Journal of Computer and System Sciences 95 (2018) 69–85

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Designing deterministic polynomial-space algorithms by color-coding multivariate polynomials [☆]

Gregory Gutin [a,*], Felix Reidl [a], Magnus Wahlström [a], Meirav Zehavi [b]

[a] *Royal Holloway, University of London, TW20 0EX, UK*
[b] *University of Bergen, Norway*

ARTICLE INFO

Article history:
Received 12 June 2017
Received in revised form 19 December 2017
Accepted 30 January 2018

Keywords:
Polynomial space
Fixed-parameter tractable
Deterministic
Kirchoff matrices
Pfaffians

ABSTRACT

We introduce an enhancement of color coding to design deterministic polynomial-space parameterized algorithms. Our approach aims at reducing the number of random choices by exploiting the special structure of a solution. Using our approach, we derive polynomial-space $O^*(3.86^k)$-time (exponential-space $O^*(3.41^k)$-time) deterministic algorithm for k-Internal Out-Branching, improving upon the previously fastest exponential-space $O^*(5.14^k)$-time algorithm for this problem. (The notation O^* hides polynomial factors.) We also design polynomial-space $O^*((2e)^{k+o(k)})$-time (exponential-space $O^*(4.32^k)$-time) deterministic algorithms for k-Colorful Out-Branching on arc-colored digraphs and k-Colorful Perfect Matching on planar edge-colored graphs. In k-Colorful Out-Branching, given an arc-colored digraph D, decide whether D has an out-branching with arcs of at least k colors. k-Colorful Perfect Matching is defined similarly. To obtain our polynomial-space algorithms, we show that $(n, k, \alpha k)$-splitters ($\alpha \geqslant 1$) and in particular (n, k)-perfect hash families can be enumerated one by one with polynomial delay using polynomial space.

1. Introduction

In this paper, we modify color coding to treat multivariate polynomials, and thus design an improved deterministic polynomial-space algorithm for k-Internal Out-Branching (k-IOB). Before we elaborate on this problem and our contribution, let us first review related previous works that motivate our study. In recent years, several powerful algebraic techniques have been developed to design *randomized* polynomial-space parameterized algorithms. The first approach was introduced by Koutis [1], strengthened by Williams [2], and is nowadays known as the multilinearity detection technique [3]. Roughly speaking, an application of this technique consists of reducing the problem at hand to one where the objective is to decide whether a given polynomial has a multilinear monomial, and then employing an algorithm for the latter problem as a black box.

One of the huge breakthroughs brought about by this line of research was Björklund's [4] proof that Hamiltonian Path is solvable in time $\mathcal{O}^*(1.66^n)$ by a randomized algorithm, improving upon the 50 year old $\mathcal{O}^*(2^n)$-time[1] algorithm [5]. The ex-

☆ Gutin was partially supported by Royal Society Wolfson Research Merit Award, Reidl and Wahlström by EPSRC grant EP/P007228/1, and Zehavi by ERC Grant Agreement no. 306992.

* Corresponding author.
E-mail address: g.gutin@rhul.ac.uk (G. Gutin).

[1] We use the common notation $\mathcal{O}^*$ to hide factors polynomial in the input size.

https://doi.org/10.1016/j.jcss.2018.01.004

istence of a *deterministic* $\mathcal{O}^*((2-\varepsilon)^n)$-time algorithm for HAMILTONIAN PATH, for a fixed $\varepsilon > 0$, is still a major open problem. Further, Björklund's result is on undirected graphs, and the existence of an $\mathcal{O}^*((2-\varepsilon)^n)$-time algorithm for HAMILTONIAN PATH on digraphs, for a fixed $\varepsilon > 0$, is another interesting open problem.

Shortly afterwards, Björklund et al. [6] have transformed the ideas in [4] into a powerful technique to design randomized polynomial-space algorithms, referred to as *narrow sieves*. This technique is also based on the analysis of polynomials, but it is applied quite differently. Here one associates a monomial with each "potential solution" in such a way that actual solutions correspond to unique monomials while incorrect solutions appear in pairs. Thus, the polynomial summing these monomials, when evaluated over a field of characteristic 2, is not identically 0 if and only if the input instance of the problem at hand is a yes-instance. In this context, the relevance of the Matrix Tree Theorem was already noted by Gabizon et al. [7].

The narrow sieves technique, proven to be of wide applicability on its own, later branched into several new methods. The one most relevant to our study was developed by Björklund et al. [8] and was translated into the language of determinants by Wahlström [9]. Here, the studied problem was S-CYCLE (or S-PATH), where the goal is to determine whether an input graph contains a cycle that passes through all the vertices of an input set S of size k. Wahlström [9] considered a determinant-based polynomial (computed over a field of characteristic 2), and analyzed whether there exists a monomial where the variable-set representing S is present. Very recently, Björklund et al. [10] utilized the Matrix Tree Theorem to improve an FPT algorithm for k-IOB, where we are asked to decide whether a given digraph has a k-internal out-branching. Recall that an *out-tree* T is an orientation of a tree with only one vertex of in-degree zero (called the *root*). A vertex of T is a *leaf* if its out-degree in T is zero; non-leaves are called *internal vertices*. An *out-branching* of a digraph D is a spanning subgraph of D, which is an out-tree, and an out-branching is *k-internal* if it has at least k internal vertices.

Björklund et al. [10] cleverly transformed k-IOB into a new problem, where the goal is to decide whether a given polynomial (computed over a field of characteristic 2 to avoid subtractions) has a monomial with at least k distinct variables.

In this paper, we present an easy-to-use[2] modification of color coding for designing deterministic polynomial-space parameterized algorithms, inspired by the principles underlying the above mentioned techniques. (A slight modification of our approach can be used to design faster, exponential-space algorithms, but we believe that the main value of the approach is for polynomial-space algorithms.) We will show that our approach brings significant speed-ups to algorithms for k-IOB. Roughly speaking, our approach can be applied as follows.

- Identify a polynomial such that it has a monomial with at least k distinct variables (called a *witnessing monomial*) if and only if the input instance of the problem at hand is a yes-instance. It should be possible to efficiently evaluate the polynomial (black box-access is sufficient here).
- Color the variables of the polynomial with k colors using a polynomial-delay perfect hash-family. To improve the running time of this step, we apply a problem-specific *coloring guide* to reduce the number of 'random' colors. Given a k-coloring, we obtain a smaller polynomial by identifying all variables of the same color.
- Use inclusion-exclusion to extract the coefficient of a colorful monomial from the reduced polynomial. By the usual color-coding arguments, if the coefficient is not equal to zero then the original polynomial contained a witnessing monomial.

While we were unable to obtain non-trivial coloring guides to the following problems, even limited application of our approach is useful for designing polynomial-space algorithms for these problems. It would be interesting to obtain non-trivial coloring guides for the problems.

Colorful Out-Branchings and Matchings Every subgraph-search problem can be extended quite naturally by imposing additional constraints on the solution, for example by letting the input graph have labels or weights. One class of such constraints states that a required subgraph of an edge-colored graph has to be *k-colorful*, i.e. to contain edges of at least k colors.

One prominent problem is RAINBOW MATCHING[3] (also known as MULTIPLE CHOICE MATCHING), defined in the classical book by Garey and Johnson [11]. Itai et al. [12] showed, already in 1978, that RAINBOW MATCHING is NP-complete on bipartite graphs. Three decades later, Le and Pfender [13] revisited this problem and showed that it is NP-hard on several restricted graph classes, which include (among others) paths, complete graph and P_4-free bipartite graphs in which every color is used at most twice. Further examples of subgraph problems with color constraints can be found in a survey by Mikio and Xueliang [14]. In this paper, we focus on two color-constrained problems: given an edge-colored graph and an integer k, we ask for either a k-colorful spanning tree/outbranching or a k-colorful perfect matching.

[2] In particular, no dynamic programming/recursive algorithms are required.
[3] In the problem, given an edge-colored graph G and an integer k, the aim is to decide whether G has a k-colorful matching of size k.