TABLE OF CONTENTS

1 Introduction

In recent years, there has been rapid advances in the algorithms for combinatorial problems. After several decades of no progress, Björklund et al. (2017) managed to design an algorithm that solves the Hamiltonian problem, i.e., finding whether a given graph contains a path which contains all vertices, significantly faster than the previously fastest algorithms. This development resulted in faster algorithms for several other problems. All this progress has been due to the development in algebraic methods for solving combinatorial problems, namely the development in algebraic fingerprinting, which will be the focus of this paper.

1.1 Preliminaries

[to do: Define here everything that should be defined.] (Combinatorial problem, fields, multilinearity, O*, ...)

1.2 Algebraization of combinatorial problems

Designing an algebraic algorithm for a problem begins by the algebraization of that problem, i.e., transforming the original combinatorial question into a question regarding some algebraic property of some algebraic object built from the elements of the original problem. The recent success has been found by using multivariate polynomials; we construct a multivariate polynomial such that it encodes all combinations of the combinatorial problem, the solutions and non-solutions, as monomials, with the solutions being multilinear. Thus, finding whether a solution exists to the combinatorial problem has been transformed into finding whether the constructed polynomial contains a multilinear monomial.

Koutis and Williams (2015) showcased an example of this in their article with the parameterized problem of k-3D matching. From a set of triples, this problem asks for a k-matching, i.e., a subset of the input with disjoin elements. This combinatorial problem can be transformed into an algebraic form by viewing each element of each input triple as a unique variable, transforming the input triples into multivariate monomials by taking the product of the elements, summing those monomials together, and constructing the final polynomial by raising the sum to the power of k. Thus, the problem of finding a k-matching transforms into a problem of finding a multilinear monomial within the constructed polynomial. A decision problem version of this can be answered by finding whether a multilinear monomial exists.

The detection of linear monomials is a very important issue since a fast solution will make every algorithm, that utilizes this algebraization, faster. However, when the problem domain is large, with n variables in an N-degree polynomial, the number of possible monomials is $\binom{n+N}{n}$. This motivates the detection of linear monomials without fully expanding the polynomial into a sum of monomials, which will be the topic of the next section.

2 Detecting multilinear monomials

To tackle the issue with polynomials of large degree and dimension, we focus on an important observation: only the multilinear monomials in the polynomial are important, and everything else can be discarded. This implies that while expanding the polynomial, i.e., multiplying the terms with each other according to the degree of the polynomial and forming the eventual sum of products – the expanded polynomial, a monomial can be discarded as soon as one its variables is squared.

2.1 Dynamically expanding the polynomial

2.2 Color coding for faster evaluation

REFERENCES

Andreas Björklund, Thore Husfeldt, Petteri Kaski, Mikko Koivisto. *Narrow sieves for parameterized paths and packings*. Journal of Computer and System Sciences. p. 119-139. 2017

Ioannis Koutis, Ryan Williams. *Algebraic Fingerprints for faster algorithms*. Annual ACM-SIAM Symposium on Discrete Algorithms. CA, USA. p. 1671-1680. 2015