FILOMAT (Niš) **13** (1999), 1–13

AN ALGORITHM FOR CONSTRUCTING GRAPHS WITH GIVEN EIGENVALUES AND ANGLES

Dragan Stevanović

Abstract. Let the eigenvalues of a graph and the angles between eigenspaces and the co-ordinate axes of the corresponding real vector space be given for a graph. Cvetković [2] gave a method of constructing a graph which is the supergraph of all graphs with given eigenvalues and angles. Based on this, we describe a branch & bound algorithm for constructing all graphs with given eigenvalues and angles.

1. Introduction

Let G be the graph on n vertices with adjacency matrix A. Let $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ constitute the standard orthonormal basis for R^n. Then A has spectral decomposition $A = \mu_1 P_1 + \mu_2 P_2 + \ldots + \mu_m P_m$, where $\mu_1 > \mu_2 > \ldots > \mu_m$ and P_i represents the orthogonal projection of R^n onto $\mathcal{E}(\mu_i)$ (moreover $P_i^2 = P_i = P_i^T$, $i = 1, \ldots, m$; and $P_i P_j = O$, $i \neq j$). The nonnegative quantities $\alpha_{ij} = \cos \beta_{ij}$, where β_{ij} is the angle between $\mathcal{E}(\mu_i)$ and $\mathbf{e}_j$, are called *angles* of G. Since P_i represents the orthogonal projection of R^n onto $\mathcal{E}(\mu_i)$ we have $\alpha_{ij} = \|P_i \mathbf{e}_j\|$. The sequence α_{ij} ($j = 1, 2, \ldots, n$) is the ith *eigenvalue angle sequence*; α_{ij} ($i = 1, 2, \ldots, m$) is the jth *vertex angle sequence.*

Cvetković ([1]) gave the first algorithm for construction of trees with given eigenvalues and angles. Then in [2] he gave a method that uses only eigenvalues and angles to construct the graph which is a supergraph of all graphs with given eigenvalues and angles. Such a supergraph is the *quasi-graph* in general case, which is described in Section 5. If we also know the eigenvalues and angles of the complementary graph, we can construct the *fuzzy image* of a graph, which enhances the quasi-graph. In the case of trees, that supergraph is the *quasi-bridge graph*, whose construction is

Received October 14, 1998

2000 *Mathematics Subject Classification.* 05C50.

Key words and phrases. Graphs, eigenvalues, angles.

much simpler than that of quasi-graph and fuzzy image. It is described in Section 4.

Further, in [3] Cvetković gave a lower bound on the distance between vertices based on eigenvalues and angles of graph. In Section 3 we give a new lower bound, similar to this one and show that the two are independent of each other. We also present some statistical data on random graphs.

Based on the lower bound on distance and the supergraph of all graphs with given eigenvalues and angles, in Section 6 we give a branch & bound algorithm to construct all graphs with given eigenvalues and angles.

2. Preliminary Lemmas

If a graph or vertex invariant can be determined provided the eigenvalues and angles are known, then the invariant is called *EA-reconstructible* ([1]). The basic property of angles is given in the next lemma.

Lemma 1. ([7]) *The number of closed walks of length s starting and terminating at vertex j is given by $\sum_{i=1}^{m} \mu_i^s \alpha_{ij}^2$.*

Corollary 2. ([7]) *The degree d_j of the vertex j, and the number t_j of triangles containing the vertex j, are given by*

$$d_j = \sum_{i=1}^{m} \alpha_{ij}^2 \mu_i^2, \quad t_j = \frac{1}{2} \sum_{i=1}^{m} \alpha_{ij}^2 \mu_i^3.$$

A partition of the vertex set of G is called *admissible* if no edge of G connects vertices from different parts; and subgraphs induced by the parts of an admissible partition are called *partial graphs* (thus a partial graph is a union of components, and the components are induced by the parts of the finest admissible partition). The spectra and angles of these partial graphs are called the *partial spectra* and *partial angles* corresponding to the original partition.

Lemma 3. ([6]) *Given the eigenvalues, angles and an admissible partition of the graph G, the corresponding partial spectra and partial angles of G are determined uniquely.*

Theorem 4. ([6]) *Given the eigenvalues and angles of a graph G, there is a uniquely determined admissible partition of G such that*

(i) in each partial graph all components have the same index, and

(ii) any two partial graphs have different indices.

For further properties of angles, see the monograph [7].

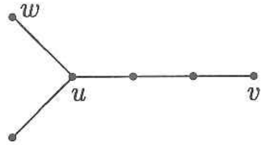

Figure 1: The tree from example.

3. Lower bounds on distance

From the spectral decomposition of A we have $a_{jk}^{(s)} = \sum_{i=1}^{m} \mu_i^s P_i\mathbf{e}_j \cdot P_i\mathbf{e}_k$. Since $|P_i\mathbf{e}_j \cdot P_i\mathbf{e}_k| \leq \|P_i\mathbf{e}_j\| \cdot \|P_i\mathbf{e}_k\|$ we get $a_{jk}^{(s)} \leq \sum_{i=1}^{m} |\mu_i^s|\, \alpha_{ij}\, \alpha_{ik}$. Let $d(j,k)$ be the distance between vertices j and k in G.

Lemma 5. ([3]) *If* $g = \min\, \{s : \sum_{i=1}^{m} |\mu_i^s|\, \alpha_{ij}\, \alpha_{ik} \geq 1\}$, *then* $d(j,k) \;\geq\; g$.

Lemma 6. *If* $g = \min\, \{s : \sum_{i=1}^{m} |\mu_i^{s+2}|\, \alpha_{ij}\, \alpha_{ik} \geq d_j + d_k + \delta_{s-1} - s\}$, *where* δ_{s-1} *is the sum of* $s-1$ *smallest degrees of vertices other than* j *and* k, *then* $d(j,k) \;\geq\; g$.

Proof Let $j = w_0, w_1, \ldots, w_{d(j,k)} = k$ be the shortest path between j and k. The number of paths of the length $d(j,k)+2$ between j and k which have the form $j = w_0, \ldots, w_i, u, w_i, \ldots, w_{d(j,k)} = k$, where u is arbitrary neighbor of w_i, is at least $d_j + d_k + \delta_{d(j,k)-1} - d(j,k)$. Hence, $d(j,k) \;\geq\; g$. □

Example. For the tree shown in Fig. 1, from Lemma 5 we have $d(u,v) \geq 2$, while from Lemma 6 it follows that $d(u,v) \geq 3$. On the other hand, from Lemma 5 it follows that $d(w,v) \geq 3$, while Lemma 6 gives only that $d(w,v) \geq 1$. This shows that the lower bounds given in these lemmas are independent of each other. In order to get a better lower bound, one must then take the greater of the values given by these lemmas. □

The average value of the lower bound (obtained from lemmas 5 and 6) in connected random graphs having from 10 to 40 vertices and given number of edges ($n-1$, n, $2n$, $3n$, $n\log n$, $n\sqrt{n}$ i $n^2/4$) is shown in Table 1. For each number of edges, 100 connected random graphs were taken. From this table it can be seen that the average value of the lower bound increases with the number of vertices in graphs having $O(n)$ edges, while it decreases in graphs with at least $O(n\log n)$ edges. The exact boundary where this average value is constant is somewhere between $O(n)$ and $O(n\log n)$.

vertices	$n-1$	n	$2n$	$3n$	$n\log n$	$n\sqrt{n}$	$n^2/4$
10	2,06	1,94	1,19	1,03	1,11	1,02	1,07
15	2,33	2,26	1,32	1,07	1,11	1,01	1,02
20	2,50	2,40	1,41	1,10	1,10	1,01	1,00
25	2,67	2,56	1,46	1,12	1,09	1,00	1,00
30	2,79	2,68	1,51	1,13	1,07	1,00	1,00
35	2,87	2,81	1,55	1,14	1,06	1,00	1,00
40	2,99	2,90	1,58	1,16	1,06	1,00	1,00

Table 1: Average value of lower bound on distance.

The average value of the lower bound for all pairs of vertices at given distance in connected random graphs with 30 vertices and given number of edges is shown in Table 2. As before, for each number of edges, 100 connected random graphs were taken. The symbol "—" in d^{th} row means that in the set of random graphs there was none with diameter $\geq d$. From this table we can see that in all cases the ratio between the lower bound and the distance decreases with the increase of distance. It also shows that the lower bound is almost unusable in graphs with at least $O(n\sqrt{n})$ edges.

4. Quasi-bridge graphs

The following theorem is proved in [2].

Theorem 7. *Let uv be a bridge of a graph G. Then $P_G^2 + 4P_{G-u}P_{G-v}$ is a square.*

The necessary condition for two vertices u and v to be joined by a bridge, provided by this theorem, is called the *bridge condition.* The *quasi-bridge graph* $QB(G)$ of the graph G is defined as the graph with the same vertices as G, with two vertices adjacent if and only if they fulfil the bridge condition. Define a quasi-bridge as an edge of $QB(G)$. If G is a tree, then we obviously have that G is a spanning tree of $QB(G)$.

The bridge condition is not sufficient for the existence of the bridge. On the other hand, there are trees for which the equality $QB(G) = G$ holds. Such examples are the stars S_n and the double stars $DS_{m,n}$ with $m \neq n$ (see [9]).

Here we deal with the number of quasi-bridges in trees. In Table 3 we give the statistical results obtained by determining the number of quasi-bridges in random trees which had from 5 to 30 vertices, where for each

distance	$n-1$	n	$2n$	$3n$	$n \log n$	$n\sqrt{n}$	$n^2/4$
1	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2	1,78	1,76	1,31	1,11	1,06	1,00	1,00
3	2,24	2,20	1,57	1,31	1,21	1,01	—
4	2,61	2,55	1,98	1,89	1,95	—	
5	2,95	2,88	2,55	2,33	—		
6	3,23	3,16	3,13	—			
7	3,47	3,42	4,00				
8	3,64	3,65	—				
9	3,79	3,90					
10	3,91	4,21					
11	3,76	4,16					
12	3,41	4,10					
13	3,79	4,00					
14	4,00	—					
15	—						

Table 2: Average value of lower bound for vertices at given distance.

number of vertices we randomly chose 100 trees. The number of vertices is shown in the first column, while the minimum, maximum and the average number of quasi-bridges in these trees are shown in the second, third and fourth column, respectively.

Data from this table show that many small random trees satisfy $QB(G) = G$. On the other hand, Cvetković [1] showed that for almost every tree G there is a nonisomorphic cospectral mate G' with the same angles. Hence, both G and G' must be spanning trees of $QB(G')$, and for almost every tree $QB(G) \neq G$.

Although it cannot be seen from Table 3 it is the case that $e(QB(T)) = \Theta(e(T)^2)$, where $e(G)$ is the number of edges of G. One example is shown in Fig. 2. It is a rooted tree of depth 2 where the root has a descendants, and each of them has exactly one descendant. All neighbors of the root are similar, hence have the same vertex-deleted characteristic polynomial. The same holds for all leaves. Since the bridge condition is satisfied for at least one pair of vertices consisting of a neighbor of a root and a leaf, it is also satisfied for all pairs of vertices consisting of an arbitrary neighbor of a root and an arbitrary leaf. Hence the tree T^* from Fig. 2 is a spanning subgraph of $QB(T)$. Further examples consist of rooted trees *regular* in the following

vertices	min	max	average
5	4	6	4.66
6	5	9	6.74
7	6	12	6.90
8	7	13	8.76
9	8	20	9.36
10	9	21	10.96
11	10	18	11.16
12	11	23	12.18
13	12	30	13.34
14	13	25	14.61
15	14	28	15.48
16	15	33	16.70
17	16	24	17.46
18	17	31	18.66
19	18	30	19.70
20	19	31	21.00
21	20	34	21.52
22	21	35	22.74
23	22	38	23.94
24	23	37	24.70
25	24	46	26.54
26	25	49	27.42
27	26	44	28.70
28	27	39	29.18
29	28	50	30.48
30	29	47	31.70

Table 3: Quasi-bridges in random trees.

sense: all nodes on the same level have the same number of descendants.

5. Quasi-graphs and fuzzy images

The following theorem is proved in [2].

Theorem 8. *Let G be a graph with n vertices and m edges, and let uv be an edge of G. Then there exists a polynomial $q(x)$ of degree at most $n-3$ such that*

$$(x^n - (m-1)x^{n-2} + q(x))P_G(x) + P_{G-u}(x)P_{G-v}(x) \text{ is a square.}$$

The necessary condition for two vertices u and v to be adjacent, provided by this theorem, is called the *edge condition.* The *quasi-graph* $Q(G)$ of the graph G is defined as the graph with the same vertices as G, with two vertices adjacent if and only if they fulfil the edge condition. Obviously, any graph is spanning subgraph of its quasi-graph.

If G is regular and both G and $\overline{G}$ are connected then from the eigenvalues and angles of G we also know the eigenvalues and angles of $\overline{G}$ [6]. Now, the edge condition in $\overline{G}$ is a necessary condition for non-adjacency in G, and any two distinct vertices of G are adjacent either in $Q(G)$ or in $Q(\overline{G})$. If they are adjacent in one and not adjacent in the other, then their status coincides with that in $Q(G)$. Thus, the *fuzzy image* $FI(G)$ is defined as the graph with

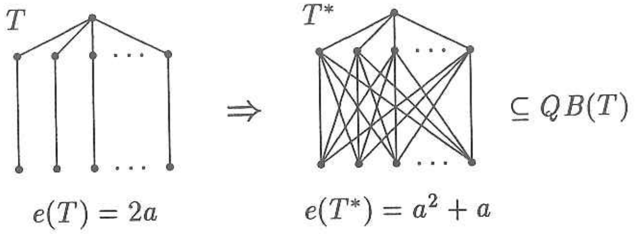

Figure 2: $QB(T)$ can have many edges.

the same vertex set as G and two kinds of edges, solid and fuzzy. Vertices u and v of $FI(G)$ are joined by a fuzzy edge if they are adjacent in both $Q(G)$ and $Q(\overline{G})$, otherwise they are joined by a solid edge if they are adjacent in $Q(G)$ and they are non-adjacent if they are non-adjacent in $Q(G)$.

Except for small values of n (up to 4), it is very difficult to use the edge condition practically. Since the coefficients of the characteristic polynomials are integers, we can use the following weaker corollary.

Corollary 9. *Let G be a graph with n vertices, and let uv be an edge of G. Then $P_{G-u}(n)P_{G-v}(n)$ is quadratic residue modulo $P_G(n)$ for every $n \in Z$.*

Using Corollary 9 involves a loss of information. Indeed, for regular graphs we got that usually only a few pairs of vertices are not joined by a fuzzy edge in the fuzzy image. Thus the problem of implementing the edge condition still remains.

6. The Constructing Algorithm

The algorithm presented at the end of this section is of branch & bound type. It does not assume anything about graph connectedness, but in the case of non-connected graphs, we can simplify the construction. Namely, we can find the partial eigenvalues and angles from Theorem 4. Then we construct all partial graphs with the corresponding eigenvalues and angles, after which we have to construct all the combinations of the partial graphs obtained.

Before the algorithm enters the main loop, it determines the graph G^* that is the supergraph of all graphs with the given eigenvalues and angles. Let G denote the putative graph with given eigenvalues and angles, with vertex set $V(G)$ and edge set $E(G)$. In the case of trees we have $G^* = QB(G)$, in the case of regular connected graphs with connected complements we have

$G^* = FI(G)$, while in other cases $G^* = Q(G)$. Then for each pair (u, v) of vertices, the algorithm determines the lower bound $d_l(u, v)$ on the distance in G between vertices u and v, based on Lemmas 5 and 6.

At level 0 we choose the vertex v_1 arbitrarily, and pass to level 1, where we enter the main loop. When we come to level i ($i \geq 1$) we have already constructed the subgraph G_i induced by vertices $v_1, v_2, \ldots, v_i$. Then we choose the vertex v_{i+1} from the set of remaining vertices, select its neighbors from the set $\{v_1, v_2, \ldots, v_i\}$, and pass to the next level, until G is constructed in whole.

The fact that at each level we know the induced subgraph of G provides us with the possibility of using the following well-known theorem.

Theorem 10. (see, for example, [8], p. 119) *Let A be a Hermitian matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and let B be one of its principal submatrices. If the eigenvalues of B are $\nu_1 \geq \nu_2 \geq \ldots \nu_m$ then $\lambda_i \geq \nu_i \geq \lambda_{n-m+i}$ ($i = 1, 2, \ldots, m$).*

The inequalities of Theorem 10 are known as *Cauchy's inequalities* and the whole theorem as the *Interlacing theorem.* If Cauchy's inequalities are not satisfied for G_i, then we are not on track, i.e. G_i can not be an induced subgraph of G and we must return to the previous level.

Suppose that the algorithm is currently at level i. Let $d_v^{G_i}$ denote the degree of v in G_i. If for some vertex u of G_i it is the case that

$$d_u^{G_i} + |\{v \in V(G) - V(G_i) : (u, v) \in E(G^*)\}| = d_u,$$

then we say that u is *forced at level* i, because u must be adjacent in G to all the vertices from $V(G) - V(G_i)$ to which it is adjacent in G^*.

Next we have to select the vertex v_{i+1}. Consider an arbitrary vertex $v \in V(G) - V(G_i)$, for which we introduce the following parameters. In the case that $G^* = FI(G)$ let

$$\begin{aligned} s_v^- &= |\{u \in V(G_i) : (u, v) \textit{is a solid edge in } FI(G)\}| \\ s_v^+ &= |\{u \in V(G) - V(G_i) : (u, v) \textit{is a solid edge in } FI(G)\}|, \\ o_v &= |\{u \in V(G_i) : (u, v) \in E(G^*), u \textit{is forced and } (u, v) \textit{is fuzzy edge}\}|. \end{aligned}$$

In other cases, let $s_v^- = s_v^+ = 0$ and

$$o_v = |\{u \in V(G_i) : (u, v) \in E(G^*) \textit{ and } u \textit{ is forced}\}|.$$

Finally, let

$$\begin{aligned} f_v^- &= |\{u \in V(G_i) : (u,v) \in E(G^*)\}| - s_v^- - o_v, \\ f_v^+ &= |\{u \in V(G) - V(G_i) : (u,v) \in E(G^*)\}| - s_v^+. \end{aligned}$$

Of course, it follows that v must be adjacent to $s_v^- + o_v$ vertices of G_i. Based on these parameters, we can determine the smallest number m_v and the largest number M_v of the remaining vertices of G_i that may be set as the neighbors of v in G_{i+1}:

$$(1) \qquad m_v = \max\{0, d_v - (s_v^- + s_v^+) - f_v^+ - o_v\},$$

$$(2) \qquad M_v = \min\{f_v^-, d_v - s_v^- - o_v\}.$$

Vertex v may be adjacent to at most $f_v^+ + s_v^+$ vertices from $V(G) - V(G_i)$, and since it has degree d_v in G, it follows that $m_v \geq d_v - (s_v^- + s_v^+) - f_v^+ - o_v$. Since m_v is non-negative, equation (1) holds. On the other hand, v must be adjacent to $s_v^- + o_v$ vertices from $V(G_i)$. Then the inequality $M_v \leq f_v^-$ follows from the fact that v may be adjacent to a vertex of G_i only if it is adjacent to that vertex in G^*, while the inequality $M_v \leq d_v - s_v^- - o_v$ holds since v is adjacent to at most d_v vertices of G_i. If for any v it is the case that $M_v < m_v$ we have to return to previous level.

If we select the vertex v as v_{i+1} then the number of neighborhoods of v in the set $\{v_1, v_2, \ldots, v_i\}$ that have to be examined at this level is equal to

$$\begin{aligned} N_v &= \binom{f_v^-}{m_v} + \binom{f_v^-}{m_v+1} + \ldots + \binom{f_v^-}{M_v} \\ &= \binom{f_v^-}{m_v} + \binom{f_v^-}{m_v}\frac{f_v^- - m_v}{m_v+1} + \ldots + \binom{f_v^-}{m_v}\frac{(f_v^- - m_v)\cdot\ldots\cdot(f_v^- - M_v + 1)}{(m_v+1)\cdot\ldots\cdot M_v} \\ &= \binom{f_v^-}{m_v}\left(1 + \frac{f_v^- - m_v}{m_v+1}\left(1 + \ldots + \frac{f_v^- - M_v + 2}{M_v - 1}\left(1 + \frac{f_v^- - M_v + 1}{M_v}\right)\ldots\right)\right). \end{aligned}$$

Hence, as the vertex v_{i+1} we choose the vertex v which minimizes the value N_v (in the algorithm we use the value $\log N_v$ computed using Stirling's formula).

Once the vertex v_{i+1} is selected, we have to specify its neighbors from the set $\{v_1, \ldots, v_i\}$ in order to construct the graph G_{i+1} completely. Let us introduce the following conditions that are applied in the algorithm:

(i) *the degree condition at level i* is satisfied if for every vertex v of G_{i+1} the degree of v in G_{i+1} is not greater than the degree of v in G (see Corollary 2).

(ii) *the triangle condition at level i* is satisfied if for vertex v of G_{i+1} the number of triangles in G_{i+1} containing v is not greater than the number of triangles of G containing v (see Corollary 2).

(iii) *the distance-3 condition at level i* is satisfied if for every pair (u, v) of vertices of G_{i+1} for which $d_l(u, v) \geq 3$ holds, u and v do not have a common neighbor in G_{i+1}.

Denote by F_{i+1} the set of forced vertices from $\{v_1, \dots, v_i\}$ that are neighbors of v_{i+1} in G^*. In case that $G^* = FI(G)$ we also put into F_{i+1} those vertices from $\{v_1, \dots, v_i\}$ that are connected by a solid edge to v_{i+1} in $FI(G)$. Of course, vertices from F_{i+1} must be adjacent to v_{i+1} in G_{i+1}. If any of the above conditions is not satisfied when we join v_{i+1} to the vertices from F_{i+1} then we have to return to the previous level. The neighborhood $S \subseteq \{v_1, \dots, v_i\}$ of v_{i+1} consisting of nonforced vertices must be such that also none of above conditions is broken. If any of the conditions is broken, we have to select the lexicographically nearest neighborhood satisfying them. This new neighborhood can not be a superset of one that does not satisfy them, due to the monotonicity of the conditions. If there is no such neighborhood, we have to return to the previous level.

We return to the previous level in the backward phase of the algorithm. Notice that when we are looking for the next neighborhood of nonforced vertices to be examined we allow it to be the superset of the current one.

The Constructing Algorithm
Input: Eigenvalues and angles of a graph.
Output: All graphs with given eigenvalues and angles.

```
begin
  find the supergraph $G^*$
  for all pairs $(u, v)$ find $d_l(u, v)$
  choose vertex $v_1$

  i=1
1. Forward phase
  while $i > 0$
    if $i$ is equal to the number of vertices then
      if the constructed graph has given eigenvalues and angles
      then print the graph
    else
      check whether any of the vertices $v_1, \ldots, v_i$ is forced at this level
      for each $v \in V(G) - V(G_i)$ find the smallest $m_v$ and
      the largest $M_v$ possible for the degree of $v$ in $G_{i+1}$
      if Cauchy's inequalities hold for $G_i$
        and $m_v \leq M_v$ for each $v \in V(G) - V(G_i)$
      then
        select vertex $v_{i+1}$
        determine the set $F_{i+1}$
        if $F_{i+1}$ does not break any condition then
          find the lexicographically smallest neighborhood $S_{i+1}$ of $v_{i+1}$
          consisting of non-forced vertices of $G_i$ ($m_{v_{i+1}} \leq |S| \leq M_{v_{i+1}}$)
          while $S_{i+1}$ does not satisfy the conditions
            find the next such neighborhood $S_{i+1}$ of $v_{i+1}$
            that is not a superset of the previous one
          if $S_{i+1}$ exists then
            $G_{i+1} \leftarrow G_i \cup \{v_{i+1}\} \times (F_{i+1} \cup S_{i+1})$
            $i \leftarrow i + 1$
            goto 1.

    $i \leftarrow i - 1$
  2. Backward phase
    while $i > 1$
      if some vertex became forced at level $i + 1$
      then it is no longer forced
      find the next set $S_{i+1}$ of non-forced neighbors
      that may be a superset of the previous one
      while $S_{i+1}$ does not satisfy the conditions
        find the next set $S_{i+1}$ of non-forced neighbors
        that is not a superset of the previous one
      if $S_{i+1}$ exists then
        $G_{i+1} \leftarrow G_i \cup \{v_{i+1}\} \times (F_{i+1} \cup S_{i+1})$
        $i \leftarrow i + 1$
        goto 1.
      else
        $i \leftarrow i - 1$
        goto 2.
  end.
```

We implemented this algorithm and it may be obtained from the author.

In [5] it is found that there are 58 pairs of cospectral graphs with the same angles on 10 vertices. However, the graphs from some 29 pairs are the complements of graphs from other 29 pairs. We tested the algorithm on those pairs of graphs which edge density is less than 1/2. For testing we used *Pentium MMX* machine on 200 MHz. The average of running times in seconds for both graphs in each pair is given in Table 4. From this table we can see that the running time depends mostly on the difference between the number of edges in the graph and the supergraph G^*.

To speed up calculations, the algorithm checks the fulfillness of Cauchy's inequalities after constructing the induced subgraph on $[2n/3]$ vertices. This checking is very time-consuming and it is the best to use it only on one level during the construction, but not too early since it will rarely happen that the Cauchy's inequalities are not fulfilled when less than half of the graph is constructed. The choice of $[2n/3]$ vertices for these graphs gave the improvement in running time ranging from 5% to 2000% compared to the case when the Cauchy's inequalities are not checked.

The distance-3 condition was of no help in this case since there are no pairs of vertices in these graphs for which the lower bound on distance is at least 3. However, the lower bound on distance helps to remove many edges from the supergraph G^*.

The hardest pair of graphs is no. 11, which is the only pair of regular graphs. All pairs of their vertices are joined by fuzzy edge in the fuzzy image and the lower bound on distance is trivial, so that the algorithm has to check all the graphs on 10 vertices with given vertex degrees and the numbers of triangles passing through them.

Acknowledgements. This paper is part of author's M.Sc. thesis, written under the supervision of Prof. Dragoš Cvetković at the University of Niš. The author is greatly indebted to the anonymous referees for several helpful comments which led to the considerable improvement of paper.

References

[1] Cvetković D., *Constructing trees with given eigenvalues and angles*, Linear Algebra and Appl., 105 (1988), 1–8

[2] Cvetković D., *Some possibilities of constructing graphs with given eigenvalues and angles*, Ars Combinatoria, 29A (1990), 179–187

[3] Cvetković D., *Some comments on the eigenspaces of graphs*, Publ. Inst. Math. (Beograd), 50(64) (1991), 24–32

[4] Cvetković D., Doob M., *Some developments in the theory of graph spectra*, Linear and Multilinear Algebra, 18 (1985), 153–181

pair	time	pair	time	pair	time
1	361	11	>10h	21	1057
2	64	12	260	22	5
3	914	13	614	23	7061
4	818	14	3428	24	269
5	35	15	2	25	375
6	71	16	45	26	454
7	6	17	31	27	15122
8	257	18	6836	28	1046
9	43	19	64	29	60
10	15287	20	920		

Table 4: The average running times for pairs of graphs from

[5] Cvetković D., Lepović M., *Cospectral graphs with the same angles and with a minimal number of vertices*, Univ. Beograd. Publ. Elektrotehn. Fak., Ser. Mat. 8 (1997), 88–102

[6] Cvetković D., Rowlinson P., *Further properties of graph angles*, Scientia (Valparaiso), 1 (1988), 41–51

[7] Cvetković D., Rowlinson P., Simić S., *Eigenspaces of Graphs*, Encyclopedia of Mathematics and Its Appl., Vol. 66, Cambridge University Press, 1997

[8] Marcus M., Minc H., *A Survey of Matrix Theory and Matrix Inequalities*, Allyn and Bacon, Inc., Boston, Mass., 1964

[9] Stevanović D., *Construction of graphs with given eigenvalues and angles* (Serbian), M.Sc. Thesis, University of Niš, Niš, 1998

Department of Mathematics, Faculty of Sciences and Mathematics, University of Niš
Ćirila i Metodija 2, 18000 Niš, Serbia, Yugoslavia
E-mail: dragance@filfak.filfak.ni.ac.yu