



ELSEVIER

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcsImproved time complexity analysis of the Simple Genetic Algorithm [☆]Pietro S. Oliveto ^{a,*}, Carsten Witt ^b^a Department of Computer Science, University of Sheffield, United Kingdom^b DTU Compute, Technical University of Denmark, Denmark

ARTICLE INFO

Article history:

Received 30 October 2013

Received in revised form 14 November 2014

Accepted 9 January 2015

Available online xxxx

Keywords:

Simple Genetic Algorithm

Crossover

Runtime analysis

ABSTRACT

A runtime analysis of the Simple Genetic Algorithm (SGA) for the ONEMAX problem has recently been presented proving that the algorithm with population size $\mu \leq n^{1/8-\varepsilon}$ requires exponential time with overwhelming probability. This paper presents an improved analysis which overcomes some limitations of the previous one. Firstly, the new result holds for population sizes up to $\mu \leq n^{1/4-\varepsilon}$ which is an improvement up to a power of 2 larger. Secondly, we present a technique to bound the diversity of the population that does not require a bound on its bandwidth. Apart from allowing a stronger result, we believe this is a major improvement towards the reusability of the techniques in future systematic analyses of GAs. Finally, we consider the more natural SGA using selection with replacement rather than without replacement although the results hold for both algorithmic versions. Experiments are presented to explore the limits of the new and previous mathematical techniques.

Crown Copyright © 2015 Published by Elsevier B.V. All rights reserved.

1. Introduction

For many years it has been a challenge to analyze the time complexity of Genetic Algorithms (GAs) using stochastic selection together with crossover and mutation. We have recently presented a first step towards a systematic analysis of GAs through a runtime analysis of the *Simple Genetic Algorithm* (SGA) for ONEMAX [20]. The main result was the proof that the SGA has exponential runtime with overwhelming probability for population sizes up to $\mu \leq n^{1/8-\varepsilon}$ for some arbitrary small constant ε and problem size n .

The main novelties of the work were two. On the one hand, we provided a rigorous proof that the SGA cannot optimize ONEMAX in polynomial time (1). The inefficient hillclimbing performance of the SGA due to the loss of selection pressure was well known in the Evolutionary Computation (EC) community since the algorithm has been well studied in the literature. In fact, [6] reports experimental results in his seminal book showing the loss of selection pressure of the algorithm and suggesting fitness scaling mechanisms to solve the problem. Nevertheless a rigorous proof was yet not available. On the other hand, the major driving force was to obtain a first basis of mathematical techniques towards systematic runtime analyses of GAs using at the same time mutation, crossover and stochastic selection (2). Undoubtedly significant progress

[☆] An extended abstract of this paper without full proofs appeared in the Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation (GECCO'13) [21].

* Corresponding author.

E-mail addresses: P.Oliveto@sheffield.ac.uk (P.S. Oliveto), cawi@dtu.dk (C. Witt).

¹ Supported in part by EPSRC grant EP/H028900/1.

has been achieved in recent years in the runtime analysis of EAs [1,9]. Nowadays, the performance of simple EAs can be analyzed on well-known combinatorial optimization problems [17]. Furthermore, major advances have been achieved in the analysis of population-based EAs with stochastic selection through techniques such as the *simplified negative-drift theorem* [19], the *negative drift in populations theorem* [12] and the *fitness levels for non-elitist populations* technique [13]. However, these techniques cannot be directly applied to the analyses of more realistic GAs incorporating a crossover operator. Several results were indeed available proving that crossover is useful [2,3,10,11,15,18,23,24], but they rely heavily on elitist selection operators. Moreover, mostly only upper bounds on the running time of crossover-based algorithms were available.

In this paper we present an improved runtime analysis of the **SGA** as a first step towards overcoming the limitations of our previous analysis. The first limitation was the bound on the population size $\mu = O(n^{1/8-\varepsilon})$ for the results to hold. The analysis presented here allows population sizes up to a power of 2 larger. Another significant limitation was the necessity for a bound on the so-called *bandwidth* of the population from which a measure on the diversity of the population was derived. The bandwidth was defined as $h - \ell$ where h is the best ONEMAX value in a population and ℓ the worst ONEMAX value, while diversity s was defined as the number of *non-converged* bit positions, that is both bit values are taken by individuals of the population. The whole analysis depended on the fact that if the diversity of the population is sufficiently low, then the behavior of fitness proportional selection is very close to that of uniform selection. The crucial observation to derive a bound on the diversity s was that $h - \ell \leq s$, i.e., the bandwidth cannot be larger than the number of non-converged bits. However, it is a non-trivial task to achieve a bound on the bandwidth for not too small population sizes μ . Furthermore, the bandwidth is heavily dependent on the problem at hand which seriously limits the generality and reusability of the technique. In this paper we present a new technique to bound the diversity of the population that does not require a bound on its bandwidth. Apart from allowing a stronger runtime result, we believe that this constitutes a major improvement towards the reusability of the presented technique in future analyses of the **SGA**.

Roughly speaking, in our previous work we measured the number of one-bits that individuals of the population have at a given position i in the bitstring at time t with a random variable X_t^i and showed that, if the population size is not too large, X_t^i has a very similar behavior to that of a martingale (i.e., the expected value of the random variable remains the same from one step to the next; see [25] for an introduction to martingales). This enabled us to define a potential function $Y_t^i := (X_t^i - \mu/2)^2$ exhibiting a positive drift (i.e., the bits converge). However, for larger population sizes the positive drift does not necessarily hold since the X_t^i -process starts to resemble a submartingale (i.e., the expected value of the random variable can increase from one step to the next) and the Y_t^i -process might drift towards 0 even if the X_t^i -process increases. The proof strategy presented herein defines a different potential function Y_t^i such that a positive drift can be proved even if the underlying X_t^i -process closely resembles a submartingale. This allows the proof of exponential runtime up to population sizes $\mu \leq n^{1/4-\varepsilon}$. From the analysis an intuition can be derived that for larger populations (i.e., $\mu = \Omega(\sqrt{n})$) the bit positions no longer converge sufficiently disabling the effectiveness of the mathematical techniques presented herein and in our previous work [20]. To this end, in aid of future research, we present some experiments showing how the diversity increases rapidly when the population size reaches values around $\mu = c\sqrt{n}$ for various constants $c > 0$.

The final improvement compared to our previous work [20] is a slight variation in the algorithm. We change the selection operator to select individuals from the population *with replacement* rather than *without replacement*. We feel that the chosen selection operator is the more natural variant, hence redefined the algorithm. In any case, since the probability of choosing an individual for selection twice is $O(1/\mu)$, this does not really affect the analysis. The results in the paper would also hold for the variant in [20], as could that variant also be used in this paper.

The rest of the paper is structured as follows. In Section 2 we discuss previous related work, define the **SGA** precisely and outline the new proof strategy in greater detail. In Section 3.1 we discuss the submartingale property of the random variable X_t^i and present a lower bound on the drift of the new potential function Y_t^i . Using the drift we derive an upper bound on the time for many bits to “almost converge” (i.e., to achieve low diversity) in Section 3.2. Finally in Section 3.3, we can apply the machinery from [20] to prove exponential time for the **SGA** with population sizes up to $\mu \leq n^{1/4-\varepsilon}$. In the Conclusions we present the experiments focusing on understanding at what population size the diversity starts increasing rapidly together with final remarks.

2. Algorithm and proof strategy

Our results are a continuation of previous work. [8] performed the first runtime analysis of fitness proportional selection (f.p.s.) by considering only one individual and bitwise mutation. This work was extended by [16] to consider arbitrary population sizes again on a mutation-only EA. In particular, it was proved that the runtime of an EA using f.p.s. and bitwise mutation for ONEMAX is exponential with overwhelming probability (w.o.p.) whatever the polynomial population size. Also if the population is not too large (i.e., logarithmic in the problem size), then the algorithm cannot optimize any function with unique optimum in polynomial time w.o.p. Finally, in [20] we presented the first analysis of the complete **SGA** for ONEMAX using selection without replacement and proving exponential runtime for population sizes up to $\mu \leq n^{1/8-\varepsilon}$ for some arbitrary small constant ε and problem size n . The well-known **SGA** with the more natural selection with replacement is displayed in Fig. 1.

The algorithm is initialized with a parent population P consisting of μ individuals chosen uniformly at random. In each generation a new population of size μ is created. Each individual for the new population is created by following three

Download English Version:

<https://daneshyari.com/en/article/6876007>

Download Persian Version:

<https://daneshyari.com/article/6876007>

[Daneshyari.com](https://daneshyari.com)