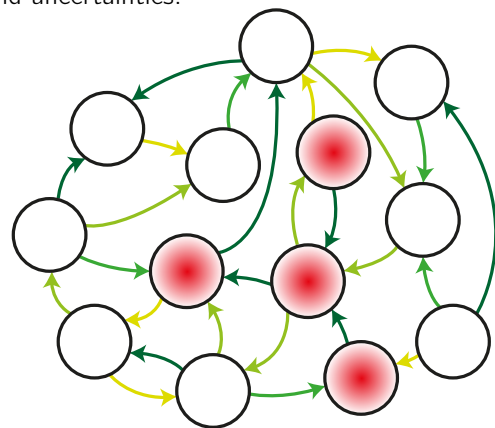


Distributed Evolutionary Search

Motivation There is hardly any company that has no need for optimization; whether it is the scheduling of employees, routing of trains or trucks, cutting parts from wooden beams with as little waste as possible or one of many other tasks. These optimization problems are typically specific to the application area and come with various constraints and uncertainties.

Evolutionary Search is a modern tool for optimizing such hard problems. By taking ideas from nature (such as using populations of solutions, mutating or combining them and selecting the best for a new generation), these algorithms are surprisingly successful in many different application domains where no tailored solutions exist. These simple algorithms cannot compete with state-of-the-art solvers in their specific domains (for example SAT-solvers), but they are very easy to apply to a wide range of optimization problems where no specific solvers are available, at very low implementation cost. Furthermore, they are extensible in the sense that performance can be improved by spending time improving the algorithm with additional heuristics and ideas.



To make use of massive parallelization in optimization algorithms, *distributed* evolutionary search spreads populations over multiple cores, called *islands* (borrowing another term from natural evolution). The islands optimize in parallel, sending information to neighboring islands when they make progress. Which islands neighbor which other islands is defined by an underlying *topology*. The goal is to have a *linear speedup* in terms of the number of cores used, thus achieving the optimal performance gain from additional hardware.

Purpose of the project We are interested in analyzing the efficiency of this approach, using techniques from the analysis of randomized algorithms, in particular the analysis of random processes.

For this project, we are interested in the following questions.

1. How efficient are different topologies at spreading information?
2. What use is a slow spread of information?

The first question is always dependent on the setting of what needs to be optimized; typically, a complete graph as underlying topology gives the fastest spread in information. The second question asks for a setting in which a fast spread of information is not desirable, in other words, a setting in which the underlying topology should not be the complete graph but rather, for example, a ring or torus topology.

The first question has gotten a preliminary answer in [1]; however, the offered bounds are not tight and can be improved. The second question has not yet been addressed in the literature. Also, the cost of communication has not been sufficiently considered (the complete graph as topology has a high communication overhead, the ring topology only has a very small overhead).

What we expect from you We expect basic proficiency in the analysis of random algorithms, a familiarity with graph-theoretic concepts and the curiosity and willingness to delve into an interesting research topic within Theoretical Computer Science. Our main goal is a rigorous understanding of distributed evolutionary algorithms and we expect you to contribute theoretical results.

What you can expect from us We will gently introduce you to the field and accompany you all along the interesting journey. This will be a team effort, and we aim at publishing our results at a renowned international conference (the deadline we have in mind is typically in early February).

How to contact us You're welcome to visit us on floor A-1 or send us an e-mail:

Prof. Dr. Tobias Friedrich > firstname.lastname@hpi.de
Dr. Timo Kötzing

“ Islands are natural workshops of evolution.



– Richard Dawkins

[1] Golnaz Badkobeh, Per Kristian Lehre, Dirk Sudholt: Black-box Complexity of Parallel Search with Distributed Populations, FOGA 2015. <http://dx.doi.org/10.1145/2725494.2725504>.