| | |
|---|---|
| Title: | Bidimensionality And Kernels |
| Name: | Daniel Lokshtanov[1] |
| Affil./Addr. | Department of Informatics, University of Bergen, Norway |
| Keywords: | Kernelization; Bidimensionality; Graph Algorithms; Parameterized Complexity; Polynomial Time Pre-processing |
| SumOriWork: | 2010; Fomin, Lokshtanov, Saurabh, Thilikos |

# Bidimensionality And Kernels

Daniel Lokshtanov[1]
Department of Informatics, University of Bergen, Norway

## Years aud Authors of Summarized Original Work

2010; Fomin, Lokshtanov, Saurabh, Thilikos

## Keywords

Kernelization; Bidimensionality; Graph Algorithms; Parameterized Complexity; Polynomial Time Pre-processing

## Problem Definition

The theory of Bidimensionality simultaneously provides subexponential time parameterized algorithms and efficient approximation schemes for a wide range of optimization problems on planar graphs, and more generally, on classes of graphs excluding a fixed graph $H$ as a minor. It turns out that Bidimensionality also provides *linear kernels* for a multitude of problems on these classes of graphs. The results stated here unify and generalize a number of kernelization results for problems on planar graphs and graphs of bounded genus, see [2] for a more thorough discussion.

**Kernelization.** *Kernelization* is a mathematical framework for the study of polynomial time pre-processing of instances of computationally hard problems. Let $\mathcal{G}$ be the set of all graphs. A *parameterized graph problem* is a subset $\Pi$ of $\mathcal{G} \times \mathbb{N}$. An *instance* is a pair $(G, k) \in \mathcal{G} \times \mathbb{N}$. The instance $(G, k)$ is a "yes"-instance of $\Pi$ if $(G, k) \in \Pi$ and a "no"-instance otherwise. A *strict kernel with ck vertices* for a parameterized graph problem $\Pi$ and constant $c > 0$ is an algorithm $\mathcal{A}$ with the following properties.

- $\mathcal{A}$ takes as input an instance $(G, k)$, runs in polynomial time, and outputs another instance $(G', k')$.
- $(G', k')$ is a "yes"-instance of $\Pi$ if and only if $(G, k)$ is.
- $|V(G')| \leq c \cdot k$ and $k' \leq k$.

A *linear kernel* for a parameterized graph problem is a strict kernel with $ck$ vertices for some constant $c$. We remark that our definition of a linear kernel is somewhat simplified compared to the classic definition [8], but that it is essentially equivalent. For a discussion of the definition of a kernel we refer to the textbook of Cygan et al. [4].

**Graph Classes.** Bidimensionality theory primarily concerns itself with graph problems where the input graph is restricted to be in a specific *graph class*. A *graph class* $\mathcal{C}$ is simply a subset of the set $\mathcal{G}$ of all graphs. As an example, the set of all planar graphs is a graph class. Another example of a graph class is the set of all *apex* graphs. Here a graph $H$ is *apex* if $H$ contains a vertex $v$ such that deleting $v$ from $H$ leaves a planar graph. Notice that every planar graph is apex.

A graph $H$ is a *minor* of a graph $G$ if $H$ can be obtained from $G$ by deleting vertices, deleting edges, or contracting edges. Here *contracting* the edge $\{u, v\}$ in $G$ means identifying the vertices $u$ and $v$ and removing all self loops and double edges. If $H$ can be obtained from $G$ just by contracting edges then $H$ is a *contraction* of $G$.

A graph class $\mathcal{C}$ is *minor closed* if every minor of a graph in $\mathcal{C}$ is also in $\mathcal{C}$. A graph class $\mathcal{C}$ is *minor-free* if $\mathcal{C}$ is minor closed and there exists a graph $H \notin \mathcal{C}$. A graph class $\mathcal{C}$ is *apex-minor-free* if $\mathcal{C}$ is minor closed and there exists an apex graph $H \notin \mathcal{C}$. Notice that $H \notin \mathcal{C}$ for a minor closed class $\mathcal{C}$ implies that $H$ can not be a minor of any graph $G \in \mathcal{C}$.

**CMSO Logic.** CMSO logic stands for *Counting Monadic Second Order* logic, a formal language to describe properties of graphs. A *CMSO-sentence* is a formula $\psi$ with variables for single vertices, vertex sets, single edges and edge sets, existential and universal quantifiers ($\exists$ and $\forall$), logical connectives $\vee$, $\wedge$ and $\neg$, as well as the following operators:

- $v \in S$ where $v$ is a vertex variable and $S$ is a vertex set variable. The operator returns true if the vertex $v$ is in the vertex set $S$. Similarly CMSO has an operator $e \in X$ where $e$ is an edge variable and $X$ is an edge set variable.
- $v_1 = v_2$ where $v_1$ and $v_2$ are vertex variables. The operator returns true if $v_1$ and $v_2$ are the same vertex of $G$. There is also an operator $e_1 = e_2$ to check equality of two edge variables $e_1$ and $e_2$.
- $\mathbf{adj}(v_1, v_2)$ is defined for vertex variables $v_1$ and $v_2$ and returns true if $v_1$ and $v_2$ are adjacent in $G$.
- $\mathbf{inc}(v, e)$ is defined for a vertex variable $v$ and edge variable $e$. $\mathbf{inc}(v, e)$ returns true if the edge $e$ is incident to the vertex $v$ in $G$, in other words, if $v$ is one of the two endpoints of $e$.
- $\mathbf{card}_{p,q}(S)$ is defined for every pair of integers $p$, $q$ and vertex or edge set variable $S$. $\mathbf{card}_{p,q}(S)$ returns true if $|S| \equiv q \mod p$. For an example $\mathbf{card}_{2,1}(S)$ returns true if $|S|$ is odd.

When we quantify a variable we need to specify whether it is a vertex variable, edge variable, vertex set variable or edge set variable. To specify that an existentially quantified variable $x$ is a vertex variable we will write $\exists x \in V(G)$. We will use $\forall e \in E(G)$ to universally quantify edge variables and $\exists X \subseteq V(G)$ to existentially quantify vertex set variables. We will always use lower case letters for vertex and edge variables, and upper case letters for vertex set and edge set variables.

A graph $G$ on which the formula $\psi$ is true is said to *model* $\psi$. The notation $G \models \psi$ means that $G$ models $\psi$. As an example, consider the formula

$$\begin{aligned}\psi_1 = \quad & \forall v \in V(G)\ \forall x \in V(G)\ \forall y \in V(G)\ \forall z \in V(G) : \\ & (x = y) \vee (x = z) \vee (y = z) \vee \neg\mathbf{adj}(v, x) \vee \neg\mathbf{adj}(v, y) \vee \neg\mathbf{adj}(v, z)\end{aligned}$$

The formula $\psi_1$ states that for every four (not necessarily distinct) vertices $v$, $x$, $y$, $z$, if $x$, $y$ and $z$ are distinct, then $v$ is not adjacent to all of $\{x, y, z\}$. In other words, a graph $G$ models $\phi_1$ if and only if the degree of every vertex $G$ is at most 2. CMSO can be used to express many graph properties, such as $G$ having a hamiltonian cycle, $G$ being 3-colorable, or $G$ being planar.

In CMSO one can also write formulas where one uses *free variables.* These are variables that are used in the formula but never quantified with an $\exists$ or $\forall$ quantifier. As an example, consider the formula:

$$\psi_{DS} = \forall u \in V(G)\ \exists v \in V(G) : (v \in S) \wedge (u = v \vee \mathbf{adj}(u, v))$$

The variable $S$ is a free variable in $\psi_{DS}$ because it used in the formula, but is never quantified. It does not make sense to ask whether a graph $G$ models $\psi_{DS}$ because when we ask whether the vertex $v$ is in $S$, the set $S$ is not well defined. However, if the set $S \subseteq V(G)$ is provided together with the graph $G$, we can evaluate the formula $\psi_{DS}$. $\psi_{DS}$ will be true for a graph $G$ and set $S \subseteq V(G)$ if, for every vertex $u \in V(G)$, there exists a vertex $v \in V(G)$ such that $v$ is in $S$ and either $u = v$ or $u$ and $v$ are neighbors in $G$ In other words, the pair $(G, S)$ models $\psi_{DS}$ (written $(G, S) \models \psi_{DS}$) if and only if $S$ is a dominating set in $G$ (i.e. every vertex not in $S$ has a neighbor in $S$).

**CMSO-Optimization Problems.** We are now in position to define the parameterized problems for which we will obtain kernelization results. For every CMSO formula $\psi$ with a single free vertex set variable $S$, we define the following two problems.

**$\psi$-CMSO-Min (Max)**:
INPUT*: Graph G and integer k.*
QUESTION*: Does there exist a vertex set $S \subseteq V(G)$ such that $(G, S) \models \psi$ and $|S| \le k$ ($|S| \ge k$ for Max).*

Formally, $\psi$-CMSO-Min (Max) is a parameterized graph problem where the "yes" instances are exactly the pairs $(G, k)$ such that there exists a vertex set $S$ of size at most $k$ (at least $k$) and $(G, S) \models \psi$. We will use the term *CMSO-optimization problems* to refer to $\psi$-CMSO-Min (Max) for some CMSO formula $\psi$.

Many well-studied and not so well-studied graph problems are CMSO-optimization problems. Examples include Vertex Cover, Dominating Set, Cycle Packing, the list goes on and on (see [2]). We encourage the interested reader to attempt to formulate the problems mentioned above as CMSO-optimization problems. We will be discussing CMSO-optimization problems *on planar graphs* and on minor-free classes of graphs.

Our results are for problems where the input graph is promised to belong to a certain graph class $\mathcal{C}$. We formalize this by encoding membership in $\mathcal{C}$ in the formula $\psi$. For an example, $\psi_{DS}$-CMSO-Min is the well-studied Dominating Set problem. If we want to restrict the problem to planar graphs, we can make a new CMSO logic formula $\psi_{\text{planar}}$ such that $G \models \psi_{\text{planar}}$ if and only if $G$ is planar. We can now make a new formula

$$\psi'_{DS} = \psi_{DS} \wedge \psi_{\text{planar}}$$

and consider the problem $\psi'_{DS}$-CMSO-Min. Here $(G, k)$ is a "yes" instance if $G$ has a dominating set $S$ of size at most $k$ *and* $G$ is planar. Thus, this problem also forces us to check planarity of $G$, but this is polynomial time solvable and therefore not an issue with repsect to kernelization. In a similar manner one can restrict any CMSO-optimization problem to a graph class $\mathcal{C}$, as long as there exists a CMSO formula $\psi_{\mathcal{C}}$ such that $G \models \psi_{\mathcal{C}}$ if and only if $G \in \mathcal{C}$. Luckily, such a formula is known to exist for every minor-free class $\mathcal{C}$. We will say that a parameterized problem $\Pi$ is a problem *on the graph class* $\mathcal{C}$ if, for every "yes" instance $(G, k)$ of $\Pi$, the graph $G$ is in $\mathcal{C}$.

For any CMSO-Min problem $\Pi$ we have that $(G,k) \in \Pi$ implies that $(G,k') \in \Pi$ for all $k' \geq k$. Similarly, for a CMSO-Max problem $\Pi$ we have that $(G,k) \in \Pi$ implies that $(G,k') \in \Pi$ for all $k' \leq k$. Thus the notion of "optimality" is well defined for CMSO-optimization problems. For the problem $\Pi = \psi$-CMSO-Min, we define

$$OPT_\Pi(G) = \min \{k \ : \ (G,k) \in \Pi\}\,.$$

If no $k$ such that $(G,k) \in \Pi$ exists, $OPT_\Pi(G)$ returns $+\infty$. Similarly, for the problem $\Pi = \psi$-CMSO-Max,

$$OPT_\Pi(G) = \max \{k \ : \ (G,k) \in \Pi\}\,.$$

If no $k$ such that $(G,k) \in \Pi$ exists, $OPT_\Pi(G)$ returns $-\infty$. We define $SOL_\Pi(G)$ to be a function that given as input a graph $G$ returns a set $S$ of size $OPT_\Pi(G)$ such that $(G,S) \models \psi$, and returns **null** if no such set $S$ exists.

**Bidimensionality** For many problems it holds that contracting an edge can not increase the size of the optimal solution. We will say that such problems are contraction closed. Formally a CMSO-optimization problem $\Pi$ is *contraction-closed* if for any $G$ and $uv \in E(G)$, $OPT_\Pi(G/uv) \leq OPT_\Pi(G)$. If contracting edges, deleting edges and deleting vertices can not increase the size of the optimal solution, we say that the problem is *minor-closed*.

Informally, a problem is *bidimensional* if it is minor closed and the value of the optimum grows with both dimensions of a grid. In other words, on a $(k \times k)$-grid the optimum should be approximately quadratic in $k$. To formally define bidimensional problems we first need to define the $(k \times k)$-grid $\boxplus_k$, as well as the related graph $\Gamma_k$.

For a positive integer $k$, a $k \times k$ *grid*, denoted by $\boxplus_k$, is a graph with vertex set $\{(x,y) \ : \ x,y \in \{1,\ldots,k\}\}$. Thus $\boxplus_k$ has exactly $k^2$ vertices. Two different vertices $(x,y)$ and $(x',y')$ are adjacent if and only if $|x-x'|+|y-y'| = 1$. For an integer $k > 0$, the graph $\Gamma_k$ is obtained from the grid $\boxplus_k$ by adding in every grid cell the diagonal edge going up and to the right, and making the bottom right vertex of the grid adjacent to all border vertices. The graph $\Gamma_9$ is shown in Fig. 1.

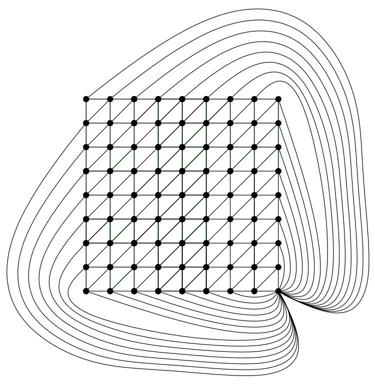

**Fig. 1.** The graph $\Gamma_9$.

We are now ready to give the definition of bidimensional problems. A CMSO-optimization problem $\Pi$ is *contraction-bidimensional* if it is contraction-closed, and there exists a constant $c > 0$ such that $OPT_\Pi(\Gamma_k) \geq ck^2$. Similarly, $\Pi$ is *minor-bidimensional* if it is minor-closed, and there exists a constant $c > 0$ such that $OPT_\Pi(\boxplus_k) \geq ck^2$.

As an example, the Dominating Set problem is contraction-bidimensional. It is easy to verify that contracting an edge may not increase the size of the smallest dominating set of a graph $G$, and that $\Gamma_k$ does not have a dominating set of size smaller than $\frac{(k-2)^2}{7}$.

**Separability** Our kernelization algorithms work by recursively splitting the input instance by small separators. For this to work, the problem has to be somewhat well behaved in the following sense. Whenever a graph is split along a small separator into two independent sub-instances $L$ and $R$, the size of the optimum solution for the graph $G[L]$ is relatively close to the size of the intersection between $L$ and the optimum solution to the original graph $G$. We now proceed with a formal definition of what it means for a problem to be well behaved.

For a set $L \subseteq V(G)$ we define $\partial(L)$ to be the set of vertices in $L$ with at least one neighbor outside $L$. A CMSO-optimization problem $\Pi$ is *linear-separable* if there exists a constant $c \geq 0$ such that for every set $L \subseteq V(G)$ we have

$$|SOL_\Pi(G) \cap L| - c \cdot |\partial(L)| \leq OPT_\Pi(G[L]) \leq |SOL_\Pi(G) \cap L| + c \cdot |\partial(L)|.$$

For a concrete example, we encourage the reader to consider the Dominating Set problem, and to prove that for Dominating Set the inequalities above hold. The crux of the argument is to augment optimal solutions of $G$ and $G[L]$ by adding all vertices in $\partial(L)$ to them.

## Key Results

We can now state our main theorem.

**Theorem 1.** *Let $\Pi$ be a separable CMSO-optimization problem on the graph class $\mathcal{C}$. Then, if $\Pi$ is minor-bidimensional and $\mathcal{C}$ is minor-free, or if $\Pi$ is contraction-bidimensional and $\mathcal{C}$ is apex-minor-free, $\Pi$ admits a linear kernel.*

The significance of Theorem 1 is that it is, in general, quite easy to formulate graph problems as CMSO-optimization problems, and prove that the considered problem is bidimensional and separable. If we are able to do this, Theorem 1 immediately implies that the problem admits a linear kernel on all minor-free graph classes, or on all apex-minor-free graph classes. As an example, the Dominating Set problem has been shown to have a linear kernel on planar graphs [1], and the proof of this fact is quite tricky. However, in our examples, we have shown that Dominating Set is a CMSO-Min problem, that it is contraction-bidimensional, and that it is separable. Theorem 1 now implies that Dominating Set has a linear kernel not only on planar graphs, but on all apex-minor-free classes of graphs! One can go through the motions and use Theorem 1 to give linear kernels for quite a few problems. We refer the reader to [9] for a non-exhaustive list.

We remark that the results stated here are generalizations of results obtained by Bodlaender et al. [2]. Theorem 1 is proved by combining "algebraic reduction rules" (fully developed by Bodlaender et al. [2]) with new graph decomposition theorems (proved in [9]). The definitions here differ slightly from the definitions in the original work [9] and appear here in the way they will appear in the journal version of [9].

## Cross-References

Bidimensionality
Data Reduction for Domination in Graphs

## Recommended Reading

1. Alber J, Fellows MR, Niedermeier R (2004) Polynomial-time data reduction for dominating set. J ACM 51(3):363–384
2. Bodlaender HL, Fomin FV, Lokshtanov D, Penninkx E, Saurabh S, Thilikos DM (2013) (Meta) Kernelization. CoRR abs/0904.0727, URL `http://arxiv.org/abs/0904.0727`
3. Borie RB, Parker RG, Tovey CA (1992) Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. Algorithmica 7(5&6):555–581
4. Cygan M, Fomin FV, Kowalik Ł, Lokshtanov D, Marx D, Pilipczuk M, Pilipczuk M, Saurabh S (2015, to appear) Parameterized Algorithms. Springer
5. Demaine ED, Hajiaghayi M (2005) Bidimensionality: new connections between FPT algorithms and PTASs. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, pp 590–601
6. Demaine ED, Hajiaghayi M (2008) The bidimensionality theory and its algorithmic applications. Comput J 51(3):292–302
7. Demaine ED, Fomin FV, Hajiaghayi M, Thilikos DM (2005) Subexponential parameterized algorithms on graphs of bounded genus and $H$-minor-free graphs. J ACM 52(6):866–893
8. Downey RG, Fellows MR (2013) Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer
9. Fomin FV, Lokshtanov D, Saurabh S, Thilikos DM (2010) Bidimensionality and kernels. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, pp 503–510
10. Fomin FV, Lokshtanov D, Raman V, Saurabh S (2011) Bidimensionality and EPTAS. In: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, pp 748–759