Journal of Symbolic Computation 77 (2016) 78–126

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Continuous amortization and extensions: With applications to bisection-based root isolation

Michael A. Burr [1]

Department of Mathematical Sciences, Clemson University, Clemson, SC 29634, United States

ARTICLE INFO

Article history:
Received 10 April 2015
Accepted 29 November 2015
Available online 28 January 2016

MSC:
68W40
68W30
14Q20

Keywords:
Continuous amortization
Subdivision algorithms
Bisection algorithms
Recursion tree
Bit-complexity

ABSTRACT

Subdivision-based algorithms recursively subdivide an input region until the smaller subregions can be processed. It is a challenge to analyze the complexity of such algorithms because the work they perform is not uniform across the input region. Continuous amortization was introduced in Burr et al. (2009) as a way to bound the complexity of subdivision-based algorithms. The main features of this new technique are that (1) the technique can be applied, uniformly, to a variety of subdivision-based algorithms, (2) the technique considers a function directly related to the subdivision-based algorithm under consideration, and (3) the output of the technique is often explicitly expressed in terms of the intrinsic complexity of the problem instance.

In this paper, the theory of continuous amortization is generalized and applied in several directions. The theory is generalized (1) to allow the domain to be higher dimensional or an abstract measure space, (2) to allow more general subdivisions than bisections, and (3) to bound the value of general functions on the regions of the final partition. The theory is applied to seven examples of subdivision-based algorithms. These applications include (1) bounding the number of subdivisions performed by algorithms for isolating real and complex roots of polynomials, (2) bounding the bit-complexity of subdivision-based algorithms for isolating the real roots of polynomials, and (3) bounding the expected runtime of an algorithm for approximating a biased coin. In each of

E-mail address: burr2@clemson.edu.

[1] This work was partially supported by a grant from the Simons Foundation (#282399 to Michael Burr) and NSF Grant CCF-1527193.

http://dx.doi.org/10.1016/j.jsc.2016.01.007

these applications, by using continuous amortization, we achieve or improve the best-known complexity bounds.

1. Introduction

Subdivision algorithms act on regions in continuous domains by recursively subdividing an input region until it has been broken into small subregions which can be processed. Subdivision algorithms can be thought of as natural generalizations, to continuous domains, of divide-and-conquer algorithms from discrete domains. Continuous amortization (Burr et al., 2009) is a technique for computing the complexity of subdivision algorithms. *The main contribution of continuous amortization is that it provides a way to study the subdivision algorithm itself, instead of the algorithm's subdivision tree. This technique induces thinking about subdivision algorithms directly and can be applied, uniformly, to many subdivision algorithms.* Continuous amortization provides a formula for bounding the complexity of subdivision algorithms in much the same way that the master theorem (Cormen et al., 2001) and the Akra–Bazzi theorem (Akra and Bazzi, 1998) provide formulae for discrete divide-and-conquer algorithms.

The current paper is divided into two parts. In the first part (Section 3), the theory of continuous amortization is generalized in three ways. In particular, continuous amortization is generalized (1) by allowing more general underlying spaces (including higher dimensional real spaces and arbitrary measure spaces), (2) by allowing the subdivisions to come from more general families that just bisection, and (3) by computing upper bounds on function evaluation (including bit complexity and expected time complexity) on the leaves of the subdivision tree. These extensions lay the theoretical work for the application of continuous amortization to a large variety of algorithms, while retaining the simplicity and straight-forward formula of the original formulation.

In the second part of the paper (Sections 5, 6, 7, and 8), we provide a collection of 7 examples of applications of continuous amortization. These examples are mostly bisection-based root isolation algorithms, and *continuous amortization allows us to bound the complexity of these algorithms in terms of the intrinsic geometry of the problem instance.* As a corollary, we show bounds for the benchmark problem of isolating all the roots of a polynomial that, in the worst case, either match or improve upon the state-of-the-art complexity bounds. These examples show how one technique can be applied, uniformly, to bound the complexity of different algorithms. In particular, the technique is applied to (1) bound the size of the subdivision tree-size for root isolation algorithms in one real or complex dimension (in particular, we improve the worst-case bound on the complexity of the size of the subdivision tree of `CEVAL` (Sagraloff and Yap, 2011a, 2011b) by a factor of $(\ln n)^2$ in most cases for the benchmark problem), (2) bound the bit-complexity of root isolation algorithms in one real dimension (in particular, we improve the worst-case bound on the bit-complexity of the `SqFreeEVAL` algorithm (Burr and Krahmer, 2012) by a factor of $(\ln L + \ln n)^2$ for the benchmark problem), and (3) bound the expected time of an algorithm for approximating a biased coin with an unbiased coin. These examples are meant to be case studies for the continuous amortization technique, illustrating how to apply and how to use the technique. This collection of examples is designed to be a guide for other researchers who will find the continuous amortization technique useful.

In Section 2, we discuss a simple bisection-based algorithm for one-dimensional domains and recall the technique of continuous amortization. In Section 3, we present the main theoretical contribution of this paper, where we generalize the theory of continuous amortization. In Section 4, we collect known results on root isolation which are used in the remainder of the paper. In Sections 5, 6, 7, and 8, we provide a collection of new examples illustrating applications of continuous amortization. In Section 9, we conclude with some possible future work in the study and application of continuous amortization.