Theoretical Computer Science 562 (2015) 330–345

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Combinatorics for smaller kernels: The differential of a graph

Sergio Bermudo [a,1], Henning Fernau [b,*]

[a] Department of Economy, Quantitative Methods and Economic History, Pablo de Olavide University, Carretera de Utrera Km. 1, 41013 Sevilla, Spain

[b] Fachbereich 4—Abteilung Informatikwissenschaften, Universität Trier, D-54286 Trier, Germany

ARTICLE INFO

Article history:
Received 15 September 2012
Received in revised form 16 July 2014
Accepted 1 October 2014
Available online 8 October 2014
Communicated by G. Ausiello

Keywords:
Parameterized algorithm
Kernelization
Approximation algorithm
Combinatorial bounds
Differential of a graph

ABSTRACT

Let $G=(V,E)$ be a graph of order n and let $B(D)$ be the set of vertices in $V\setminus D$ that have a neighbor in the vertex set D. The *differential of* D is defined as $\partial(D)=|B(D)|-|D|$ and the *differential of a graph* G, written $\partial(G)$, is equal to $\max\{\partial(D): D\subseteq V\}$. If G is connected and $n\geq 3$, $\partial(G)\geq n/5$ is known. This immediately leads to a linear vertex kernel result (in the terminology of parameterized complexity) for the problem of deciding whether $\partial(G)\geq k$, taking k as the parameter. We then establish a new combinatorial result which establishes that $\partial(G)\geq n/4$ if G is a connected graph of order $n\geq 6$ and if G contains no induced path of five vertices whose midpoint is a cut vertex and whose endpoints have degree one. This technical combinatorial theorem can be used to derive an even smaller linear vertex kernel for general graphs. Also, we show that the related maximization problem allows for a polynomial-time factor-$\frac{1}{4}$ approximation algorithm.

1. Introduction

The development of kernelization algorithms is one of the driving forces in the theory of fixed-parameter tractability. In a nutshell, the idea of kernelization can be seen as a means of providing a mathematical (combinatorial) analysis of preprocessing.

Sometimes, it is possible to interpret known combinatorial bounds on graph parameters as also providing kernel bounds. This in itself is not a new idea. As we will see, analyzing the known worst-case examples can lead to new preprocessing rules that help improve those kernel bounds. We exemplify this strategy by proving kernel bounds that are better than the ones that could be immediately produced by employing known combinatorial bounds for the problem of finding a differential set of size at least k. More precisely, we derive a kernel of order at most $4k$. We also show that our kernel bounds are *optimal* with respect to our reduction rules. By this statement, we mean that there are infinite families of irreducible graphs that show that no better kernel bound claims are possible based on the proposed reductions. Finally, we explain how our combinatorial proofs also yield a polynomial-time algorithm for approximating the differential of a graph up to a factor of $\frac{1}{4}$.

* Corresponding author.
E-mail addresses: sbernav@upo.es (S. Bermudo), fernau@uni-trier.de (H. Fernau).

[1] Partially supported by Ministerio de Ciencia y Tecnología (MTM 2012-35107) and by the Junta de Andalucía (FQM-260).

http://dx.doi.org/10.1016/j.tcs.2014.10.007
0304-3975/

2. Definitions from graph theory and parameterized algorithms

We will use standard notions from graph theory throughout this paper. For instance, $N(v)$ denotes the set of vertices that are neighbors of the vertex v and we write $v \sim u$ when the vertices v and u are adjacent. If U is some set of vertices from $G = (V, E)$, then $G[U]$ denotes the graph induced by U, and $G - U = G[V \setminus U]$. For some vertex v, we also write $G - v$ instead of $G - \{v\}$. We use $n(G)$ to denote the *order* of graph $G = (V, E)$, i.e., $n(G) = |V|$. Following [17], if D is a vertex set, then $N(D) = \bigcup_{x \in D} N(x)$ is the neighborhood of D, while $B(D) = N(D) \setminus D$ is the *boundary* of D, collecting only the *proper neighbors* of vertices from D, and $C(D) = V \setminus (D \cup B(D))$.

The *differential of* D is defined as $\partial(D) = |B(D)| - |D|$ and the *differential of a graph* G, written $\partial(G)$, is equal to $\max\{\partial(D) : D \subseteq V\}$. A set D satisfying $\partial(D) = \partial(G)$ is also called a *differential set*, or ∂*-set* for short. The graph parameter ∂ was introduced in [17], where several basic properties were derived. As explained in [2,17], the differential of a graph can be seen as a simplified deterministic model of influencing a network representing a social group, aiming at maximizing the economic or political benefit of those who want to influence the network. Several paper have appeared on the differential of a graph, most of purely combinatorial nature. We only refer to [2,5] and the literature quoted therein. More recently, it has become clear that the sum of the differential and the Roman domination number [9,13] of a graph equals its order [4]; this fact also offers new motivations to study computational aspects of the differential of a graph, as we do in this paper.

In this paper, we are interested in the following related decision problem DIF: Given a graph G and an integer k, decide whether $\partial(G) \geq k$. We will consider this question from the viewpoint of parameterized complexity; see [11]. More precisely, we will discuss problem kernel results with respect to this parameter k. Due to the connections of problem kernels and approximation, it should be mentioned that the intractability and approximability of the decision respectively maximization problems have been first discussed in [3], leading, for instance, to MAX SNP completeness results for the Maximum Differential problem on graphs of bounded degree (and also NP-completeness results for DIF, although these can be also obtained by the mentioned relation to Roman domination).

There is a subtle difference between minimization and maximization problems when it comes to interpreting linear kernel results as approximation results. For instance, as shown by Hochbaum in [14], the famous Nemhauser–Trotter theorem [19] that basically provides a linear kernel for Vertex Cover (as later observed in the parameterized complexity community) can be also interpreted as a factor-2 approximation for Minimum Vertex Cover, namely by considering all vertices from the kernel (plus the ones added to the cover by the reduction rules), and similar observations are possible for many other minimization problems, as long as the whole kernel provides a trivial solution. However, this interpretation is no longer possible for maximization problems. As indicated, we only got constant-factor approximability results for Maximum Differential for graphs of bounded degree, and it was yet unclear what kind of results are to be expected in the case of general graphs. The main thrust in the proofs of the kernel results is extremal combinatorics, which in itself does not immediately provide algorithmic results. In our case, we will however be able to read the proofs also as an approximation algorithm. Hence, we can conclude that, in general graphs, the maximum differential of a graph can be approximated up to a factor of $\frac{1}{4}$. Our earlier results show that better approximations are possible for bounded-degree graphs; more specifically, in (sub-)cubic graphs, there is a polynomial-time approximation algorithm giving a factor of $\frac{35}{134}$, and for graphs of degree at most four, we have an approximation algorithm with a factor of $\frac{35}{201}$. So, while the specific algorithm is still a bit better for subcubic graphs, the new general algorithm is better for other degree bounds. However, we will also show how to further improve on the approximation ratios for degree-bounded graphs. This will lead us, for instance, to a factor-$\frac{1}{2}$ approximation on subcubic graphs.

We are mostly interested in the notion of a *(problem) kernel*, as kernelization is known to be equivalent to fixed-parameter tractability. In our case, we are looking for a function $g : \mathbb{N} \to \mathbb{N}$ and a polynomial-time algorithm that transforms an instance (G, k) of DIF into an equivalent instance (G', k') such that the size of G' is bounded above by $g(k)$ and $k' \leq k$. Such an algorithm is also known as a *kernelization (algorithm)*. Notice that we will also allow kernelizations to directly answer YES or NO, which would formally correspond to producing trivial YES- or NO-instances of DIF, respectively. We will first show a kernelization providing a kernel (G', k') where the order of G' is bounded by $5k'$, which is based on known combinatorial bounds. Then, using more sophisticated kernelization rules and some new combinatorial insights into the differential of a graph, we improve this bound to $4k'$. This also comprises the main algorithmic result of this paper. The main technical contributions (Theorems 3.4, 6.1 and Proposition 6.3) are however of a purely combinatorial nature. This also means that the basic structure of our kernelization algorithm is very simple: apply some reduction rules to exhaustion and then either reply YES or conclude that the order of the graph is upper-bounded by four times its differential. This result is complemented by showing an infinite family of irreducible graphs that prove that the analysis of the kernel size of our kernelization algorithm is tight with respect to the proposed reduction rules. Finally, we also give FPT algorithms for our problem, more precisely, we show that it can be decided in time $\mathcal{O}^*(c^k)$ if a given graph G has a differential set of size at least k, for some constant $c \approx 9$. Here, the $\mathcal{O}^*$-notation suppresses polynomial factors. Hence, an $\mathcal{O}^*(c^k)$ algorithm for DIF has running time $\mathcal{O}(c^k \cdot p(n(G)))$ for some polynomial p on input G and k.

Recall that a graph consisting of one central vertex c and d neighbors that in turn have no further neighbors other than c is also known as a *star* $S_d = K_{1,d}$. If $S = (V, E)$ is an S_d star with center c, then $V \setminus \{c\}$ will be also called *ray vertices* and the edges will be termed *rays*. We also denote an S_d star S by $S = \{c; v_1, \ldots, v_d\}$ to indicate that c is its center and $v_1, \ldots, v_d$ are its ray vertices. We will call an S_d star *big* if $d \geq 2$.