Table des matières

Dynamic programming on bipartite tree decompositions

Lars Jaffke, Department of Informatics, University of Bergen, lars.jaffke@uib.no
Laure Morelle, LIRMM, Université de Montpellier, laure.morelle@lirmm.fr
Ignasi Sau, LIRMM, Université de Montpellier, ignasi.sau@lirmm.fr
Dimitrios M. Thilikos, LIRMM, Université de Montpellier, sedthilk@thilikos.info

We revisit a graph width parameter that we dub *bipartite treewidth*, along with its associated graph decomposition that we call *bipartite tree decomposition*. Bipartite treewidth can be seen as a common generalization of treewidth and the odd cycle transversal number.

Intuitively, a bipartite tree decomposition is a tree decomposition whose bags induce almost bipartite graphs and whose adhesions contain at most one vertex from the bipartite part of any other bag, while the width of such decomposition measures how far the bags are from being bipartite.

Adapted from a tree decomposition originally defined by Demaine, Hajiaghayi, and Kawarabayashi [SODA 2010] and explicitly defined by Tazari [Theor. Comput. Sci. 2012], bipartite treewidth appears to play a crucial role for solving problems related to *odd-minors*, which have recently attracted considerable attention.

As a first step toward a theory for solving these problems efficiently, we develop dynamic programming techniques to solve problems on graphs of small bipartite treewidth. For such graphs, we provide a number of para-NP-completeness results, FPT-algorithms, and XP-algorithms, as well as several open problems. In particular, we show that K_t-Subgraph-Cover, Weighted Vertex Cover/Independent Set, Odd Cycle Transversal, and Maximum Weighted Cut are FPT parameterized by bipartite treewidth. We also provide the following complexity dichotomy when H is a 2-connected graph, for each of the H-Subgraph-Packing, H-Induced-Packing, H-Scattered-Packing, and H-Odd-Minor-Packing problems : if H is bipartite, then the problem is para-NP-complete parameterized by bipartite treewidth while, if H is non-bipartite, then the problem is solvable in XP-time.