Mixing Color Coding-Related Techniques

Meirav Zehavi

Department of Computer Science, Technion IIT, Haifa 32000, Israel
meizeh@cs.technion.ac.il

Abstract. Narrow sieves, representative sets and divide-and-color are three breakthrough techniques related to color coding, which led to the design of extremely fast parameterized algorithms. We present a novel family of strategies for applying mixtures of them. This includes: (a) a mix of representative sets and narrow sieves; (b) a faster computation of representative sets under certain separateness conditions, mixed with divide-and-color and a new technique, called “balanced cutting”; (c) two mixtures of representative sets and a new technique, called “unbalanced cutting”. We demonstrate our strategies by obtaining, among other results, significantly faster algorithms for $k$-INTERNAL OUT-BRANCHING and WEIGHTED 3-SET $k$-PACKING, and a general framework for speeding-up the previous best deterministic algorithms for $k$-PATH, $k$-TREE, $r$-DIMENSIONAL $k$-MATCHING, GRAPH MOTIF and PARTIAL COVER.

1 Introduction

A problem is fixed-parameter tractable (FPT) with respect to a parameter $k$ if it can be solved in time $O^*(f(k))$ for some function $f$, where $O^*$ hides factors polynomial in the input size. The color coding technique, introduced by Alon et al. [1], led to the discovery of the first single exponential time FPT algorithms for many subcases of SUBGRAPH ISOMORPHISM. In the past decade, three breakthrough techniques improved upon it, and led to the development of extremely fast FPT algorithms for many fundamental problems. This includes the combinatorial divide-and-color technique [7], the algebraic multilinear detection technique [20,21,35] (which was later improved to the more powerful narrow sieves technique [2,3]), and the combinatorial representative sets technique [16].

Divide-and-color was the first technique that resulted in (both randomized and deterministic) FPT algorithms for weighted problems that are faster than those relying on color coding. Later, representative sets led to the design of deterministic FPT algorithms for weighted problems that are faster than the randomized ones based on divide-and-color. The fastest FPT algorithms, however, rely on narrow sieves. Unfortunately, narrow sieves is only known to be relevant to the design of randomized algorithms for unweighted problems[1].

We present novel strategies for applying these techniques, combining the following elements (see Section 3).

1 More precisely, when used to solve weighted problems, the running times of the resulting algorithms have exponential dependencies on the length of the input weights.
- Mixing narrow sieves and representative sets, previously considered to be two independent color coding-related techniques.
- Under certain “separateness conditions”, speeding-up the best known computation of representative sets.
- Mixing divide-and-color-based preprocessing with the computation in the previous item, speeding-up standard representative sets-based algorithms.
- Cutting the universe into small pieces in two special manners, one used in the mix in the previous item, and the other mixed with a non-standard representative sets-based algorithm to improve its running time (by decreasing the size of the partial solutions it computes).

To demonstrate our strategies, we consider the following well-studied problems.

**k-Internal Out-Branching (k-IOB):** Given a directed graph \( G = (V, E) \) and a parameter \( k \in \mathbb{N} \), decide if \( G \) has an out-branching (i.e., a spanning tree with exactly one node of in-degree 0) with at least \( k \) internal nodes.

**Weighted k-Path:** Given a directed graph \( G = (V, E) \), a weight function \( w : E \to \mathbb{R} \), \( W \in \mathbb{R} \) and a parameter \( k \in \mathbb{N} \), decide if \( G \) has a simple directed path on exactly \( k \) nodes and of weight at most \( W \).

**Weighted 3-Set k-Packing ((3, k)-WSP):** Given a universe \( U \), a family \( S \) of subsets of size 3 of \( U \), a weight function \( w : S \to \mathbb{R} \), \( W \in \mathbb{R} \) and a parameter \( k \in \mathbb{N} \), decide if there is a family \( S' \subseteq S \) of \( k \) disjoint sets and weight at least \( W \).

The k-IOB problem is NP-hard since it generalizes HAMILTONIAN PATH. It is of interest, for example, in database systems [10], and for connecting cities with water pipes [31]. Many FPT algorithms were developed for k-IOB and related variants (see, e.g., [8,9,13,14,18,22,30,32,36]). We solve it in deterministic time \( O^*(5.139^k) \) and randomized time \( O^*(3.617^k) \), improving upon the previous best deterministic time \( O^*(6.855^k) \) [32] and randomized time \( O^*(4^k) \) [936]. To this end, we establish a relation between certain directed trees and paths on 2 nodes. This shows how certain partial solutions to k-IOB can be completed efficiently via a computation of a maximum matching in the underlying undirected graph.

We also present a unified approach for speeding-up standard representative sets-based algorithms. It can be used to modify the previous best deterministic algorithms (that already rely on the best known computation of representative sets) for k-PATH, k-TREE, r-DIMENSIONAL k-MATCHING ((r, k)-DM), GRAPH MOTIF WITH DELETIONS (GM_D) and PARTIAL COVER (PC), including their weighted variants, which run in times \( O^*(2.6181^k) \) [1532], \( O^*(2.6181^k) \) [1532], \( O^*(2.6181^{r-1}^k) \) [17], \( O^*(2.6181^{2k}) \) [29] and \( O^*(2.6181^k) \) [32], to run in times \( O^*(2.5961^k) \), \( O^*(2.5961^k) \), \( O^*(2.5961^{r-1}^k) \), \( O^*(2.5961^{2k}) \) and \( O^*(2.5961^k) \), respectively. To demonstrate our approach, we use WEIGHTED k-PATH.

In the past decade, (3, k)-WSP and (3, k)-SP enjoyed a race towards obtaining the fastest FPT algorithms (see [31,34,11,12,19,20,23,24,33,34,37]). We solve (3, k)-WSP in deterministic time \( O^*(8.097^k) \), improving upon \( O^*(12.155^k) \), which is both the previous best running time of an algorithm for (3, k)-WSP and the previous best running time of a deterministic algorithm for (3, k)-SP [37]. The full version [38] also solves \( P_2 \)-PACKING, a special case of (3, k)-WSP.