Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 12, Number 1 (2016), pp. 33-55

http://www.ripublication.com

Connection of Two Nodes of A Transportation Network by Two Disjoint Paths

Ilya Abramovich Golovinskii[1]
National Research University of Electronic Technology (MIET), 124498, Bld. 1, Shokin Square, Zelenograd, Moscow, Russia.

Abstract

The problem of the connection of two vertices of an undirected graph by two paths which do not intersect in intermediate vertices is considered. The problem is of importance for the operative control of the reliability of transportation networks. An algorithm to obtain such a pair of paths is proposed. The solution is based on the use of the fundamental cycle basis of a graph. Reasons of the failure to find a solution without the use of special methods are given.

Keywords: transportation network, operative control of reliability, biconnected graph, fundamental set of cycles of a graph.

Introduction

Problem statement:

The problem given in the title of the paper is of importance for the operative control of transportation networks with variable configuration. Elements of such a network may be put out of operation for scheduled maintenance or due to accidents. The latter may require putting reserve elements into operation if they are available. The traffic control service must in this case ensure the necessary level of functioning of the network and monitor its reliability. In a man-machine control system not only decisions themselves must be efficient but efficiency is also required for the interaction of the man and the system that is easy to understand visualization of data, convenient controls, ergonomic man-machine interfaces are needed.
The functional source of the problem may be illustrated by the following example. Let us assume that the traffic control service needs to produce a one-way transport connection between two nodes (vertices) of their traffic network without intersections in intermediate nodes. The problem has a solution if the nodes which need to be

connected belong to the same biconnected component of the traffic network graph. To remind, an undirected connected graph containing more than two vertices is said to be (vertex) biconnected if it remains connected after the removal of any one vertex. A biconnected subgraph of an undirected graph is said to be its biconnected component if it is not contained in a larger (vertex) biconnected subgraph of the original graph. According to the Whitney theorem (Whitney, 1932; Harary, 1969) an undirected graph is biconnected if and only if any two of its distinct vertices can be connected by two simple paths having no other common vertices than those being connected (Note 1). The latter is equivalent to the existence of a simple cycle passing through the two vertices given (Note 2).

At the same time the existence of a simple cycle passing through two graph vertices means that they belong to a biconnected subgraph of this graph (as any simple cycle is biconnected). From this it follows that both vertices belong to the same biconnected component of the graph. Thus, the problem given in the title of the paper has a solution if and only if two graph vertices belong to the same biconnected component of the graph.

While stating the existence of the pair of paths connecting two graph vertices without intersections in intermediate vertices the theory of biconnected graphs does not however propose a practically useful method to build such paths (Berge, 1958; Ore, 1962; Harary, 1969; Bondy and Murty, 1982; Diestel, 2000). The present paper describes a method to build such a pair practically.

The construction of two such paths is also important for the monitoring of the reliability of a transportation network. An absence of intermediate common nodes in the both paths means an independence of the communication taking place via these paths. If communication via one of the paths in the pair is broken there may be a possibility of communication via the other path. A large bibliography on the reliability of networks can be found (Colboum, 1987; Shier, 1991; etc.).

Ergonomical issues of the problem under consideration arise from the fact that graphical schemes of transportation networks may be difficult to perceive. Two paths without intersection in intermediate nodes may visually appear as crossed. Even if the transportation network graph is planar it may look like having crossing edges. For example the graph shown in Fig.1 is biconnected and planar, however this is not obvious when looking at the figure. Moreover, this graph is a simple cycle which is also not immediately visible. It is not easy to see by eye that any pair of vertices in the figure is connected by two paths without intersections in intermediate vertices.

This example illustrates that simply looking at the graphical representation of a network without a computer analysis of its topology may lead to time delays needed to better understand the topology or to wrong logical conclusions. For an operative control of a complex transportation network with a variable topology formal algorithms are needed which produce the correct result independent on the visual appearance of the graph.

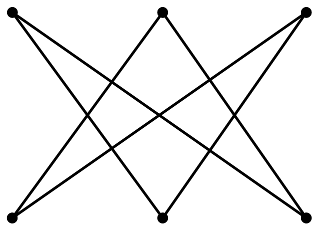

Figure 1: A graph consisting of a single simple cycle which is not obvious visually.

Task description:

We say that two paths connecting two graph vertices are *disjoint* if they do not intersect in intermediate vertices. In the general case the problem of their construction has many solutions however without a special algorithm even finding a single solution may result in unsolvable difficulties. For example one may attempt to search for the pair of paths by first selecting one path connecting the given vertices which is easy to do. Then the path found is removed from the graph without removing its end vertices. In the remaining part of the graph an attempt is then made to connect these vertices again. Though it may happen that the graph breaks into several connected components and the vertices which need to be connected belong to different components.

Such a situation is shown in Fig.2. Solid and dashed lines and their connection points in Fig.2 form a biconnected graph. Let us assume that its vertices α and β have to be connected by a pair of disjoint paths. It is easy to see that this can be done and even in more than one way. However if one takes the path depicted by the thick dashed line in Fig.2 as the first path of the pair then it becomes impossible to find the second path connecting α and β. When the intermediate vertices of the first path are removed from the initial graph then the edges depicted by the thin dashed lines are removed too. The remaining graph contains only the edges depicted by solid lines. It consists of two connected components, with the vertices α and β belonging to different components.

The way the difficulty described above should be addressed is building the both paths at the same time. For this the connecting sequence from which the paths are built should consist of adjacent simple cycles rather than simple edges. Adjacency of two simple cycles means that they have a common edge. All simple cycles in this sequence are joined and their common edges are removed. This results in a simple cycle containing the vertices given provided certain rules of building cycles are fulfilled. The application of the general algorithm of connecting two vertices by a simple cycle is demonstrated below using the graph in Fig.2 and a pair of its vertices α and β.

As the problem of connecting two vertices of a graph by a simple cycle has a solution precisely when the both vertices belong to the same biconnected component of the graph the first step of the solution is to calculate this component or to determine its absence. The known Tarjan-Hopcroft method (Tarjan, 1972; Hopcroft and Tarjan, 1973; Tarjan, 1974) described in many graph manuals (Reingold *et al.*, 1977; Aho *et* *al.*, 1983; Sedgewick, 2002; McConnell, 2007) can be used for this. According to the method all articulation points of the graph have to be found first and then the biconnected component required or its absence can be determined (Note 3). Therefore it is more advisable to use another method – *method of fundamental cycles* proposed in (Golovinskii and Tumakov, 2013; Tumakov and Golovinskii, 2013). It allows one to determine whether the given edge x is a bridge and if not then obtain the biconnected component containing this edge (Note 4). In contrast to the Tarjan-Hopcroft method it is not necessary to calculate all articulation points of the graph. Any edge belonging to an arbitrary path from α to β can be taken for x.

If it is found that the vertices α and β belong to the same biconnected component then the pair of the paths desired should be built. The algorithm of the construction is based on fundamental cycles too. The same system of fundamental cycles as the one found to calculate the biconnected component containing α and β can be used in this algorithm.

In the paper only finite transportation networks are considered. Their graphs are undirected. The direction is introduced on the spanning trees of the graphs. Some details of the solution presented are out of the scope of the paper and are therefore omitted. Due to the same reason the complexity of the algorithm is not analysed either.

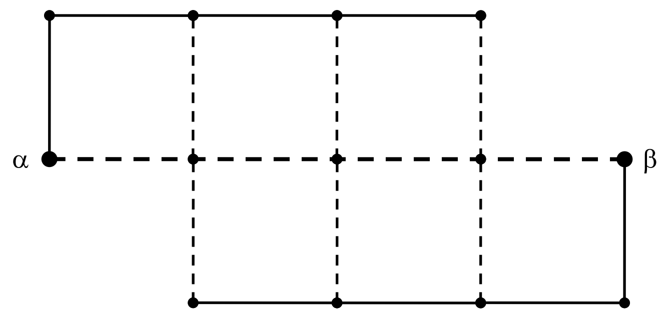

Figure 2: A failed attempt to connect two vertices α and β of a biconnected graph by two simple disjoint paths.

Methods

Construction of the fundamental set of cycles and the structural marking of a graph:

The efficiency of the use of the fundamental system of cycles originates from the fact that this system forms the basis of a linear (over a two-element field) space. The elements of the space are all possible unions of simple cycles with disjointing sets of edges (Berge, 1958; Swamy and Thulasiraman, 1981; Lipski, 1982). The elements of such a basis can be directly operated in calculations. As the term “fundamental cycle” appears many times in the rest of the paper it is called “f-cycle” for brevity.

Recall the definition of the fundamental system of cycles. In a connected graph G an arbitrary spanning tree T is built. The edges of G belonging to T are called branches and the remaining edges - chords (with respect to the spanning tree T). An arbitrary vertex ρ of G is selected as the root of T. If x is a chord then two simple paths P_1 and P_2 run from ρ to the ends of x through T. While moving from ρ the edges of P_1 and P_2 may initially coincide and then separate starting from some vertex φ. The parts of P_1 and P_2 from the vertex φ to the ends of the chord x together with this chord form a simple cycle in the graph G. It is referred to as the *fundamental cycle* (f-cycle) of the graph G with respect to the spanning tree T. Each f-cycle is determined by its own chord. The vertex φ is referred to as the *focus* of the f-cycle. The parts of P_1 and P_2 belonging to a given f-cycle are referred to as its *sides*. One of the sides of an f-cycle may be empty. Then its focus coincides with one of the ends of its chord. The branch of an f-cycle incident to its focus is referred to as the initial one.

If a graph G is not a connected one then the spanning tree, its chords and f-cycles are determined for each connected component of G independently on the other connected components. The set of all f-cycles of a graph is referred to as its fundamental system of cycles (with respect to the specified spanning forest of the graph). A number of papers deal with the methods of the construction of fundamental systems of cycles (Welch, 1966; Gotlieb and Corneil, 1967; etc).

If the fundamental system of cycles of a graph is built then any simple cycle of this graph can be built from the f-cycles of this system by a single operation – symmetric difference of the sets of cycle edges (Harary, 1969; Lipski, 1982). The calculation of a symmetric difference of two simple cycles (not necessarily fundamental) with common edges results in a “mutual cancellation” of these edges. If common edges of two simple cycles form a simple path then their symmetric difference is a simple cycle too (see Fig.3). In the general case a symmetric difference of two simple cycles is a union of several simple cycles with disjoint sets of edges.

Let us write a cycle consisting of edges $a_1,a_2,\ldots,a_k$, as the sum of its edges using the symbol "⊕" to denote addition: $a_1 \oplus a_2 \oplus \ldots \oplus a_k$. The operation of symmetric difference of the sets of cycle edges is interpreted as the addition of linear forms of this kind with the coefficients in the field $\mathbf{Z}_2$ consisting of two elements 0 and 1. This addition is designated by the symbol "⊕" too. The field $\mathbf{Z}_2$ is the residue field modulo 2. Its summation rules are defined as $0\oplus0=1\oplus1=0$; $0\oplus1=1\oplus0=1$ and the multiplication rules as $0\cdot0=0\cdot1=1\cdot0=0$; $1\cdot1=1$. A sum of two cycles over $\mathbf{Z}_2$ results in the coefficients of the common edges being equal to $1\oplus1=0$. Therefore the common edges cancel. The result is the same as the symmetric difference of the sets of cycle edges.

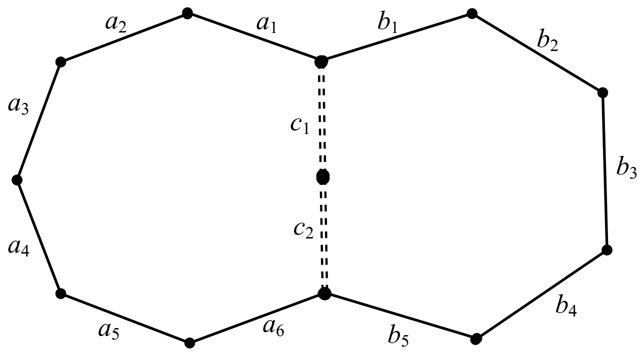

Figure 3: The symmetric difference of two simple cycles. The common edges c_1 and c_2 cancel.

In the example of Fig.3 two simple cycles $a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus c_1 \oplus c_2$ и $b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus c_1 \oplus c_2$ have a common simple path consisting of two edges c_1 и c_2. The result of the addition is a simple cycle $a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5$.

Let us assume that the given graph is connected and that by the beginning of the search for a simple cycle connecting two vertices the spanning tree of the graph and the fundamental system of cycles with respect to this tree have been already built as they are needed for the calculation of the biconnected component of the graph. For an efficient treatment of f-cycles each graph vertex and edge is assigned a structural identifier when the spanning tree is built. An assignment of structural identifiers is called a *structural marking* of the graph. It is needed to connect a pair of vertices by a simple cycle as well. When performing structural marking the root of the spanning tree can be chosen arbitrarily. But the algorithm of this connection requires that the root should be coincident with one of the vertices being connected. If there is no such coincidence then the root should be moved to one of the vertices being connected. Consequently the orientation of the spanning tree and structural marking has to be updated. This is easy to do.

Let us assume that the root of the spanning tree T of a given graph G is moved from the vertex ρ to the vertex β. Then the orientation of the branches along the path from ρ to β through the spanning tree has to be reversed. The orientation of the other branches remains unchanged. The associated change of the structural marking is performed according to simple rules which are omitted for brevity. Let us assume that in the test example these changes have already been done and the root of the spanning tree is β (see Fig. 4).

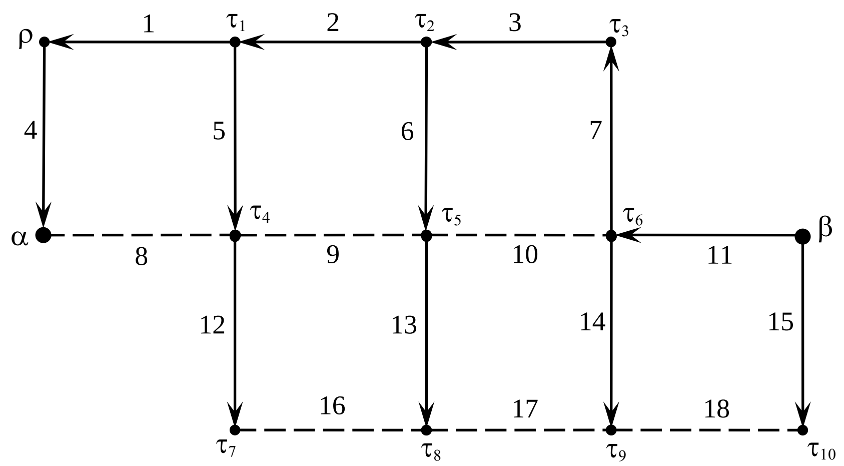

Figure 4: The test graph with a directed spanning tree.

Let us specify the orientation of the branches of the tree *T* as directed from the root β. This orientation determines the partial order relation on the set of the vertices of the directed rooted tree *T* and on the set of the branches of *T*. The order relation is denoted by the symbol "<".

The partial order allows one to assign each vertex and edge a structural identifier containing all information about the location of the vertex or branch in the hierarchy of the vertices and branches. Let us assume that initially all graph vertices and edges are assigned a unique identifier. It is convenient to identify the edges by integer numbers. In the test example in Fig.4 the unique identifiers of the vertices are the symbols $\alpha,\beta,\rho,\tau_1,\tau_2,\tau_3,\tau_4,\tau_5,\tau_6,\tau_7,\tau_8,\tau_9,\tau_{10}$ and the edges – the numbers from 1 to 18. From them hierarchical structural identifiers of all vertices and branches are built in the following way.

The structural identifier of a vertex represents the path through the spanning tree from its root β to the vertex. The root β is assigned the structural identifier ∅. It represents the empty path through the spanning tree from the root β to itself. If the path from the root β to the vertex σ consists of the sequence of branches $a_1,a_2,\ldots,a_k$ then the vertex σ is assigned a structural identifier made up from the same sequence of symbols but separated by a dot $a_1.a_2.\ldots.a_k$. The same structural identifier is assigned to the last branch a_k of the path. In the test example the path from the root β to the vertex τ_5 consists of the branches 11, 7, 3 and 6 (see Fig. 4). Therefore the vertex τ_5 and the branch 6 are assigned the structural identifier "11.7.3.6". In this way all vertices of a tree *T* and of a graph *G* and the branches of the tree *T* are assigned unique structural identifiers.

The chords of f-cycles are shown in Fig.4 by the dashed lines. From the structural identifiers of the ends of the chord one can find the sides of the f-cycle determined by this chord. The number (that is the initial numeric identifier as opposed to the structural identifier) of the chord is then assigned to the f-cycle determined by this chord.

Table 1 gives the structural marking of the vertices and branches of the test graph. The identifiers of the chords of the test graph are given in Table 2. While Table 2 contains the chords it thereby describes the f-cycles of the test graph too. The sides of the f-cycles are described in Table 2 by the structural identifiers of the ends of its chord. In the identifier of the side of an f-cycle the structural identifier of its focus is specified in square brackets. It is the path from the root of the spanning tree to the focus of the f-cycle. An empty path is denoted by the symbol ∅. Outside the square brackets the identifier of the side of an f-cycle specifies the sequence of the side branches of the f-cycle belonging to this side. Knowing the edges which are members of the sides of the f-cycle the last column of Table 1 can be filled.

Table 1: The structural marking of the test graph

Branch No.	Branch start	Branch end	Structural identifier of the branch and its end	Numbers of the f-cyles containing the branch
1	τ_1	ρ	11.7.3.2.1	8
2	τ_2	τ_1	11.7.3.2	9, 16
3	τ_3	τ_2	11.7.3	10, 17
4	ρ	α	11.7.3.2.1.4	8
5	τ_1	τ_4	11.7.3.2.5	8, 9, 16
6	τ_2	τ_5	11.7.3.6	9, 10, 16, 17
7	τ_6	τ_3	11.7	10, 17
11	β	τ_6	11	18
12	τ_4	τ_7	11.7.3.2.5.12	16
13	τ_5	τ_8	11.7.3.6.13	16, 17
14	τ_6	τ_9	11.14	17, 18
15	β	τ_{10}	15	18

Table 2. The fundamental system of cycles of the test graph

Chord No.	Chord vertex 1	Chord vertex 2	F-cycle side 1	F-cycle side 2	F-cycle focus
8	α	τ_4	[11.7.3.2].1.4	[11.7.3.2].5	11.7.3.2
9	τ_4	τ_5	[11.7.3].2.5	[11.7.3].6	11.7.3
10	τ_5	τ_6	[11].7.3.6	[11].∅	11
16	τ_7	τ_8	[11.7.3].2.5.12	[11.7.3].6.13	11.7.3
17	τ_8	τ_9	[11].7.3.6.13	[11].14	11
18	τ_9	τ_{10}	[∅].11.14	[∅].15	∅

Unifocal families of fundamental cycles:

The algorithm of connecting two graph vertices by a simple cycle is based on some structure properties of the fundamental system of cycles of this graph. Let a biconnected graph G have a directed spanning tree T. We shall say that two f-cycles determined by the spanning tree T are *confocal* if their foci coincide. A set of f-cycles determined by directed rooted spanning tree T and having the focus in the vertex φ is called a *unifocal family of fundamental cycles* with the focus φ and is denoted by $\Omega(\varphi)$.

If two f-cycles have a common edge then this edge can only be a branch. We shall say that two f-cycles are adjacent if they have a common branch. Let us consider the structure of the adjacency of f-cycles in a unifocal family.

Let two adjacent f-cycles F_1 and F_2 have a common focus φ and a common branch a. Any branch of the spanning tree which precedes the branch a and the beginning of which does not precede the focus φ is common to the f-cycles F_1 and F_2 too. Therefore two confocal f-cycles have a common edge if and only if they have a common branch beginning from their common focus. This means that the structure of the adjacency of a unifocal family of f-cycles is fully determined by their common branches beginning from their common focus.

This structure of adjacency can be represented by a bigraph (bipartite graph), which we denote by $D(\varphi,V_1,V_2)$ or $D(\varphi)$. Let the parts V_1 и V_2 of this bigraph represent in a one-to-one manner the branches beginning from the focus φ and the f-cycles with the focus φ, respectively. Let Δ be the one-to-one mapping of the union of the set of f-cycles with the focus φ and the set of the branches beginning from φ onto the set of the vertices of the bigraph $D(\varphi,V_1,V_2)$. If a is a branch coming out from φ then $\Delta(a) \in V_1$. If the focus of an f-cycle F is φ then $\Delta(F) \in V_2$. Let us assume by definition that the bigraph $D(\varphi,V_1,V_2)$ contains the edge $(\Delta(a),\Delta(F))$ if and only if the branch a belongs to the f-cycle F. We shall say that the bigraph $D(\varphi,V_1,V_2)$ is the *adjacency bigraph* of the f-cycles of the unifocal family $\Omega(\varphi)$.

The bigraph $D(\varphi,V_1,V_2)$ is built by the selection of those cycles from the table of f-cycles whose focus is φ. In the test example this is Table 2. From the analysis of the identifiers of these f-cycles their initial branches can be found.

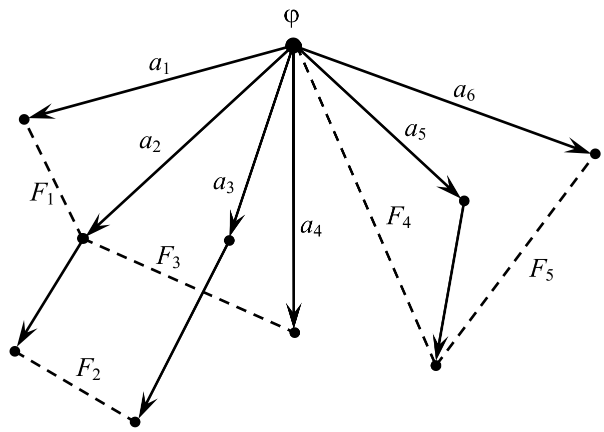

Figure 5: An example of a unifocal family of f-cycles.

An example of a unifocal family of f-cycles is shown in Fig.5. The chords of f-cycles are depicted by the dashed lines. Fig.6 gives the bigraph $D(\varphi, V_1, V_2)$ representing the structure of the adjacency of the f-cycles of this family. The dashed lines in Fig.6 correspond to the parts of $D(\varphi, V_1, V_2)$. The symbol Δ and the parentheses are omitted in the denotations of the verttices $\Delta(a_1), \Delta(a_2), \ldots, \Delta(F_1), \Delta(F_2), \ldots$

The sequence $(F_1, F_2, \ldots, F_K)$ of confocal f-cycles of a graph G in which each two neighbouring f-cycles have a common edge can be depicted as a "fan" (see Fig. 7). We shall say that it is a *fan* and the common focus of all f-cycles of a fan is a *fan focus*. In the adjacency bigraph $D(\varphi, V_1, V_2)$ the simple path $\Delta(a_0)$-$\Delta(F_1)$-$\Delta(a_1)$-$\Delta(F_2)$-$\Delta(a_2)$-…-$\Delta(F_K)$-$\Delta(a_K)$ corresponds to the fan $(F_1, F_2, \ldots, F_K)$ where $a_0, a_1, a_2, \ldots, a_K$ are branches, $F_1, F_2, \ldots, F_K$ are f-cycles. The fan (F_1, F_2, F_3, F_4) is shown in Fig. 7.

The sum

$$F_1 \oplus F_2 \oplus \ldots \oplus F_K$$

over the field $\mathbf{Z}_2$ results in the cancellation of the common edges of adjacent cycles as each such edge is present strictly in two cycles. The set of the remaining edges is called a *fan boundary*. In Fig. 7 the fan boundary is marked by the bold lines.

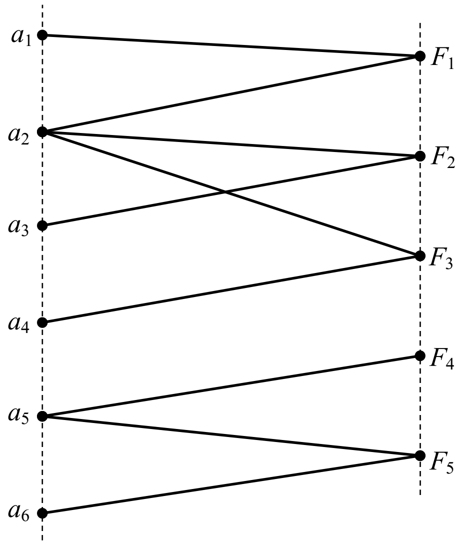

Figure 6: Bigraph $D(\varphi, V_1, V_2)$ representing the structure of the adjacency of the unifocal family of f-cycles shown in Fig. 5.

In the main algorithm three following theorems are used. Their proofs are outside the scope of the paper.

Theorem 1. The boundary of a fan is a simple cycle.

Theorem 2. Let a biconnected graph have a directed rooted spanning tree T, φ be a vertex different from the root of tree T and $D(\varphi, V_1, V_2)$ be an adjacency bigraph of the f-cycles of a unifocal family $\Omega(\varphi)$. Then each connected component of the bigraph $D(\varphi, V_1, V_2)$ has such a vertex ζ that $\Delta^{-1}(\zeta)$ is a branch of the f-cycle the focus of which lies on the directed rooted tree T closer to its root than φ.

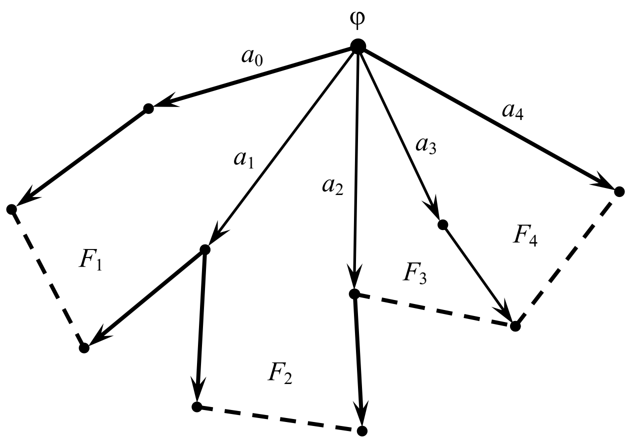

Figure 7: The fan of confocal f-cycles F_1, F_2, F_3, F_4.

Fig. 8 shows the same unifocal family as in Fig. 5 but together with two those f-cycles the existence of which is stated by Theorem 2. These two f-cycles F' and F'' are marked by the bold lines.

Theorem 3. Let a biconnected graph have a directed rooted spanning tree T, φ be a vertex different from the root of the tree T, x be a branch coming into the vertex φ. If the set A(φ) of the branches coming out of the vertex φ is not empty then there exists an f-cycle containing the branch x and one of the branches of the set A(φ).

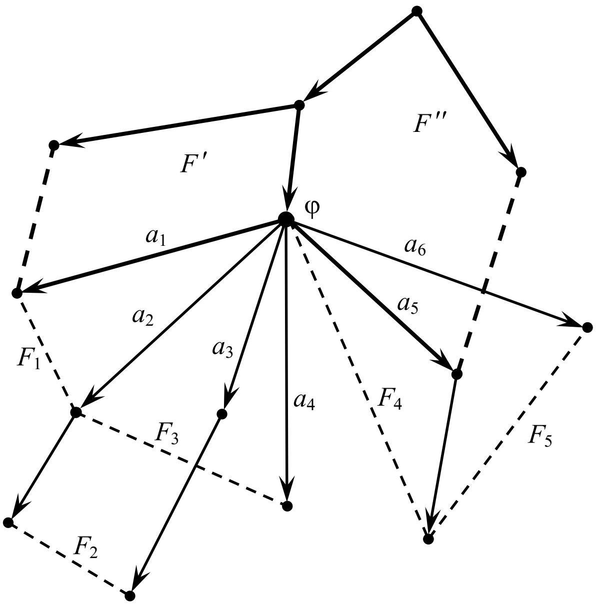

Figure 8: Illustration to Theorem 2.

Algorithm of connecting two graph vertices by a simple cycle

General description and proof:

Assume that a directed rooted spanning tree T of the connected graph G with a root β have been built, the structural marking of the graph G has been done and its results have been put into two tables similar to Tables 1 and 2. By P denote an oriented path through the tree T connecting the vertex α to the root β and directed from β to α. Let us begin to build a pair of the paths sought from α and gradually move to β along the path P. Since the graph G is a biconnected one the degree of the vertex α is not less than two. As α≠β then one of the edges incident to the vertex α is an incoming branch. Let us take another edge incident to the vertex α and by α′ denote the other end of this edge. Let us build two simple paths from the vertices α and α′ to the vertex β such that these paths intersect only in β. For this let us gradually split the path P into two “parallel” paths moving from α towards β. The splitting is performed by adjoining f-cycles.

If at least one chord is incident to the vertex α then let us denote it by x_0 and the f-cycle corresponding to this chord by F_1. If no chord is incident to the vertex α then it has at least one outcoming branch. According to Theorem 3 one of the branches coming out of α together with a branch coming into α belong to some f-cycle. Let us then take the branch coming out of from α and belonging to this f-cycle as x_0 and denote the f-cycle itself as F_1.

If x_0 is a branch then the number of the f-cycle F_1 can be found from the number (initial numeric identifier) of the branch x_0 in the table similar to Table 1. If x_0 is a chord then the number of the f-cycle F_1 is the same as the number of this chord. By φ_1 denote the focus of the f-cycle F_1. If $\varphi_1=\beta$ then the problem of connecting the vertices α and β by a simple cycle has been solved: a simple cycle F_1 containing the vertices α and β has been found.

Consider the case $\varphi_1 \neq \beta$. The focus φ_1 lies on the directed path P belonging to tree T closer to the tree root β than the vertex α. The cycle F_1 contains two simple paths connecting the ends of the edge x_0 to the focus φ_1 and intersecting only in φ_1. One of these paths is a part of the path P and the other one is formed by the remaining edges of the cycle F_1 less the edge x_0. By x_1 denote the edge incident to the vertex φ_1 in this second path. This edge can be either a chord or a branch. The situation is shown in Fig. 9 in the case when x_0 and x_1 are branches. By y_1 denote the edge of the first path incident to the vertex φ_1. It is a branch coming out of φ_1. The only branch coming into φ_1 is denoted as z_1.

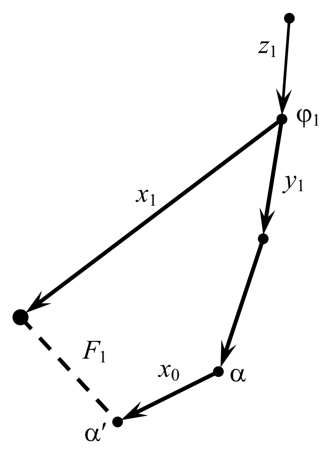

Figure 9: An f-cycle $F1$ in the case when x_0 and x_1 are branches.

According to Theorem 3 there is such a f-cycle F_2 which contains the branch z_1 and one of the branches coming out of φ_1. This outcoming branch is denoted by w_1. It may coincide either with y_1 or with x_1 or may coincide with none of them. Among such f-cycles some ones have a focus located on a directed path P most closely to the root β.

If among the f-cycles with this property there is a cycle having $w_1=y_1$ or $w_1=x_1$ then let us select it as F_2. Otherwise let us select the cycle F_2 such that its branch w_1 have the property $\Delta(w_1) \in C_1$ where C_1 is the connected component of the bigraph $D(\varphi_1)$ for which $\Delta(y_1) \in C_1$. Such an f-cycle exists according to Theorem 2. Recall that the bigraph $D(\varphi_1)$ represents the adjacency relation between the f-cycles of a unifocal family $\Omega(\varphi_1)$ by the branches coming out of φ_1. If $\Delta(y_1) \in C_1$ and x_1 is a branch then $\Delta(x_1) \in C_1$ too. By φ_2 denote the focus of the f-cycle F_2.

Consider the cases $w_1=y_1$ and $w_1=x_1$ first. In these cases the simple cycle Q_2 connecting the vertices α and φ_2 is given by the formula $Q_2 = F_1 \oplus F_2$. The branch w_1 and possibly some other branches following it on the spanning tree are common to the cycles F_1 and F_2. The cycle Q_2 is a simple one because the common branches of the cycles F_1 and F_2 form a simple path belonging to one side of the f-cycle F_1 and one side of the f-cycle F_2. Fig. 10a illustrates this situation when x_1 is a chord, Fig. 10b – when x_1 is a branch common to F_1 and F_2. The simple cycle Q_2 is marked by the bold lines. The double dashed lines depict the branches which cancel in the sum $F_1 \oplus F_2$.

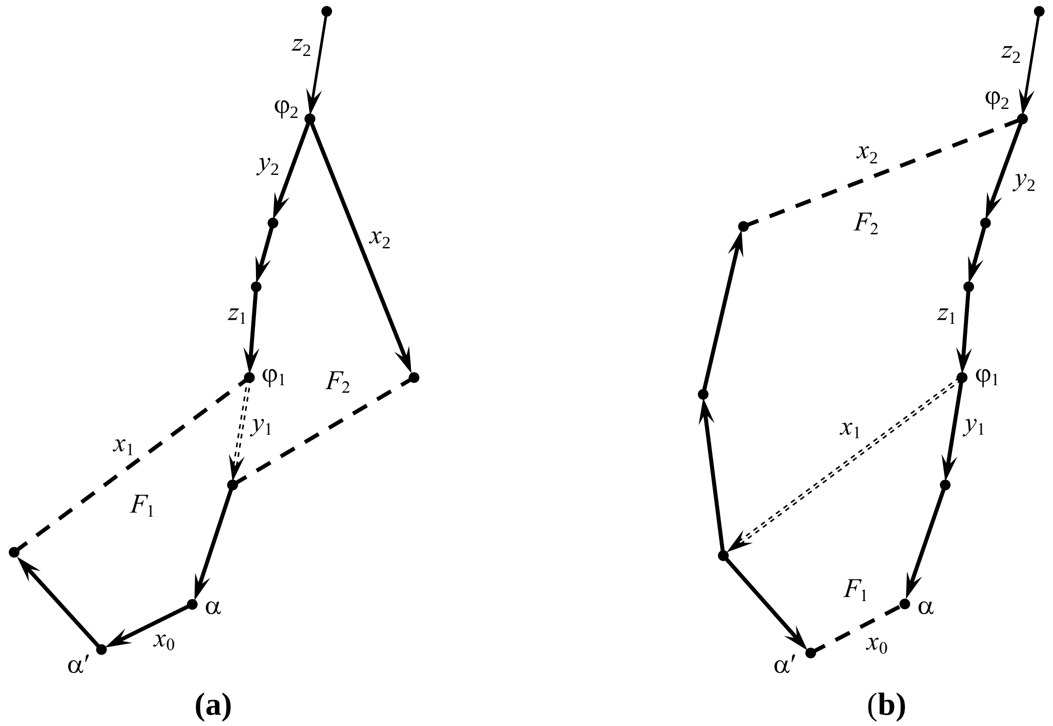

Figure 10: The case of adjacent f-cycles F_1 and F_2: a - x_1 is a chord, b - x_1 is a branch.

Consider the case when the branch w_1 coincide with neither y_1 nor x_1. As the f-cycles F_1 and F_2 do not have a common branch one has to connect them into one simple cycle by an intermediate chain of pairwise-adjacent f-cycles of the unifocal family $\Omega(\varphi_1)$. This gives a fan with the focus φ_1 the boundary of which contains the branches y_1 and w_1.

The fan sought can be found by analyzing the topology of the bigraph $D(\varphi_1)$. Fig. 11 presents the situation when the f-cycles F_1 and F_2 are connected by the use of the fan $(f_{1,0},f_{1,1},f_{1,2})$. The adjacency bigraph of the f-cycles of this fan is shown in Fig. 12. The symbol Δ and the parentheses are omitted in the denotations of the bigraph vertices.

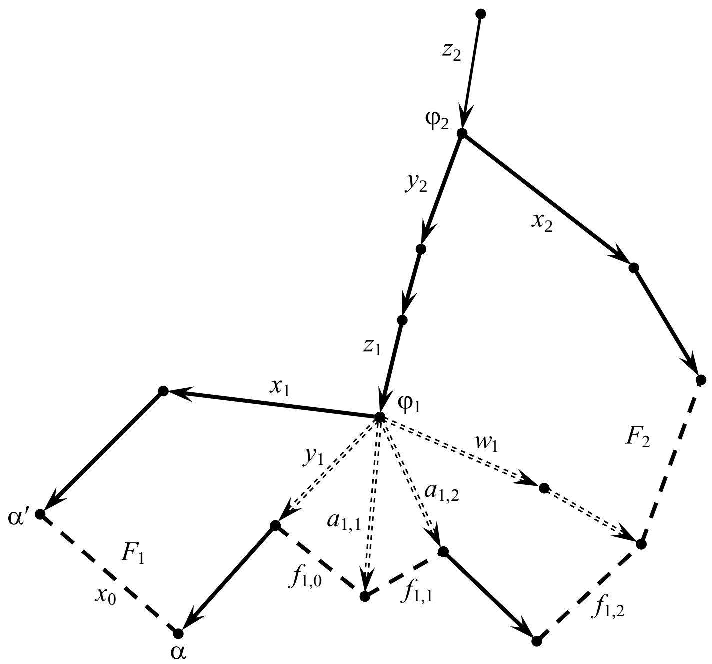

Figure 11: The chain y_1-$f_{1,0}$-$a_{1,1}$-$f_{1,1}$-$a_{1,2}$-$f_{1,2}$-w_1 of pairwise-adjacent f-cycles of the fan $(f_{1,0},f_{1,1},f_{1,2})$ connecting the branches y_1 and w_1.

Let us find a simple path connecting the vertices $\Delta(y_1)$ and $\Delta(w_1)$ in the connected component C_1 of the bigraph $D(\varphi_1)$. It can be obtained by building a spanning tree of the graph C_1. Let us assume that the path sought has been built:

$$\Delta(y_1) - \Delta(f_{1,0}) - \Delta(a_{1,1}) - \Delta(f_{1,1}) - \Delta(a_{1,2}) - \Delta(f_{1,2}) - \ldots - \Delta(a_{1,k_1}) - \Delta(f_{1,k_1}) - \Delta(w_1)\ .$$

Here $f_{1,0},f_{1,1},f_{1,2},\ldots,f_{1,k_1}$ are the pairwise-adjacent f-cycles with the common focus φ_1; $a_{1,1},a_{1,2},\ldots,a_{1,k_1}$ are the branches with the common origin φ_1 which are common in the adjacent f-cycles. The sequence of the confocal f-cycles $F_1,f_{1,0},f_{1,1},f_{1,2},\ldots,f_{1,k_1}$ forms a fan (see Fig. 11). According to Theorem 1 the sum of these f-cycles over the field $\mathbf{Z}_2$ is a simple cycle. The f-cycle F_2 is added to it over $\mathbf{Z}_2$. This gives the graph

$$Q_2 = F_1 \oplus f_{1,0} \oplus f_{1,1} \oplus \ldots \oplus f_{1,k_1} \oplus F_2\ , \tag{1}$$

containing the vertices α and φ_2 (see Fig. 11).

From building the fan (F_1,$f_{1,0}$,$f_{1,1}$,$f_{1,2}$,…,f_{1,k_1}) the f-cycle F_2 has only those common branches with the simple cycle $F_1 \oplus f_{1,0} \oplus f_{1,1} \oplus \ldots \oplus f_{1,k_1}$ which belong to the intersection $f_{1,k_1} \cap F_2$. With the other terms of the sum specified the f-cycle F_2 does not have common vertices. Therefore the graph Q_2 given by the formula (1) is a simple cycle.

The situation is illustrated in Fig. 11 for k_1=2. The edges of the cycle Q_2 are marked by the bold lines. The symbols $f_{1,0}$, $f_{1,1}$ and $f_{1,2}$ in Fig. 11 denote the chords of the corresponding f-cycles.

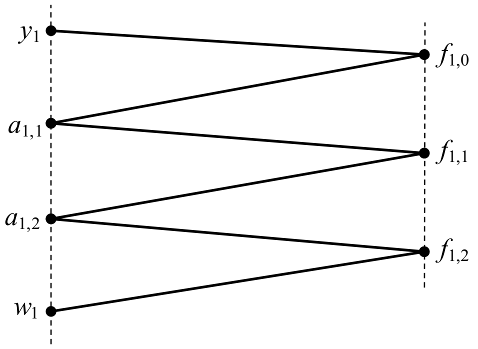

Figure 12: The adjacency bigraph of the f-cycles of the fan ($f_{1,0}$,$f_{1,1}$,$f_{1,2}$) for the example of Fig. 11.

If the focus φ_2 of the f-cycle F_2 is coincident with β then the problem of connecting the vertices α and β by a simple cycle has been solved: the simple cycle sought is Q_2. Otherwise one can apply the same speculations and steps with respect to the vertex φ_2 as were done for the vertex φ_1. By y_2 denote the branch coming out of φ_2 and belonging to the simple cycle Q_2. The other edge of Q_2 incident to the vertex φ_2 is denoted by x_2. It can be a chord or a branch coming out of φ_2. The branch coming into the vertex φ_2 is denoted by z_2.

Let $D(\varphi_2)$ be the bigraph representing the adjacency relation between the f-cycles of a unifocal family $\Omega(\varphi_2)$. By C_2 denote such the connectivity component of the bigraph $D(\varphi_2)$ that $\Delta(y_2) \in C_2$. According to Theorem 2 there is an f-cycle containing the branch z_2 and such a branch w_2 coming out of φ_2 that $\Delta(w_2) \in C_2$. Consider two cases separately: 1) there is such an f-cycle among these f-cycles that w_2=y_2 or w_2=x_2; 2) there is no such an f-cycle among these f-cycles that w_2=y_2 or w_2=x_2.

In the first case let us arbitrarily select one of the f-cycles the coinciding foci of which lie most closely to β from all f-cycles having w_2=y_2 or w_2=x_2. By F_3 denote this f-cycle and by φ_3 denote its focus: $\varphi_3 < \varphi_2$. Let us add the f-cycle F_3 over $\mathbf{Z}_2$ to the simple cycle Q_2: $Q_3 = Q_2 \oplus F_3$. In the case of w_2=x_2 the cycle Q_3 is obviously a simple one as the edges common to the f-cycle F_3 and the cycle Q_2 are exactly the same as the ones common to F_3 and F_2. This is the branch x_2 and possibly the branches on the directed rooted tree T continuously following x_2.

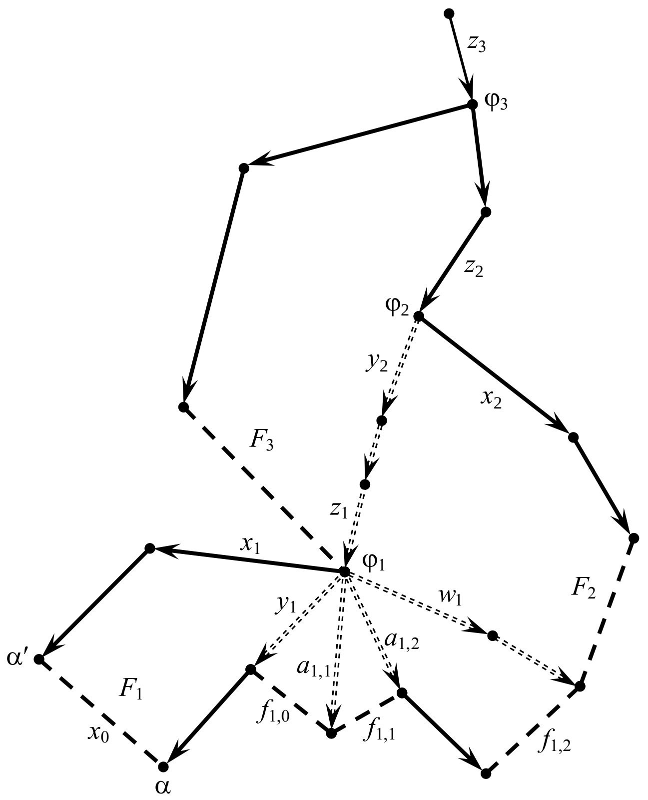

Figure 13: The case of the f-cycle F_3 attached to the f-cycle F_2 from the side belonging to the path on the spanning tree from the root β to α.

The situation is somewhat more complicated for $w_2=y_2$ (see Fig. 13). In this case while summing $Q_2 \oplus F_3$ the f-cycle F_3 is attached to the f-cycle F_2 from that side which belongs to the path P going from β to α. Only the branches between φ_2 and φ_1 can be common to the f-cycles F_3 and F_2. This follows from the fact that if a branch coming out of φ_1 belongs to the f-cycle F_3 then it contradicts to the selection of the f-cycle F_2 as having the minimum focus among the f-cycles containing a branch coming out of φ_1. Therefore all edges of the set $F_3 \cap F_2$ belong to the cycle Q_2. While summing the simple cycles Q_2 and F_3 over $\mathbf{Z}_2$ the edges of the set $F_3 \cap F_2$ cancel out and the resulting cycle Q_3 happens to be a simple one too.

Fig. 13 shows the worst from the possible configurations: the chain of the common branches of the cycles F_3 and F_2 may finish in the vertex φ_1 but may not pass it. Even in this case the graph Q_3 is a simple cycle.

If however the branch w_2 coincides with neither x_2 nor y_2 then among all f-cycles containing the branch coming into φ_2 and the branch coming out of φ_2 one has to select that one the focus of which lies most closely to β. By F_3 denote this f-cycle, by φ_3 denote its focus, so that $\varphi_3 < \varphi_2$. Based on the analysis of the topology of the structural bigraph $D(\varphi_2)$ one can build the fan $(f_{2,0}, f_{2,1}, f_{2,2}, \ldots, f_{2,k_2})$ with the focus φ_2

connecting the branches y_2 and w_2. If it happens that the side of the f-cycle $f_{2,0}$ starting from the branch y_2 also contains the branch y_1 (that is passes over the vertex φ_1) then the f-cycle F_2 has to be replaced by $f_{2,0}$. This means one goes back to the previous step of the solution of the problem and takes for F_2 the f-cycle $f_{2,0}$ which was used in the construction of the fan connecting the branches y_2 and w_2. If such a replacement takes place then one has to remove the first term $f_{2,0}$ from the fan $(f_{2,0},f_{2,1},f_{2,2},\ldots,f_{2,k_2})$. Then cycle Q_3 is obtained by the formula

$$Q_3 = Q_2 \oplus f_{2,1} \oplus f_{2,2} \oplus \ldots \oplus f_{2,k_2} \oplus F_3 . \qquad (2)$$

The simplicity of the cycle Q_3 obtained in such a way results from the following. The sum $f_{2,1}\oplus f_{2,2}\oplus\ldots\oplus f_{2,k_2}$ is the boundary of the fan of confocal f-cycles and therefore is a simple cycle (Theorem 1). The f-cycle F_3 has only those common branches with this simple cycle which belong to the intersection $f_{2,k_2}\cap F_3$. There are no common vertices of the f-cycle F_3 with the other terms of the sum specified. Therefore the sum $f_{2,1}\oplus f_{2,2}\oplus\ldots\oplus f_{2,k_2}\oplus F_3$ is a simple cycle. It has only those common branches with the cycle Q_2 which are common between this sum and F_2. From this it follows that the entire sum (2) is a simple cycle.

If however a side of the f-cycle $f_{2,0}$ starting from the branch y_2 does not contain the branch y_1 (that is does not pass over the vertex φ_1) then the replacement of F_2 to $f_{2,0}$ does not have to be done. In this case the cycle Q_3 is obtained by the formula

$$Q_3 = \mathrm{Q}_2 \oplus f_{2,0} \oplus f_{2,1} \oplus f_{2,2} \oplus \ldots \oplus f_{2,k_2} \oplus F_3 . \qquad (3)$$

The simplicity of the cycle Q_3 obtained from the formula (3) results from the following. The sum $f_{2,0}\oplus f_{2,1}\oplus f_{2,2}\oplus\ldots\oplus f_{2,k_2}$ is the boundary of the fan of confocal f-cycles and therefore is a simple cycle (Theorem 1). The f-cycle F_3 has only those common branches with this simple cycle which belong to the intersection $f_{2,k_2}\cap F_3$. There are no common branches of the f-cycle F_3 with the other terms of the sum specified. Therefore the sum $S = f_{2,0}\oplus f_{2,1}\oplus f_{2,2}\oplus\ldots\oplus f_{2,k_2}\oplus F_3$ is a simple cycle. As $f_{2,0}$ does not contain the branch y_1 it has no common branches with F_1. The simple cycle S does not have them either. Therefore $Q_2\cap S = F_2\cap S$ from which the simplicity of the cycle Q_3 obtained from the formula (3) follows.

If the focus φ_3 coincides with the root β of the directed rooted tree T then the problem has been solved: the simple cycle Q_3 contains the vertices α and β. Otherwise one has to continue the process of the adjoining of f-cycles: one finds f-cycles $F_4,F_5,\ldots$, connects them by fans if necessary and obtains graphs $Q_4,Q_5,\ldots$. The proof that these graphs are simple cycles is done in the same way as for the cycle Q_3. On the k-th step one obtains the focus φ_k of the f-cycle F_k lying on the directed path from β to α closer than the focus φ_{k-1} of the f-cycle F_{k-1}. For some $N > 0$ it happens that $\varphi_N = \beta$. The simple cycle Q_N is then the solution of the problem.

The application of the algorithm to the test example:

The reasoning of the general algorithm seems to be somewhat cumbersome but its application turns out to be relatively simple. Let us apply the general algorithm to the test graph depicted in Fig. 4 in order to connect the vertices α and β by a simple cycle. The process of the adjoining of f-cycles is specified in Table 3. It happens to consist of four steps only that is four f-cycles have to be summed to come to the solution. The result of the process is shown in Fig. 16.

Step 1. Chord 8 is incident to the vertex α. One assumes $F_1=F(8)$. The focus of the f-cycle $F(8)$ is τ_1 that is $\varphi_1=\tau_1$. Since $\varphi_1\neq\beta$ the process is continued.

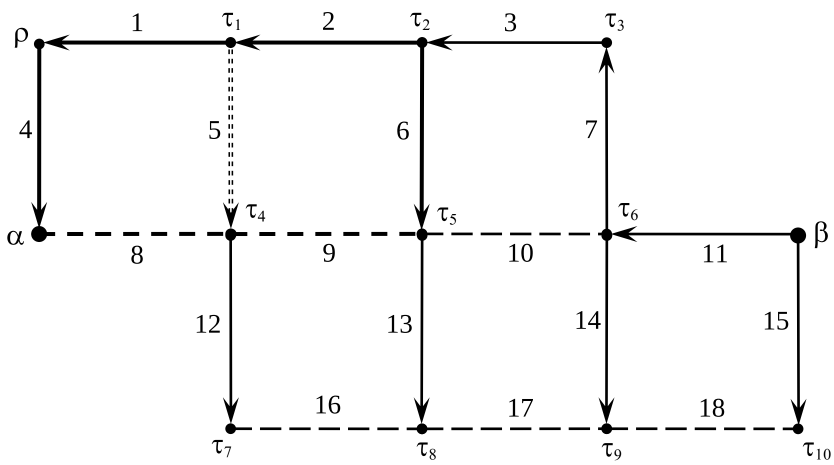

Figure 14: The simple cycle Q_2 in the test example.

Step 2. No chord is incident to the vertex $\varphi_1=\tau_1$. But from this vertex branch 5 comes out. From Table 1 we find that it belongs to f-cycles $F(9)$ and $F(16)$. From Table 2 we see that the common focus τ_2 of these f-cycles lies on the directed path from β to α closer to β than $\varphi_1=\tau_1$. Therefore we set $F_2=F(9)$ and $\varphi_2=\tau_2$. The simple cycle $Q_2=F_1\oplus F_2$ is marked by the bold lines in Fig. 14. It contains the vertices α and τ_2. Since $\varphi_2\neq\beta$ the process is continued.

Step 3. No chord is incident to the vertex $\varphi_2=\tau_2$. From this vertex branch 6 comes out belonging to two f-cycles $F(10)$ and $F(17)$ having a common focus τ_6. It is more convenient to select f-cycle $F(17)$ as it has a common branch 14 together with f-cycle $F(18)$ the focus of which is located closer to β than φ_2. F-cycle $F(10)$ does not have such property. Actually the focus of f-cycle $F(18)$ coincides with β. Therefore we take $F(17)$ as F_3. Consequently $\varphi_3=\tau_6$ and $Q_3=Q_2\oplus F_3$. The simple cycle Q_3 is marked by the bold lines in Fig. 15. It connects the vertices α and τ_6. Since $\varphi_3\neq\beta$ the process is continued.

Step 4. From the vertex $\varphi_3=\tau_6$ branch 14 comes out belonging to f-cycle $F(18)$ the focus of which is β. We set $F_4=F(18)$ and $Q_4=Q_3\oplus F_4$. The simple cycle $Q_4=F_1\oplus F_2\oplus F_3\oplus F_4$ connects the vertices α and β and is the solution of the problem. It is shown by the bold lines in Fig. 16.

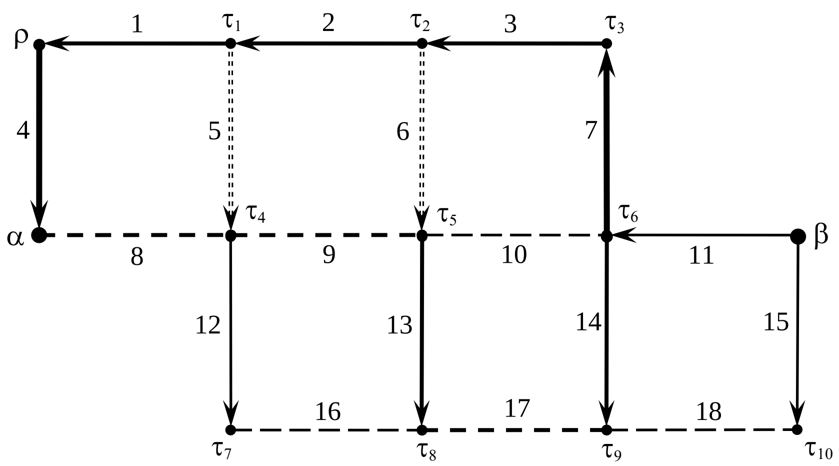

Figure 15: The simple cycle Q_3 in the test example.

Table 3: The construction of a simple cycle connecting the vertices α and β of the test graph

Step No.	Number of the branch or chord coming out of the focus	Number of the adjoining f-cycle	Focus of the adjoining f-cycle	Structural identifier of the focus of the adjoining f-cycle
1	8	8	τ_1	11.7.3.2
2	5	9	τ_2	11.7.3
3	6	17	τ_6	11
4	14	18	β	∅

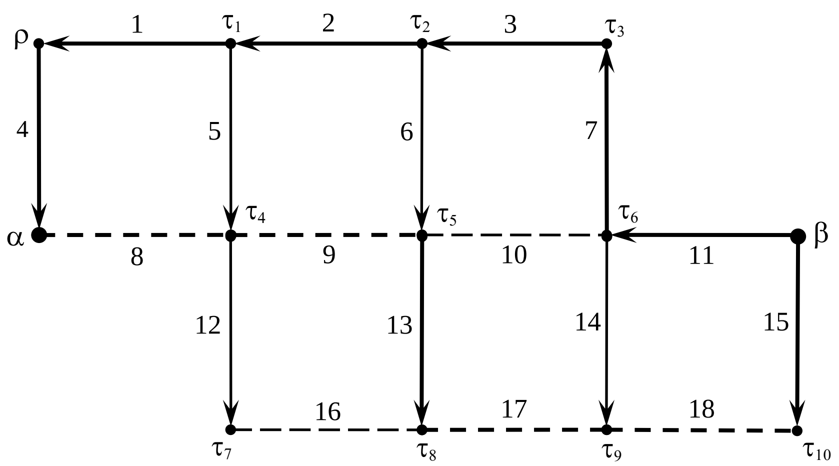

Figure 16: The result of the construction of a simple cycle connecting the vertices α and β of the graph of the test example.

Conclusion

One of the basic methods of increasing the reliability of a transportation network is to provide redundancy of the connections between its nodes. The simplest indication of the existence of such redundancy is the presence of vertex biconnectivity of the graph of the network. Such a network is referred to as non-separable because it remains connected when a node or a branch is removed from it. Theory states that in a vertex biconnected graph any two vertices α and β can be connected by a simple cycle that is by two paths having no other common vertices than α and β. However the theorem of the existence of a connecting simple cycle does not give a practically useful method of its construction. A simple example in the paper demonstrates that an attempt to build a pair of such paths by intuition may easily fail.

The general correct (that is not relying on simplified euristic attempts subject to failures) solution proposed in the paper is based on the theory of fundamental systems of cycles. The algorithm is reduced to the construction of such a chain of fundamental cycles $F_1,F_2,\ldots,F_N$ in which $\alpha \in F_1$, $\beta \in F_N$ and each two neighbouring fundamental cycles have a common edge. Because of a special way to select the cycles $F_1,F_2,\ldots,F_N$ the sum $F_1 \oplus F_2 \oplus \ldots \oplus F_N$ (where the symbol "$\oplus$" designates a summation of the coefficients in a two-element field $\mathbf{Z}_2$) turns out to be a simple cycle. This simple cycle contains the vertices α and β. An efficient solution is produced due to a deep analysis of the topology of the graph. The solution demonstrates a practical significance of the method of fundamental cycles for the analysis and monitoring of transport networks reliability.

Acknowledgment

This work was financially supported by the Ministry of Education and Science of the Russian Federation. The paper contains the results of the research work performed within the project «The development of an adaptive and coordinating device for the operating control of switchovers in a smart electric grid», project No. RFMEFI57514X0077, in the National Research University of Electronic Technology (MIET).

References

[1] Aho, A.V., Hopkroft, J.E., & Ullman, J.D. (1983). *Data Structures and Algorithms*. Addison-Wesley.

[2] Berge C. (1958). *Théorie des graphs et ses applications*. Dunod, Paris.

[3] Bondy, J.A., & Murty, U.S.R. (1982). *Graph Theory with Applications*. Elsevier Science.

[4] Colboum, C. J. (1987). *The Combinatorics of Network Reliability*. Oxford University Press.

[5] Diestel, R. (2000). *Graph Theory*. Springer.

[6] Golovinskii, I.A., & Tumakov, A.V. (2013). Graph biconnection analysis by means of joining fundamental cycles. *Information technologies of modelling and control 2*(80), 92-105 (Russian).
[7] Gotlieb, C. C., & Corneil, D. G. (1967). Algorithms for finding a fundamental set of cycles for an undirected linear graph. *Communications of the ACM*. *10*(12), 780–783. // DOI: 10.1145/363848.363861.
[8] Harary, F. (1969). *Graph Theory*. Addison-Wesley.
[9] Hopcroft, J.E., & Tarjan, R.E. (1973). Efficient algorithms for graph manipulation. // *Comm. ACM*, *16* (6), 372-378.
[10] Lipski, W. (1982). *Kombinatoryka dla programistów*. Wydawnictwa Naukowo-Techniczne, Warszawa.
[11] McConnell, J. (2007). *Analysis of Algorithms: An Active Learning approach (2nd Edition)*. Jones & Bartlett Publishers.
[12] Ore, O. *Theory of Graphs*. (1962). American Mathematical Society.
[13] Reingold, E.M., Nievergelt, J., & Deo, N. (1977). *Combinatorial Algorithms: Theory and Practice*. Prentice Hall.
[14] Sedgewick, R. (2002). *Algorithms in C++ Part 5: Graph Algoritms (3rd Edition) (Pt.5)*. Addison-Wesley.
[15] Shier, D. R., (1991). *Network Reliability and Algebraic Structures*. Clarendon Press Oxford.
[16] Swamy, M.N.S., & Thulasiraman, K. (1981). *Graphs, Networks and Algorithms*. Whiley Interscience.
[17] Tarjan, R.E. (1972). Depth first search and linear graph algorithms. // *BIAM J. Computing*, *1*(2), 146-160.
[18] Tarjan, R.E. (1974). A note on finding the bridges of a graph. *Information Processing Letters*, *2*(6), 160-161.
[19] Tumakov, A.V., & Golovinskii, I.A. (2013). Graph biconnection analysis: foundation and variations of the method of fundamental cycles. *Information technologies of modelling and control*, *2*(80), 123-137 (Russian).
[20] Welch, J. T., Jr. (1966). A mechanical analysis of the cyclic structure of undirected linear graphs. *J. ACM, 13*, (2), 205-210.
[21] Whitney, H. (1932). Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1), 150–168.

Notes :

Note 1. A path connecting vertices A and B of an undirected graph is such a sequence of graph edges that each pair of the neighbouring edges has exactly one common vertex. A path determines the sequence of vertices passed when moving from A to B. A path is called simple if the vertices in the sequence do not repeat.

Note 2. A path connecting vertices A and B is called a cycle if A = B. A cycle is called simple if all its vertices have a degree of 2.

Note 3. A graph vertex is called an articulation point if it is incident to two edges which do not belong to the same biconnected component.

Note 4. A graph edge is called a bridge if it does not belong to any biconnected component.