0-1 Algebraic Approach Algorithm for the Determination of Component Minimal Complete Subgraph

Silviu Bârză
Faculty of Mathematics and Informatics
Spiru Haret University
Bucharest, Romania
silviu.barza@gmail.com

Abstract
Giving a graph G we wish to determine the component minimal complete subgraph of G. In previous papers we gave algorithms in terms of sets representation of graph G. The goal of this paper is to present the same problem in 0-1 algebraic approach. We do not give the theorem that characterized the algorithm we propose which will be presented in a future paper.
Keywords: *subgraphs, complete graphs*
ACM/AMS Classification: 05C30

1. General definitions and results

Firstly I wish to give some of the graph theory definitions (see [Bârză and Morogan 2008]) and their corresponding results in 0-1 algebraic representation of graphs. Also I wish to take the results related to component minimal complete subgraphs (see [Bârză, 2012]) and to specify them in terms of 0-1 representation of graphs.

Definition 1.1 (sets) *Let*

$$V = \{x_1, x_2, \ldots, x_n\}$$

be a finite and non-empty set and

$$E = \{\{x, y\} \,|x, y \in X, x \neq y\}\,.$$

*The pair $G = (V, E)$ is a **graph**, elements in V are named **vertices** and element in E are named **edge**.*

Definition 1.1 (0-1 algebraic) *Let*

$$V = \{x_1, x_2, \ldots, x_n\}$$

*be a finite and non-empty set and $A \in \mathcal{M}_n(\{0, 1\})$ a matrix so that $a_{i,i} = 0$ for any i, $1 \leq i \leq n$ and $a_{i,j} = a_{j,i}$ for any i, j, $1 \leq i, j \leq n$, $i \neq j$ (A is a symetric matrix). The pair $G = (V, A)$ is a **graph**, the elements of V are*

*named **vertices**, and we say that x_ix_j is an **edge** in G if and only if $a_{i,j} = 1$, $1 \leq i, j \leq n$, $i \neq j$.*

Definition 1.2 (sets) *Let $G = (V, E)$ be a graph. A sequence of vertices $y_1, y_2, \ldots, y_k$ is named **walk** in G if $\{y_i, y_{i+1}\} \in E$ for any $i = 1, 2, \ldots, k-1$. If $y_1 = y_k$ the walk is named **loop**. The walk (and the loop) is specified as*

$$L = [y_1, y_2, \ldots, y_k].$$

Definition 1.2 (0-1 algebraic) *Let $G = (V, A)$ be a graph. A sequence of vertices $y_1, y_2, \ldots, y_k$ is named **walk** in G if $a_{i,i+1} = 1$ for any $i = 1, 2, \ldots, k-1$. If $y_1 = y_k$ the walk is named **loop**. The walk (and the loop) is specified as*

$$L = [y_1, y_2, \ldots, y_k].$$

Also we can say that the lenght of the walk L is $k - 1$.

The relationship between sets and 0-1 algebraic representation is given by the following result:

Proposition 1.1 *Let G be a graph with sets representation $G = (V, E)$ and with $0-1$ algebraic representation $G = (V, A)$, then for any i, j, $1 \leq i, j \leq n$, we have*

$$a_{i,j} = \begin{cases} 1 & if\ \{x_i.x_j\} \in E \\ 0 & if\ \{x_i.x_j\} \notin E \end{cases}.$$

In graph theory literature, if we consider a graph $G = (V, E)$ in sets representation, the matrix A from 0-1 algebraic representation of G is designated by A_G and it is called ***adiacence matrix*** of G.

On $\{0, 1\}$ we consider the computing operation

$$\oplus : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$$

defined by

$$x \otimes y = \begin{cases} 0 & \text{if } x = y = 0 \\ 1 & \text{otherwise} \end{cases}$$

and

$$\otimes : \{0, 1\} \times \{0, 1\} \to \{0, 1\}$$

defined by

$$x \otimes y = \begin{cases} 0 & \text{if } x = y = 1 \\ 1 & \text{otherwise} \end{cases}$$

for any $x, y \in \{0, 1\}$.

We extend these operations to work on matrices. So we consider

$$\bigoplus : \mathcal{M}_n(\{0, 1\}) \times \mathcal{M}_n(\{0, 1\}) \to \mathcal{M}_n(\{0, 1\})$$

defined by $A \bigoplus B = C$ for any $A, B \in \mathcal{M}_n(\{0, 1\})$, $A = (a_{i,j})_{i,j=\overline{1,n}}$, $B = (b_{i,j})_{i,j=\overline{1,n}}$, $C = (c_{i,j})_{i,j=\overline{1,n}}$, where $c_{i,j} = a_{i,j} \oplus b_{i,j}$ and

$$\bigotimes : \mathcal{M}_n(\{0, 1\}) \times \mathcal{M}_n(\{0, 1\}) \to \mathcal{M}_n(\{0, 1\})$$

defined by $A \bigotimes B = C$ for any $A, B \in \mathcal{M}_n(\{0,1\})$, $A = (a_{i,j})_{i,j=\overline{1,n}}$, $B = (b_{i,j})_{i,j=\overline{1,n}}$, $C = (c_{i,j})_{i,j=\overline{1,n}}$, where

$$\begin{aligned} c_{i,j} &= a_{i,1} \otimes b_{1,j} \oplus a_{i,2} \otimes b_{2,j} \oplus \ldots \oplus a_{i,n} \otimes b_{n,j} \\ &= \sum_{k=1}^{n} a_{i,k} \otimes b_{k,j} \end{aligned} .$$

Using the operations $\oplus$ and $\otimes$ defined on $\mathcal{M}(\{0,1\})$ we can compute $A^2 = (q_{ij})_{i,j=\overline{1,n}}$ so that if $A \otimes A = (d_{ij})_{i,j=\overline{1,n}}$, then

$$q_{ij} = \begin{cases} d_{ij}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

and $A^p = A \otimes A^{p-1}$, and we have the following result.

Proposition 1.2 *Let $G = (V, A)$ be o graph in $0-1$ algebraic representation and $x_i, x_j \in V$. There exists a walk L from x_i to x_j with $l(L) = k$ if and only if in $A^k = (b_{ij})_{i,j=\overline{1,n}}$ we have $b_{ij} = 1$.*

Considering the graph $G = (V, E)$ in sets representation, we can associate to the graph the matrix $W_G \in \mathcal{M}(\{0.1\})$ so that if $W_G = (w_{ij})_{i,j=\overline{1,n}}$, then for any i, j, $1 \leq i, j \leq n$,

$$w_{ij} = \begin{cases} 1, & \text{if there exists a walk from } x_i \text{ to } x_j \\ 0. & \text{otherwise} \end{cases} .$$

The matrix W is called ***walk matrix*** of graph G. We have the following result.

Proposition 1.3 *Let $G = (V, E)$ be a graph in sets representation, then*

$$W_G = A_G \oplus A_G^2 \oplus \ldots \oplus A_G^{n-1} = \sum_{i=1}^{n-1} A_G^i.$$

Corollary 1.1 *Let $G = (V, E)$ be a graph in $0-1$ algebraic representation and $W = (w_{ij})_{i,j=\overline{1,n}} = \sum_{i=1}^{n-1} A^i$. Then for any $x_i, x_j \in V$ there exists a walk from x_i to x_j if and only if $w_{ij} = 1$.*

Corollary 1.2 *Let $G = (V, E)$ be a graph in $0-1$ algebraic representation and for any k, $1 \leq k \leq n-1$, we consider the matrix $A^k = (a_{ij}^k)_{i,j=\overline{1,n}}$. Then $L = [x_{i_1}, x_{i_2}, \ldots, x_{i_s}]$ is a walk in G if and only if, for any $p = 2, \ldots, s$ we have $a_{i_1,i_p}^{p-1} = 1$.*

Definition 1.3 (sets) *Let $G = (V, E)$ be a graph. If $W \subset V$ and*

$$F = \{\{x, y\} \in E | x, y \in W\}$$

*then the graph $H = (W, F)$ is named **subgraph** of G.*

Definition 1.3. (0-1 algebraic) *Let $G = (V, A)$ be a graph in 0-1 algebraic representation. If $W = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ and $B \in \mathcal{M}(\{0,1\})$,*

$B = (b_{t,s})_{t,s=\overline{1,k}}$ *is the matrix defined by* $b_{t,s} = a_{i_t,i_s}$ *for any* $1 \leq t, s \leq k$, *then the graph* $H = (W, B)$ *is named* ***subgraph*** *of* G.

Definition 1.4 (sets) *Let* $G = (V, E)$ *be a graph. If for any* $x, y \in V$ *there exists a walk from* x *to* y *then* G *is named* ***connected****. If* G *is not connected,* G *is named* ***disconencted****.*

Definition 1.4. (0-1 algebraic) *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation and* $W = \sum_{i=1}^{n-1} A^p = (w_{ij})_{i,j=\overline{1,n}}$ *with* A^2 *computed as above. If for any* $i \neq j$, $1 \leq i, j \leq n$, $w_{ij} = 1$ *then* G *is named* ***connected****. If* G *is not connected, then* G *is named* ***disconnected****.*

Definition 1.5 (sets) *Let* $G = (V, E)$ *be a graph. A subgraph* $H = (W, F)$ *of* G *which is connected and there does not exist a walk from* x *to* y *for any* $x \in W$ *and* $y \in V - W$ *is named* ***component*** *of* G.

Definition 1.5. (0-1 algebraic) *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation. A subgraph* $H = (W, B)$ *of* G *which is connected so that for any* $y \in V - W$ *the subgraph* $H = (W \bigcup \{y\}, C)$ *is disconnected is named* ***component*** *of* G.

Proposition 1.4 *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation. There exists a partition of* V *with the sets* $V_1, V_2, \ldots, V_k$ *(*$V_1 \cup V_2 \cup \ldots \cup V_k = V$ *and* $V_i \cap V_j = \oslash$ *for any* $i, j = 1, 2, \ldots, k$, $i \neq j$*) so that subgraphs* $G = (V_i, B_i)$ *are components in* G.

Definition 1.6 (sets) *Let* $G = (V, E)$ *be a graph. If* G *is connected and it does not contain loops then the graph is named* ***tree****.*

Definition 1.6. (0-1 algebraic) *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation and* $W = \sum_{i=1}^{n-1} A^p = (w_{ij})_{i,j=\overline{1,n}}$ *with* A^2 *computed as above. If* G *is connected and for any* j, $1 \leq j \leq n$, $w_{jj} = 0$, *then* G *is named* ***tree****..*

Propozition 1.5 *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation with* $A = (a_{ij})_{i,j=\overline{1,n}}$. G *is a tree if ond only if*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = 2(|V| - 1).$$

Definition 1.7 (sets) *Let* $G = (V, E)$ *be a graph with* $|V| = p \geq 3$. *If for any* $x, y \in V$, $x \neq y$, $\{x, y\} \in E$ *then* G *is named* ***complete*** *and we write it as* K_p.

Definition 1.7. (0-1 algebraic) *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation with* $A = (a_{ij})_{i,j=\overline{1,n}}$ *and* $p = |V| \geq 3$. *If for any* $i \neq j$, $1 \leq i, j \leq p$, $a_{ij} = 1$, *then* G *is named* ***complete*** *and we write it as* K_p.

Definition 1.8 (sets) *Let* $G = (V, E)$ *be a graph. If for any* $x \in V$ *there exist* $y, z \in V$, $y \neq z$ *so that* $\{\{x, y\}, \{x, z\}, \{y, z\}\} \subset E$ *we say that* G *is a* ***minimal complete graph****.*

Definition 1.8. (0-1 algebraic) *Let* $G = (V, A)$ *be a graph in 0-1 algebraic representation. If for any* $x_i \in V$ *there exist* $x_j, x_k \in V$, $i \neq j$, $i \neq km$ $j \neq k$, *so that* $a_{ij} = a_{ik} = a_{jk} = 1$, *we say that* G *is a* ***minimal complete graph****.*

Definition 1.8 (0-1 algebraic) is equivalent with the next definition: *Let $G = (V, A)$ be a graph in 0-1 algebraic representation. If for any $x \in V$ there exist $y, z \in V$, $x \neq y$, $x \neq z$, $y \neq z$, so that the subgraph $H = (\{x, y, z\}, B)$ is a complete graph, then we say that G is a **minimal complete graph**.*

Definition 1.9 *Let G be a graph. If any component in G is a minimal complete graph then G is named **component minimal complete graph**.*

Definition 1.10 (sets) *Let $G = (V, E)$ be a graph and $x \in V$. If we consider the set $C = \{y \in V | \{x, y\} \in E\}$ then the number $\omega(x) = |C|$ is named the **degree** of x.*

Definition 1.10. (0-1 algebraic) *Let $G = (V, A)$ be a graph in 0-1 algebraic representation and $x_i \in V$. The number $\omega(x_i) = \sum_{j=1}^{n} a_{ij}$ is named **the degree** of x_i in G.*

Let CMC_G designate the component minimal complete subgraph for the graph G, if such a subgraph exists. If G has not CMC_G we write

$$CMC_G = \emptyset.$$

Proposition 1.6 *Let $G = (V, E)$ (or $G = (V, A)$) be a graph in sets representation (in 0-1 representation) and*

$$X = \{x \in V | \omega(x) = 0\}$$

with $V - X \neq \oslash$. Let consider the subgraph $H = (V - X, F)$ ($H = (V - X, B)$) of G. Then, G has a component minimal complete subgraph if and only if H has a component minimal complete subgraph.
In addition, we have

$$CMC_G = CMC_H.$$

Proposition 1.7 *Let $G = (V, E)$ (or $G = (V, A)$) be a graph in sets representation (in 0-1 representation) and*

$$X = \{x \in V | \omega(x) = 1\}.$$

Let us consider the subgraph $H = (V - X, F)$ ($H = (V - X, B)$) of G. Then, G gas a component minimal complete subgraph if and only if H has a component minimal complete subgraph and

$$CMC_G = CMC_H.$$

.

Proposition 1.8 *Let $G = (V, E)$ (or $G = (V, A)$) be a graph in sets representation (in 0-1 representation) and let $CMC_G = (U, F)$ ($CMC_G = (U, B)$) be the component minimal complete subgraph for G. Then for any $x \in U$ in G we have $\omega(x) \geq 2$.*

Proposition 1.9 *Let G be a graph. If G ia a tree then $CMC_G = \emptyset$.*

Proposition 1.10 *Let $G = (V, E)$ be a graph formed with the components $G_1, G_2, \ldots, G_k$. Then*

$$CMC_G = \bigcup_{i=1}^{k} CMC_{G_i}.$$

2. Algorithms in sets representation of graphs

In [Bârză, 2013] and [Bârză, 2014] I have presented three algorithms for the determination of component minimal complete subgraph, namely, an algorithm for the determination of CMC_G without eliminations, and an algorithm for the determination of CMC_G with vertices and edges elimination, and the algorithm for the determination of CMC_G working on components.

Let $G = (V, E)$ be a grap $k = |E|$ and so, $E = \{m_1, m_2, \ldots, m_k\}$. We have considered the algorithm for determination of CMC_G without vertices and edges elimination which is specified as the following process:

Step 1. Let $U = \oslash$, $F = \oslash$ and $i = 1$.

Step 2. If $i = k-2$ then STOP otherwise continue.

Step 3. Let $j = i + 1$.

Step 4. If $j = k - 1$ then $i \leftarrow i + 1$ and go to Step 2, otherwise continue.

Step 5. If $m_i \cap m_j = \oslash$ then go to Step 9, otherwise continue.

Step 6. Let $p = (m_i \cup m_j) \backslash (m_i \cap m_j)$.

Step 7. If $p \notin \{m_{j+1}, \ldots, m_k\}$ then go to Step 9, otherwise continue.

Step 8. Let $U \leftarrow U \cup m_i \cup m_j$ and $F \leftarrow F \cup \{m_i, m_j, p\}$.

Step 9. Let $j \leftarrow j + 1$ and go to Step 4.

For this algorithm we have shown that:

Theorem 2.1. *Let $G = (V, E)$ be a graph with $|E| = k$. Then by applying the algorithm given above we obtain CMC_G and the algorithm has the complexity $O(k^3)$.*

The second algorithm I have presented in [Bârză, 2013], the algorithm for the determination of CMC_G with vertices and edges elimination, it was given by the following process applied to $H = (X, T)$ as original given graph:

Step 1. Let $Y = \{x \in X | \omega(x) = 0\} \cup \{x \in X | \omega(x) = 1\}$.

Step 2. If $Y = \oslash$ then go to Step 4, otherwise continue.

Step 3. Let $X \leftarrow X/Y$, $T \leftarrow T/\{m \in T | m \cap Y \neq \oslash\}$ and go to Step 1.

Step 4. Let $V = X$ $E = T$, $k = |E|$ and the graph $G = (V, E)$ for which we consider that $E = \{m_1, m_2, \ldots, m_k\}$.

Step 5. Let $U = \oslash$, $F = \oslash$ and $i = 1$.

Step 6. If $i = k-2$ then STOP otherwise continue.

Step 7. Let $j = i + 1$.

Step 8. If $j = k - 1$ then $i \leftarrow i + 1$ and go to Step 2, otherwise continue.

Step 9. If $m_i \cap m_j = \oslash$ then go to Step 9, otherwise continue.

Step 10. Let $p = (m_i \cup m_j) \backslash (m_i \cap m_j)$.

Step 11. If $p \notin \{m_{j+1}, \ldots, m_k\}$ then go to Step 9, otherwise continue.

Step 12. Let $U \leftarrow U \cup m_i \cup m_j$ and $F \leftarrow F \cup \{m_i, m_j, p\}$.

Step 13. Let $j \leftarrow j + 1$ and go to Step 4.

For this algortihm I have shown that:

Theorem 2.2. *Let $H = (X, T)$ be a graph. Then by applying the algorithm given above we obtain CMC_H and the algorithm has the complexity $O(k^3)$, where $k \leq |T| - |\{x \in X | \omega(x) = 1\}|$ and it represents the number of remaining edges after the elimination process made in the first three steps of the algorithm.*

The thrird algorithm I have presented in [Bârză, 2014], the algorithm for the determination of CMC_G working on components, is given by the following process:

Step 1. Let $Y \leftarrow \{x \in V | \omega_G(x) = 0\}$. Let $V \leftarrow V - Y$.

Step 2. Let $U \leftarrow \emptyset$, $F \leftarrow \emptyset$, $T \leftarrow \emptyset$, $A \leftarrow V$, and $X \leftarrow E$.

Step 3. If $A = \emptyset$ then STOP; $CMC_G = (U, F)$, otherwise continue.

Step 4. Take $x \in A$, let $B \leftarrow \{x\}$, $C \leftarrow \emptyset$, and $H \leftarrow \emptyset$.

Step 5. If $B = \emptyset$ then goto Step 9, otherwise continue.

Step 6. Take $x \in B$.

Step 7. For any $m \in X$, if $x \in m$ then let $X \leftarrow X - \{m\}$, $H \leftarrow H \bigcup \{m\}$ and $B \leftarrow B \bigcup m$, otherwise continue.

Step 8. Let $B \leftarrow B - \{x\}$, $C \leftarrow C \bigcup \{x\}$ and goto Step 5.

Step 9. Let $A \leftarrow A - C$.

Step 10. If $|H| = |C| - 1$ then goto Step 3, otherwise continue.

Step 11. Let $Y \leftarrow \left\{x \in C | \omega_{(C,H)}(x) = 1\right\}$.

Step 12. If $Y = \emptyset$ then goto Step 14 otherwise continue.

Step 13. Let $Z \leftarrow \{m \in H | m \bigcap Y \neq \emptyset\}$. Let $H \leftarrow H - Z$, $C \leftarrow C - Y$ and goto Step 11.

Step 14. Consider that $H = \{m_1, m_2, \ldots, m_k\}$. Let $i \leftarrow 1$.

Step 15. If $i = k - 1$ then goto Step 3, otherwise continue.

Step 16. Let $j \leftarrow i + 1$

Step 17. If $j = k$ then $i \leftarrow i + 1$ and goto Step 15, otherwise continue.

Step 18. If $m_i \bigcap m_j = \emptyset$ then goto Step 22, otherwise continue.

Step 19. Let $p \leftarrow (m_i \bigcup m_j) - (m_i \bigcap m_j)$

Step 20. If $p \notin \{m_{j+1}, \ldots, m_m\}$ then goto Step 22, otherwise continue.

Step 21. Let $U \leftarrow U \bigcup m_i \bigcup m_j$, $F \leftarrow F \bigcup \{m_i, m_j, p\}$ and $T \leftarrow T \bigcup \{m_i \bigcup m_j\}$.

Step 22. Let $j \leftarrow j + 1$ and goto Step 17.

For this algorithm I have shown that:

Theorem 2.3. *Let $G = (V, E)$ be a graph and $m = |E|$. We consider that the graph G is forms with p components $G_1, G_2, \ldots, G_p$ and for each component $G_i = (V_i, E_i)$, $(1 \leq i \leq p)$ the number of edges is $k_i = |E_i|$. The algorithm presented realizes the determination of CMC_G and has the complexity given by:*

$$O(\sum_{1 \leq i \leq p,\, k_i \neq 0,\, k_i \neq |V_i| - 1} k_i^3).$$

3. The algorithm for the determination of CMC_G of G in 0-1 algebraic approach

In this section we wish ti give the algorithm for the determination of CMC_G in which one works with the instrument of algebra using 0-1 representation of graphs.

Because we have shown that the better algorithm is the algorithm for the determination of CMC_G working on components, in our approach we will consider only this one to build our new process in algebraic approach.

Because we want to work on component, firstly we must remember the vertices that are already used in our process. To do so, we must use a 0-1 vector defined via

$$used_i = \begin{cases} 1 & \text{if vector } x_i \text{ is already used} \\ 0 & \text{otherwise} \end{cases}.$$

At the beginning of our process we shall initialize this vector with all components equal to 0. The process must stop when all the components of this vector is equal to 1, or else when $\sum_{i=1}^{n} used_i = n$.

In considering the algorithm for sets representation, the first part eliminates the isolated vertices, which are the vertexes x_i with $\omega(x_i) = 0$. Then doing here the same operation we shall compute $\omega(x_i) = \sum_{j=1}^{n} a_{ij}$ and if this value is equal to 0 then, to eliminate x_i means to consider x_i as an already used vertex which is made doing $used_i = 1$.

To continue, we have to choose between computing the walks matrix or to take the components one by one. To compute the walks matrix we must perform a lot of multiplying and adding operations on a matrix of order n. To reduce the performed operation we consider that a better choose is to take the components one by one and to use a process in which we have only adding operations.

To determine a component of G we can start with the first vertex x_i for which $used_i = 0$ and to consider its line in A as a separate vector with elements $l_{new_j} = a_{ij}$ for any $j = \overline{1,n}$. Also we consider the vector with components $l_{old_j} = 0$ for any $j = \overline{1,n}$.

Now, starting with $l = l_{new}$, for any j for which $l_{old_j} = 0$ and $l_{new_j} = 1$, we add to l the line j from A, so $l_k = l_k + a_{jk}$ for any $k = \overline{1,n}$.

This subprocess will stop when $l = l_{new}$, or else when $\sum_{k=1}^{n} l_k = \sum_{k=1}^{n} l_{new_k}$. If $l \neq l_{new}$ we consider that $l_{old} = l_{new}$ and $l_{new} = l$ and perform again this subprocess.

When this subprocess ends, l_{new} indicates the vertices of the component of G which includes the vertex x_i.

In the algorithm considered in sets approach, we determined the CMC for a component if and only if the component is not a tree. To do the same operation here, we use the proposition 1.5 in which we consider the sum only for i and j so that $l_{new_i} = l_{new_j} = 1$ and instead of V we use the value $\sum_{k=1}^{n} l_{new_k}$ which represents the numer of vertexes in current component of G. So we have the condition

$$\sum_{\substack{i=1 \\ l_{new_i}=1}}^{n} \sum_{\substack{j=1 \\ l_{new_j}=1}}^{n} a_{ij} = 2(\sum_{k=1}^{n} l_{new_k} - 1).$$

If this condition is true, then the current component of G is a tree and so its CMC is empty. If this condition is not true, then we have to determine its CMC and for this operation we will use the definition 1.8 (0-1 algebraic).

When we end the process for current component we add the vector l_{new} to the vector $used$ and if the stop condition is not true we pass to the next component of G.

To identify the K_3 component in G we will consider the set of sets $\{x, y, z\}$ which will be empty at the beginning of the algorithm. When algorithm stops we will generate the graph $CMC_G = (U, B)$ using this set.

On the basis of the above considerations we propose the following process, which will be called as *the 0-1 algebraic algorithm for the determination of* CMC_G *working on components*:

Step 1. For $i = \overline{1,n}$ let $used_i \leftarrow 0$; let $k3n \leftarrow 0$.

Step 2. For $i = \overline{1,n}$ if $\sum_{j=1}^{n} a_{ij} = 0$ then let $used_i \leftarrow 1$.

Step 3. If $\sum_{i=1}^{n} used_i = n$ then goto step 12, otherwise continue.

Step 4. For $i = \overline{1,n}$ let $l_{old_i} \leftarrow 0$; let $i \leftarrow 1$.

Step 5. if $used_i = 1$ then $i \leftarrow i + 1$ and goto step 5, otherwise for $j = \overline{1,n}$ let $l_{new_j} \leftarrow 0$; let $l_{new_i} \leftarrow 1$.

Step 6. If $\sum_{k=1}^{n} l_{old_k} = \sum_{k=1}^{n} l_{new_k}$ then goto step 8, otherwise let $l_{old} \leftarrow l_{new}$ and $l \leftarrow l_{new}$.

Step 7. $j = \overline{1,n}$ if $l_{old_j} = 0$ and $l_{new_j} = 1$ then for $k = \overline{1,n}$ $l_k \leftarrow l_k \oplus a_{jk}$; let $l_{new} \leftarrow l$.

Step 8. If $\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} = 2(\sum_{i=1}^{n} l_{new_i} - 1)$ then for $i = \overline{1,n}$ $used_i \leftarrow used_i + l_{new_i}$; goto step 3, otherwise continue.

Step 9. For $i = \overline{1,n}$ if $l_{new_i} = 1$ perform step 10; At the end for $i = \overline{1,n}$ $used_i \leftarrow used_i + l_{new_i}$; goto step 3.

Step 10. For $j = \overline{i+1, n-1}$ if $l_{new_j} = 1$ perform step 11.

Step 11. For $k = \overline{j+1, n}$ if $l_{new_k} = 1$ then if $a_{ij} + a_{ik} + a_{jk} = 3$ then let $k3n \leftarrow k3n + 1$, $v_{k3n,1} \leftarrow i$, $v_{k3n,2} \leftarrow j$ and $v_{k3n,3} \leftarrow k$.

Step 12. Let $p \leftarrow 1$; for $i, j = \overline{1,n}$ let $b_{ij} \leftarrow 0$. (If $k3n \neq 0$ then the vector v indicates the minimal complete subgraphs of G. If $k3n = 0$ then $CMC_G = \emptyset$).

Step 13. If $p > k3n$ then STOP, otherwise continue.

Step 14. $i \leftarrow v_{p,1}$, $j \leftarrow v_{p,2}$, $k \leftarrow v_{p,3}$, $b_{ij} \leftarrow 1$, $b_{ji} \leftarrow 1$, $b_{ik} \leftarrow 1$, $b_{ki} \leftarrow 1$, $b_{jk} \leftarrow 1$, $b_{kj} \leftarrow 1$, $p \leftarrow p + 1$ and goto step 13.

The above process is clearly a finite one and so it is an algorithm.

Because the analyse of this process is complicated, we will do this operation in another paper in which we will present also the theorem for the characterization of this algorithm.

References

1. Bamdy, J.A., Murty U.S.R., *Graph Theory*, "Springer-Verlag", 2007.
2. Bârză, S., and Morogan, L.M., *Algoritmica grafurilor*, Editura Fundatiei *Romania de Maine*, Bucharest, 2008.
3. Bârză, S., *Some Consideration about Component Minimal Complete Subgraph*, Analele Universităţii *Spiru Haret*, Seria Matematică-Informatică, anul VII, nr. 2, Editura Fundaţiei *România de Mâine*, Bucureşti 2012.
4. Bârză, S., *Set Approach Algorithms for Determination of Component Minimal Complete Subgraph*, Analele Universităţii *Spiru Haret*, Seria Matematică-Informatică, anul VIII, nr. 1, pp. 65-71, Editura Fundaţiei *România de Mâine*, Bucureşti 2013.
5. Bârză, S., *A Better Set Approach Algorithms for Determination of Component Minimal Complete Subgraph*, Analele Universităţii *Spiru Haret*, Seria Matematică-Informatică, anul IX, nr. 1, pp. 5-12, Editura Fundaţiei *România de Mâine*, Bucureşti 2014.
6. Berge, C., *Graph Theory and Application*, Romanian edition, Editura Tehnica, Bucharest, 1971.
7. Popescu, D.R., *Combinatorică şi teoria grafurilor*, Editura Societăţii de Ştiinţe matematice din România, Bucharest, 2005.
8. Tomescu, I., *Combinatorică şi teoria grafurilor*, Editura Universităţii Bucureşti, Bucharest, 1990.