ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

A. Galluccio, G. Proietti

TOWARDS A 4/3–APPROXIMATION ALGORITHM FOR BICONNECTIVITY

R. 506 Giugno 1999

Anna Galluccio - Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: galluccio@iasi.rm.cnr.it.

Guido Proietti - Dipartimento di Matematica Pura ed Applicata, Università di L'Aquila, Via Vetoio, 67010 L'Aquila, Italy. E-mail: proietti@univaq.it.

Istituto di Analisi dei Sistemi ed Informatica, CNR
viale Manzoni 30
00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

Finding a minimum size 2-vertex connected spanning subgraph of a graph G with n vertices and m edges is known to be NP-hard even if a Hamiltonian path of G is given as part of the input. In this paper we propose an $O(n+m)$ time and space algorithm which approximates the optimal solution for the above problem by a factor of no more than 4/3. The best known algorithm for the case in which a Hamiltonian path is not given is due to Garg et al., and has an approximation guarantee of 3/2. The ratio of their algorithm does not decrease when it is applied to the special case in which a Hamiltonian path is given as part of the input.

1. Introduction

The problem of finding a minimum size 2-vertex connected (simply biconnected, in the following) spanning subgraph of a biconnected graph G is one of the classical problems in computer science and combinatorial optimization. It is known to be *NP*-hard, since its decision version contains as a special case the *Hamiltonian cycle* problem (i.e., the problem of deciding whether a graph G contains a simple cycle that includes all the vertices), which is well-known to be *NP*-complete [3]. Due to its relevance and to the great number of applications it finds in different fields, several approximation algorithms for solving this problem have been devised in the past few years. Khuller and Vishkin [8] introduced the notions of *carving* of a graph to establish approximation factors of no more than 5/3. Their algorithm has been improved by Garg et al. [4], who lowered the approximation ratio to 3/2. For an exhaustive survey on vertex-connectivity problems, the interested reader can refer to [7].

A question which naturally arises is that of studying whether the approximation guarantee can be improved once the input of the problem is enriched. In 1976, Papadimitriou and Steiglitz [9] proved that the problem of determining whether a graph contains a Hamiltonian cycle remains *NP*-complete even if a Hamiltonian path is given as part of the input. It follows that the problem of determining whether a graph admits a biconnected spanning subgraph of size $k \geq n$, once a Hamiltonian path is given as part of the input, is *NP*-complete as well. In this paper we consider the optimization version of this latter problem, that is, given a biconnected graph G and a Hamiltonian path in it, find a biconnected spanning subgraph of G whose size is minimum. We refer to this problem as the *MBSH problem* and we show that it can be solved in $O(n + m)$ time and space and with an approximation guarantee of 4/3. It is not hard to see that the algorithm proposed by Garg et al. [4], will not guarantee an approximation factor better than 3/2 when adapted to the latter problem (the adaptation essentially consists of setting the generic depth first search tree used there to be just the given Hamiltonian path).

From an application point of view, our algorithm has a practical impact in *chain communication networks* (or *one-to-one communication networks*), where we have a distinguished source vertex r sending messages to a sink vertex s through a chain of vertices $\langle v_1 = r, v_2, \ldots, v_n = s\rangle$. Suppose we have a set of potential additional links (v_i, v_j), with $1 \leq i < j+1 \leq n$, such that the graph resulting from the chain now enriched of the additional edges is biconnected. Then, one might be interested in making the communication between r and s immune to vertex failures (except for the failing of r and s), by using a minimum number of links. Our algorithm solves this problem in linear time and space with an approximation factor of 4/3.

The algorithm starts by computing an *open ear decomposition* (OED) of a biconnected spanning subgraph of G. Recall that a graph is biconnected if and only if it has an OED [10]. After performing a series of gluing operations on the ears, the algorithm outputs a refined OED whose ears are either long or poorly adjacent to the rest of the graph. Hence, it will be possible to show that each refined ear needs at least 3 edges (in amortized sense) to be biconnected to the rest of the graph. From this, the approximation factor of 4/3 will be derived.

2. Preliminaries

2.1. Basic definitions

Let $G = (V, E)$ be an unweighted, undirected graph, where V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Let $n \geq 3$ and m denote the number of vertices and the number of edges, respectively. A graph $H = (V(H), E(H))$ is called a *subgraph* of G if $V(H) \subseteq V$ and $E(H) \subseteq E$. If $V(H) \equiv V$ then H is called a *spanning subgraph* of G.

A *simple path* P (or a *path* for short) in G is a subgraph with $V(P) = \{v_1, \ldots, v_k | v_i \neq v_j \text{ for } i \neq j\}$ and $E(P) = \{(v_i, v_{i+1}) | 1 \leq i < k\}$, also denoted as $\langle v_1, v_2, \ldots, v_k \rangle$. Path P is said to go from v_1 to v_k, called the *endvertices* of P, passing through the *internal* vertices $v_2, v_3, \ldots, v_{k-1}$. The number of vertices belonging to P will be denoted as $|P|$, while the restriction of P to the *subpath* $\langle v_i, v_{i+1}, \ldots, v_j \rangle, 1 \leq i < j \leq k$, will be denoted as $P(v_i, v_j)$. A *cycle* is a path whose endvertices coincide.

A spanning path $T = \langle v_1, \ldots, v_n \rangle$ of G is called a *Hamiltonian path*. Edges in $E(T)$ are called *path edges*, while the remaining edges of G are called *cycle edges*.

A graph G is *2-vertex connected* (or simply *biconnected*) if, given any three distinct vertices u, v, w of G, there exists a path from u to w not passing through v.

An *ear decomposition* $C_0, P_1, \ldots, P_k$ of G is a partition of its edges into sets $E(C_0), E(P_1), \ldots, E(P_k)$ such that:

(i) C_0 is a cycle;

(ii) P_1 is a path having both endvertices, but no internal vertex, in $V(C_0)$;

(iii) P_i, $2 \leq i \leq k$, is a path having both endvertices in $V_{i-1} = V(C_0) \cup V(P_1) \cup \ldots \cup V(P_{i-1})$ and having no internal vertex in V_{i-1}.

The paths P_i are called *ears*. A *t-ear* is an ear consisting of t vertices. If the endvertices of an ear are distinct we say that the ear is *open*.

A graph is biconnected if and only if it admits an open ear decomposition (OED) [10]. In the following, we shall denote by $\mathcal{E}(G) = C_0 + P_1 + \ldots + P_k$ a biconnected spanning subgraph of G whose ear decomposition is $C_0, P_1, \ldots, P_k$.

2.2. The initial open ear decomposition

Let $T = \langle v_1, v_2, \ldots v_n \rangle$ be a Hamiltonian path of G. To simplify notations, vertex v_i will be in the following identified by i. We start by computing in $O(n + m)$ time the value [2]

$$L(i) = \min\{j | j \leq i \wedge \exists k \geq i : (j, k) \in E\}$$

for all $i \in V$. Then, we decompose T in $O(n + m)$ time and space in a set of subpaths $\mathcal{I} = \{I_1, I_2, \ldots, I_k\}$, defined as follows:

$$I_1 = \langle a_0 \equiv 1, \ldots, b_1 \rangle, \text{ where } b_1 = \max\{i | L(v_i) = 1\};$$

$$I_2 = \langle a_1, \ldots, b_2 \rangle, \text{ where } b_2 = \max\{i | a_0 < L(i) < b_1\} \text{ and } a_1 = L(b_2);$$

$$\vdots$$

$$I_j = \langle a_{j-1}, \ldots, b_j \rangle, \text{ where } b_j = \max\{i | b_{j-2} \leq L(i) < b_{j-1}\} \text{ and } a_{j-1} = L(b_j);$$

$$\vdots$$

$$I_k = \langle a_{k-1}, \ldots, b_k \equiv n \rangle, \text{ where } a_{k-1} = L(n).$$

Note that, by definition, the edges (a_{j-1}, b_j), $j = 1, \ldots, k$ are cycle edges. Starting from $\mathcal{I}$ we identify, in $O(n)$ time and space, the set of paths

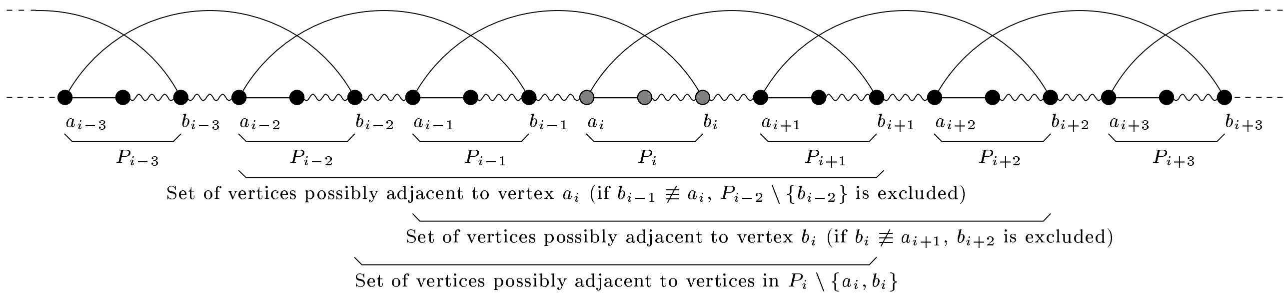

Figure 1: Possible adjacencies of (grey) vertices in P_i: splines denote paths, possibly empty (in such a case, the endvertices of the spline coincide).

$$\mathcal{P} = \{P_j | P_j = \langle a_j, \ldots, b_j \rangle, j = 1, \ldots, k-1\}.$$

Let $P_i \cap P_j$ denote the set of vertices $V(P_i) \cap V(P_j)$. The following properties of paths in $\mathcal{P}$ are easy to show (see Figure 1):

(P1): $|P_i| \geq 2, i = 1, \ldots, k-1$.

(P2): $P_i \cap P_{i+1} \subseteq \{b_i\}, i = 1, \ldots, k-2$.

(P3): $P_i \cap P_j = \emptyset, 1 \leq i < j+1 \leq k-1$.

(P4): (Adjacencies of a_i) For $3 \leq i \leq k-1$, if $a_i \not\equiv b_{i-1}$, then a_i is adjacent only to vertices in $T(b_{i-2}, b_{i+1})$, otherwise a_i is adjacent to a_{i-2} and may be also adjacent to vertices in $T(a_{i-2}+1, b_{i-2}-1)$. Adjacencies of a_i for $i = 1, 2$ can be easily inferred.

(P5): (Adjacencies of b_i) For $1 \leq i \leq k-2$, if $b_i \not\equiv a_{i+1}$, then b_i is adjacent only to vertices in $T(a_{i-1}, b_{i+2}-1)$, otherwise b_i is also adjacent to b_{i+2}. Adjacencies of b_{k-1} can be easily inferred.

(P6): (Adjacencies of an internal vertex of P_i) For $3 \leq i \leq k-2$, an internal vertex v of P_i can be adjacent only to vertices in $T(b_{i-2}, b_{i+1})$. Adjacencies of v for $i = 1, 2, k-1$ can be easily inferred.

Let $C_0 = (V_0, E_0)$, where

$$V_0 = V \setminus \bigcup_{i=1}^{k-1} \{a_i + 1, \ldots, b_i - 1\} \qquad E_0 = \left(E(T) \setminus \bigcup_{i=1}^{k-1} E(P_i) \right) \cup \bigcup_{i=1}^{k} (a_{i-1}, b_i).$$

It is not hard to see that C_0 is a cycle, and together with the paths in $\mathcal{P}$ defines a (planar) OED of a biconnected spanning subgraph $\mathcal{E}(G)$ of G (see Figure 2). If an ear $P_i = \langle a_i, \ldots, b_i \rangle$ has its first vertex coinciding with the last vertex of P_{i-1}, that is $a_i \equiv b_{i-1}$, we say that P_i is *special*, otherwise it is *regular*.

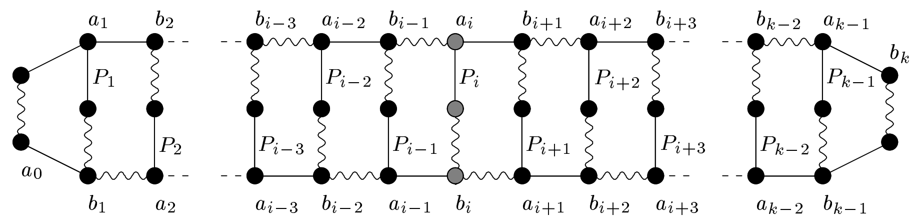

Figure 2: The biconnected spanning subgraph $\mathcal{E}(G)$: splines denote paths, possibly empty.

3. The refinement algorithm

The algorithm starts from $\mathcal{E}(G) = C_0 + P_1 + \ldots + P_{k-1}$ and produces a new biconnected spanning subgraph of G, say $\mathcal{E}'(G) = D_0 + Q_1 + \ldots + Q_p$, $p \leq k-1$, whose special 3-ears are poorly adjacent in G to the rest of the ears. We shall show that each $Q_j, j = 1, \ldots, p$ needs at least 3 edges (in amortized sense) to be biconnected to the rest of the graph. It is worth noting that if an ear in $\mathcal{E}(G)$ is a regular t-ear, with $t \geq 3$, then at least t edges are necessary to biconnect its t vertices to the rest of G. Therefore, to obtain an approximation ratio of 4/3, it suffices to handle 2-ears and special 3-ears in $\mathcal{E}(G)$.

3.1. High level description of the algorithm

The algorithm sets initially $D_0 := C_0$, and then considers one after the other all the ears in $\mathcal{E}(G)$. At the i-th step, the partial biconnected subgraph $D_0 + Q_1 + \ldots + Q_{j-1}$, $j \leq i$, has already been constructed and the new ear P_i of $\mathcal{E}(G)$ is considered:

1. if P_i is a 2-ear, then P_i contains a single edge, and it is simply discarded from $\mathcal{E}(G)$;
2. if P_i is a special 3-ear (in the sense that its first vertex coincides with the last vertex of the last created refined ear, say Q_{j-1}), then we consider its adjacencies in G with Q_{j-1}, as well as with the next ear P_{i+1} (if any) and with D_0, and we define four procedures aiming to eliminate P_i from $\mathcal{E}(G)$;
3. otherwise, P_i is simply added to the final solution.

The four procedures mentioned in the Step 2. are called GLUEDOWN, GLUEUP, STRETCH and SWITCH and they are given in detail in the Appendix. Here, we provide a high-level description of them and we make use of sampling figures to illustrate how they work. In the rest of the paper, the first (last) vertex of Q_j will be denoted with α_j (β_j). So, the 3-ear $P_i = \langle a_i, v, b_i \rangle$ considered at the i-th step of the algorithm is special if $a_i \equiv \beta_{j-1}$.

The procedure GLUEDOWN either glues together the special 3-ear P_i with Q_{j-1}, where $|Q_{j-1}| \leq 5$, or eliminates P_i and transforms Q_{j-1} into a regular 3-ear. Figure 3 illustrates the various cases of the procedure (for each case, the left picture represents $\mathcal{E}(G)$ and the right one represents $\mathcal{E}'(G)$).

The procedure GLUEUP either eliminates P_i and transforms P_{i+1} into a regular t-ear, $t = 3$ or 4, or eliminates both of them by elongating D_0. The operations performed are similar to the one depicted in case (c) of Figure 3, and are described in detail in the Appendix.

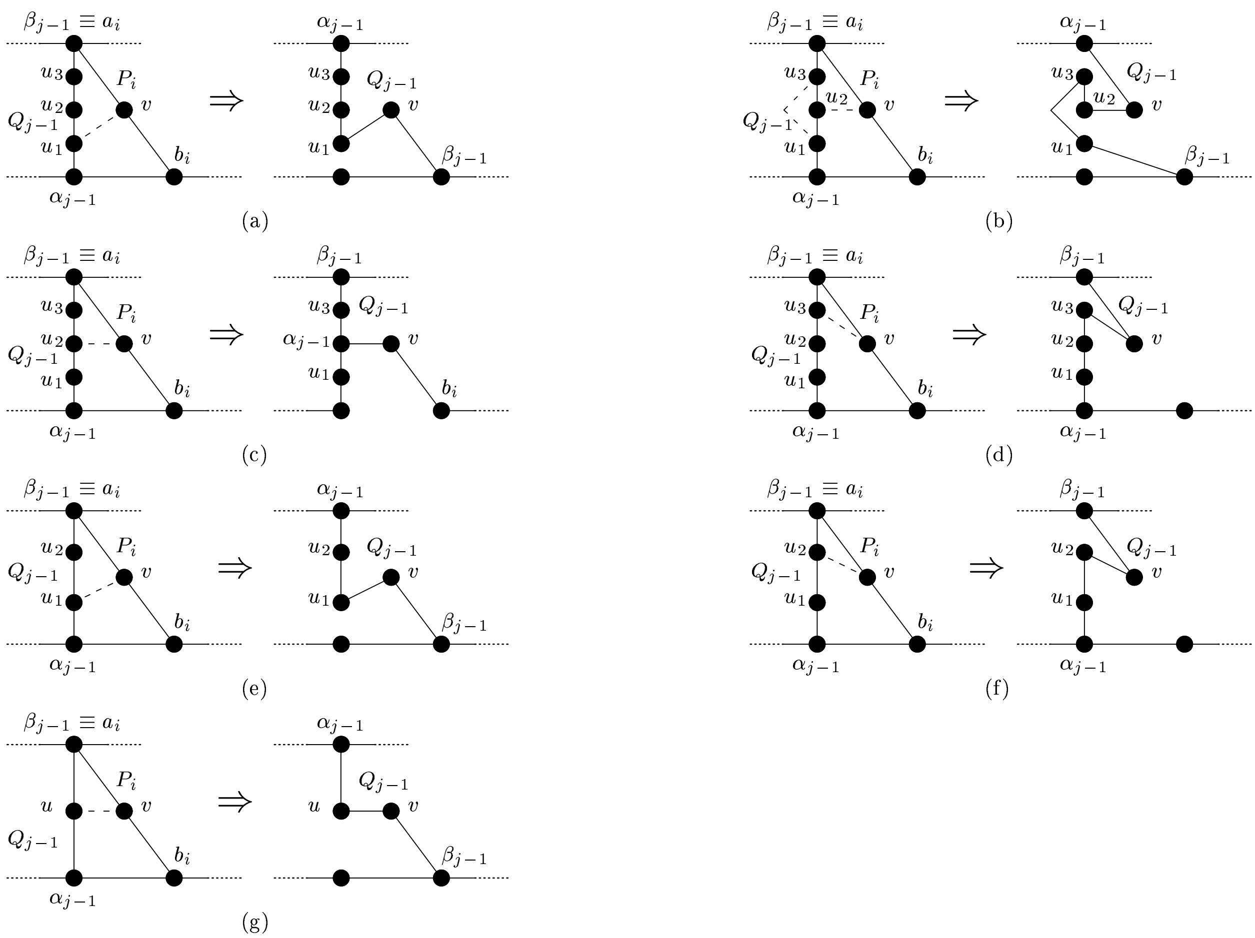

Figure 3: Procedure GLUEDOWN(Q_{j-1}, P_i): dashed edges are in $E(G) \setminus E(\mathcal{E}(G))$. Cases (a)-(g) are associated with cases (a)-(g) of the procedure (see the Appendix).

The procedure STRETCH eliminates the special 3-ear P_i and elongates D_0 by inserting v, as shown in Figure 4.

Finally, the procedure SWITCH eliminates P_i by inserting v in D_0 and elongates Q_{j-1} so that it will contain at least 4 vertices. Figure 5 depicts samples of the execution of this procedure.

The algorithm to find a refined ear decomposition $\mathcal{E}'(G)$ is given in the following:

Algorithm *Refined Open Ear Decomposition*;
Input: A graph G and a biconnected spanning subgraph $\mathcal{E}(G) = C_0 + P_1 + \ldots + P_{k-1}$ of G;
Output: A refined biconnected spanning subgraph $\mathcal{E}'(G) = D_0 + Q_1 + \ldots + Q_p$ of G, with $p \leq k-1$.

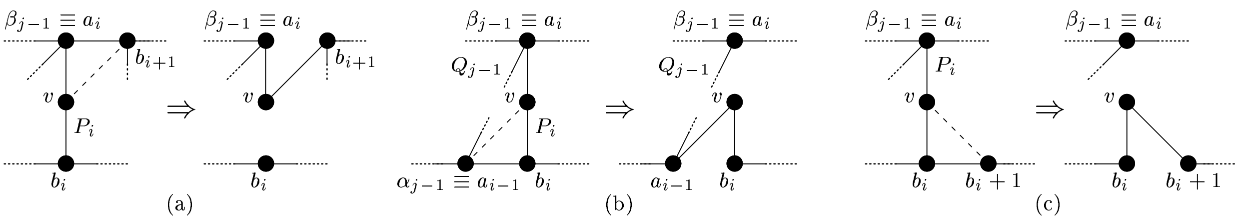

Figure 4: Procedure STRETCH(D_0, v): dashed edges are in $E(G) \setminus E(\mathcal{E}(G))$. Cases (a)-(c) are associated with cases (a)-(c) of the procedure (see the Appendix).

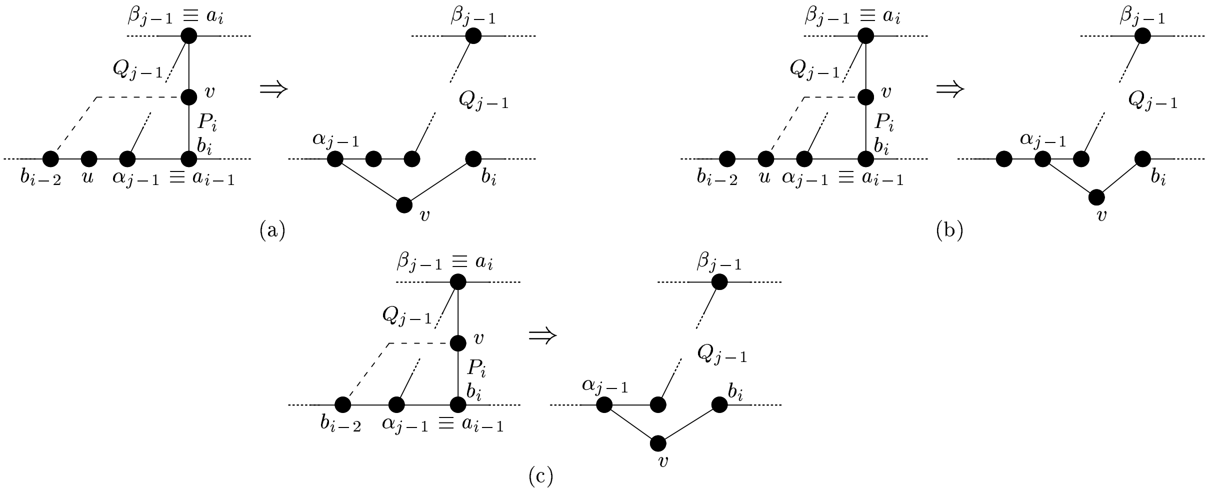

Figure 5: Procedure SWITCH(D_0, Q_{j-1}, v): dashed edges are $E(G) \setminus E(\mathcal{E}(G))$. Cases (a)-(c) are associated with cases (a)-(c) of the procedure (see the Appendix).

Procedure ROED($G, \mathcal{E}(G)$);
    $D_0 := C_0$; $Q_1 := P_1$; $j := 2$; $\alpha_1 := a_1$; $\beta_1 := b_1$; $b_0 := a_0$;
    for each path $P_i, i = 2, \ldots, k-1$ such that $|P_i| > 2$ **do begin**
        if $P_i = \langle a_i, v, b_i \rangle$ is special **then**
            case adjacencies of v **of**
                v is adjacent to some internal vertex of Q_{j-1} **and** $|Q_{j-1}| \leq 5$: GLUEDOWN(Q_{j-1}, P_i);
                v is adjacent to some internal vertex of P_{i+1} **and** $|P_{i+1}| \leq 5$: GLUEUP(P_{i+1}, P_i);
                $\langle a_i, v, b_{i+1} \rangle$ **or** $\langle b_i, v, a_{i-1} \rangle$ **or** $\langle b_i, v, b_i + 1 \rangle$ is a triangle: STRETCH(D_0, v);
                v is adjacent to $D_0(b_{i-2}, a_{i-1} - 1)$ **and** $|D_0(b_{i-2}, a_{i-1})| \leq 3$: SWITCH($D_0, Q_{j-1}, v$);
                all other adjacencies: **begin** $Q_j := P_i$; $j := j + 1$ **end**;
            end case;
        else begin $Q_j := P_i$; $j := j + 1$ **end**; $\{Q_j$ is created with $|Q_j| \geq 3\}$
    end;
end ROED.

3.2. Analysis of the algorithm

The output of the algorithm is a new biconnected spanning subgraph $\mathcal{E}'(G) = D_0 + Q_1 + \ldots + Q_p$ of G, where no new special 3-ear has been created. More specifically, if $Q_j, 2 \leq j \leq p$, is a special 3-ear of $\mathcal{E}'(G)$, then there exists an index $i \geq j$ such that $Q_j = P_i$.

The next lemma will show that the special 3-ears remaining from the original OED $\mathcal{E}(G)$ are those ones having their internal vertex poorly adjacent in G to the rest of the graph. In the following, we will denote by L_i the path $C_0(b_{i-2}, a_{i-1})$ and by U_i the path $C_0(b_i, a_{i+1})$, $3 \leq i \leq k-2$. Moreover, we set $L_2 = C_0(a_0, a_1)$ and $U_{k-1} = C_0(b_{k-1}, a_{k-1})$. These paths can be easily identified by looking at Figure 2.

Given two paths P and P', let in the following $P \cup P'$ denote the set of vertices $V(P) \cup V(P')$. Observe first that, from property (P6), the internal vertex v of a special 3-ear $Q_j = P_i$ can be adjacent only to vertices belonging to $L_i \cup P_{i-1} \cup P_i \cup U_i \cup P_{i+1}$, $2 \leq i \leq k-1$.

If $Q_j = P_i$ is a special 3-ear, we say that a vertex u is a *credit vertex* for Q_j if $u \in L_i \cup U_i \cup Q_{j-1} \cup Q_{j+1}$, u is not the endvertex of any ear of $\mathcal{E}'(G)$ and, moreover, it is uniquely associated with Q_j. Notice that, by definition, $L_i \equiv U_{i-2}$, and therefore L_i (or U_i) can contain (at most) two credit vertices.

Lemma 3.1. *The algorithm ROED($G,\mathcal{E}(G)$) yields a refined biconnected spanning subgraph $\mathcal{E}'(G) = D_0 + Q_1 + \ldots + Q_p$ of G such that for each special 3-ear $Q_j = \langle \alpha_j \equiv \beta_{j-1}, v, \beta_j \rangle = P_i, 2 \leq j \leq p, j \leq i \leq k-1$, the following properties hold:*

(Q1): v is not adjacent to b_{i+1}, a_{i-1} and $b_i + 1$;

(Q2): if v is adjacent to some internal vertex of Q_t, $t = j-1$ or $t = j+1$, then Q_t contains a credit vertex for Q_j;

(Q3): if v is adjacent to some vertex of L_i, then L_i contains a credit vertex for Q_j;

(Q4): if v is adjacent to some vertex of U_i other than b_i, then U_i contains a credit vertex for Q_j.

Proof. Property (Q1) follows trivially from the procedure STRETCH of the algorithm.

To prove (Q2), observe first that, by the procedures GLUEDOWN and GLUEUP of the algorithm, $|Q_t| \geq 6$. Then at least one internal vertex of Q_t can be used as a credit vertex for Q_j.

Concerning (Q3), if v is adjacent to a vertex u in L_i, then $|L_i| \geq 4$, since otherwise the procedure SWITCH would have been applied, and Q_j would have not been a special 3-ear for $\mathcal{E}'(G)$. Hence, L_i contains at least two vertices which are not envertices of any ear of $\mathcal{E}'(G)$, and then L_i contains a credit vertex for Q_j.

Finally, concerning (Q4), if v is adjacent to a vertex u in U_i other than b_i, then $|U_i| \geq 3$ since otherwise the procedure STRETCH would have been applied. Now, if $|U_i| \geq 4$, then analogously to the previous case, U_i contains a credit vertex for Q_j. On the other hand, if $|U_i| = 3$, i.e., $U_i = \langle b_i, u, a_{i+1} \rangle$, then the only chance is that v is adjacent to a_{i+1}, since otherwise procedure STRETCH would have been applied. If Q_{j+2} is a special 3-ear, then its internal vertex is not adjacent to any vertex of $L_{i+2} \equiv U_i$, since otherwise the procedure SWITCH would have been applied to the ear Q_{j+2}. This implies that $u \in U_i$ is a credit vertex for Q_j. The same holds if Q_{j+2} is not a special 3-ear, since in this case u can be a credit vertex only for Q_j. This completes the proof. ∎

Lemma 3.2. *Let $\mathcal{E}'(G) = D_0 + Q_1 + \ldots + Q_p$ be the output produced by ROED($G, \mathcal{E}(G)$). Let $H_{\text{OPT}} = (V, E_{\text{OPT}})$ be a minimum size biconnected spanning subgraph of G. Then*

$$|E_{\text{OPT}}| \geq max\{n, 3p\}.$$

Proof. If $n \geq 3p$ then the inequality follows trivially. Then, let us show that $|E_{\text{OPT}}| \geq 3p$ when $n < 3p$. As first, observe that each vertex in a biconnected graph has at least one edge entering and one edge leaving. Now, in order to count the edges of E_{OPT} exactly once, we will count only half edge entering and half edge leaving each vertex under consideration.

Let us consider the j-th ear of the system, $Q_j = \langle \alpha_j, v, \ldots, \beta_j \rangle, 1 \leq j \leq p$. If Q_j is special (i.e., $\beta_{j-1} \equiv \alpha_j$), then we set $Q'_j = \langle v, \ldots, \beta_j \rangle$, otherwise we set $Q'_j = Q_j$. Clearly, the paths Q'_j are pairwise disjoint.

If $|Q'_j| \geq 3$, then at least 3 new edges are needed to biconnect Q'_j to the rest of the graph. Hence, we restrict ourselves to the case when $Q'_j = \langle v, \beta_j \rangle$, namely Q_j is a special 3-ear. Notice that the first ear is never special, and then Q'_1 contains at least 3 vertices, while the p-th ear has at least 3 vertices in amortized sense, since it is easy to see that in $D_0(\alpha_p, \beta_p)$ there exists at least one vertex that can be uniquely associated with Q_p as a credit vertex; thus, both Q_1 and Q_p need at least 3 edges to be biconnected.

We shall prove that each special 3-ear $Q_j = P_i$, $2 \leq j \leq p-1$, needs at least 3 edges to be biconnected to the rest of the graph. But these three edges will be distributed differently among the vertices of Q'_j, depending on the adjacencies of v. More specifically, we shall prove that if v is not adjacent to vertices of G other than α_j and β_j, then one full edge entering v and one full edge leaving v will be counted in E_{OPT}, plus half edge entering and half edge leaving β_j. On the other hand, if v is adjacent to vertices of G different from α_j and β_j, then Q'_j will borrow a credit vertex u from $L_i \cup U_i \cup Q_{j-1} \cup Q_{j+1}$, and the three edges needed to biconnect $Q'_j \cup \{u\}$ to the rest of the graph will be distributed so that each vertex has half edge entering and half edge leaving.

Suppose now that the claim does not hold and let Q'_j be the special 3-ear which fails to satisfy it and has index $j \geq 2$ as small as possible. Since Q_j is a special 3-ear, it was special also before being considered by the algorithm ROED($G, \mathcal{E}(G)$). This means that there exists an index i such that $P_i = Q_j$. We consider now the two cases:

Case 1. v is adjacent only to α_j and β_j.

Note first that, by the choice of the index j, only half edge entering and half edge leaving $\alpha_j \equiv \beta_{j-1}$ have been counted in E_{OPT}. The rest of the proof is based on the key observation that, from properties (P4) and (P6), vertices of index greater than $\beta_j \equiv b_i$ are not adjacent to vertices of index smaller than $\alpha_j \equiv a_i$, since $a_i \equiv \beta_{j-1} \equiv b_{i-1}$. Hence, all the paths from a vertex $x > b_i$ to a vertex $y < a_i$ must use vertices in Q_j.

The edges (α_j, v) and (v, β_j) need to be inserted in E_{OPT} in order to biconnect v to the rest of the graph. Now, suppose that one of the half edges counted for α_j coincides with half of (α_j, v). If the remaining half edge of α_j connects it to a vertex greater than or equal to b_i, then, from the above observation, all paths from a vertex $x > b_i$ to a vertex $y < a_i$ use the vertex b_i. Thus, b_i is a cutvertex of H_{OPT}, a contradiction.

On the other hand, if the remaining half edge of α_j connects it to a vertex smaller than a_i, then, again, from the above observation, all paths from a vertex $x > b_i$ to a vertex $y < a_i$ use the vertex b_i. Thus, b_i is a cutvertex of H_{OPT}, a contradiction. It follows that half of the edge (α_j, v) has not been counted as a leaving half edge of α_j: since it has to be added in order to biconnect v to the rest of the graph, we have to count it as a full edge for v. Similarly, we can prove that half of (v, β_j) cannot be counted as half edge entering or leaving β_j. The reasoning is analogous to the previous one with the role of a_i and b_i interchanged. It follows that the half edges leaving and entering β_j are different from (v, β_j). This proves the claim.

Case 2. v is adjacent to some vertex u different from α_j and β_j.

Let us recall that, from property (P6), v may be adjacent only to vertices belonging to $L_i \cup P_{i-1} \cup P_i \cup U_i \cup P_{i+1}$.

First, suppose that the vertex u is an internal vertex of P_{i-1}. Since P_i is special, it is easy to see that $V(P_{i-1}) \subseteq V(Q_{j-1})$, and then $u \in Q_{j-1}$. Hence, from property (Q2) of Lemma 3.1, we have that Q_{j-1} contains a credit vertex for Q_j.

On the other hand, if u is an internal vertex of P_{i+1}, then from the fact that P_i is special, it follows that $|P_{i+1}| > 5$, since otherwise P_i would have been eliminated by procedure GLUEUP. Hence, it is easy to see that $V(P_{i+1}) \subseteq V(Q_{j+1})$, and then $u \in Q_{j+1}$. Therefore, from property (Q2) of Lemma 3.1, we have that Q_{j+1} contains a credit vertex for Q_j.

Suppose now that u is not an internal vertex of P_{i-1} and P_{i+1}, that is $u \in L_i \cup U_i \cup \{b_{i+1}\}$. From property (Q1) of Lemma 3.1, it must be $u \not\equiv b_{i+1}$. Henceforth, $u \in L_i \cup U_i$, and from properties (Q3) and (Q4) of Lemma 3.1, we have that at least one of these paths contains a credit vertex for Q_j.

It follows that for any special 3-ear we have to consider three new vertices which need to be biconnected with the rest of the graph, the vertices v and β_j plus one credit vertex for Q_j. Therefore, since each of these vertices needs at least half edge entering and half edge leaving, at least 3 new edges need to be counted in E_{OPT}. ∎

Theorem 3.3. *Given a biconnected graph $G = (V, E)$ with $n \geq 3$ vertices and m edges, and a Hamiltonian path T in G, there exists an $O(n+m)$ time and space algorithm yielding a biconnected spanning subgraph $\mathcal{E}'(G) = (V, E')$ of G such that the size of E' is at most 4/3 times the size of E_{OPT}, where $H_{\mathrm{OPT}} = (V, E_{\mathrm{OPT}})$ is a biconnected spanning subgraph of G of minimum size. The bound is asymptotically tight.*

Proof. The initial biconnected spanning subgraph $\mathcal{E}(G)$ of G can be computed in $O(n+m)$ time and space. The algorithm ROED$(G, \mathcal{E}(G))$ requires $O(n+m)$ time, since it performs on $O(n)$ special 3-ears a constant number of operations which can be executed in $O(1)$ time, and space occupancy is trivially $O(m+n)$. Moreover, it outputs a refined biconnected spanning subgraph $\mathcal{E}'(G) = (V, E')$ of G, such that $|E'| = n + p$.

By Lemma 3.2, we have that $|E_{\mathrm{OPT}}| \geq max\{n, 3p\}$, and hence

$$\frac{|E'|}{|E_{\mathrm{OPT}}|} \leq \frac{n+p}{max\{n, 3p\}}.$$

Thus, if $n \geq 3p$ we have

$$\frac{|E'|}{|E_{\mathrm{OPT}}|} \leq \frac{n+p}{n} \leq \frac{\frac{4}{3}n}{n} = \frac{4}{3}$$

while if $n < 3p$, we get

$$\frac{|E'|}{|E_{\mathrm{OPT}}|} \leq \frac{n+p}{3p} \leq \frac{4p}{3p} = \frac{4}{3}.$$

The above bound is asymptotically tight. In fact, consider the graph G and the biconnected spanning subgraph $\mathcal{E}(G)$ depicted in Figure 3. Notice that $\mathcal{E}'(G) \equiv \mathcal{E}(G)$ and then $k - 1 = p$.

Moreover, $n = 3k = 3(p+1)$ and G is Hamiltonian (dotted cycle). Hence, $\frac{|E'|}{|E_{\mathrm{OPT}}|} = \frac{n+p}{n} = \frac{3(p+1)+p}{3(p+1)} = 1 + \frac{p}{3(p+1)}$, which tends to 4/3 for large p. ∎

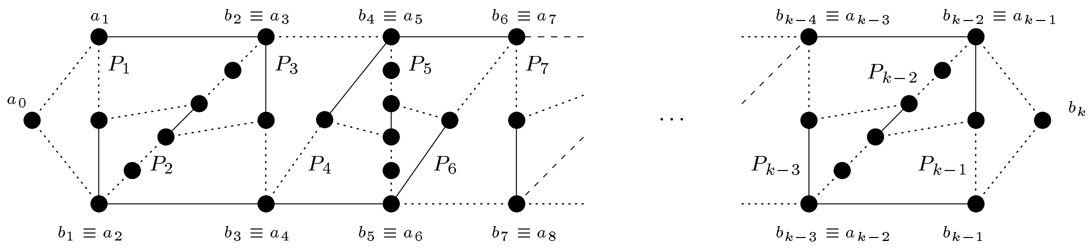

Figure 6: A graph G for which the approximation ratio tends to 4/3.

4. Conclusions

In this paper we have presented a 4/3-approximation algorithm for the MBSH problem which runs in $O(n+m)$ time and space.

The problem of finding a similar result for the general case in which the Hamiltonian path (if any) is not given, remains open. Since a Hamiltonian path corresponds to a depth first search (DFS) spanning tree of G, one could try to apply the technique developed in this paper to solve the general case. The idea would be to apply our algorithm to each branch of a generic DFS spanning tree of G and then collect the outputs from the different subproblems. However, in doing this, more complex adjacencies among ears arise, and the ear decomposition refinement becomes harder. The extension of our algorithm to the general case is currently under investigation.

Appendix

In this Appendix, we provide a detailed description of procedures GLUEDOWN, GLUEUP, STRETCH and SWITCH defined in the previous section.

Procedure GLUEDOWN(Q_{j-1}, P_i);
  case adjacencies of v **of** {In each of the following cases P_i is eliminated}
(a) $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, u_3, \beta_{j-1} \rangle$ **and** $(v, u_1) \in E$:
    $Q_{j-1} = \langle \alpha_{j-1} := a_i, u_3, u_2, u_1, v, \beta_{j-1} := b_i \rangle$; {$Q_{j-1}$ is elongated: note that its last vertex is now b_i}
  $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, u_3, \beta_{j-1} \rangle$ **and** $(v, u_2) \in E$:
(b)   **if** $(u_1, u_3) \in E$ **then** $Q_{j-1} = \langle \beta_{j-1}, v, u_2, u_3, u_1, \alpha_{j-1} \rangle$
(c)   **else begin**
      $D_0 = D_0 \setminus \langle \alpha_{j-1}, b_i \rangle \cup \langle \alpha_{j-1}, u_1, u_2, v, b_i \rangle$; {$D_0$ is elongated}
      $Q_{j-1} = \langle \alpha_{j-1} := u_2, u_3, \beta_{j-1} \rangle$; {$Q_{j-1}$ becomes a regular 3-ear}
    end;
(d) $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, u_3, \beta_{j-1} \rangle$ **and** $(v, u_3) \in E$:
    $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, u_3, v, \beta_{j-1} \rangle$; {$Q_{j-1}$ is elongated}
(e) $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, \beta_{j-1} \rangle$ **and** $(v, u_1) \in E$:
    $Q_{j-1} = \langle \alpha_{j-1} := a_i, u_2, u_1, v, \beta_{j-1} := b_i \rangle$; {$Q_{j-1}$ is elongated; note that its last vertex is now b_i}

(f) $Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, \beta_{j-1} \rangle$ **and** $(v, u_2) \in E$:
$Q_{j-1} = \langle \alpha_{j-1}, u_1, u_2, v, \beta_{j-1} \rangle$; {$Q_{j-1}$ is elongated}
(g) $Q_{j-1} = \langle \alpha_{j-1}, u, \beta_{j-1} \rangle$ **and** $(v, u) \in E$:
$Q_{j-1} = \langle \alpha_{j-1}, u, v, \beta_{j-1} \rangle$; {$Q_{j-1}$ is elongated}
end case;
end GLUEDOWN.

Procedure GLUEUP(P_{i+1}, P_i);
case adjacencies of v **of** {In each of the following cases P_i is eliminated and D_0 is elongated}
(a) $P_{i+1} = \langle a_{i+1}, u_1, u_2, u_3, b_{i+1} \rangle$ **and** $(v, u_1) \in E$:
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_1)\} \cup P_{i+1}(u_1, b_{i+1})$; {$P_{i+1}$ is eliminated}
(b) $P_{i+1} = \langle a_{i+1}, u_1, u_2, u_3, b_{i+1} \rangle$ **and** $(v, u_2) \in E$: **begin**
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_2)\} \cup P_{i+1}(u_2, b_{i+1})$;
$P_{i+1} = \langle a_{i+1}, u_1, u_2 \rangle$; **end**; {$P_{i+1}$ becomes a regular 3-ear}
(c) $P_{i+1} = \langle a_{i+1}, u_1, u_2, u_3, b_{i+1} \rangle$ **and** $(v, u_3) \in E$: **begin**
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_3)\} \cup (u_3, b_{i+1})$;
$P_{i+1} = \langle a_{i+1}, u_1, u_2, u_3 \rangle$; **end**; {$P_{i+1}$ becomes a regular 4-ear}
(d) $P_{i+1} = \langle a_{i+1}, u_1, u_2, , b_{i+1} \rangle$ **and** $(v, u_1) \in E$:
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_1)\} \cup P_{i+1}(u_1, b_{i+1})$;
(e) $P_{i+1} = \langle a_{i+1}, u_1, u_2, , b_{i+1} \rangle$ **and** $(v, u_2) \in E$: **begin**
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_2)\} \cup (u_2, b_{i+1})$;
$P_{i+1} = \langle a_{i+1}, u_1, u_2 \rangle$; **end**; {$P_{i+1}$ becomes a regular 3-ear}
(f) $P_{i+1} = \langle a_{i+1}, u_1, , b_{i+1} \rangle$ **and** $(v, u_1) \in E$:
$D_0 := D_0 \backslash (a_i, b_{i+1}) \cup \{(a_i, v), (v, u_1)\} \cup (u_1, b_{i+1})$;
end case;
end GLUEUP.

Procedure STRETCH(D_0, v);
case adjacencies of v **of** {In each of the following cases P_i is eliminated and D_0 is elongated}
(a) $(v, b_{i+1}) \in E : D_0 = D_0 \setminus \langle a_i, b_{i+1} \rangle \cup \langle a_i, v, b_{i+1} \rangle$;
(b) $(v, a_{i-1}) \in E : D_0 = D_0 \setminus \langle a_{i-1}, b_i \rangle \cup \langle a_{i-1}, v, b_i \rangle$;
(c) $(v, b_i + 1) \in E$: $D_0 = D_0 \setminus \langle b_i, b_i + 1 \rangle \cup \langle b_i, v, b_i + 1 \rangle$;
end case.
end STRETCH.

Procedure SWITCH(D_0, Q_{j-1}, v);
case structure of $D_0(b_{i-2}, a_{i-1})$ **of** {In each of the following cases P_i is eliminated}
$D_0(b_{i-2}, a_{i-1}) \equiv \langle b_{i-2}, u, a_{i-1} \rangle$: **begin**
case adjacencies of v **of**
(a) $(v, b_{i-2}) \in E$: **begin**
$D_0 = D_0 \setminus \langle b_{i-2}, u, a_{i-1}, b_i \rangle \cup \langle b_{i-2}, v, b_i \rangle$;
$Q_{j-1} = \langle \alpha_{j-1} := b_{i-2}, u, a_{i-1}, \ldots, \beta_{j-1} \rangle$; **end;** {$Q_{j-1}$ is elongated; its first vertex is now b_{i-2}}
(b) $(v, u) \in E$: **begin**
$D_0 = D_0 \setminus \langle u, a_{i-1}, b_i \rangle \cup \langle u, v, b_i \rangle$;
$Q_{j-1} = \langle \alpha_{j-1} := u, a_{i-1}, \ldots, \beta_{j-1} \rangle$; **end;** {$Q_{j-1}$ is elongated; its first vertex is now u}
end case.
(c) $D_0(b_{i-2}, a_{i-1}) \equiv \langle b_{i-2}, a_{i-1} \rangle$: **begin**
$D_0 = D_0 \setminus \langle b_{i-2}, a_{i-1}, b_i \rangle \cup \langle b_{i-2}, v, b_i \rangle$;
$Q_{j-1} = \langle \alpha_{j-1} := b_{i-2}, \alpha_{j-1} \equiv a_{i-1}, \ldots, \beta_{j-1} \rangle$; {$Q_{j-1}$ is elongated; its first vertex is now b_{i-2}}
end;
end case.
end SWITCH.

References

[1] J. Cheriyan, A. Sebö and Z. Szigeti, An improved approximation algorithm for minimum size 2-edge connected spanning subgraphs, *Proc. of the 6th Conf. on Integer Programming and Combinatorial Optimization, LNCS 1412*, (1998) 126–136.

[2] S. Even and R.E. Tarjan, Computing an st-numbering, *Theor. Comp. Science*, **2** (1976) 339–344.

[3] M.R. Garey and D.S. Johnson, *Computers and intractability: a guide to the theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.

[4] N. Garg, V.S. Santosh and A. Singla, Improved approximation algorithms for biconnected subgraphs via better lower bounding techniques, *Proc. of the 4th ACM-SIAM Symp. on Discrete Algorithms*, (1993) 103–111.

[5] M. Grötschel, C.L. Monma and M. Stoer, Design of survivable networks, in: *Handbooks in OR and MS, Vol. 7*, Elsevier (1995) 617–672.

[6] A. Itai and M. Rodeh, The Multi-Tree approach to reliability in distributed networks, *Information and Computation*, **79** (1988) 43–59.

[7] S. Khuller, Approximation algorithms for finding highly connected subgraphs, in *Approximation Algorithms for NP-Hard Problems*, Dorit S. Hochbaum Eds., PWS Publishing Company, Boston, MA, 1996.

[8] S. Khuller and U. Vishkin, Biconnectivity approximations and graph carvings, *J. ACM*, **41**(2) (1994) 214–235.

[9] C.H. Papadimitriou and K. Steiglitz, Some complexity results for the Traveling Salesman Problem, *Proc. of the 8th ACM Symp. on Theory of Computing*, (1976) 1-9.

[10] H. Whitney, Nonseparable and planar graphs, *Trans. Amer. Math. Soc.*, **34** (1932) 339-362.