Journal of Combinatorial Theory, Series B 122 (2017) 815–843

Contents lists available at ScienceDirect

Journal of Combinatorial Theory, Series B

www.elsevier.com/locate/jctb

Irrelevant vertices for the planar Disjoint Paths Problem

Isolde Adler [a], Stavros G. Kolliopoulos [b,1], Philipp Klaus Krause [c,2], Daniel Lokshtanov [d,3], Saket Saurabh [e,d,3], Dimitrios M. Thilikos [f,g,1]

[a] *School of Computing, University of Leeds, Leeds, United Kingdom*
[b] *Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece*
[c] *Institut für Informatik, Goethe-Universität, Frankfurt, Germany*
[d] *Department of Informatics, University of Bergen, Norway*
[e] *The Institute of Mathematical Sciences, CIT Campus, Chennai, India*
[f] *Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece*
[g] *AlGCo project-team, LIRMM, CNRS, Montpellier, France*

A R T I C L E I N F O

Article history:
Received 26 July 2013
Available online 31 October 2016

Keywords:
Graph Minors
Treewidth
Disjoint Paths Problem

A B S T R A C T

The Disjoint Paths Problem asks, given a graph G and a set of pairs of terminals $(s_1, t_1), \dots, (s_k, t_k)$, whether there is a collection of k pairwise vertex-disjoint paths linking s_i and t_i, for $i = 1, \dots, k$. In their $f(k) \cdot n^3$ algorithm for this problem, Robertson and Seymour introduced the *irrelevant vertex technique* according to which in every instance of treewidth greater than $g(k)$ there is an "irrelevant" vertex whose removal creates an equivalent instance of the problem. This fact is based on the celebrated *Unique Linkage Theorem*, whose – very technical – proof gives a function $g(k)$ that

E-mail addresses: i.m.adler@leeds.ac.uk (I. Adler), sgk@di.uoa.gr (S.G. Kolliopoulos), philipp@informatik.uni-frankfurt.de (P.K. Krause), daniello@ii.uib.no (D. Lokshtanov), saket@imsc.res.in (S. Saurabh), sedthilk@thilikos.info (D.M. Thilikos).

[1] Co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: "*Thalis. Investing in knowledge society through the European Social Fund*".
[2] Supported by a fellowship within the FIT-Programme of the German Academic Exchange Service (DAAD) at NII, Tokyo and by DFG-Projekt GalA, grant number AD 411/1-1.
[3] Supported by "Rigorous Theory of Preprocessing, ERC Advanced Investigator Grant 267959" and "Parameterized Approximation, ERC Starting Grant 306992".

http://dx.doi.org/10.1016/j.jctb.2016.10.001

is responsible for an immense parameter dependence in the running time of the algorithm. In this paper we give a new and self-contained proof of this result that strongly exploits the combinatorial properties of planar graphs and achieves $g(k) = O(k^{3/2} \cdot 2^k)$. Our bound is radically better than the bounds known for general graphs.

1. Introduction

One of the most studied problems in graph theory is the Disjoint Paths Problem (DPP): *Given a graph G and a set $\mathcal{P}$ of k pairs of terminals, $(s_1, t_1), \ldots, (s_k, t_k)$, decide whether G contains k vertex-disjoint paths $P_1, \ldots, P_k$ where P_i has endpoints s_i and t_i*, $i = 1, \ldots, k$. In addition to its numerous applications in areas such as network routing and VLSI layout, this problem has been the catalyst for extensive research in algorithms and combinatorics [27]. DPP is NP-complete, along with its edge-disjoint or directed variants, even when the input graph is planar [16–18,28]. The celebrated algorithm of Roberson and Seymour solves it however in $f(k) \cdot n^3$ steps, where f is some computable function [22]. This implies that, when we parameterize DPP by the number k of pairs of terminals, the problem is fixed-parameter tractable. The Robertson–Seymour algorithm is the central algorithmic result of the Graph Minors series of papers, one of the deepest and most influential bodies of work in graph theory.

The basis of the algorithm in [22] is the so-called *irrelevant-vertex technique* which can be summarized very roughly as follows. As long as the input graph G violates certain structural conditions, it is possible to find a vertex v that is *solution-irrelevant:* every collection of paths certifying a solution to the problem can be rerouted to an *equivalent* one, that links the same pairs of terminals, but in which the new paths avoid v. One then iteratively removes such irrelevant vertices until the structural conditions are met. By that point the graph has been simplified enough so that the problem can be attacked via dynamic programming.

The following two structural conditions are used by the algorithm in [22]: (i) G excludes a clique, whose size depends on k, as a minor and (ii) G has treewidth bounded by some function of k. When it comes to enforcing Condition (ii), the aim is to prove that in graphs without big clique-minors and with treewidth at least $g(k)$ there is always a solution-irrelevant vertex. This is the most complicated part of the proof and it was postponed until the later papers in the series [23,24]. The bad news is that the complicated proofs also imply an *immense* parametric dependence, as expressed by the function f, of the running time on the parameter k. This puts the algorithm outside the realm of feasibility even for elementary values of k.

The ideas above were powerful enough to be applicable also to problems outside the context of the Graph Minors series. During the last decade, they have been applied to