Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Discrete Algorithms 4 (2006) 181–199

JOURNAL OF DISCRETE ALGORITHMS

www.elsevier.com/locate/jda

# On the analysis of a dynamic evolutionary algorithm [☆]

Thomas Jansen, Ingo Wegener [*]

*FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany*

Available online 2 February 2005

**Abstract**

Evolutionary algorithms are applied as problem-independent optimization algorithms. They are quite efficient in many situations. However, it is difficult to analyze even the behavior of simple variants of evolutionary algorithms like the $(1+1)$ EA on rather simple functions. Nevertheless, only the analysis of the expected run time and the success probability within a given number of steps can guide the choice of the free parameters of the algorithms. Here static $(1+1)$ EAs with a fixed mutation probability are compared with dynamic $(1+1)$ EAs with a simple schedule for the variation of the mutation probability. The dynamic variant is first analyzed for functions typically chosen as example-functions for evolutionary algorithms. Afterwards, it is shown that it can be essential to choose the suitable variant of the $(1+1)$ EA. More precisely, functions are presented where each static $(1+1)$ EA has exponential expected run time while the dynamic variant has polynomial expected run time. For other functions it is shown that the dynamic $(1+1)$ EA has exponential expected run time while a static $(1+1)$ EA with a good choice of the mutation probability has polynomial run time with overwhelming probability.
© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Evolutionary algorithms; Run time analysis; Mutation probability; Time-dependent parameter setting

☆ This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

* Corresponding author.
*E-mail addresses:* jansen@ls2.cs.uni-dortmund.de (T. Jansen), wegener@ls2.cs.uni-dortmund.de (I. Wegener).

## 1. Introduction

The design and analysis of problem-specific optimization algorithms is a well-established subject. Many tools for the analysis of deterministic or randomized algorithms have been presented and applied. General randomized search heuristics (like simulated annealing or evolutionary algorithms) are problem-independent optimization algorithms. The idea is to design optimization algorithms, which are robust, i.e., they have a good, although not optimal behavior for many of the "typical problems". Nevertheless, it is quite obvious that a problem-specific algorithm will outperform a problem-independent search heuristic. Therefore, it is useful to add problem-specific modules to search heuristics when applying them to problems with a known structure.

However, some people underestimate the need for randomized search heuristics. In the following two scenarios, randomized search heuristics are a good choice. If a company has to solve an optimization problem where no problem-specific algorithm is known, there are often not enough resources (time, money, experts) to design a problem-specific algorithm and it is better to apply a randomized search heuristic. The second scenario is called black-box optimization. This is the case in many engineering applications. There is no full information about the problem instance since, e.g., some parameters are unknown. This excludes problem-specific algorithms but randomized search heuristics are applicable in this scenario. For details of this scenario see Droste et al. [7].

Hence, we claim that it is necessary to analyze the behavior of randomized search heuristics on selected problems in order to understand their advantages and disadvantages. We do not claim that these randomized search heuristics in their pure form outperform problem-specific algorithms. We concentrate on the maximization of discrete or pseudo-Boolean functions $f:\{0,1\}^n \to \mathbb{R}$ and we analyze the dynamic $(1+1)$ EA which is defined in the following way.

**Algorithm 1.** Dynamic $(1+1)$ EA for functions $f:\{0,1\}^n \to \mathbb{R}$.

1. Choose a sequence $p_t(n) \in (0, 1/2)$ called mutation probabilities for step $t$.
2. Choose $x \in \{0,1\}^n$ uniformly at random. Set $t = 1$.
3. Let $y$ be the result of flipping each bit in $x$ independently with probability $p_t(n)$ (mutation).
4. If $f(y) \geqslant f(x)$, set $x := y$ (selection).
5. Increase $t$ by 1.
6. Stop if some stopping criterion is fulfilled. Otherwise, continue at step 3.

We do not specify a stopping criterion since we investigate the dynamic $(1+1)$ EA as infinite stochastic process which never stops. The dynamic $(1+1)$ EA can be generalized to a dynamic $(\mu+\lambda)$ EA. Then $\mu$ search points are stored. In order to create $\lambda$ new search points (called children) we have to choose $\lambda$ parents from the $\mu$ existing search points (with possible repetitions). Since there are many selection procedures for this step, we do not describe the dynamic $(\mu+\lambda)$ EA in detail. In any case, the $\lambda$ best search points of the multiset of parents and children are chosen to be stored (to build the next generation). There has to be a rule to break ties.