

Comparing Global and Local Mutations on Bit Strings

Benjamin Doerr
Max-Planck-Institut für
Informatik
Campus E 1 4
Saarbrücken, Germany

Thomas Jansen
Technische Universität
Dortmund
Otto-Hahn-Straße 14
Dortmund, Germany

Christian Klein
Max-Planck-Institut für
Informatik
Campus E 1 4
Saarbrücken, Germany

ABSTRACT

Evolutionary algorithms operating on bit strings usually employ a global mutation where each bit is flipped independently with some mutation probability. Most often the mutation probability is set fixed in a way that on average exactly one bit is flipped in a mutation. A seemingly very similar concept is a local one realized by an operator that flips exactly one bit chosen uniformly at random.

Most known results indicate that the global approach leads to run-times at least as good as the local approach. The draw-back is that the global approach is much harder to analyze. It would therefore be highly useful to derive general principles of when and how results for the local operator extend to the global ones.

In this paper, we show that there is little hope for such general principles, even under very favorable conditions. We show that there is a fitness function such that the local operator from each initial search point finds the optimum in small polynomial time, whereas the global operator for almost all initial search points needs a weakly exponential time.

Categories and Subject Descriptors

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

General Terms

Theory, Algorithms

Keywords

Evolutionary Computation, Randomized Local Search, Mutation, Analysis

This work benefited from being presented and discussed at the Dagstuhl seminar 08051 on Theory of Evolutionary Algorithms.

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07...\$5.00.

1. INTRODUCTION

Evolutionary algorithms (EAs) are typically described as robust general problem solvers. They are able to perform a global search different from gradient-descent methods or hill-climbers, which easily are trapped in local optima.

It is in fact easy to prove that evolutionary algorithms find a global optimum with probability converging to 1 with time if they make use of a positive mutation operator, i. e., a mutation that changes any point in the search space to any other point in the search space with positive probability. If an EA operates on bit strings of fixed length n , the most commonly used mutation operator is standard bit mutation. With standard bit mutation, each bit is flipped independently with a fixed mutation probability p_m . The most recommended choice for the mutation probability is $p_m = 1/n$.

Clearly, with mutation probability $p_m = 1/n$, on average exactly 1 bit is flipped in each mutation. Therefore, it seems to be a small change to replace standard bit mutation by a local mutation operator that flips exactly one bit chosen uniformly at random. However, with such a local mutation operator the EA may now get stuck in a local optimum, and consequently, the probability to finally reach the global optimum might no longer converge to 1 with time. In addition, most results indicate that the local operator in case of convergence leads to similar run-times as the global one. Therefore, one might be tempted to believe that the global operator generally is superior to the local one. Since typically rigorous analyses for the global operator are much harder than for the local one, general results of how a good optimization behavior of the local operator extends to the global one would be highly desirable.

When analyzing such general phenomena of evolutionary algorithms one often considers particularly simple evolutionary algorithms to facilitate a rigorous analysis. The probably most simple example is the well-known $(1+1)$ EA. It uses a population of size only 1, produces only 1 offspring using standard bit mutation and a plus-selection. Thus, the parent x is replaced by its offspring y if and only if $f(y) > f(x)$ holds (assuming that we want to maximize the fitness function f). If we replace standard bit mutation by a local mutation operator that flips exactly one bit chosen uniformly at random, we obtain an algorithm that is well known as randomized local search (RLS). Since RLS is a hill-climber and no evolutionary algorithm, the $(1+1)$ EA is right on the borderline and a comparison of RLS and the $(1+1)$ EA is a comparison between an evolutionary algorithm and a simpler search heuristic. This is one motivation for comparing the performance of these two algorithms in a rigorous way.

