Univ. Beograd. Publ. Elektrotehn. Fak.
Ser. Mat. 14 (2003), 37–48.

THE EIGENSPACE OF THE EIGENVALUE –2 IN GENERALIZED LINE GRAPHS AND A PROBLEM IN SECURITY OF STATISTICAL DATABASES

Ljiljana Branković, Dragoš Cvetković

We introduce the notion of the L-core of a graph what enables a simple description of some properties of the eigenspace of the eigenvalue -2 in generalized line graphs and an elegant formulation of the solution of a problem in the security of data in statistical databases.

0. INTRODUCTION

During the visit of the second author to the University of Newcastle, School of Electrical Engineering and Computer Science, Newcastle, Australia, in November and December 2002, the authors of this note realized that they, with some coauthors [**7**], [**2**], [**1**], [**10**], have independently discovered some facts on the eigenvectors of the eigenvalue -2 in generalized line graphs in quite different settings. While papers [**7**], [**10**] continue earlier mathematical research on graphs with least eigenvalue -2, works [**2**], [**1**] deal with a practical problem arising in the study of the security of data in statistical databases.

In this paper we introduce the notion of the L-core of a graph. This notion enables a useful formulation of necessary and sufficient conditions for a class of query sets in statistical databases to be compromise-free. We show how the L-core can be determined from eigenvalues and angles of the corresponding generalized line graph and formulate an algorithm to determine the L-core starting from the graph itself.

2000 Mathematics Subject Classification: Primary 05C50, Secondary 68P15
Keywords and Phrases: Graph theory, graph spectra, statistical databases.

1. PRELIMINARIES

Let $G = (V, E)$ be a simple graph with n vertices. The characteristic polynomial $\det(xI - A)$ of the adjacency matrix A of G is called the *characteristic polynomial of* G and denoted by $P_G(x)$. The eigenvalues of A (i.e. the zeros of $\det(xI - A)$) and the spectrum of A (which consists of the n eigenvalues) are also called the *eigenvalues* and the *spectrum* of G, respectively. The eigenvalues of G are usually denoted by $\lambda_1, \lambda_2, \dots, \lambda_n$; they are real because A is symmetric. We shall assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and use the notation $\lambda_i = \lambda_i(G)$ for $i = 1, 2, \dots, n$. The least eigenvalue $\lambda_n(G)$ of a graph G will also be denoted by $\lambda(G)$.

The eigenvalues of A are the numbers λ satisfying $A\mathbf{x} = \lambda\mathbf{x}$ for some non-zero vector $\mathbf{x} \in \mathbf{R}^n$. Each such vector $\mathbf{x}$ is called an *eigenvector* of the matrix A (or of the labelled graph G) belonging to the eigenvalue λ. The relation $A\mathbf{x} = \lambda\mathbf{x}$ can be interpreted in the following way: if $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ then $\lambda x_u = \sum_{v \sim u} x_v$ where the summation is over all neighbours v of the vertex u.

If λ is an eigenvalue of A then the set $\{\mathbf{x} \in \mathbf{R}n : A\mathbf{x} = \lambda\mathbf{x}\}$ is a subspace of $\mathbf{R}^n$, called the *eigenspace* of λ and denoted by $\mathcal{E}(\lambda)$. Such eigenspaces are called *eigenspaces* of G.

Let now G be a graph on n vertices with distinct eigenvalues $\mu_1, \mu_2, \dots, \mu_m$ ($\mu_1 > \mu_2 > \cdots > \mu_m$) and let $S_1, S_2, \dots, S_m$ be the corresponding eigenspaces. Let $\{e_1, e_2, \dots, e_n\}$ be the standard (orthonormal) basis of $\mathbf{R}^n$. The numbers $\alpha_{pq} = \cos\beta_{pq}(p = 1, 2, \dots, m; q = 1, 2, \dots, n)$, where β_{pq} is the angle between S_p and e_q, are called *graph angles*. The sequence α_{pq} ($q = 1, 2, \dots, n$) is called the *eigenvalue angle sequence* corresponding to the eigenvalue μ_p ($p = 1, 2, \dots, m$).

Let $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ ($i = 1, 2, \dots, n$) be orthonormal eigenvectors of G. Define $M_p = \{j \mid Ax_j = \mu_p x_j\}$. We have

$$\alpha_{pq}^2 = \sum_{j \in M_p} x_{jq}^2 \tag{1}$$

for squares of angles of G. This formula holds for any choice of orthonormal eigenvectors of G ([**9**], p. 76).

An overview of results on graph angles is given in [**9**] including the characterizing properties of graph angles.

As usual, K_n, C_n and P_n denote respectively the *complete graph*, the *cycle* and the *path* on n vertices. Further, $K_{m,n}$ denotes the *complete bipartite* graph on $m + n$ vertices. The *cocktail-party graph* $CP(n)$ is the unique regular graph with $2n$ vertices of degree $2n - 2$; it is obtained from K_{2n} by deleting n mutually non-adjacent edges.

A connected graph with n vertices is said to be *unicyclic* if it has n edges. It is called *even* (*odd*) if its unique cycle is even (odd). An *orchid* is a unicyclic graph with an odd cycle, or a tree in which one pendant edge is doubled (i.e. replaced by a pendant double edge called also a 2-*cycle* or a *petal*). We use the term *supercycle* to mean either an odd cycle or a 2-cycle (petal). An *orchid garden* is a graph whose components are orchids. A graph consisting of two supercycles connected by a path, possibly of length 0, is called an *odd dumbbell*.

The *line graph* $L(H)$ of any graph H is defined as follows. The vertices of $L(H)$ are the edges of H and two vertices of $L(H)$ are adjacent whenever the corresponding edges of H have a vertex of H in common.

A *generalized line graph* $L(H; a_1, \dots, a_n)$ is defined for graphs H with vertex set $\{1, \dots, n\}$ and non-negative integers $a_1, \dots, a_n$ by taking the graphs $L(H)$ and $CP(a_i)$ $(i = 1, \dots, n)$ and adding extra edges: a vertex e in $L(H)$ is joined to all vertices in $CP(a_i)$ if i is an end-vertex of e as an edge of H. We include as special cases an ordinary line graph $(a_1 = a_2 = \cdots = a_n = 0)$ and the cocktail-party graph $CP(n)$ $(n = 1$ and $a_1 = n)$.

An *exceptional* graph is a connected graph with least eigenvalue greater than or equal to -2 which is not a generalized line graph.

2. THE EIGENSPACE OF THE EIGENVALUE –2 IN GENERALIZED LINE GRAPHS

According to [**10**] we shall describe the eigenspace of the eigenvalue -2 in generalized line graphs and give a link to the mentioned problem in security of statistical databases.

If G is a connected graph which is neither a tree nor an orchid, then the least eigenvalue of $L(G)$ is -2 [**11**]. A *foundation* of G is a spanning tree of G if G is bipartite and a spanning orchid garden in G if G is non-bipartite. (Note that in the latter case all components in a foundation are unicyclic graphs with odd cycles.)

We now turn to generalized line graphs. If λ is the least eigenvalue of such a graph then $\lambda \geq -2$: this and other properties of generalized line graphs are described in [**7**]. In particular the multiplicity of λ is determined when $\lambda = -2$.

The following result of M. Doob and D. Cvetković [**12**] is of interest in our considerations.

Theorem 1. *If G is a connected graph with least eigenvalue greater than -2 then one of the following holds:*
(i) $G = L(T; 1, 0, \dots, 0)$ *where T is a tree;*
(ii) $G = L(H)$ *where H is a tree or an odd unicyclic graph;*
(iii) *G is one of the* 20 *exceptional graphs on* 6 *vertices (represented by the root system E_6);*
(iv) *G is one of the* 110 *exceptional graphs on* 7 *vertices (represented by the root system E_7);*
(v) *G is one of the* 443 *exceptional graphs on* 8 *vertices (represented by the root system E_8).*

We have already noted the role of graphs of type (ii) in line graphs; below we explain the role of graphs of type (i) and (ii) in generalized line graphs.

Consider a generalized line graph $L(G; a)$, where G is connected and $\sum_{i=1}^{n} a_i > 0$. The *root graph* of $L(G; a)$ is defined in [**7**] as the multigraph H obtained from G by adding a_i pendant double edges (petals) at vertex v_i for each $i = 1, \dots, n$. Then $L(G; a) = L(H)$ if we understand that in $L(H)$ two vertices are adjacent if and only if the corresponding edges in H have exactly one vertex in common. In the case that $L(H)$ has least eigenvalue -2 (i.e. $L(H)$ is not of type (i)) we say that the subgraph F of H is a *foundation* for H if F is a spanning orchid garden of H.

EXAMPLE. Let H be the root multigraph of the generalized line graph $L(K_3; 1, 1, 0)$. Thus H consists of a triangle with a pair of double edges added to two vertices of a triangle. All non-isomorphic foundations of H are shown in Fig. 1.

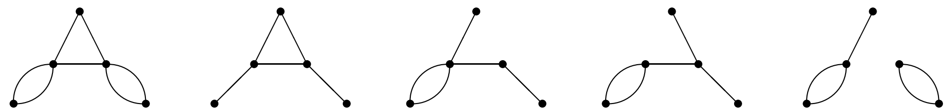

Fig. 1. A multigraph and its foundations.

Next we show how foundations of root graphs can be used to construct a basis for the eigenspace of -2 in generalized line graphs. This generalizes DOOB's construction [**11**] of such a basis in the case of line graphs. There are $m - n + \sum_{i=1}^{n} a_i$ edges of H not in F, and since F is an orchid garden three possibilities arise when such an edge e is added to F: (1) the edge closes an even cycle, (2) the edge closes a supercycle (i.e. an odd cycle or doubles one pendant edge), (3) the edge joins a vertex of one orchid to a vertex of another orchid (thus forming an odd dumbbell). We now ascribe weights to the edges of H as follows. In case (1) all weights are 0 except for 1 and -1 alternately on edges of the even cycle. In cases (2) and (3), $F+e$ contains a unique shortest path P between vertices of two different supercycles, and we first ascribe weights of 2 and -2 alternately to the edges of P.

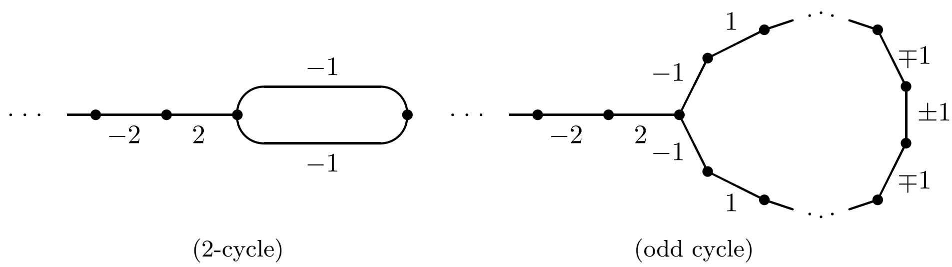

Fig. 2. The construction of eigenvectors.

To within a unique choice of sign, weights are ascribed to the edges of the two supercycles as illustrated in Fig. 2, and all remaining weights are 0. (In all cases the construction may be seen as ascribing weights ± 1 alternately to the edges in a closed trail, with the assumption that double edges are assigned the same value; in edges traversed twice, the values are added.) In each case, the weights of edges in H are taken as co-ordinates of a vector whose entries are indexed by the corresponding vertices of $L(H)$. We call this vector the *characteristic vector* of the subgraph $F+e$, and we say that $F+e$ is *constructed* from F. It is straightforward to check that each characteristic vector is an eigenvector of $L(H)$ (that is, an eigenvector of $L(G;a)$) corresponding to -2. The $m-n+\sum_{i=1}^{n} a_i$ characteristic vectors are linearly independent because each of the aforementioned closed trails contains an edge not present in any of the others. Since we know from [**7**] that the multiplicity of -2 is equal to $m-n+\sum_{i=1}^{n} a_i$, these vectors generate the whole eigenspace. In this way the following result has been proved in [**10**].

Theorem 2. *The eigenspace for the eigenvalue -2 of a generalized line graph is generated by the characteristic vectors of subgraphs constructed from any foundation of the corresponding root graph.*

Corollary 1. *The eigenspace for the eigenvalue -2 of a generalized line graph is generated by the characteristic vectors of even cycles and odd dumbbells of the corresponding root graph.*

Corollary 2. *A connected generalized line graph has least eigenvalue equal to -2 if and only if the corresponding root graph contains either an even cycle or two supercycles connected by a path (possibly of length 0), i.e. an odd dumbbell.*

In the proof of Theorem 2, the characteristic vector of $F+e$ is an eigenvector of $L(H)$ with the property that the co-ordinate corresponding to e is non-zero and the co-ordinates corresponding to all other vertices outside $L(F)$ are zero. Deletion of these zero co-ordinates results in a vector which spans the eigenspace of -2 in $L(F+e)$.

An equivalent description of the eigenspace of the eigenvalue -2 has been obtained in [**2**], [**1**] with another terminology when treating a problem in computer science. We explain briefly the view point of [**2**], [**1**].

Let R be the vertex-edge incidence matrix of a graph H. It is known (cf., e.g., [**6**], Theorem 3.38 on p. 107) that a non-zero vector x is an eigenvector of -2 in $L(H)$ if and only if $Rx=0$. This equivalence extends to generalized line graphs with suitable chosen matrix R. In this case the matrix contains also entries equal to -1 (for one of the two edges of each petal; cf. [**13**] or [**5**], p. 52). Instead of petals, the authors of [**2**], [**1**] used “semi-edges”, i.e. edges having only one end vertex. This causes that the matrix R contains only one entry equal to 1 in the column corresponding to a semi-edge. The eigenspace of the eigenvalue -2 in generalized line graphs is, in fact, constructed in [**2**], [**1**] by finding vectors x which satisfy $Rx=0$ for the matrix R formed in the described way.

3. A PROBLEM IN SECURITY OF STATISTICAL DATABASES

Statistical databases are those that allow only statistical access to their records. Individual values are typically deemed confidential and are not to be disclosed, either directly or indirectly. Thus, users of a statistical database are restricted to statistical types of queries, such as SUM, AVG, MIN, MAX, etc. Moreover, no sequence of answered queries should enable a user to obtain any of the confidential individual values. However, if a user is able to reveal a confidential individual value, the database is said to be *compromised*. Statistical databases that cannot be compromised are called *secure*.

To illustrate these concepts, we shall make use of an abstract model of statistical databases, represented as a two-dimensional table. Each row in the table corresponds to an individual (a person or an organization), and each column describes a property of these individuals. The rows are usually referred to as 'records' while columns are referred to as 'attributes'.

For example, in a "Company" database, rows (records) could correspond to employees and columns (attributes) might include properties such as "Name", "Address", "Position" and "Salary". For the sake of this example, we assume that the Company database is used for statistical purposes only and that the attribute "Salary" is considered to be confidential. In that case users are not allowed to pose any query that reveals and individual salary, e.g., "What is the salary of the Senior programmer Adam Peterson?". They can, however, pose any statistical query, e.g., "What is the average salary of all System analysts under 35?" or "What is the minimum salary of all female Managers?".

We restrict our attention to a case where all queries have the same form as the above examples, that is, a statistic is calculated on the confidential attribute (i.e., "Salary"), while the values of other attributes are used to select a subset of records on which a statistic is to be calculated (the so-called "query set"). We now show how the database can be compromised. Firstly, if a statistical query is based on a single record, the answer to the query reveals an individual salary, and the database is compromised. Thus the first step towards secure statistical databases is to disable statistical queries based on individual records. Secondly, it is often possible to calculate an individual value from more than one statistical query.

For example, if there is only one female Manager in our "Company" database, her salary can be revealed from the following two queries: "What is the sum of salaries of all the Managers?" and "What is the sum of salaries of all the male Manager?".

The security problem of statistical databases is to ensure that no sequence of queries leads to a database compromise. At the same time, it is very important not to overly restrict the set of queries available to users, as otherwise the database may prove to be of little or no use to them. Thus, there are two conflicting requirements that need to be satisfied: security versus usability (i.e., the percentage of queries that are available to users).

There are several security mechanisms described in the literature, but most of them are either not secure or overly restrictive. One exception is the so-called "Audit Expert" first proposed in [**4**]. This method concentrates on the SUM queries. The database security system keeps track of all previously answered queries and each new query is answered only if, when consider together with all previously answered queries, does not lead to a database compromise. A version of Audit Expert, called "Hybrid Audit Expert" is considered in [**1, 2**]. There users can specify an initial collection of queries that are of particular importance to them. The system then checks whether or not the specified queries cause a compromise; if not, all the queries are made available to the users. Then the system specifies some new queries which are carefully chosen so as to maximize the total number of answerable queries and hence the usability of the database. From this point on, only the specified queries (either by users or by the system) and the ones that are linearly dependent on them are available to the users.

In this paper we focus our attention on recognizing whether a user-specified collection of queries is compromise-free or not. We consider a restricted case where the query collection can be described as a graph. Surprisingly, the results from [**1, 2**] show an amazing connection between compromise-free query collections and graphs with least eigenvalue –2.

We next give a mathematical formulation of Audit Expert. In a database of m records, an answered SUM query can be thought of as a linear equation

$$r_i = r_{i1}x_1 + r_{i2}x_2 + \cdots + r_{im}x_m,$$

where $r_{ij} = 1$ if the record j is in the query i and $r_{ij} = 0$ otherwise; x_j is the value of the confidential attribute X of the record j; r_i is the answer to the query i. Then a set Q of n answered queries can be viewed as a system of n linear equations in m variables $x_1, x_2, \ldots, x_m$, with right hand sides r_i, $1 \leq i \leq n$. Denoting by $R = [r_{ij}]$ the corresponding $n \times m$ coefficient matrix, this linear system is simply $R\mathbf{x}^T = \mathbf{r}^T$ where $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_n)$. The matrix R is called the *query matrix.*

A malicious user may try to compromise a database, that is, to reveal some of all of the confidential values x_i, based on the query matrix R and vector $\mathbf{r}$. This can happen if and only if the normalized matrix (triangular matrix obtained by the Gaussian elimination) equivalent to R has a row containing exactly one non-zero entry, as it was shown in [**4**]. In such a case we say that the query matrix R (and the query set Q) is *compromised.* In the opposite case, R and Q are *compromise-free.* One can prove that this happens if and only if for each i, $1 \leq i \leq m$, there exists a vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ with $v_i \neq 0$ such that $R\mathbf{v}^T = \mathbf{0}$.

We shall now concentrate on the restricted case where each record in the database is contained in at most two queries of a query set Q. Such query sets are said to be of *type* α. Then the query matrix R corresponds to the vertex edge incidence matrix of a graph H, where queries correspond to vertices and records correspond to edges. We shall say that the graph H is *associated* to the query set Q. To cater for the situation in which a record belongs to only one query, we allow "semiedges" in our graph H. These correspond to columns in the incidence matrix with exactly one nonzero entry. Semiedges have been considered in [**2**], [**1**] to be cycles of length 1 (while loops were not allowed in the graph H) and in this capacity a semiedge could be a part of an odd dumbbell.

We say that the graph H is *compromised* or *compromise-free* depending on the status of the corresponding query set Q. As already said in a more general context, the graph H is *compromise-free* if for each edge f there exists an m-dimensional column vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ such that $R\mathbf{x} = \mathbf{0}$ but $x_f \neq 0$. Any such vector will be called *f-capturing*.

The following theorem was proved in [**1, 2**].

Theorem 3. *A graph H is compromise-free if and only if each edge of H is contained either in an even cycle or in an odd dumbbell of H.*

By analyzing the proof of Theorem 3, given in [**2**], one can see that the construction of f-capturing vectors is essentially the same as the construction of eigenvectors of the eigenvalue -2 in the generalized line graph of the root graph obtained from H by replacing each semiedge by a petal. Namely, the authors of [**2**] assign the weight -2 to a semiedge what is the same as assigning weights -1 to each of the edges of the corresponding petal (cf. Fig. 2 here). The necessity part of the proof in [**2**] shows that the whole eigenspace of -2 is generated by the vectors arising from even cycles and odd dumbbells. However, Theorem 3 and its proof in [**2**] do not say anything about the dimension of the eigenspace.

It is interesting to note that the original Doob's description [**11**] in 1973 of the eigenspace of -2 in line graphs in terms of even cycles and odd dumbbells has been extended to generalized line graphs by Cvetković, Doob and Simić [**7**] in 1981 in terms of the chain groups, not explicitly dealing with cycles and dumbbells. The independent discovery of Branković, Miller and Širáň [**2**] in 1996 put implicitly some light on the description of the eigenspace in generalized line graphs a bit before Cvetković, Rowlinson and Simić in 2001 (the paper was submitted in 1998) using the star complement technique and the notion of of a foundation and without being aware of [**2**] gave the entire description of the eigenspace expressed by Theorem 2.

However, Theorem 3 contains further content which gives rise to new mathematical considerations in Sections 4 and 5. This will give also a reformulation of Theorem 3 in terms of generalized line graphs and their eigenspaces of the eigenvalue -2.

4. The L-CORE

Let $G = L(H)$ be a generalized line graph with least eigenvalue greater than or equal to -2. An edge u of the (root multi)graph H is called *strong* if it is contained in an even cycle or in an odd dumbbell. Edges of H which are not strong are called *weak*.

Definition. *The L-core of a (root multi)graph H is the subgraph of H induced by its strong edges.*

The following propositions are straightforward.

Proposition 1. *The L-core of a graph H is empty if and only if H contains neither even cycles nor odd dumbbells.*

Proposition 2. *An edge u of a graph H is strong if and only if there exists an eigenvector of $L(H)$ for the eigenvalue -2 whose coordinate corresponding to u is different from 0.*

The proof follows from Corollary 1 to Theorem 2.

Proposition 3. *The L-core of a graph H is empty if and only if $L(H)$ has the least eigenvalue greater than -2.*

Connected graphs with this property are enumerated in Theorem 1, parts (i) and (ii).

Our main observation inspired by the results of [**2**], [**1**], is given by the following theorem.

Theorem 4. The L-core of a graph H is induced by the edges of H which correspond to the non-zero elements in the eigenvalue angle sequence of the eigenvalue -2 in $L(H)$.

The proof follows from Proposition 2 and formula (1) for angles of a graph.

This theorem extends known results on the reconstructibility of a graph from its eigenvalues and angles (cf. [**9**], Section 5.3, and [**8**]).

Now we can formulate the solution of our query problem in the following elegant way.

Theorem 5. Let Q be a query set of type α for a statistical database and let H be the graph associated with Q. Then Q is compromise-free if and only if H coincides with its L-core.

This theorem shows that the theory of graphs with least eigenvalue -2, a well developed mathematical theory, has also some applications beyond mathematics. For a recent application of the theory to convex quadratic programming see [**3**].

5. ALGORITHMS FOR FINDING AN L-CORE

According to Theorem 5 we can decide whether a query set of type α is compromise-free if we look at the L-core of the corresponding root graph H. Hence, algorithms for finding an L-core are of practical interest.

A spectrally based algorithm for finding am L-core is provided by Theorem 4. One should find eigenvectors of -2 in $L(H)$ and apply formula (1) to obtain angles. This algorithm is of complexity $O(n^3)$. If we want to decide whether Q is compromise-free on the basis of the results of [**4**] by applying the Gaussian elimination then we again get the same complexity $O(n^3)$.

Thus it is of interest to find the L-core of a graph H by non-spectral means using directly the structure of H. After some introductory remarks we shall formulate such an algorithm.

We can assume that H is connected since otherwise we would consider each component separately.

A connected graph, different from a tree, consists of a *central part* and of some trees attached to the central part. The central part of a connected graph is defined as the maximal connected subgraph without vertices of degree 1. All edges in attached trees are weak and they do not belong to the L-core of H. However, all petals do belong to the central part. The central part can be obtained by successive deletion of vertices of degree 1 as far as they are present in the process of deletion. For a tree the central part is not defined.

The central part can contain some bridges. A bridge is a strong edge if it belongs to an odd dumbbell. The two odd cycles of the dumbbell should be in different blocks of H so that the bridge belongs to a path connecting the cycles. After deleting weak bridges the central part splits in some components. Each such component should be checked whether it contains some weak edges stemming from odd cycles.

In order to formulate our algorithm we need a definition and we have to prove a proposition.

The *block graph* $B(G)$ of a graph G has vertices corresponding to blocks of G with two vertices being adjacent if the corresponding blocks have a common cut point.

Proposition 4. *Let G be a connected graph containing at least one bridge. If all bridges of G are strong, then G coincides with its L-core.*

Proof. Let u be a bridge of G. Let A and B be the two components of $G - u$. Since u is strong, it is contained in an odd dumbbell. This implies that both A and B contain an odd cycle. For any edge v of G one of the following holds (i) v is a bridge, (ii) v is contained in an even cycle, (iii) v is contained in an odd cycle C. In case (i) v is strong since all bridges of G are strong. In case (ii) v is strong by definition. In case (iii) cycle C belongs either to A or B. Suppose it is in A and consider an odd cycle D of B. Cycles C and D are disjoint. Since G is connected, cycles C and D can be joined by a path thus forming an odd dumbbell. Hence, v is strong.

This completes the proof.

Now we can formulate the following algorithm for finding the L-core of a graph H.

1. Find components of H and consider only those which contain at least one cycle.

2. For each such component find its central part and delete attached trees.

3. For each central part find blocks and, in particular, bridges. Construct the block graph of the central part under consideration. Mark the blocks containing an odd cycle. (Since a petal is a block for itself, such blocks will be marked).

4. Find weak bridges, using information on marked blocks, and delete such bridges thus splitting the central part in some components. At most one component contains marked blocks since otherwise some deleted bridges are strong. Components without marked blocks belong to the L-core of H.

5. If there is no component with marked blocks we have identified the L-core. Otherwise consider the component with marked blocks. If the component contains a bridge, the component belongs to the L-core by Proposition 4.

6. Finally, in the case that the component considered contains no bridges, delete edges of odd cycles which do not belong to even cycles. In particular, a petal which is the only odd cycle should be always deleted. The considered graph could split further into some components. Components containing edges belong to the L-core.

The complexity of the described algorithm can be estimated as $O(n^2)$.

It would be interesting to study components in an L-core. From the algorithm described it follows that bipartite graphs without bridges (but possibly with cut points) can be components of an L-core. If a component of an L-core contains odd cycles, then it cannot contain attached trees but can have strong bridges. Not all such graphs can be components of an L-core since they can contain weak edges stemming from some odd cycles. We do not have any useful characterization of graphs that can appear in this case.

REFERENCES

1. L. Branković: *Usability of secure statistical databases.* PhD Thesis, Newcastle, Australia, 1998.
2. L. Branković, M. Miller, J. Širáň: *Graphs,* $(0,1)$*-matrices and usability of statistical databases.* Congressus Numerantium, **120** (1996), 169–182.
3. D. M. Cardoso: *Convex quadratic programming approach to the maximum matching problem.* J. Global Opt., **21** (2001), 91–106.
4. F. Y. Chin, G. Ozsoyoglu: *Auditing and inference control in statistical databases.* IEEE Transactions on Software Engineering, **SE8** (6), (1982), 574–582.
5. D. Cvetković, M. Doob, I. Gutman, A. Torgašev A: *Recent Results in the Theory of Graph Spectra.* North-Holland, Amsterdam, 1988.
6. D. Cvetković, M. Doob, H. Sachs: *Spectra of Graphs*, 3rd edition. Johann Ambrosius Barth Verlag, Heidelberg, 1995.
7. D. Cvetković, M. Doob, S. Simić: *Generalized line graphs.* J. Graph Theory, **5** (1981), No.4, 385–399.
8. D. Cvetković, P. W. Fowler, P. Rowlinson, D. Stevanović: *Constructing fullerene graphs from their eigenvalues and angles.* Linear Algebra Appl. **356** (2002), 37–56.
9. D. Cvetković, P. Rowlinson, S. K. Simić: *Eigenspaces of Graphs.* Cambridge University Press, 1997.

10. D. Cvetković, P. Rowlinson, S. K. Simić: *Graphs with least eigenvalue* -2; *the star complement technique.* J. Algebraic Combinatorics, **14** (2001), 5–16.
11. M. Doob: *An interrelation between line graphs, eigenvalues, and matroids.* J. Combinatorial Theory, Ser. B, **15** (1973), 40–50.
12. M. Doob, D. Cvetković: *On spectral characterizations and embedding of graphs.* Linear Algebra and Appl. **27** (1979), 17–26.
13. A. J. Hoffman: $-1-\sqrt{2}$. Combinatorial Structures and Their Applications. Proc. of the Calgary Intern. Conf. on Combinatorial Structures and their Applications held at the Univ. of Calgary, June, 1969, Guy R., Hanani H., Sauer N., Schönheim J., Gordon and Breach Sci. Publ., New York, 1970, 173–176.

School of Electrical Engineering
and Computer Science,
University of Newcastle,
Newcastle,
Australia
E–mail: lbrankov@cs.newcastle.edu.au

(Received September 22, 2003)

Faculty of Electrical Engineering,
University of Belgrade,
P.O.Box 35–54,
11120 Belgrade,
Serbia and Montenegro
E–mail: ecvetkod@etf.bg.ac.yu