Discrete Applied Mathematics 161 (2013) 1154–1161

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

Towards optimal kernel for connected vertex cover in planar graphs

Łukasz Kowalik [a], Marcin Pilipczuk [a,*], Karol Suchan [b,c]

[a] Institute of Informatics, University of Warsaw, Poland
[b] Universidad Adolfo Ibáñez, Chile
[c] WMS, AGH - University of Science and Technology, Poland

ARTICLE INFO

Article history:
Received 16 January 2012
Received in revised form 25 August 2012
Accepted 3 December 2012
Available online 20 December 2012

Keywords:
Kernelization
Planar graphs
Connected vertex cover

ABSTRACT

We study the parameterized complexity of the connected version of the vertex cover problem, where the solution set has to induce a connected subgraph. Although this problem does not admit a polynomial kernel for general graphs (unless NP $\subseteq$ coNP/*poly*), for planar graphs Guo and Niedermeier [ICALP'08] showed a kernel with at most $14k$ vertices, subsequently improved by Wang et al. [MFCS'11] to $4k$. The constant 4 here is so small that a natural question arises: could it be already an optimal value for this problem? In this paper we answer this question in the negative: we show a $\frac{11}{3}k$-vertex kernel for Connected Vertex Cover in planar graphs. We believe that this result will motivate further study in the search for an optimal kernel.

In our analysis, we show an extension of a theorem of Nishizeki and Baybars [Takao Nishizeki, Ilker Baybars, Lower bounds on the cardinality of the maximum matchings of planar graphs, Discrete Mathematics 28 (3) (1979) 255–267] which might be of independent interest: every planar graph with $n_{\geq 3}$ vertices of degree at least 3 contains a matching of cardinality at least $n_{\geq 3}/3$.

1. Introduction

Many NP-complete problems, while most likely they cannot be fully solved efficiently, admit kernelization algorithms, i.e. efficient algorithms which replace input instances with an equivalent, but often much smaller one. More precisely, a *kernelization algorithm* takes an instance I of size n and a parameter $k \in \mathbb{N}$, and after a time polynomial in n it outputs an instance I' (called a *kernel*) with a parameter k' such that I is a yes-instance iff I' is a yes-instance, $k' \leq k$, and $|I'| \leq f(k)$ for some function f depending only on k. The most desired case is when the function f is polynomial, or even linear (then we say that the problem admits a polynomial or linear kernel). In such a case, when the parameter k is relatively small, the input instance, possibly very large, is "reduced" to a small one. Intuitively, kernelization aims at finding the core difficulty in the input instance. The output instance can then be processed in many ways, including approximation algorithms or heuristics. For small values of k and small kernels, one can often even use an exact (exponential-time) algorithm.

A typical example of the above phenomenon is the well-known Vertex Cover problem, which admits a kernel with at most $2k$ vertices [4], where k is the size of an optimum vertex cover in the input instance. However, for some problems, reducing to a linear number of vertices seems a hard task, e.g. for the Feedback Vertex Set problem, the best known result is the $4k^2$-vertex kernel of Thomassé [16]. Even worse, there are many natural problems (examples include Dominating Set or Steiner Tree) for which it is proved that they do not admit a polynomial kernel, unless some widely believed complexity

* Corresponding author. Fax: +48 22 55 44 400.
E-mail addresses: kowalik@mimuw.edu.pl (Ł. Kowalik), malcin@mimuw.edu.pl (M. Pilipczuk), karol.suchan@uai.cl (K. Suchan).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2012.12.001

hypothesis fails (*FPT* $\neq W[2]$ in the first case and NP $\subsetneq$ coNP/*poly* in the second one). This motivates investigation of polynomial kernels in natural restrictions of graph classes. Note that it is of particular interest to guarantee the output instance belongs to the same class as the input instance.

A classic example is the $335k$-vertex kernel for the Dominating Set problem in planar graphs due to Alber et al. [1] (note that in planar graphs the number of edges is linear in the number of vertices, thus kernels with a linear number of vertices are in fact linear kernels). Later, this work was substantially generalized (see e.g. [9]), and researchers obtained a number of linear kernels for planar graphs. Let us mention the $112k$-vertex kernel for Feedback Vertex Set by Bodlaender and Penninkx [2], the linear kernel for Induced Matching by Moser and Sikdar [14] or the $624k$-vertex kernel for the Maximum Triangle Packing by Guo and Niedermeier [11].

Observe that the constants in the linear functions above are crucial: since we deal with NP-complete problems, in order to find an exact solution in the reduced instance, most likely we need exponential time (or at least superpolynomial, because for planar graphs $2^{O(\sqrt{k})}$-time algorithms are often possible), and these constants appear in the exponents. Motivated by this, researchers seek for linear kernels with constants as small as possible. For example, by now there is known a $67k$-vertex kernel for Dominating Set [3], a $28k$-vertex kernel for Induced Matching [8] or a $75k$-vertex kernel for Maximum Triangle Packing [17].

In this work, we study the Connected Vertex Cover problem, a variant of the classical Vertex Cover: we are given a planar graph $G = (V, E)$ and a parameter k, and we ask whether there exists a vertex cover S (i.e. a set $S \subseteq V$ such that every edge of G has an endpoint in S) of size at most k which induces a connected subgraph of G. This problem is NP-complete also in planar graphs [10], and, contrary to its simpler relative, it does not admit a polynomial kernel in arbitrary graphs [6]. However, Guo and Niedermeier [11] showed a $14k$-vertex kernel for planar graphs. Very recently, it was improved to $4k$ by Wang et al. [17]. The constant 4 here is already so small that a natural question arises: could it be the optimal value for this problem? In this paper we answer this question in the negative: we show a $\frac{11}{3}k$-vertex kernel for Connected Vertex Cover in planar graphs.

Let us recall that in the analysis of the $4k$-vertex kernel by Wang et al. [17], the vertices of the graph are partitioned into three parts: vertices of degree one, the solution, and the rest of the graph, and it is proven that, after applying a few reduction rules, the sizes of these parts can be bounded by k, k and $2k$, respectively. We present (in Lemma 3.1) a deeper analysis of these bounds and we show that an instance where all the bounds are close to being tight is somewhat special. This analysis is the main technical contribution of this paper. We believe that this result will motivate further study in the search for an optimal kernel.

As a by-product of our analysis, we show an extension of a theorem of Nishizeki and Baybars [15] which might be of independent interest: every planar graph with $n_{\geq 3}$ vertices of degree at least 3 contains a matching of cardinality at least $n_{\geq 3}/3$.

Organization of the paper. The paper is organized as follows. In Section 2 we present our kernelization algorithm along with a proof of its correctness. In Section 3 we show that our algorithm outputs a kernel with the number of vertices bounded by $\frac{11}{3}k$. Finally, in Section 4 we describe an example which shows that our analysis is tight (and hence improving on the kernel size would require adding new reduction rules to the algorithm).

Terminology and notation. We use standard fixed parameter complexity and graph theory terminology, see e.g. [7,5]. For brevity, we call a vertex of degree d a *d-vertex* and if a vertex v has a d-vertex w as a neighbor we call w a d-neighbor. By $N_G(v)$ we denote the set of neighbors of v. For a set of vertices $X \subseteq V(G)$, we define $N_G(X) = (\bigcup_{x \in X} N_G(x)) \setminus X$. We omit the subscript in $N_G(v)$ and $N_G(X)$ when it is clear from the context. For a graph G and a subset of its vertices S, by $G[S]$ we denote the subgraph of G induced by S.

2. Algorithm

In this section we present our kernelization algorithm. It works in three phases. In what follows, (G_0, k_0) denotes the input instance.

2.1. Phase 1

Phase 1 is a typical kernelization algorithm. A set of rules is specified and in each rule the algorithm searches the graph for a certain configuration. If the configuration is found, the algorithm performs a modification of the graph (sometimes, also of the parameter k), typically decreasing the size of the graph. Each rule has to be *correct*, which means that the new graph is planar and the graph before application of the rule has a connected vertex cover of size at most k if and only if the new graph has a connected vertex cover of size not greater than the new value of k. We apply the rules in order, i.e. Rule i can be applied only if, for every $j < i$, Rule j does not apply. We keep our graph simple, i.e. if after application of a rule a multiple edge appears, we replace it by a single one. The first three rules come from [17].

Rule 1. If a vertex v has more than one 1-neighbors, then remove all these neighbors except for one.

Rule 2. For a 2-vertex v with $N(v) = \{u, w\}$ and $uw \in E(G)$, contract the edge uw and decrease the parameter k by one.