Chapter 14

METHODS FOR THE ANALYSIS OF EVOLUTIONARY ALGORITHMS ON PSEUDO-BOOLEAN FUNCTIONS*

Ingo Wegener

Abstract Many experiments have shown that evolutionary algorithms are useful randomized search heuristics for optimization problems. In order to learn more about the reasons for their efficiency and in order to obtain proven results on evolutionary algorithms it is necessary to develop a theory of evolutionary algorithms. Such a theory is still in its infancy. A major part of a theory is the analysis of different variants of evolutionary algorithms on selected functions. Several results of this kind have been obtained during the last years. Here important analytical tools are presented, discussed, and applied to well-chosen example functions.

1. Introduction

Evolutionary algorithms are randomized search heuristics with many applications, e.g., in optimization, adaptation, classification, control systems, or learning. Here we focus on optimization (for an overview on the whole area we refer to Bäck, Fogel, and Michalewicz, 1997; Fogel, 1995; Goldberg, 1989; Holland, 1975; Schwefel, 1995). Despite the many successful experiments with evolutionary algorithms a theory on evolutionary algorithms is still in its infancy. This holds in particular if one compares the state of the art with the situation on problem-specific deter-

---

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center “Computational Intelligence” (SFB 531).
ministic exact optimization algorithms (Cormen, Leiserson, and Rivest, 1990), deterministic approximation algorithms (Hochbaum, 1997), or randomized optimization and approximation algorithms (Motwani and Raghavan, 1995). One reason is that evolutionary algorithms have been developed by engineers, while the other disciplines have been created by theoreticians (leading sometimes to a lack of experimental knowledge). Moreover, the fundamental idea of evolutionary algorithms is to obtain robust problem-independent search heuristics with a good behavior on many problems from a large variety of problems (this statement remains true, although many evolutionary algorithms also use problem-specific components). This variety of problems makes the analysis of evolutionary algorithms much harder than the analysis of problem-specific algorithms (which often are designed in order to make an analysis possible). Nevertheless, progress on the design and the application of evolutionary algorithms will gain a lot from a theoretical foundation. Nowadays, we are able to analyze evolutionary algorithms without crossover on many functions and evolutionary algorithms with crossover on some functions. The functions are not examples from real-world applications but example functions describing some typical issues of functions (fitness landscapes) or are chosen to show some extreme behavior of evolutionary algorithms. Also very strange functions can be useful in order to disprove widely accepted conjectures or to show the differences between variants of evolutionary algorithms. Altogether, we find a list of interesting theoretical results on evolutionary algorithms, in particular, during the last years. The purpose of this contribution is to present some of the most important tools for such results.

In Section 2, we discuss differences between discrete and non-discrete state spaces and why we investigate the optimization of pseudo-boolean functions $f : \{0, 1\}^n \rightarrow \mathbb{R}$. The aim of an analysis of an evolutionary algorithm is the investigation of certain performance measures. This paper focusses on expected run times and the success probability within reasonable time bounds. The reasons for this decision are presented in Section 3. Since several example functions are used for different purposes all these functions are defined in Section 4. The following three sections show how tail inequalities (Section 5), the coupon collector’s theorem (Section 6), and results on the gambler’s ruin problem (Section 7) can be applied to the analysis of evolutionary algorithms. Another main idea is to measure the progress of an evolutionary algorithm not with respect to the considered fitness function but with some cruder scale. In Section 8 and Section 9, upper and lower bounds on the expected run time of evolutionary algorithms are proved using levels based on intervals of fitness values. The method of using so-called potential functions for the