An Algorithm for Cyclic Edge Connectivity of Cubic Graphs

Zdeněk Dvořák   Jan Kára   Daniel Král'
Ondřej Pangrác

Department of Applied Mathematics and
Institute for Theoretical Computer Science (ITI)*
Charles University
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{rakdver,kara,kral,pangrac}@kam.mff.cuni.cz

Abstract

The cyclic edge connectivity is the size of the smallest edge cut in a graph such that at least two of the parts of the graph are not acyclic. We present an algorithm running in time $O(n^2 \log^2 n)$ for computing the cyclic edge connectivity of n-vertex cubic graphs.

1 Introduction

Cyclic connectivity as a graph parameter was introduced by Tait [6] already in 1880. A *cyclic edge cut* of a graph G is an edge cut such that at least two components of the new graph contain a cycle. If G is not connected and at least two of its components contain a cycle, then an empty set of edges form a cyclic edge cut. *Cyclic edge connectivity* of a graph G is the size of the smallest cyclic edge cut. If G is connected, then the smallest cyclic edge cut splits G in exactly two components. A graph may have no cyclic edge cuts at all, graphs K_4, K_5 and $K_{3,3}$ are examples of such graphs. Cyclic edge connectivity has been studied, e.g., in [5] for planar graphs or

*Institute for Theoretical Computer Science is supported by Ministry of Education of Czech Republic as project LN00A056.

in [4] in relation to other connectivity parameters. Results on a structure of so-called atoms of cyclic connectivity for the class of cubic graphs, the graph class which mainly interest us, can be found in [2].

The real importance of cyclic vertex and edge connectivity is for classes of graphs of bounded degree. Usual connectivity parameters are bounded for such classes of graphs and hence they provide only a little information about the level of connectivity. A prominent class of graphs with bounded maximum degree is a class of *cubic graphs*, i.e., graphs with all vertex degrees equal to three. Cyclic edge connectivity is in particular important for snarks, cubic graphs which are not three-edge-colorable [3]. Despite this, there is only a little of previous work on algorithms for cyclic edge connectivity of either cubic graphs or graphs in general. The only published result is an $O(n^3 \log n)$-algorithm by Lou et. al.[1] but it turned out that the published version of the algorithm is flawed.

We present in this paper two algorithms for computing cyclic edge connectivity of cubic graphs, one running in time $O(n^3 \log n)$ and the other in time $O(n^2 \log^2 n)$. Both our algorithms are simple to implement and their running time bounds do not involve any hidden large constants. So, either of them can be said to be practical. The main tool for constructions of our algorithms is a structural result obtained in Section 3 which is of its own interest: Consider a cubic graph G with cyclic edge connectivity κ. Then, G either contains a cycle of length κ or there exists a cyclic edge cut of size κ such that each of the components contain a so-called full tree of depth $\Omega(\log \kappa)$ (Theorem 1). We refer the reader for a definition of a full tree to Section 2, but it might suffice to say at the moment that a full tree of depth d contains a complete binary tree of depth d as a subgraph. We remark that there is a short proof of the existence of a full tree of depth $\Omega(\log \kappa)$ in this setting, but we need that its activity (see Section 2 for a formal definition of this parameter) is at least $\kappa + 1$. So an arbitrary function of order $\Omega(\log \kappa)$ does not suffice for our purposes. Even this statement can be proved easily if κ is sufficiently large, but if we want to design practical algorithms, we need that the activity of such a tree is at least $\kappa + 1$ for all values of κ. It turned out that actually small values of κ (between 6 and 25) require quite a lot of work and the proof is even a little technical in some aspects.

We present our $O(n^3 \log n)$-algorithm in Section 4. The algorithm first computes the girth g of an input graph G. The girth is an upper bound on the cyclic edge connectivity of a cubic graph if the number of vertices is at least 8 (Lemma 1). Then, the algorithm computes minimum edge separations for all pairs of vertex disjoint full trees of depth at most $O(\log g)$

(the number of such pairs is $O(n^2 \log^2 \log n)$). The correctness of such an algorithm follows straightforwardly from Theorem 1.

Next, we present an $O(n^2 \log^2 n)$-algorithm in Section 5. Again, the algorithm first computes the girth g of an input graph G. If the number of vertices of the input graph is small (at most 242), we run the $O(n^3 \log n)$-algorithm for computing the edge connectivity. Otherwise, we find g edge-disjoint large subgraphs of G. We mean by large subgraphs such subgraphs whose activity is at least g. Since g is an upper bound on the cyclic edge connectivity, we want to test the existence of a cyclic edge cut of size at most $g-1$. If such a cyclic edge cut exists, then one of these g subgraphs is disjoint from it. Hence it is enough to compute minimum edge separations between these g subgraphs and vertex disjoint full trees of depth at most $O(\log g)$. The number of such pairs is $O(n \log n \log \log n)$ which reduces the time complexity of the algorithm compared to the algorithm from Section 4.

Both our algorithms assume that an input cubic graph contains at least 8 vertices. The main reason for this is that a cubic graph with at least 8 vertices always contains a cyclic edge cut (Lemma 1), which is not the case of cubic graphs on 2, 4 and 6 vertices (consider a triple edge, K_4 and $K_{3,3}$). This does not harm usability of our algorithms: There is only a single cubic graph on two vertices (a triple edge) and there are only three cubic graphs on four vertices (two triple edges, two double edges joined by a matching and a complete graph on four vertices). The only cubic graph on six vertices for which our algorithms do not work properly is $K_{3,3}$. These are only five cases and they can be easily checked in the very beginning of our algorithms.

2 Notation

In this section, we introduce notation and definitions used throughout the paper.

In this paper, we allow graphs to have parallel edges but all the considered graphs are loopless. We write $V(G)$ and $E(G)$ for the vertex set and the edge set of a graph G, respectively. $G[W]$ denotes the subgraph of G induced by the vertex set W, $W \subseteq V(G)$. A graph G is *cubic* if each vertex of it has a degree three.

An algorithmic procedure which is widely used in computer science and we use it several times throughout the paper is a procedure for *breadth-first search (BFS)* of a graph. You start at a vertex v_0 of G and label v_0 by zero. Then, label neighbors of v_0 by one (this is the first iteration). At iteration k, label unlabelled neighbors of labelled vertices by the number k. Continue until all the vertices of a graph are labelled. It is easy to see that the labels are distances of the vertices from v_0. We mean by a *BFS-graph* of depth d the graph induced by all the edges uv such that u is labelled by at most $d-1$ and v by at most d. Note that a BFS-graph of depth d need not to be a subgraph of G induced by the vertices of distance at most d from v_0 (the edges joining two vertices at the distance d are missing). If the BFS-graph is acyclic, we call it a *BFS-tree.* The vertex v_0 is said to be a *root* of the BFS-graph (BFS-tree). The vertices labelled with the number k form a *level* k of it. We also sometime abuse a little this notation and we root the BFS-graph at an edge. In such case, both the end-vertices of the edge are labelled by zero and the rest of the vertices are labelled in the manner described above.

We often use arguments related to cleverly chosen BFS-graphs to prove upper bounds on the girth of a graph. Observe the following simple facts: If the girth of G is at least $2k+1$, then a BFS-graph of depth k rooted at any vertex of G is acyclic. If the girth of G is at least $2k+2$, then the vertices of a BFS-graph of depth k rooted at any vertex of G induce an acyclic subgraph of G. We demonstrate the just presented concepts in the proof of the next lemma. The lemma was proven under a slightly different notation in [1], but we include its short proof for the sake of completeness.

Lemma 1 *Let G be a cubic graph of order n and girth g, $n \geq 8$. Then, $g \leq 2\lceil \log(n/3+1) \rceil$. Moreover, G contains a cyclic edge cut of size g.*

Proof: Assume that $g > 2\lceil \log(n/3+1) \rceil$ and let $d = \lceil \log(n/3+1)) \rceil$. Then, the BFS-graph of depth d rooted at any vertex of G is acyclic. Since G is cubic, such a BFS-tree of depth d contains $1+3+3\cdot 2+\ldots+3\cdot 2^{d-1} = 1+3(2^d-1)$ vertices. This is impossible because $1+3(2^d-1) > n$.

Consider a cycle of length g in G. The edges incident with the vertices of the cycle (which are not contained in the cycle) form a cut in G. We prove that both parts of the graph contain a cycle. The part consisting of a cycle of length g is obviously not acyclic. The other part of the graph contains $n-g$ vertices and $3n/2-2g$ edges. We prove that $3n/2-2g \geq n-g$. The existence of a cycle in this part of the graph then follows straightforwardly. The desired inequality is equivalent to $g \leq n/2$, but this inequality follows from the bound $g \leq 2\lceil \log(n/3+1) \rceil)$ and $n \geq 8$.

■

Edge cuts are understood in this paper as decompositions of a vertex set of G into two sets. Hence the cut is denoted by (A, B) where A and B are disjoint subsets of $V(G)$ such that $A \cup B = V(G)$. The cut itself is formed by the edges joining a vertex of A and a vertex of B and its size is equal to the number of such edges. An edge cut is said to be *cyclic* if both $G[A]$ and $G[B]$ contain a cycle.

We consider a problem of finding cyclic edge cuts in cubic graphs. The definition introduced in this paragraph and the lemma proven in the sequel give us a tool to prove that an edge cut is a cyclic edge cut. Let G' be a connected subgraph of G. Then the *activity* of G' is equal to $\sum_{v \in V(G')} (\deg_G v - \deg_{G'} v)$. Note that the activity is at least the number of edges incident with the vertices of $V(G')$ not included to G' but it might be larger (the edges not belonging to G' which join two vertices of G' are counted "twice"). A motivation for this definition is obvious from the following lemma:

Lemma 2 *Let G_1 and G_2 be vertex-disjoint connected subgraphs of a cubic graph G, each of activity at least $k+1$. Any edge cut separating G_1 and G_2 which is of size at most k is a cyclic edge cut.*

Proof: Let (A, B) be an edge cut of size at most k such that $V(G_1) \subseteq A$ and $V(G_2) \subseteq B$ and let E_{AB} be the edges joining A and B. We prove that $G[A]$ contain a cycle. The case of $G[B]$ is symmetric. If the subgraph of G induced by $V(G_1)$ is not acyclic, then the claim is trivial. Otherwise, G_1 is an induced subtree of G.

Assume now for the sake of contradiction that $G[A]$ is a forest. Let E_1 be the edges incident with G_1. Since G_1 is an induced subtree of G, the activity of G_1 is equal to $|E_1|$ and hence $|E_1| \geq k+1$. Then the number of leaves of $G[A]$ not contained in $V(G_1)$ must be at least $|E_1 \setminus E_{AB}| \geq k+1-|E_1 \cap E_{AB}|$. Each of the leaves of $G[A]$ which is not contained in $V(G_1)$ is incident with at least two edges of E_{AB}. Hence the size of E_{AB} has to be at least $2(k+1-|E_1 \cap E_{AB}|) + |E_1 \cap E_{AB}| = 2k+2-|E_1 \cap E_{AB}| \geq k+2$ which is impossible.

∎

The graphs for which we most often use Lemma 2 will be trees. A *binary tree* of depth d is a tree rooted at a vertex v with levels $0, 1, \ldots, d$ such that each vertex of levels $0, 1, \ldots, d-1$ has two children. The number of vertices of the last level is 2^d and the activity of a binary tree of depth d which is a subgraph of a cubic graph is $1 + 2^{d+1}$. A *full tree* of depth d is a tree rooted at a vertex v with levels $0, 1, \ldots, d$ such that the vertex v have three children and each vertex at a level between 1 and $d-1$ has two children. The number of vertices of the last level is $3 \cdot 2^{d-1}$ and the activity of a full tree of depth d which is a subgraph of a cubic graph is $3 \cdot 2^d$. We also consider special trees T_a and $T_{a,b}$ in the rest: A tree T_a is a tree rooted at a vertex with three children such that a out of its three children have two children. The remaining vertices of T_a are leaves. Hence T_a has $4+2a$ vertices. A tree $T_{a,b}$ is a tree rooted at a vertex with three children such that there are a vertices at level 1 with two children and b vertices at level 2 with two children. The remaining vertices of it are again leaves. Note that it must be $2a \geq b$. There might be several non-isomorphic trees $T_{a,b}$ for certain combinations of a and b, but we always use this notation for such $T_{a,b}$ which are determined by a and b. The number of vertices of $T_{a,b}$ is $4+2(a+b)$. Note also that $T_a = T_{a,0}$.

3 Structural Results

The following lemma was proved in [1] but we include its short proof for the sake of completeness:

Lemma 3 *Let G be a connected cubic graph with a cyclic edge connectivity κ and let (A, B) be cyclic edge cut of size κ. Then, both $G[A]$ and $G[B]$ are connected graphs with minimum degree two. The number of degree-two vertices in $G[A]$ or in $G[B]$ is exactly κ.*

Proof: $G[A]$ and $G[B]$ must be clearly connected graphs. Assume that G contains a vertex v incident with two edges of a cyclic edge cut (A, B) and $v \in A$. Consider a cut (A', B') such that $A' = A \setminus \{v\}$ and $B' = B \cup \{v\}$. Both $G[A']$ and $G[B']$ contain a cycle (v has degree one in $G[A]$ and hence it is contained in no cycle of $G[A]$ and $B \subseteq B'$) but the size of the cut (A', B') is smaller than the size of the cut (A, B) because G is cubic. Thus, if (A, B) is a cyclic edge cut of size κ, no vertex is incident with two edges of the cut. The lemma now follows straightforwardly.

■

Lemma 4 *Let H be a connected graph with n_2 vertices of degree two and n_3 vertices of degree three. If H does not contain a full tree of depth d, then H contains a cycle of length at most $2d$ or $n_3 \leq (2^d - 2)n_2$.*

Proof: If $n_3 = 0$, the lemma is clearly true. Otherwise, consider a vertex v of degree three in H. If the BFS-graph of depth d rooted at v is not acyclic, then H contains a cycle of length at most $2d$. Otherwise, the BFS-graph of depth d rooted at v is a tree, but it cannot be a full tree. Hence, there is a non-leaf vertex of degree two contained in it. Hence the distance between v and the nearest degree-two vertex is at most $d-1$. Since the choice of v was arbitrary, we may conclude that each vertex of degree three is at distance at most $d-1$ from a vertex of degree two. The number of neighbors at distance at most $d-1$ from a vertex of degree two is at most 2^d-2 for a fixed degree two vertex. The bound of $(2^d-2)n_2$ follows from the fact that the number of degree-two vertices is n_2.

■

Besides a general Lemma 4, we will need also the following two specialized lemmas:

Lemma 5 *Let H be a connected graph with n_2 vertices of degree two and n_3 vertices of degree three. If H does not contain a tree T_2, then H contains a cycle of length at most 4 or $n_3 \leq n_2$.*

Proof: Consider a vertex v of degree three adjacent to other two vertices v' and v'' of degree three. If v, v' and v'' together with their neighbors do not form a tree T_2, then H has a cycle of length at most 4. If H neither contains a tree T_2 nor a cycle of length at most 4, then each vertex of degree three is adjacent to at least two vertices of degree two. A simple counting argument yields the desired inequality $n_3 \leq n_2$.

■

Lemma 6 *Let H be a connected graph with n_2 vertices of degree two and n_3 vertices of degree three. If H does not contain a tree $T_{3,1}$, then H contains a cycle of length at most 7 or $n_3 \leq 8n_2/3$.*

Proof: By the girth assumption, the subgraph induced by the vertices of distance at most 3 from a vertex of degree three must be tree. Since H does not contain a tree $T_{3,1}$, each vertex of degree three is either adjacent to a vertex of degree two or all its neighbors have degrees three and all the neighbors of its neighbors have degree two. We use a simple discharging argument to prove the claim. Each vertex of degree two sends a single unit to its two neighbors and it sends $1/6$ of a unit to the four vertices at distance two from it. Each vertex of degree three receives at least a whole unit. Thus $n_3 \leq (2+4/6)n_2 = 8n_2/3$.

■

Theorem 1 *Let G be a cubic graph with a cyclic edge connectivity κ, $\kappa \geq 1$. Then at least one of the following holds:*

1. *G contains a cycle of length κ.*
2. *G contains a cyclic edge cut of size κ such that each of the two parts contains a full tree of depth $d = \lceil \log_2 \frac{\kappa+1}{3} \rceil$.*

Proof: Assume that G does not contain a cycle of length κ, i.e., the girth of G is at least $\kappa+1$. If none of cyclic edge cuts (A,B) of size κ has the property that both $G[A]$ and $G[B]$ contain the full tree, then consider a cyclic edge cut (A,B) of size κ such that A does not contain a full tree of depth d and the number of vertices of A is as small as possible. Let $H = G[A]$. Clearly, $H \neq C_\kappa$ and the girth of H (a subgraph of G) is at least $\kappa+1$. By Lemma 3, H is connected, its minimum degree is two and the number of degree-two vertices of H is exactly κ. Let $n_2 = \kappa$ be the number of vertices of degree two (in the rest, we use both n_2 and κ for this same number depending on which of the two quantities we emphasize) and n_3 the number of vertices of degree three of H.

We first deal with small values of κ:

- $\kappa = 1, 2$ and $d = 0$
  A 0-tree is a single vertex and hence the claim holds trivially.

- $\kappa = 3, 4, 5$ and $d = 1$
  In order to prove that H contains a 1-tree, it is enough to show that H contains a vertex of degree 3 because H contains no parallel edges due to the assumption on its girth. This follows trivially from the assumptions that H is not a cycle, it is connected and its minimum degree is two.

Let κ be at least six in the rest of this proof. Let $\overline{H}$ be the cubic graph obtained from H by suppressing all the vertices of degree two. If $\overline{H}$ is a triple edge, then one of the three edges corresponds to a path with at least $\kappa/3$ vertices of degree two. The paths corresponding to the remaining two edges of $\overline{H}$ form a cycle of length at most $2+2\kappa/3 \leq \kappa$ which is impossible because the girth of H is at least $\kappa+1$. Thus $n_3 \geq 4$ (n_3 must be a non-zero even number greater than two).

We show in this paragraph that H does not contain two adjacent vertices of degree two. Assume that u and v are two adjacent degree-two vertices in H. Let uu' and vv' be the edges of the cut in G incident with u and v. Let u'' and v'' be the neighbors of u and v respectively different from u, u', v and v'. Consider the cut (A', B') where the edges uu' and vv' are replaced by the edges uu'' and vv''. Obviously, $A' = A \setminus \{u, v\}$, $B \subseteq B'$ and thus $G[B']$ contains a cycle. The graph $G[A']$ is the graph $G[A] \setminus \{u, v\}$. Hence it contains at most two vertices of degree one (these might be the vertices u'' and v'' if their degrees are two in $G[A]$), at least two vertices of degree three (this follows form $n_3 \geq 4$) and the remaining vertices have degrees equal to two. Thus $G[A']$ is not acyclic and (A', B') is a cyclic edge cut of size κ. But this contradicts our assumption that the number of vertices of A is the least possible.

We now show that $n_3 \geq 6$. If $n_3 = 4$, then $\overline{H}$ contains a cycle of length two or three. This cycle corresponds to a cycle of length at most six in H which is impossible because the girth of H is at least $\kappa + 1 \geq 7$. Similarly, $\overline{H}$ must be triangle-free. Note that if $n_3 = 6$, then the fact that $\overline{H}$ is triangle-free implies $\overline{H} = K_{3,3}$. We can conclude that $n_3 \geq 8$ unless $\overline{H} = K_{3,3}$.

We also show that H does not contain a vertex of degree three adjacent to three vertices of degree two. Assume the opposite. Let v be a vertex of degree three, x, y and z the neighbors of v of degree two and x', y' and z' the other neighbors of x, y and z in H, respectively. Note that all the vertices v, x, y, z, x', y' and z' are mutually different because of the girth assumption. Let xx'', yy'' and zz'' be the edges of the cut incident with the vertices x, y and z. Consider the (A', B') where the edges xx'', yy'' and zz'' are replaced by the edges xx', yy' and zz'. Obviously, $A' = A \setminus \{v, x, y, z\}$, $B \subseteq B'$ and thus $G[B']$ contains a cycle. The minimum degree of $G[A']$ is two because the vertices x', y' and z' are of degree three in H. Thus $G[A']$ cannot be acyclic. We conclude that (A', B') is a cyclic edge cut of size of κ which is impossible because of the choice of A.

We are now ready to deal with the remaining cases:

- $\kappa = 6, 7$ and $d = 2$
  Assume first that H does not contain a tree T_2. Then $n_3 \leq 7$ by Lemma 5. This implies that $\overline{H} = K_{3,3}$. Consider first the case $\kappa = 6$. If $\overline{H}$ contains a vertex incident with two edges which do not correspond

to paths with degree-two vertices, then a four-cycle of $\overline{H}$ containing these two edges correspond to a cycle of length at most six in H. Otherwise, the three edges which do not correspond to paths with degree-two vertices form a matching in $\overline{H} = K_{3,3}$. There is also a four-cycle of $\overline{H}$ containing two edges (of the matching) not corresponding to paths of degree-two vertices in this case. This cycle corresponds to a cycle of length at most six in H. Consider now the remaining case $\kappa = 7$. $\overline{H}$ contains an edge not corresponding to a path with degree-two vertices and thus there is a four-cycle in $\overline{H}$ with such an edge. This cycle corresponds to a cycle of length at most seven in H. In either of the cases, we have obtained a contradiction with the girth assumption on H.

Assume next that H does not contain a full tree of the depth two but it contains a tree T_2. Then $n_3 \leq 2n_2$ by Lemma 4. Let $w_1, w_2, \dots, w_\kappa$ be the vertices of degree two of H and let H_3 be a subgraph of H induced by the vertices of degree three in H (note that $H_3 = H \setminus \{w_1, w_2, \dots, w_\kappa\}$). Since H does not contain a full tree of the depth two, the maximum degree of H_3 is two. On the other hand, its minimum degree is one. We further distinguish several cases:

- H_3 is 2-regular.
  H_3 must be a single cycle of length 2κ, otherwise one of the components of H_3 is a cycle of length at most κ. Let $v_1, v_2, \dots, v_{2\kappa}$ be the vertices of H_3 in the cyclic order. No vertices of the cycle at distance at most $\kappa - 2$ cannot be adjacent to the same vertex w_i in H because of the girth assumption. Hence, we may assume that v_i is adjacent to w_i for $1 \leq i \leq \kappa - 1$. The vertex v_κ can be adjacent only either to w_1 or to w_κ; similarly the vertex $v_{2\kappa}$ can be adjacent only either to $w_{\kappa-1}$ or w_κ.

  If v_κ is adjacent to w_1 and $v_{2\kappa}$ to $w_{\kappa-1}$, then H contains a 6-cycle $v_1 w_1 v_\kappa v_{\kappa-1} w_{\kappa-1} v_{2\kappa}$ (which is not possible). If v_κ is adjacent to w_1 and $v_{2\kappa}$ to w_κ, then the other vertex among v_i adjacent to w_κ must be the vertex $v_{\kappa+1}$ (otherwise the distance between the vertices adjacent to w_κ is less than $\kappa - 1$). But then H contains a 6-cycle $v_1 w_1 v_\kappa v_{\kappa+1} w_\kappa v_{2\kappa}$ (which is impossible). We may conclude that v_κ is adjacent to w_κ.

  By symmetry, any κ consecutive vertices of the cycle are adjacent to mutually different vertices w_i. In particular, $v_{\kappa+1}$ must be adjacent to w_1 and $v_{\kappa+2}$ to w_2. Then, H contains a 6-cycle

$v_1, v_2, w_2, v_{\kappa+2}, v_{\kappa+1}, w_1$ which is again impossible because of the girth assumption.

- H_3 consists of cycles and paths.
  H_3 contains exactly a single cycle (H_3 has less than 2κ vertices and the length of a cycle of H_3 must be at least $\kappa + 1$). The cycle of H_3 contains two vertices adjacent to the same vertex w_i, $1 \le i \le \kappa$ because its length is at least $\kappa + 1$. Now, the vertex w_i with its two neighbors and the shorter arc of the cycle comprises a cycle of length at most κ which is impossible.
- H_3 consists of a single path.
  Let $v_1, v_2, \ldots, v_{2\kappa-2}$ be the path of H_3. No vertices of the path at distance at most $\kappa - 2$ can be adjacent to the same vertex w_i in H because of the girth assumption. Hence, we may assume v_1 is adjacent to w_1 and w_2, v_i is adjacent to w_{i+1} for $2 \le i \le \kappa - 1$ and v_κ is adjacent to w_1. The vertex $v_{\kappa+1}$ can be adjacent to w_2 or w_3. In the first case, the graph H contains a cycle $v_1, w_1, v_\kappa, v_{\kappa+1}, w_2$. In the second case, the graph H contains a cycle $v_1, v_2, w_3, v_{\kappa+1}, v_\kappa, w_1$. In either of the cases, the found cycle contradicts the girth assumption on H.
- H_3 consists of several paths and $\kappa = 6$.
  If H_3 consists of three or more paths, then the degrees of all the vertices in H_3 are one, but then H does not contain a tree T_2. Assume hence that H_3 consists of two paths. Thus $n_3 = 2\kappa - 4 \le 8$. Consider a vertex of degree three adjacent to at least two other vertices of degree three in H (this corresponds to a copy T_2 in H). Since the girth of H is at least 7, the BFS-graph of depth 3 rooted at v is acyclic. The numbers of vertices at levels $0, 1, 2, 3$ respectively in the BFS-tree rooted at v must be at least $1, 3, 5, 6$ respectively because H contains no two adjacent degree-two vertices. But this is impossible because $1 + 3 + 5 + 6 = 15$ but $n_2 + n_3 = 14$.
- H_3 consists of several paths and $\kappa = 7$.
  Then $n_3 \le 2\kappa - 4 \le 10$. Consider a vertex of degree three adjacent to at least two other vertices of degree three in H (this corresponds to a copy T_2 in H). Since the girth of H is at least 8, the vertices at distance at most three from v induce an acyclic subgraph of H. The numbers of vertices at levels $0, 1, 2, 3$ respectively in the BFS-tree rooted at v must be at least $1, 3, 5, 6$

respectively because H contains no two adjacent degree-two vertices. Hence there are at most two vertices at distance at least four from v. Moreover, either the level 3 of the BFS-tree rooted at v contains seven vertices or at least one of the vertices of the level 3 is of degree three. In either of the cases, there are at least 7 edges joining the vertices of the level 3 to the vertices at distance at least four from v. But this is impossible because there are at most two such vertices and each of them has degree at most three.

In either of the cases, we found a cycle of length κ in H.

- $\kappa = 8$ and $d = 2$
  Assume first that H does not contain a tree T_2, then $n_3 \leq 8$ by Lemma 5. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 4 is acyclic because the girth of H is at least 9. The numbers of vertices at levels $0, 1, 2, 3, 4$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 7$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 3 + 6 + 7 = 20$ but $n_2 + n_3 \leq 16$.

  Assume next that H does not contain a full tree of depth two but it contains a tree T_2. By Lemma 4, $n_3 \leq 16$. Consider a vertex v of degree three adjacent to at least two degree-three vertices in H. Then, the BFS-graph of depth 4 is acyclic because the girth of H is at least 9. The numbers of vertices at levels $0, 1, 2, 3, 4$ respectively in the BFS-tree rooted at v must be at least $1, 3, 5, 6, 10$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 5 + 6 + 10 = 25$ but $n_2 + n_3 \leq 24$.

- $\kappa = 9$ and $d = 2$
  Assume first that H does not contain a tree T_2, then $n_3 \leq 9$ by Lemma 5. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 4 is acyclic because the girth of H is at least 10 (even the graph induced by these vertices is a tree). The numbers of vertices at levels $0, 1, 2, 3, 4$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 6$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 3 + 6 + 6 = 19$ but $n_2 + n_3 \leq 18$.

  Assume next that H does not contain a full tree of depth two but it contains a tree T_2. By Lemma 4, $n_3 \leq 18$. Consider a vertex v

of degree three adjacent to at least two degree-three vertices in H. Then, the vertices at distance at most four from v induce a tree in H because the girth of H is at least 10. The numbers of vertices at levels $0, 1, 2, 3, 4$ respectively in the BFS-tree rooted at v must be at least $1, 3, 5, 6, 10$ respectively because H contains no two adjacent degree-two vertices. Hence there are at most 2 vertices at distance five or more from v, but the number of edges between the vertices at distance four and at distance five must be at least 10 (the number of the vertices at distance four). This is impossible because the maximum degree of H is three.

- $\kappa = 10$ and $d = 2$
  If H does not contain a full tree of depth two, then $n_3 \leq 20$ by Lemma 4. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 5 rooted at v is acyclic because the girth of H is at least 11. The numbers of vertices at levels $0, 1, 2, 3, 4, 5$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 6, 12$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 3 + 6 + 6 + 12 = 31$ but $n_2 + n_3 \leq 30$.

- $\kappa = 11$ and $d = 2$
  If H does not contain a full tree of depth two, then $n_3 \leq 22$ by Lemma 4. Consider a vertex v of degree three in H. Then, the graph induced by the vertices at distance at most five from v is acyclic because the girth of H is at least 12. The numbers of vertices at levels $0, 1, 2, 3, 4, 5$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 6, 12$ respectively because H contains no two adjacent degree-two vertices. Hence there are at most two vertices at distance six from v, but the number of edges between the vertices at distance five and at distance six must be at least 12 (the number of the vertices at distance five). This is impossible because the maximum degree of H is three.

- $\kappa = 12$ and $d = 3$
  Assume first that H does not contain a tree $T_{3,1}$. By Lemma 6, $n_3 \leq 32$. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 6 is acyclic because the girth of H is at least 13. The numbers of vertices at levels $0, 1, 2, 3, 4, 5, 6$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 6, 12, 21$ respectively because

H contains no two adjacent degree-two vertices. This is impossible because $1+3+3+6+6+12+21 = 52$ but $n_2+n_3 \leq 44$.

Assume next that H does not contain a full tree of depth three but it contains a tree $T_{3,1}$. By Lemma 4, $n_3 \leq 72$. Consider a vertex v of degree three adjacent to three degree-three vertices such that one of its neighbors is adjacent to at least two degree-three vertices (the center of $T_{3,1}$). Then, the BFS-graph of depth 6 is acyclic because the girth of H is at least 13. The numbers of vertices at levels $0,1,2,3,4,5,6$ respectively in the BFS-tree rooted at v must be at least $1,3,6,7,12,19,38$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1+3+6+7+12+19+38 = 86$ but $n_2+n_3 \leq 84$.

- $\kappa = 13$ and $d = 3$
  Assume first that H does not contain a full tree of depth two, then $n_3 \leq 26$ by Lemma 4. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 6 is acyclic because the girth of H is at least 14 (even the graph induced by these vertices is a tree). The numbers of vertices at levels $0,1,2,3,4,5,6$ respectively in the BFS-tree rooted at v must be at least $1,3,3,6,6,12,20$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1+3+3+6+6+12+20 = 51$ but $n_2+n_3 \leq 39$.

  Assume next that H does not contain a full tree of depth three but it contains a full tree of depth two. By Lemma 4, $n_3 \leq 78$. Consider a vertex v of degree three adjacent to three degree-three vertices in H. Then, the vertices at distance at most six from v induce a tree in H because the girth of H is at least 14. The numbers of vertices at levels $0,1,2,3,4,5,6$ respectively in the BFS-tree rooted at v must be at least $1,3,6,6,12,17,34$ respectively because H contains no two adjacent degree-two vertices. Hence there are at most 12 vertices at distance seven or more from v. The number of vertices at distance six from v is at least $34+l$ where l is the number of vertices of degree two at distance six from v. Then, there are at least 68 edges (two edges from each degree-three vertex) joining the vertices at distance six from v to the vertices at distance seven or more, but this is impossible because there are at most 12 such vertices and their degrees are at most three.

- $\kappa = 14, 15$ and $d = 3$
  Assume first that H does not contain a full tree of depth two, then

$n_3 \leq 30$ by Lemma 4. Consider a vertex v of degree three in H. Then, the BFS-graph of depth 7 is acyclic because the girth of H is at least 15. The numbers of vertices at levels $0, 1, 2, 3, 4, 5, 6, 7$ respectively in the BFS-tree rooted at v must be at least $1, 3, 3, 6, 6, 12, 19, 38$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1+3+3+6+6+12+19+38 = 88$ but $n_2+n_3 \leq 45$.

Assume next that H does not contain a full tree of depth three but it contains a full tree of depth two. By Lemma 4, $n_3 \leq 90$. Consider a vertex v of degree three adjacent to three degree-three vertices in H. Then, the BFS-graph of depth 7 is acyclic because the girth of H is at least 15. The numbers of vertices at levels $0, 1, 2, 3, 4, 5, 6, 7$ respectively in the BFS-tree rooted at v must be at least $1, 3, 6, 6, 12, 16, 32, 64$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 6 + 6 + 12 + 16 + 32 + 64 = 140$ but $n_2 + n_3 \leq 105$.

- $16 \leq \kappa \leq 18$ and $d = 3$
  If H does not contain a full tree of depth three, then $n_3 \leq 6\kappa \leq 108$ due to Lemma 4. The BFS-graph of depth 8 rooted at any vertex of degree three is acyclic because the girth of H is at least 17. The numbers of vertices of levels $0, 1, 2, 3, 4, 5, 6, 7, 8$ respectively in the BFS-tree of depth 8 rooted at a vertex of degree three must be at least $1, 3, 3, 6, 6, 12, 15, 30, 60$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1 + 3 + 3 + 6 + 6 + 12 + 15 + 30 + 60 = 136$ but $n_2 + n_3 \leq 126$.

- $19 \leq \kappa \leq 23$ and $d = 3$
  If H does not contain a full tree of depth three, then $n_3 \leq 6\kappa \leq 138$ due to Lemma 4. The BFS-graph of depth 9 rooted at any vertex of degree three is acyclic because the girth of H is at least 20 (even the subgraph induced by its vertices is a tree). The numbers of vertices of levels $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ respectively in the BFS-tree of depth 9 rooted at a vertex of degree three must be at least $1, 3, 3, 6, 6, 12, 12, 24, 46, 92$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1+3+3+6+6+12+12+24+46+92 = 205$ but $n_2 + n_3 \leq 161$.

- $24 \leq \kappa \leq 25$ and $d = 4$
  If H does not contain a full tree of depth four, then $n_3 \leq 14\kappa \leq 350$ due to Lemma 4. The BFS-graph of depth 11 rooted at any vertex of

degree three is acyclic because the girth of H is at least 23 (even the subgraph induced by its vertices is a tree because the girth is actually larger). The numbers of vertices of levels $0,1,2,3,4,5,6,7,8,9,10,11$ respectively in the BFS-tree of depth 11 rooted at a vertex of degree three must be at least $1,3,3,6,6,12,12,24,44,88,176,352$ respectively because H contains no two adjacent degree-two vertices. This is impossible because $1+3+3+6+6+12+12+24+44+88+176+352=727$ but $n_2+n_3 \leq 375$.

- $\kappa \geq 26$
We first bound the number of vertices of H. If H does not contain a full tree of depth d, then $n_3 \leq (2^d-2)n_2$ by Lemma 4. We bound the number of vertices of H as follows:

$$n_2+n_3 \leq (2^d-1)\kappa \leq \left(2^{\log_2 \frac{\kappa+1}{3}+1}-1\right)\kappa \leq \left(2\cdot\frac{\kappa+1}{3}-1\right)\kappa \leq \frac{2\kappa^2}{3}$$

Let $l=\lfloor\kappa/2\rfloor$. Since the girth of H is at least $\kappa+1$, the BFS-graph of depth l rooted at any vertex of degree three is acyclic. The number of vertices at level i, $1\leq i\leq l$, of this BFS-tree is at least $3\cdot 2^{\lfloor\frac{i-1}{2}\rfloor}$ because H contains no two adjacent vertices of degree two. Hence the BFS-tree contains at least the following number of vertices:

$$1+3\cdot 2^0+3\cdot 2^0+3\cdot 2^1+3\cdot 2^1+3\cdot 2^2+\ldots+3\cdot 2^{\lfloor\frac{l-1}{2}\rfloor}=$$
$$1+3\cdot 2^0+3\cdot 2^1+\ldots+3\cdot 2^{\lfloor\frac{l-2}{2}\rfloor}+3\cdot 2^0+3\cdot 2^1+\ldots 3\cdot+2^{\lfloor\frac{l-1}{2}\rfloor}=$$
$$1+3\left(2^{\lfloor l/2\rfloor}-1\right)+3\left(2^{\lceil l/2\rceil}-1\right)=3\cdot 2^{\lfloor l/2\rfloor}+3\cdot 2^{\lceil l/2\rceil}-5$$

Since l is integer, we further bound this number of vertices as follows:

$$3\cdot 2^{\lfloor l/2\rfloor}+3\cdot 2^{\lceil l/2\rceil}-5 \geq 3\cdot 2\cdot 2^{l/2}-5=6\cdot 2^{l/2}-5$$

We get using the inequality $l=\lfloor\kappa/2\rfloor\geq\kappa/2-1/2$ the following:

$$6\cdot 2^{l/2}-5\geq 6\cdot 2^{(\kappa-1)/4}-5$$

We may conclude that H must contain at least $6\cdot 2^{(\kappa-1)/4}-5$ vertices. But H contains at most $2\kappa^2/3$ vertices as proved above. This is impossible because $2\kappa^2/3<6\cdot 2^{(\kappa-1)/4}-5$ for $\kappa\geq 26$.

■

4 An $O(n^3 \log n)$-Algorithm

We are now ready to present our first algorithm for the cyclic edge-connectivity of cubic graphs:

Theorem 2 *There is an algorithm for computing cyclic edge connectivity of cubic graphs running in time $O(n^3 \log n)$.*

Proof: A pseudocode of the algorithm can be found in Figure 1. We assume that the number of vertices of the input graph G is at least 8. By Lemma 1, such a graph G always contains a cyclic edge cut. The cases when G has 2, 4 or 6 vertices can be easily handled separately as explained at the end of Section 1.

First, we find the girth g of the input graph G. This can be straightforwardly done in time $O(n^2)$ by running a BFS routine from each of the vertices of G. The girth g which is of order $O(\log n)$ (Lemma 1) is an upper bound on the cyclic edge connectivity of G. The algorithm computes the minimum edge separation between the full trees of depth d rooted at v and at w (if such two trees are vertex-disjoint) for all the pairs v and w of the vertices of G and all the values of $0 \leq d \leq \lceil \log_2 k/3 \rceil$ for k which is the size of the cyclic edge cut found so far (initially $k = g$). If the size of the edge cut is smaller than $3 \cdot 2^d$, then a found edge cut is also a cyclic edge cut by Lemma 2 and this edge cut is a new upper bound on the size of the minimum cyclic edge cut (if smaller than the cyclic edge cut found so far).

We use a simple flow algorithm to find the edge cut for each pair of full trees. We either find $3 \cdot 2^d$ edge-disjoint paths between the full trees or the edge cut of size smaller than $3 \cdot 2^d$: The algorithm at each iteration either augments the flow (increase the number of edge-disjoint paths) between the trees or finds an edge cut of the size equal to the number of the paths. The number of iterations is bounded by $3 \cdot 2^d$. Each of the iterations requires time which is linear in the number of edges of G, i.e., $O(n)$. For a fixed pair of vertices, we have to run at most $3 \cdot 2^0 + 3 \cdot 2^1 + 3 \cdot 2^2 + \ldots 3 \cdot 2^{\lceil \log_2 g/3 \rceil}$ iterations, hence the number of iterations of the flow algorithm for all the $O(\log g)$ full trees for a fixed pair of vertices is at most $O(g)$. Hence, the algorithm runs in time $O(n^3 g) = O(n^3 \log n)$. We may further improve a running time of the algorithm by using the edge-disjoint paths found between the full trees of depth d rooted at v and w as a starting set of paths between the full trees of depth $d + 1$ rooted at v and w.

The correctness of the algorithm follows from Theorem 1. Let κ be the cyclic edge connectivity of G. If $\kappa = 0$, then G is disconnected and the

Input: a cubic graph G of order at least 8
Output: a cyclic edge cut of the minimum size

```
cutsize := girth(G)
cut := edges incident with a cycle of length cutsize
for v ∈ V(G) do
   for w ∈ V(G) do
      d := -1
      paths := ∅
      repeat
         d := d + 1
         Tv := a full tree of depth d rooted at v
         Tw := a full tree of depth d rooted at w
         if Tv and Tw are not vertex-disjoint then break
         paths := findpaths (Tv, Tw, paths)
         if |paths| < 3 * 2^d and |paths| < cutsize then
            cutsize = |paths|
            cut = findcut (Tv, Tw, paths)
         fi
      while 3 * 2^d < cutsize
   endfor
endfor
output cutsize and cut
```

Figure 1: The algorithm for the cyclic edge connectivity of cubic graphs running in time $O(n^3 \log n)$. The subroutines `findpaths` and `findcut` are described in the last paragraph of the proof of Theorem 2.

algorithm clearly works correctly. If there is a cycle of length κ in G, then the minimum cyclic edge cut is equal to the girth of G and we find it in the very beginning of the algorithm. Otherwise, there exists an edge cut (A, B) of size κ such that both $G[A]$ and $G[B]$ contain full trees of depth $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ due to Theorem 1. At a certain step of the algorithm, the minimum edge separation between these two full trees was computed and its size was at most κ (the cyclic edge cut is one of edge cuts between A and B). Since the activity of a full tree of depth $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ is at least $\kappa + 1$, we found a cyclic edge cut of size κ (any cut of size κ between these full trees is cyclic by Lemma 2).

Let us say a few comments to the implementation of the algorithm in Figure 1. The algorithm uses the following subroutines: `findpaths(A,B,S)` and `findcut(A,B,S)`. The subroutine `findpaths(A,B,S)` finds the largest number of edge-disjoint paths between the vertex disjoint subgraphs A and B augmenting a provided set S of edge-disjoint paths between the subgraphs A and B. The running time of this subroutine is $O((k + 1 - k_0)n)$ where k is the number of edge-disjoint paths between A and B, k_0 is the number of paths of S and n is the number of vertices of G. The subroutine `findcut(A,B,S)` finds the edge cut between the vertex disjoint subgraphs A and B if S is a set of the largest number of edge-disjoint paths between A and B. The running time of this subroutine is $O(n)$.

■

5 An $O(n^2 \log^2 n)$-Algorithm

We first prove that if the girth or the order of a cubic graph is sufficiently large, then one can construct a large number of edge-disjoint subgraphs of a certain type in an algorithmic way:

Lemma 7 *Let G be a cubic graph of order at least 243 and girth at least five. Then G contains 12 edge-disjoint full trees of depth two. Moreover, such 12 trees can be found in time $O(n)$ where n is the number of vertices of G.*

Proof: Since the girth of G is at least five, the BFS-graph of depth two rooted at any vertex of G must be a full tree (of depth two). We find 12 vertices of G such that the distance between any pair of them is at least four. The BFS-trees of depth two rooted at such vertices are edge-disjoint and as argued above they are actually full trees of depth two.

Take an arbitrary vertex of G and mark this vertex together with all the vertices at distance at most three from it. Then take an unmarked vertex, mark it and mark also all the vertices at distances at most three from it. At each step, at most $1+3+6+12=22$ vertices get marked. Since G has at least $243 > 22*11$ vertices, we definitely find at least 12 such vertices. The just described greedy algorithm can be easily implemented in time $O(n)$.

■

Lemma 8 *Let G be a cubic graph of girth at least g, $g \geq 13$. Then G contains at least g edge-disjoint binary trees of depth $\lceil \log_2 \frac{g-1}{2} \rceil$. Moreover, such g trees can be found in time $O(n)$ where n is the number of vertices of G.*

Proof: Let $d = \lceil \log_2 \frac{g-1}{2} \rceil$ and $D = \lfloor \frac{g-1}{2} \rfloor$. Consider any edge e of the graph G. Consider the BFS-graph rooted at e of the depth D. Due to the girth assumption, the vertices of the levels $0,\ldots,D-1$ induce a tree in G. Consider the following binary trees: Let $l = \lfloor D/d \rfloor$. Take two binary trees of the depth d rooted at the two vertices of the level 0; these trees contain vertices of levels $0,\ldots,d$. Take 2^{d+1} binary trees of the depth d rooted at 2^{d+1} vertices of the level d; these trees contain vertices of levels $d,\ldots,2d$. Proceed in this manner upto the trees rooted at vertices of the level $d(l-1)$. All the constructed trees are edge-disjoint. Their number is equal to $2+2^{d+1}+\ldots+2^{d(l-1)+1} = 2\frac{2^{dl}-1}{2^d-1}$.

We check that $2\frac{2^{dl}-1}{2^d-1} \geq g$ if $g \geq 13$. If $13 \leq g \leq 18$, then $d=3$, $l=2$ and hence the number of the trees is $2\frac{2^6-1}{2^3-1} = 18$. If $19 \leq g \leq 24$, then $d=4$, $l=2$ and hence the number of the trees is $2\frac{2^8-1}{2^4-1} = 34$. Finally, if $25 \leq g \leq 26$, then $d=4$, $l=3$ and hence the number of the trees is $2\frac{2^12-1}{2^4-1} = 546$. We proceed as follows for the remaining values of g:

$$2\frac{2^{dl}-1}{2^d-1} \geq 2\frac{2^{(D/d-1)d}-1}{\frac{g-1}{-}1} = \frac{2^{D-d}-1}{g-2} \geq \frac{2^{D+1}/(g+1)-1}{g-2} =$$

$$2\frac{2^{D+1}-g-1}{g^2-g-2} \geq 2\frac{2^{g/2}-g-1}{g^2-g-2}$$

The obtained expression is greater than g if $g \geq 27$.

The trees can be clearly found algorithmically in time $O(n)$. Simply take any edge e of G, run the BFS routine and output g binary trees of depth d constructed in the above way.

■

We are now ready to present our algorithm running in time $O(n^2 \log^2 n)$:

Theorem 3 *There is an algorithm for computing cyclic edge connectivity of cubic graphs running in time $O(n^2 \log^2 n)$.*

Proof: A pseudocode of the algorithm can be found in Figure 2 (the subroutines `findpaths` and `findcut` are described at the end of the proof of Theorem 2). Let us briefly explain the algorithm: If the number of vertices of the input cubic graph G is smaller than 243, we run the algorithm of Theorem 2. Otherwise, we compute the girth g of G. This can be straightforwardly done in time $O(n^2)$ by running a BFS routine from each of the vertices of G. We create a set A_0 of g edge-disjoint subgraphs of G:

- If $2 \leq g \leq 4$, then A_0 consists of any g different edges of G.
- If $5 \leq g \leq 12$, then A_0 consists of g edge-disjoint full trees of depth 2. Such a set A_0 can be constructed in time $O(n)$ due to Lemma 7.
- If $13 \leq g$, then A_0 consists of g edge-disjoint binary trees of depth $\lceil \log_2 \frac{g-1}{2} \rceil$. Such a set A_0 can be constructed in time $O(n)$ due to Lemma 8.

Note that the activity of any subgraph contained in A_0 is at least g.

Our algorithm computes a minimum edge separation for all pairs of a subgraph $A \in A_0$ and a full tree rooted at $v \in V(G)$ of depth d for $0 \leq d \leq \lceil \log_2 k/3 \rceil$ for k which is the size of the cyclic edge cut found so far (initially $k = g$). As in the algorithm of Theorem 2, we use a simple flow algorithm to compute edge-disjoint paths between a full tree and a subgraph A. We also use the paths between the full tree of depth d and A as an initial set of paths between the full tree of depth $d+1$ and A. If the found edge cut is smaller than $3 \cdot 2^d$ (the activity of the full tree of depth d) and than g, the edge cut is cyclic (by Lemma 2) and if it smaller than the cyclic edge cut found so far, we have a new upper bound on the cyclic edge cut.

The number of iterations for a fixed pair of a subgraph A and a vertex v is $O(g) = O(\log n)$ for all the full trees rooted at v together. Each iteration

Input: a cubic graph G of order at least 8
Output: a cyclic edge cut of the minimum size

```
if |V(G)| < 243 then run O(n^3 log n)-algorithm
g := girth(G)
cutsize := g
cut := edges incident with a cycle of length cutsize
switch g of
   case 2,3,4:
      A_0 := g different edges
   case 5,6,7,8,9,10,11,12:
      A_0 := g edge-disjoint full trees of depth 2
   default:
      A_0 := g edge-disjoint binary trees of depth ⌈log_2 (g-1)/2⌉
endswitch
for v ∈ V(G) do
   for A ∈ A_0 do
      d := -1
      paths := ∅
         repeat
         d := d + 1
         T := a full tree of depth d rooted at v
         if T and A are not vertex-disjoint then break
         paths := findpaths (T, A, paths)
         if |paths| < 3 * 2^d and |paths| < cutsize then
            cutsize = |paths|
            cut = findcut (T, A, paths)
         fi
      while 3 * 2^d < cutsize
   endfor
endfor
output cutsize and cut
```

Figure 2: The algorithm for the cyclic edge connectivity of cubic graphs running in time $O(n^2 \log^2 n)$. The subroutines `findpaths` and `findcut` are described in the last paragraph of the proof of Theorem 2.

takes time $O(n)$. The number of subgraphs in A_0 is $g = O(\log n)$ and thus the running time of the whole algorithm is $O(n^2 \log^2 n)$.

We prove the correctness of our algorithm: Let κ be the size of the smallest cyclic edge cut of G. If $\kappa = 0$, then G is disconnected and the algorithm clearly finds the empty cyclic edge cut. If $\kappa = g$, then the cyclic edge cut of size g is found in the first phase of the algorithm. If $\kappa < g$, then there is a cyclic edge cut (B, C) such that both $G[B]$ and $G[C]$ contain full trees of depth $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ due to Theorem 1. Besides this, one of the graphs $G[B]$ and $G[C]$ contains a subgraph $A \in A_0$ because one of the subgraphs of A_0 does not contain an edge of the cut (the subgraphs of A_0 are edge-disjoint, their number is g and the size of the cyclic edge cut is $\kappa < g$). Assume that $G[B]$ does. Now, $G[B]$ contains a subgraph A from the set A_0 and $G[C]$ contains a full tree of depth $\lceil \log_2 \frac{\kappa+1}{3} \rceil$. At the step when we consider the pair consisting of the subgraph A and a full tree of depth $\lceil \log_2 \frac{\kappa+1}{3} \rceil$ in $G[C]$, we found a cyclic edge cut of size κ (we found an edge cut of size at most κ and such an edge cut is cyclic by Lemma 2).

■

Acknowledgement

Our attention was attracted to the problem of designing an efficient algorithm for computing cyclic edge connectivity of cubic graphs by Roman Nedela from Matej Bel University in Banska Bystrica, Slovakia. This research was initially started when the authors took part in the DIMACS/DIMATIA Research Experience for Undergraduates programme in 2001 under a supervision of Jeff Kahn from Rutgers University. This programme is a joint project of DIMACS at Rutgers University and DIMATIA at Charles University. We temporarily stopped working on this problem when the programme had been finished and we restarted our work at the end of the year 2002.

References

[1] D. Lou, L. Teng, X. Wu: A polynomial algorithm for cyclic edge connectivity of cubic graphs, Australasian Journal of Combinatorics **24** (2001), 247–259.

[2] R. Nedela, M. Škoviera: Atoms of cyclic connectivity in cubic graphs, Math. Slovace **45** (1995), 481–499.

[3] R. Nedela, M. Škoviera: Decompositions and reductions of snarks, J. Graph Theory **22** (1996), 253–279.

[4] B. Peroche: On Several Sorts of Connectivity, Discrete Mathematics **46** (1983), 267–277.

[5] M. D. Plummer: On the cyclic connectivity of planar graphs, in: Y. Alavi, D. R. Lick and A. T. White (eds.), Graph Theory and Applications, Springer-Verlag, Berlin (1972), 235–242.

[6] P. G. Tait, Remarks on the colouring of maps, Proc. Roy. Soc. Edingburg **10** (1880), 501–503.