Information Processing Letters 41 (1992) 29–33
North-Holland

27 April 1992

Detecting cycles through three fixed vertices in a graph

Herbert Fleischner

Institute of Information Processing, Austrian Academy of Sciences, Dr. Ignaz Seipel Platz 2, A-1010 Vienna, Austria

Gerhard J. Woeginger

Institut für Mathematik B, Technische Universität Graz, Kopernikusgasse 24, A-8010 Graz, Austria

Communicated by M.J. Atallah
Received 30 October 1991
Revised 24 January 1992

Abstract

Fleischner, H. and G.J. Woeginger, Detecting cycles through three fixed vertices in a graph, Information Processing Letters 42 (1992) 29–33.

We present a simple linear time algorithm for finding a cycle through three fixed vertices in an undirected graph. The algorithm is based on decompositions into triconnected components and on a combinatorial result of Lovász.

Keywords: Algorithms, graph algorithms, subgraph homeomorphism problem

1. Introduction

Given a graph $G=(V, E)$ and three distinct vertices x, y and z in V, the *three-vertex-cycle problem* (3VP for short) is to determine whether the three vertices lie on a common simple cycle, and to find such a cycle if it does exist. The corresponding problem in *directed* graphs (which consists in detecting a *directed* simple cycle) has been shown to be NP-complete by Fortune, Hopcroft and Wyllie [2]. In 1980, LaPaugh and Rivest [6] presented a linear time algorithm for the 3VP in *undirected* graphs. This algorithm is rather lengthy and involves much case analysis. Some details have been omitted due to space and readability considerations. A full version of the algorithm may be found in [5]. For informations on related problems the reader is referred to Johnson's NP-completeness column [4].

In this note, we present a very simple alternative algorithm to solve the 3VP in undirected graphs. Contrary to the purely algorithmic approach of LaPaugh and Rivest, we first explore the underlying combinatorial structure of the problem. We give a clean characterization of all unsolvable cases which is strongly based on a result of Lovász. This note is organized as follows. Section 2 gives some combinatorial preliminaries on triconnected graphs, Section 3 presents our algorithmic results.

Correspondence to: Professor G.J. Woeginger, Institut für Mathematik B, Technische Universität Graz, Kopernikusgasse 24, A-8010 Graz, Austria.

2. Combinatorial preliminaries

We begin with some definitions. All graphs treated in this paper are undirected and loopless.

A *path* P of *length* $|P|=k$ in a graph $G=(V, E)$ is a sequence of edges $e_i \in E$, $1 \leqslant i \leqslant k$, such that e_i is of the form $e_i=[x_i, x_{i+1}] \in E$. A path is called *simple* if additionally $x_i \neq x_j$ holds for $i \neq j$, except possibly $x_1=x_{k+1}$. A simple path with $x_1=x_{k+1}$ is a *simple cycle*. For $V_1, V_2, \ldots, V_k$ ($k \geqslant 2$) subsets of V, we define a V_1–V_2–$\cdots$–V_k path to be any simple path starting at some vertex in V_1, visiting V_2 through V_{k-1} and ending at some vertex of V_k. For $V' \subset V$, we denote by $G-V'$ the graph that results from removing V' and all incident edges from G.

A graph is *biconnected* if it has at least three vertices and if removing any vertex does not disconnect it. A graph is *triconnected* if it has at least four vertices and if removing any two vertices does not disconnect it.

Our algorithm will make excessive use of the following two theorems.

Theorem 1. *Let $G=(V, E)$ be a triconnected, undirected graph. Let x denote any vertex in V, and let A, B, C and D be four subsets of V with $|A|=|B|=3$, $|C|=|D|=2$. Assume moreover that any two of the sets $\{x\}$, A, B, C and D are disjoint. Then the following holds.*

(i) *There exists a simple cycle through the three vertices in A.*

(ii) *There exist three simple $\{x\}$–A paths that are pairwise vertex disjoint with the exception of the vertex x.*

(iii) *There exist three simple vertex disjoint A–B paths.*

(iv) *There exist a C–D path and a C–$\{x\}$–D path that are vertex disjoint.*

(v) *There exists a simple C–D–D–C path.*

Moreover, all these paths and cycles can be constructed in linear time.

Proof. The proofs of (i), (ii) and (iii) easily follow from Menger's Theorem and may be found, e.g., in [10].

To prove (iv), let $C=\{c_1, c_2\}$ and let $D=\{d_1, d_2\}$. We first construct two vertex disjoint C–D paths which exist by (iii) and call them P_1 and P_2. If x lies on one of these paths, we are finished. Otherwise, we use three simple $\{x\}$–$\{c_1, c_2, d_1\}$ paths Q_1, Q_2 and Q_3 that are pairwise vertex disjoint with the exception of x. Let Q_i' ($1 \leqslant i \leqslant 3$) denote the subpath of Q_i going from x to the first vertex q_i that lies also on P_1 or on P_2. Then two of the three vertices q_i must lie on the same P_j, and obviously we can use subpaths of P_1, P_2 and some of the Q_i' to derive the desired paths.

The proof of (v) is just an easy modification of the proof of (iv). Again, let $C=\{c_1, c_2\}$ and let $D=\{d_1, d_2\}$. Construct two $\{d_1\}$–C paths that are vertex disjoint with the exception of d_1 and call them P_1 and P_2. Then construct three paths Q_i connecting d_2 to the set $\{c_1, c_2, d_1\}$. Using subpaths as above completes the proof.

Finally, we observe that all paths and cycles that we used can be constructed by finding network flows (all of value two or three). It is known that this can be done in linear time (see, e.g., [1]). $\square$

Theorem 2. *Let $G=(V, E)$ be a triconnected, undirected graph. Let $e_1, e_2, e_3 \in E$ be three distinct edges in G. Then at least one of the following three statements holds.*

(a) *All three e_i are incident to some common vertex.*

(b) *$\{e_1, e_2, e_3\}$ is a separating edge set in G.*

(c) *There exists a simple cycle through e_1, e_2 and e_3.*

(*If* (c) *holds, obviously neither* (a) *nor* (b) *can be true and vice versa; statements* (a) *and* (b) *are not mutually exclusive.*) *If* (c) *holds, the cycle can be found in linear time.*

Proof. Define $e_i=[x_i, y_i]$, $i=1,\ldots,3$, and call these three edges the red edges. First we get rid of the trivial cases where G contains some vertex x which is incident to exactly two of the red edges. We show that in these cases, condition (c) must be fulfilled: Assume, without loss of generality, $x=x_1=x_2$. If $\{y_1, y_2\} \cap \{x_3, y_3\}=\emptyset$, we use the fact that $G-\{x\}$ is biconnected: There exist two vertex disjoint $\{y_1, y_2\}$–$\{x_3, y_3\}$ paths in $G-\{x\}$ which can be used to produce the desired cycle. Similarly, if $\{y_1, y_2\} \cap \{x_3, y_3\} \neq \emptyset$, then either the red edges induce a triangle (and form a cycle as described in (c)), or we use the fact that $G-\{x, y\}$ is connected (where, without loss of generality, $y:=y_2=y_3$). Then there exists a path connecting the two end-vertices of the path induced by the red edges; these two paths define a desired cycle.

Consequently, we may assume that the e_i are pairwise vertex disjoint and do not fulfill (a). This case is treated in problem 6.67 in Lovász's book [7]; there it is shown that either (b) or (c) holds. We will shortly sketch this proof, as it leads to a simple algorithm.

Lovász assumes that (c) does not hold. In a first step, he shows that there exists a simple cycle C containing two of the red edges, which does not meet the third one. Namely, let C_1 be a simple cycle through e_1 and e_2. If x_3 and y_3 both lie on C_1, then e_3, e_1, e_2 and subpaths of C_1 yield C. If x_3 lies on C_1 and y_3 does not, he considers a path from y_3 to $C_1 - \{x_3\}$. Then subpaths of this path and of C_1 together with e_3 give the desired C.

Now suppose that both x_3 and y_3 do not lie on C_1; then $C-e_1-e_2 = P_1 \cup P_2$, where P_1, P_2 are disjoint paths. In the rest of his proof, Lovász shows that either $\{e_1, e_2, e_3\}$ is a separating edge set in G (and statement (b) is true), or there are two disjoint paths connecting $\{x_3, y_3\}$ to P_1 and not meeting P_2, or there are two disjoint paths connecting $\{x_3, y_3\}$ to P_2 and not meeting P_1. Such two disjoint paths together with e_3 and a subpath of C yield a simple cycle through the red edges, and the proof is complete.

Finally, let us investigate the algorithmic consequences of Lovász's proof. If the red edges do not separate the graph (which can be checked in linear time), we start with any simple cycle through e_1 and e_2. Following the construction as described above, we either find a cycle through all three red edges or a cycle C which does not meet e_3. We assume that P_1 connects x_1 to x_2 and that P_2 connects y_1 to y_2. Since the red edges do not separate the graph, we know that there are two disjoint paths connecting $\{x_3, y_3\}$ to P_1 and not meeting P_2, or two disjoint paths connecting $\{x_3, y_3\}$ to P_2 and not meeting P_1. Hence, we only have to find two disjoint paths from $\{x_3, y_3\}$ to $\{x_1, x_2\}$ in $G - P_2$ or two disjoint paths from $\{x_3, y_3\}$ to $\{y_1, y_2\}$ in $G - P_1$. Obviously, these paths can be found in linear time by finding network flows. □

3. How to find the cycle

Now let us attack the three-vertex-cycle problem for a graph $G = (V, E)$ and three distinct vertices x, y and z. First of all, we may assume that G is biconnected. (Either x, y and z all lie in the same biconnected component of G or the problem trivially does not have a solution.) To get results for biconnected graphs, we use decompositions into split components. For standard definitions of split components and algorithms for obtaining them, see [8] and [3].

We simply recall that split components are obtained by splitting the graph at pairs of vertices: Let u and v be a pair of vertices whose removal disconnects a biconnected graph G. Let $G_1, \ldots, G_k$ be the components of $G - \{u, v\}$. Insert a copy of u and v in each graph G_i, maintaining their connections as in G, and also insert a virtual edge $[u, v]$ (even if such edge exists already in G). Continue this procedure in each of these k graphs with another pair of separating vertices until no further reduction is possible. The resulting graphs are the *split components*. They are multigraphs of three types: triconnected graphs, triangles and triple bonds (which consist of two vertices and three edges between them). Clearly, any two split components have at most two vertices in common. We will call pairs of vertices which are shared by two or more split components *connecting pairs*. Note that two different connecting pairs may have one vertex in common. We also observe that we may obtain different sets of split components depending on the sequence of pairs of separating vertices by which we proceed in producing the split components.

However, it is known (see [9]) that the split components together with the connecting pairs form a tree $T(G)$ which is defined in the following way. The vertices of the tree correspond to split components and connecting pairs, an edge represents containment of a connecting pair in a split component.

Next, we prune $T(G)$ down to the smallest subtree $T_1(G)$ that contains all vertices corresponding to split components with x, y or z inside. $T_1(G)$ is either a path or homeomorphic to

$K_{1,3}$; all end-vertices of $T_1(G)$ correspond to split components containing at least one of the vertices x, y, z.

Theorem 3. *For a biconnected graph G and three vertices x, y and z, the* 3VP *is solvable unless $T_1(G)$ is homeomorphic to $K_{1,3}$ and where the three branches of $T_1(G)$ meet at a vertex corresponding*

(a) *to some connecting pair, or*

(b) *to some triple bond, or*

(c) *to some triconnected split component G^* such that the three edges in G^* whose endpoints share the same copies of vertices with vertices in the three adjacent connecting pairs form a configuration as described in Theorem* 2(a) *or* (b).

Proof. If all three vertices are in the same split component, we apply Theorem 1(i) to this split component to obtain a cycle $C \supseteq \{x, y, z\}$ such that C is a cycle in G as well, or else can be extended to $C_1 \supseteq \{x, y, z\}$ in G by replacing virtual edges by paths in split components. If two vertices (say y and z) lie in the same split component $G(y)$ and x does not lie in $G(y)$, we consider the unique connecting pair $[u, v]$ in $T_1(G)$ adjacent to $G(y)$. Obviously, there exist two $\{u, v\}$–$\{x\}$ paths which are vertex disjoint with the exception of x and which do *not* enter $G(y)$. Inside of $G(y)$, we apply Theorem 1(v) by setting $C = \{u, v\}$ and $D = \{y, z\}$. These three paths glued together at u and v form a simple cycle through x, y and z.

It remains to consider the cases where x, y and z all lie in different split components $G(x)$, $G(y)$ and $G(z)$, respectively. If $T_1(G)$ is a path with $G(y)$ as interior vertex, consider the two connecting pairs (u, v), (u', v') adjacent to $G(y)$. If $u = u'$, we find two $\{y\}$–$\{v, v'\}$ paths in $G(y)$ which do not contain $u = u'$ (note that $G(y)$ is triconnected since it has at least four vertices), two $\{x\}$–$\{u, v\}$ paths in $G(x)$ and two $\{z\}$–$\{u', v'\}$ paths in $G(z)$. The union of these paths forms a cycle as required. If $\{u, v\} \cap \{u', v'\} = \emptyset$, we apply Theorem 1(iv) and use an analogous construction.

If $T_1(G)$ is homeomorphic to $K_{1,3}$ whose 3-valent vertex corresponds to a connecting pair, then it is obvious that no solution exists (this connecting pair would partition the simple cycle into three parts, which is impossible). We may argue in an analogous way, if $T_1(G)$ is homeomorphic to $K_{1,3}$ with a triple bond in the center. Finally, if $T_1(G)$ is homeomorphic to $K_{1,3}$ with a triconnected split component G^* as 3-valent vertex, we easily find internally disjoint path pairs from x, y and z, respectively, to the respective connecting pair which is adjacent to G^* in $T_1(G)$. Note that any two of these altogether six paths have no vertex in common except possibly one which belongs to a connecting pair, or x, y, z, respectively, if these two paths form one of the above path pairs. Moreover, the vertices of each of these connecting pairs are joined in G^* by e_1, e_2, e_3, respectively. Therefore, the existence of a cycle C through x, y, z is reduced to the existence of a cycle C_0 through e_1, e_2, e_3 in G^*. Consequently, if e_1, e_2, e_3 have the property as stated in Theorem 2(a) or (b), C_0 and thus C does not exist. □

Corollary 4. *Given a graph $G = (V, E)$ and three vertices in V, we can decide in linear time whether the three vertices lie on a common simple cycle and find such a cycle in case it exists.*

Proof. Hopcroft and Tarjan [3] show how to decompose a graph into split components in linear time. All constructions used in the proofs of Theorems 1, 2 and 3 can be implemented in linear time by using network flows. □

References

[1] S. Even and R.E. Tarjan, Network flow and testing graph connectivity, *SIAM J. Comput.* **4** (1975) 691–703.

[2] S. Fortune, J. Hopcroft and J. Wyllie, The directed subgraph homeomorphism problem, *Theoret. Comput. Sci.* **10** (1980) 111–121.

[3] J.E. Hopcroft and R.E. Tarjan, Dividing a graph into triconnected components, *SIAM J. Comput.* **2** (1973) 135–158.

[4] D.S. Johnson, The NP-completeness column: An ongoing guide, *J. Algorithms* **8** (1987) 285–303.

[5] A.S. LaPaugh, The subgraph homeomorphism problem,

Massachusetts Institute of Technology, Laboratory for Computer Science TM 99, 1978.
[6] A.S. LaPaugh and R.L. Rivest, The subgraph homeomorphism problem, *J. Comput. System Sci.* **20** (1980) 133–149.
[7] L. Lovász, *Combinatorial Problems and Exercises* (North-Holland, Amsterdam, 1978).
[8] W.T. Tutte, The factorization of linear graphs, *J. London Math. Soc.* **22** (1947) 107–111.
[9] W.T. Tutte, *Connectivity in Graphs* (University of Toronto Press, Toronto, 1966).
[10] W.T. Tutte, *Graph Theory* (Addison-Wesley, Reading, MA, 1984).