2009.

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

Selected Literature VII

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

G. Rudolph.
Takeover times and probabilities of non-generational selection rules.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00)*, pages 903–910. Morgan Kaufmann, 2000.

G. Rudolph.
Takeover times of noisy non-generational selection rules that undo extinction.
In *Proceedings of the 5th International Conference on Artificial Neural Nets and Genetic Algorithms (ICANNGA 2001)*, pages 268–271. Springer, 2001.

G. Rudolph.
Takeover time in parallel populations with migration.
In B. Filipic and J. Silc, editors, *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2006)*, pages 63–72, 2006.

J. Scharnow, K. Tinnefeld, and I. Wegener.
The analysis of evolutionary algorithms on sorting and shortest paths problems.
Journal of Mathematical Modelling and Algorithms, 3(4):349–366, 2004.

Z. Skolicki.
An Analysis of Island Models in Evolutionary Computation.
PhD thesis, George Mason University, Fairfax, VA, 2000.

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

Selected Literature VIII

D. Sudholt.
General lower bounds for the running time of evolutionary algorithms.
In *11th International Conference on Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238 of *LNCS*, pages 124–133. Springer, 2010.

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

F. Teytaud and O. Teytaud.
Log(λ) modifications for optimal parallelism.
In *11th International Conference on Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238 of *LNCS*, pages 254–263. Springer, 2010.

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

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.

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

C. Witt.
Fitness levels with tail bounds for the analysis of randomized search heuristics.
Information Processing Letters, 114(1–2):38–41, 2014.

Thank you!

Questions?