Connectivity check in 3-connected planar graphs with obstacles

Bruno Courcelle [a,1,2,3], Cyril Gavoille [a,1,3],
Mamadou Moustapha Kanté [a,1,3] and Andrew Twigg [b,4,3]

[a] *Bordeaux 1 University, LaBRI, CNRS*
351 cours de la Libration
33405 Talence, France

[b] *Computer Laboratory, Cambridge University*
CB3 0FD United Kingdom

Abstract

We define a vertex labelling for every planar 3-connected graph with n vertices from which one can answer connectivity queries. A *connectivity query* asks whether there exists in the given graph a path linking u and v that avoids a set F of edges and a set X of vertices. The vertices u, v and those of X are given by their labels. The edges of F are given by the labels of their ends. Each label has a size of $O(\log(n))$ bits. Our construction makes an essential use of *straight-line embeddings* on $n \times n$ grids of simple loop-free planar graphs. Such embeddings can be constructed in linear time by Schnyder's algorithm [7].

Keywords: Planar graph, Compact labelling scheme, Connectivity, Obstacles

[1] Supported by the GRAAL project of "Agence Nationale pour la Recherche".
[2] Member of "Institut Universitaire de France".
[3] courcell@labri.fr(contact author), gavoille@labri.fr, kante@labri.fr, andy.twigg@gmail.com.
[4] Supported by ITA project, US Army Research laboratory and the UK Ministry of Defence, W911NF-06-3-0001

Let us review the motivations for looking for compact labellings of graphs. By *compact*, we mean of order less than n, the number of vertices. In distributed computing over a communication network with underlying graph G, nodes must act according to their local knowledge only, which is updated by message passing. Due to space constraints on local memory of nodes, and limited message size, a distributed task cannot be solved in practice by representing the whole graph G in each node or in each message, but must rather manipulate more compact representations of G. Typically, the routing task may involve routing tables, that are sublinear in the size of G (preferably of poly-logarithmic size), and short addresses transmitted in the headers of messages (of poly-logarithmic size too). As surveyed in [6] many problems including routing and distance computation can be achieved using *compact labels*. If nodes or links fail in a network, then recomputation of the labels is generally required. Courcelle and Twigg studied in [4] the *forbidden-set labelling problem*, where in addition to source and destination, the query algorithm is given a set of failed nodes and edges and it must construct a path that avoids this set. These labels can be used to quickly recover from failures in a network. In this framework labellings can be updated by transmitting to all surviving nodes the list of labels of all defected nodes and links, so that surviving nodes can update their local data-structures (e.g., their routing tables).

We now state the main theorem and describe the main ideas of the construction. If G is a graph, $u,v \in V(G)$, $X \subseteq V(G)-\{u,v\}$ and $F \subseteq E(G)$, we let $\mathrm{Conn}(u,v,X,F)$ mean : there exists a path between u and v that *avoids X and F*, i.e, a path in the graph $(G-F)\backslash X$. We call this a *connectivity query* (implicitly in the subgraph of G defined by excluding X and F). We write $\mathrm{Conn}(u,v,X)$ if $F=\varnothing$. A labelling *supports a query* if it makes possible to answer it from the labels of the arguments.

Main Theorem *For every simple undirected* 3*-connected planar graph, we can construct in $O(n\log(n))$ time an $O(\log(n))$-bit labelling supporting connectivity queries.*

The problem of connectivity labelling in planar undirected graphs is easy without forbidden sets ($\log(n)$ bits suffice to identify the connected components). The problem of *reachability* in directed planar graphs is not easy. It is known to be LOGSPACE-hard [1]. It can be solved efficiently with labelling schemes. Thorup [8] shows that a planar digraph can be preprocessed in near-linear time, producing a near-linear space oracle that can answer reachability queries in constant time. The oracle can be distributed as an $O(\log(n))$ space labelling for each vertex from which we can determine if one vertex can reach another by considering their two

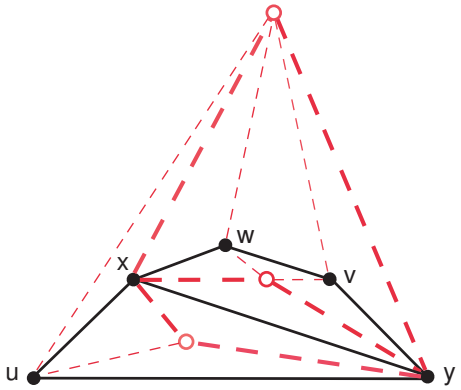

Fig. 1. The augmented graph discussed in the example.

labels only.

More difficult queries that we know how to treat for graphs of bounded clique-width ([4]) but not yet for planar graphs are *distance queries* and *routing* (explicit construction of shortest paths).

For a plane graph G, we let G^+ be the plane graph obtained by the addition of one new vertex in the middle of each face and of edges between this vertex and those of G incident with that face. If G is biconnected, the graph G^+ is simple and can be embedded in the $m \times m$-grid where $m = |V(G^+)|$. A linear-time algorithm for doing so has been given by Schnyder [7]. We fix such an embedding. For $X \subseteq V(G)$, we let its *barrier* $\mathrm{Bar}(X)$ be a set of edges of G^+ such that u and $v \in V(G) - X$ are separated by X in G iff they are separated in $\mathbb{R}^2$ by $\mathrm{Bar}(X)$, i.e., belong to different connected components of $\mathbb{R}^2 - \mathrm{Bar}(X)$. Note that $\mathrm{Bar}(X)$ is a set of straight line segments, hence $\mathbb{R}^2 - \mathrm{Bar}(X)$ is a union of connected open sets. (See [3] for the precise definition of the set $\mathrm{Bar}(X)$.)

Example Figure 1 shows a graph H with vertices u,v,w,x,y and edges represented by continuous lines, and its augmented graph H^+. Dotted lines represent the edges of H^+ that are not in H. The graph H^+ is simple since H is biconnected and it is drawn with straight-lines. The barrier of $\{x,y\}$ consists of the 6 (thick) dotted edges $(\{x,a\},\{x,b\},\{x,c\},\{y,a\},\{y,b\},\{y,c\})$ where a,b,c are the face vertices in H^+. It separates u from v and w.

If, from labels attached to the vertices of X we can deduce the set of straight-line segments forming $\mathrm{Bar}(X)$, and if we also know the coordinates of u and v, we can test whether u and v are separated in $\mathbb{R}^2$ by $\mathrm{Bar}(X)$ by means of computational geometry algorithms (De Berg et al. [2]).

To do so, to each vertex x of G, we attach, not only its own pair of coordinates

in the fixed embedding, but also those of a bounded number of neighbour vertices of G and of vertices of G^+ representing faces of G. This can be done because every planar graph is the union of 3 edge disjoint forests. This fact is used in [7].

However this sketched proof only works for 3-connected planar graphs G, or rather for those such that every 2 vertices are incident with a bounded number of faces. Hence, in particular for 3-connected graphs where each edge is subdivided by the addition of one degree 2 vertex. This is useful, because we can in this way reduce the problem for 3-connected graphs to the particular case where F is empty, i.e., where one deletes only vertices.

With many additional constructions, we can extend this result to all undirected planar connected graphs [3]. In a few words, we consider first biconnected graphs decomposed into 3-connected components, and then connected graphs decomposed into biconnected components, which gives the general theorem. These decompositions are expressed as trees. By using a labelling scheme due to Courcelle and Vanicat [5], we can recognize certain cases where u and v are separated by one or two vertices of the given set X. If those separation criteria do not apply, then we are reduced to connectivity queries in certain 3-connected components, and the geometric method of the present communication can be used. We conjecture that the main theorem extends to graphs embedded in a fixed surface.

For undirected planar graphs, distance and routing queries should also be investigated. The problem of reachability in directed planar graphs with obstacles is a difficult and important open question. Another one concerns the case where the network is locally modified : how can the labelling be updated efficiently ?

References

[1] E. Allender, S. Datta, S. Roy, *The Directed Planar Reachability Problem*. Proceedings of FSTTCS 2005, Lec. Notes Comput. Sci. **3821** (2005), 238-249.

[2] M. de Berg et al. *Point location*, Chapter 6 of *Computational Geometry*, Springer 1997, pp. 119-144.

[3] B. Courcelle, C. Gavoille, M.M. Kanté, A. Twigg, *Optimal labelling for connectivity checking in planar networks with obstacles*, Preprint, March 2008.

[4] B. Courcelle, A. Twigg, *Compact Forbidden-set Routing*, 24^{th} Annual Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science **4393** (2007), 37-48.

[5] B. Courcelle, R Vanicat, *Query efficient implementations of graphs of bounded clique-width*, Discrete Applied Mathematics. **131** (2003), 129-150.

[6] C. Gavoille, D. Peleg, *Compact and Localized Distributed Data Structures*, Distributed Computing, **16** (2003), 111-120.

[7] W. Schnyder, *Embedding planar graphs in the grid*, SODA, First ACM-SIAM Symposium on Discrete Algorithms, San Francisco, 1990, pp. 138-148.

[8] M. Thorup, *Compact oracles for reachability and approximate distances in planar digraphs*, J. ACM **51** (2004), 993-1024.