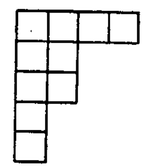
PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 39 (53), 1986, pp. 35–43

BRANCHING EXTENT AND SPECTRA OF TREES

Miłosz Michalski

Summary. Following the concepts of Ruch and Gutman [**7**] we discuss possible connections between branching extent and spectra of trees. It is suggested to what extent relations between spectra reflect partial ordering of trees according to their branching. We show how branching can be measured by certain coefficients of characteristic polynomial of a tree. The second part of the paper is devoted to the problem of constructing trees with a fixed measure of branching. A non-polynomial time algorithm is developed and its acceptably good performance in the majority of cases is documented by sample computation results.

1. Introduction. In [**7**] Ruch and Gutman introduced partial ordering of graphs according to the "degree of branching". The starting point of their investigation is the idea of a partial order (shortly PO) relation defined on the set of Young diagrams of a fixed order [**6**]. A Young diagram of order n is a pictorial representation of a partition of the integer $n \geq 1$, say $n_1 + n_2 + \cdots + n_k = n$, satisfying $n_1 \geq n_2 \geq \ldots n_k \geq 1$. The figure shows the Young diagram of order 10 corresponding to the partition $4+2+2+1+1$. A diagram can be described by its row $\gamma = \langle r_1, \ldots, r_k \rangle$ or, equivalently, column $\gamma = \langle c_1, \ldots, c_l \rangle$ partition of, "boxes" (in our example it is $(4,2,2,1,1)$ or $\langle 5,3,1,1 \rangle$; angle brackets are used to distinguish the latter case). For a diagram γ we define its row and column partial sums by



$$R_t(\gamma) = \sum_{i=1}^{t} r_i \quad \text{and} \quad C_t(\gamma) = \sum_{i=1}^{t} c_j, \quad t = 1, 2, \ldots$$

For $i > k$, $j > l$ we assume $r_i = c_j = 0$. Given two diagrams γ and δ of the same order we cay that γ is greater than δ, denoted by $\gamma \succ \delta$ iff

$$\forall t \geq 1 \qquad R_t(\gamma) \geq R_t(\delta)$$

(or equivalently $C_t(\gamma) \leq C_t(\delta)$). "$\succ$" is the PO mentioned above. By introducing the following operations

$$\gamma \sqcap \gamma' = \hat{\gamma} \quad \text{iff} \quad R_t(\hat{\gamma}) = \min\{R_t(\gamma), R_t(\gamma')\}$$
$$\gamma \sqcup \gamma' = \check{\gamma} \quad \text{iff} \quad C_t(\check{\gamma}) = \min\{C_t(\gamma), C_t(\gamma')\}$$

AMS Subject Classification (1980): Primary 05C05,05C50; Secondary 06A10

one obtains diagram lattice, [**6**]. We denote it by Ļ_n being the order of diagrams.

Let Γ_m be the class of all simple graphs with m edges and without isolated vertices. Let $G, H \in \Gamma_m$ and let $V(G) - \{v_1, \ldots, v_k\}$, $V(H) = \{w_1, \ldots, w_i\}$ be their vertex sets ordered according to the valency, $d(v_1) \geq \cdots \geq d(v_k)$ and $d(w_1) \geq \cdots \geq d(w_i)$. We say that G is more branched than H, denoted by $G \succ H$, iff

$$\forall t \geq 1 \qquad \sum_{i=1}^{t} d(v_i) \geq \sum_{i=1}^{t} d(w_i)$$

Now let $\approx$ be the equivalence relation on Γ_m defined by $G \approx H$ iff $G \succ H$ and $H \succ G$ (i.e. G and H have the same valency sequences). Then $\text{Ģ}_m = (\Gamma_m, \succ)/\approx$ is isomorphic bein subset (not a sublattice in general) of $\mathbf{C}L_{2m}$, the equivalence classes of $\approx$ being in correspondence with diagrams formed by vertex degrees. Since not all partitions of $2m$ are graphic (i.e. not all of them are degree sequences of simple graphs) the subset mentioned above is proper for details see [**7**]). However, if we restrict Γ_m $(m \geq 2)$ to the class of trees only, denoted by Θ_m, we can map $\text{Į}_m = (\Theta_m, \succ)/\approx$ isomorphically onto the full lattice Ļ_{m-1}.

Lemma 1. $\text{Į}_m \cong \text{Ļ}_{m-1}$.

Proof. Let $[T] \in \text{Į}_m$ and let $d(v_1) \geq \cdots \geq d(v_k) > d(v_{k+1}) = \cdots = d(v_{m+1}) = 1$ be the degree sequence of T. We define the mapping $\varphi : \text{Į}_m \to \text{Ļ}_{m-1}$ by $\varphi([T]) = (d(v_1) - 1, \ldots, d(v_k) - 1)$. Clearly φ is "$1-1$" and also "onto" since for any $\gamma = (r_1, \ldots, r_k) \in \text{Ļ}_{m-1}$ the degree sequence

$$r_1 + 1, \ldots, r_k + 1, \quad \underbrace{1, \ldots, 1}_{m-k+1 \text{ times}}$$

can be realized as a tree with m edges. Moreover it can be easily verified that

$$\forall [T_1], [T_2] \in \text{Į}_m [T_1] \succ [T_2] \quad \text{iff} \quad \varphi([T_1]) \succ \varphi([T_2])$$

and thus φ is an isomorphism. $\square$

In the present paper we deal with the problem of how far spectra of trees reflect their branching. The spectrum of a graph G, $Sp(G)$, is the nonincreasing sequence of the eigenvalues of the adjacency matrix $A(G)$ of G, [**2**]. By $Sp^2(G)$ we mean the nonincreasing sequence of squared elements of $Sp(G)$, i.e. the eigenvalues of $A^2(G)$. Any set of (not necessarily integer) partitions of a fixed number q can be partially ordered similarly as Young diagrams are [**5**], i.e. if $x = (x_1 \geq \cdots \geq x_k)$ $y = (y_1 \geq \cdots \geq y_l)$, $\sum x_i = \sum y_i = q$, then $x \succ y$ iff $\forall t \geq 1 \sum_{i=1}^{t} x_i \geq \sum_{i=1}^{t} y_i$. Let G be a graph with a degree sequence $d(G) = (d(v_1) \geq \cdots \geq d(v_n))$. It is a well known fact form the matrix theory, [**5**], that $Sp^2(G) \succ d(G)$. Moreover, as it has been reported by several authors, for example [**3**], [**4**], the increase of branching is usually connected with the increase of the largest eigenvalue of a graph, which has been used succesfully in the field of quantum chemistry. However, for $G, H \in \Gamma_m$, the relation $G \succ H$ implies neither $Sp^2(G) \succ Sp^{(}H)$ nor $Sp(G) \succ Sp(H)$

in general. The same concerns trees. In spite of this, we could still formulate the question whether it is possible to embed Ļ_m isomorphically into the PO set $S = (\{Sp^2(T) : T \in \Theta_m\}, \succ)$ in such a way that if $\varphi([T]) = Sp^2(T')$ then $T' \in [T]$, φ being the embedding. The inspection of the tables of tree spectra shows that it is not so, at least for $m = 7, 8, 9$, while the weaker request of homomorphic embedding is fulfillable. However, general answers for both cases need formal proofs. Presently we shall establish a bit humbler result that connects tree spectra and the branching.

II. Spectra of trees and quantitative characterization of branching. Every measurement of branching extent according to some numerical scale corresponds to a homomorphic mapping of the diagram lattice into real numbers. Unfortunately all such mappings image incomparable diagrams as comparable values. It means, in particular, that more branched graphs would always be assigned greater (or smaller) values of the function used for measuring but, in general, not conversely. Formally, $\forall \gamma, \gamma' \in \text{Ļ}_n\, \gamma \succ \gamma'$, $\gamma \neq \gamma' \Rightarrow f(\gamma) > f(\gamma')$ (or $f(\gamma) < f(\gamma')$), f being the measuring function. Presently we introduce a homomorphism $b : \text{Ļ}_n \to \mathbf{R}$ that will be related to some spectral characteristics of $n+2$ — vertex trees represented by diagrams of Ļ_n.

Let $\gamma = (r_1, \dots, r_k) \in \text{Ļ}_n$ and set $s_i = \sum_{j=1}^{k} r_j = n - R_{i-1}(\gamma)$, $i = 2, \dots, k$. Then

$$b(\gamma) \stackrel{\text{def}}{=\!=} \sum_{i=1}^{k-1} r_i s_{i+1}$$

Lemma 2. $\forall \gamma, \delta \in \text{Ļ}_n\, \gamma \succ \delta,\ \ \gamma \neq \delta \Rightarrow b(\gamma) < b(\delta)$

Proof. Observe that $\gamma \succ \delta$ means that γ can be obtained from δ by moving boxes exclusively upward, i.e. from shorter rows to longer or equal ones. Thus, it is sufficient to consider $\delta = (r_1, \dots, r_i, \dots, r_j, \dots, r_k)$ and $\delta^+ = (r_1, \dots, r_i + 1, \dots, r_j - 1, \dots, r_k)$, $\delta^+ \succ \delta$, for which we have

$$b(\delta^+) = b(\delta) + r_j - r_i - 1 < b(\delta)$$

since $r_j \leq r_i$ for $j > i$. $\square$

Given $T \in \Theta_{n-1}$, we consider its characteristic polynomial

$$P_T(\lambda) = \det(\lambda I - A(T)) = \sum_{i=0}^{n} a_i \lambda^{n-1} \tag{2.1}$$

Due to the well known "coefficients theorem" [**2**, theorem 1.3, p. 32) the coefficients a_i of (2.1) are strictly related to the number of subgraphs of T composed of several disjoint copies of K_2. In particular, a_4 is equal to the number of pairs of disjoint edges contained in T.

Theorem 1. *Let $T \in \Theta_{n-1}$, $P_T = \sum_{i=0}^{n} a_i \lambda^{n-i}$, and let $\gamma_T = (d_1, \dots, d_k) \in \text{Ļ}_{n-2}$ be the diagram representing T. Then $b(\gamma_T) = a_4$.*

Proof. To prove the theorem we have to count the number of pairs of disjoint edges in a tree T with the degree sequence $d(v_i) = d_i + 1$, $i = 1, \dots, n$, where $d_i = 0$

for $i > k$. Any fixed edge $\{x, y\}$ of T combined into pairs with others contributes the quantity $n - d(x) - d(y)$ to the value of a_4. By summation over all edges of T, where all the pairs are counted twice, we obtain

$$\begin{aligned} 2a_4 &= (n-1)n - \sum_{i=1}^{n} d(v_i)^2 = (n-2)^2 - \sum_{i=1}^{n} (d(v_i) - 1)^2 = \\ &= (n-2)^2 - \sum_{i=1}^{k} d_i^2 = \left(\sum_{i=1}^{k} d_i \right) - \sum_{i=1}^{k} d_i^2 = 2 \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d_i d_j = 2b(\gamma_T). \quad \square \end{aligned}$$

Note 1. Let $T \in \Theta_{n-1}$, $A = A(T)$ and $\lambda_i \in Sp(T)$ $i = 1, \ldots, n$. We have

$$TrA^4 = \sum_{i=1}^{n} \lambda_i^4 = 2(n-1)^2 - 4a_4$$

and hence

$$(\gamma_T) = \frac{1}{2}(n-1)^2 - \frac{1}{4} \sum_{i=1}^{n} \lambda_i^4$$

where $\gamma_z \in \mathcal{L}_{n-2}$ represents T. Thus $\sum \lambda_i^4$ can be used as a measure of branching as well.

Note 2. Other coefficients a_i of P_T are not as useful as a_4 for characterization of branching. In particular, $a_i = 0$ for i odd, $a_2 = n - 1$ for all $T \in \Theta_{n-1}$ and each of $a_{2,i}$, $i > 2$, usually takes distinct values for different trees with identical degree sequences. However, if for each diagram (degree sequence) a unique cannonical tree realization C is considered (as for example ordered caterpillar), one obtains for a fixed $i > 4$: $\gamma_C \succ \gamma_{C'}$, $\gamma_C \neq \gamma_{C'} \Rightarrow a_i \leq a_i'$ (or $a_i \geq a_i'$), where a_i and a_i' are the i-th coefficients of P_C and $P_{C'}$, respectively, and the greater i is the more often the equality holds.

Note 3. If G is an arbitrary multigraph with m edges, none of which is a loop, then the same argumentation as in the proof of theorem 1 shows the number of pairs of disjoint edges in G to be equal to $[m(m+1) - \sum_i d(v_i)^2]$. This quantity, as well as $\sum d(v_i)^2$, can be used as a measure of branching. It is not simple, however, to employ any spectral characteristics of G in this context, since the number of cycles of appropriate length is also incorporated in each coefficient a_i of P_G.

III. Construction of trees with a fixed branching. For an exhaustive study of a branching homomorphism $f : \mathcal{L}_n \to \mathbf{R}$ it may be necessary to find the reverse (in general multivalued) function f^{-1}. In our case the problem of determining the values of b^{-1} is equivalent to the one of solving the system of equations

$$\sum_{i=1}^{n} r_i = n \qquad \sum_{i=1}^{n} r_i^2 = c \tag{3.1}$$

in nonnegative integers with additional condition $r_1 \geq \cdots \geq r_n \geq 0$, where $r_i's$ are the searched components of a diagram $\gamma \in \mathcal{L}_n$ and $c = n^2 - 2b(\gamma)$, $(n, b(\gamma)$ given). One can also regard it as a problem of tree reconstruction from certain limited amount of spectral information. Actually we search only for possible degree sequences rather than for trees themselves, but once the degree sequence is known, one can easily find all the corresponding trees by means of constrained switching construction (see [**1**], [**8**]). In general, even if the whole spectrum of a tree is known, one cannot find the unique valency sequence since there are many cospectral and nonisomorphic trees, [**2**].

LEMMA 3. *Let* $\gamma = (r_1, \ldots, r_k) \in \mathcal{L}_k$ *and* $c = n^2 - 2b(\gamma)$. *Then the following inequalities hold*

$$c/n + \Delta \leq r_1 \leq 1/2 + \sqrt{c - n + 1/4} \tag{3.2}$$

where $\Delta = \Delta(c, n, r_1) \geq 0$.

Proof. Consider two diagrams $\alpha, \beta \in \mathcal{L}_n$, $\alpha = (p, 1, \ldots, 1)$, $\beta = (q, \ldots, q, s)$ $s = n \mod q$, where the numbers p and q are chosen so that

$$b(\alpha) \leq b(\gamma) \leq b(\beta) \tag{3.3}$$

Since $\alpha = \sqcap\{\xi \in \mathcal{L}_n : \xi = (x_1, \ldots, x_k), x_1 \geq p\}$
$\beta = \sqcup\{\xi \in \mathcal{L}_n : \xi = (x_1, \ldots, x_k), x_1 \leq q\}$
then we have $q \leq r_1 \leq p$. Therefore we have to minimize p and maximize q subject to (3.3). We have $n^2 - 2b(a) = p^2 + n - p \geq c$, which, by minimizing p gives the upper bound in (3.2). On the other hand $n^2 - 2b(\beta) = q(n - s) + s^2 \geq c$ and thus

$$q \leq \frac{c - s^2}{n - s} = \frac{c}{n} + \Delta$$

To complete the proof we have to show that

$$\frac{c - s^2}{n - s} - \frac{c}{n} = \frac{(c - ns)s}{(n - s)n} \geq 0$$

i.e. $ns \leq c$. The last inequality follows from

$$ns = s(n - s) + s^2 < q(n - s) + s^2 \leq c. \quad \square$$

Lemma 3 allows us to develop an algorithm that solves (3.1). Suppose we are given the values $n^{(0)} = n$ and $c^{(0)} = c = n^2 - 2b(\gamma)$. Applying (3.2) we find upper $r^{(1)}_{\max}$ and lower $r^{(1)}_{\min}$ bounds to r_1. Then for any fixed value of r_1 ($r^{(1)}_{\min} \leq r_1 \leq r^{(1)}_{\max}$) we compute the numbers $n^{(1)} = n^{(0)} - r_1$. and $c^{(1)} = c^{(0)} - r_2$, which are used similarly for the determination of the bounds $r^{(2)}_{\min}$ and $r^{(2)}_{\max}$ to r_2. For a selected r_2 ($r^{(2)}_{\min} \leq r_2 \leq \min\{r^{(2)}_{\max}, r_1\}$), we find $n^{(2)} = n^{(1)} - r_2$ and $c^{(2)} = c^{(1)} - r_2^2$ and so on. If at the i-th step we obtain $c^{(i)} = n^{(i)}$, then the sequence

$$r_1, \ldots, r_i, \underbrace{1, \ldots, 1}_{n^{(i)} \text{ times}}$$

solves (3.1). The lack of solutions with the first i components fixed as above will be detected by the occurrence of the condition $r_{\min}^{i+1} > r_{\max}^{i+1}$. By selecting other values of $r_1, \ldots, r_i$ we arrive at different solutions, if any. The correctness of the method will be evident if we prove that the conditions $n \leq c \leq n^2$ imply

$$\forall r_{\min}(n,c) \leq r \leq r_{\max}(n,c) \quad 0 \leq n-r \leq c-r^2 \leq (n-r)^2 \tag{3.4}$$

The first inequality $0 \leq n-r$ is trivial since $r_{max} = 1/2 + \sqrt{c-n+1/4} \leq n$ holds for $c \leq n^2$. The second one $n-r \leq c-r^2$ is satisfied for $1-r_{\max} \leq r \leq r_{\max}$. Finally,

$$c - r^2 \leq (n-r)^2 \tag{3.5}$$

holds for any r iff $c \leq n^2/2$. For $c > n^2/2$ we have $r_{\min} > n/2$. Since for $r \geq n/2$ the function $(n-r)^2 + r^2 - c$ increases, it is sufficient to show that (3.5) holds for $r = r_{\min}$. In practice $r_{\min}$ is set to the smallest value q for which $c \leq q(n - n \mod q) + (n \mod q))^2$ holds. In addition, in our case we have $n \mod r_{\min} = n - r_{\min}$. Therefore,

$$c - r_{\min}^2 \leq r_{\min}(n - r_{\min} - (n - r_{\min})) + (n - r_{\min})^2 = (n - r_{\min})^2 \tag{3.6}$$

which completes the proof of (3.4).

We express the algorithm in the form of a recursive Pascal procedure. The parameter i designates the step number (level of recursion), while the global vector R is used to store partial solution obtained so far, with $R_o = n$ additionally. The value of $c = n^2 - 2b$ we start with is assumed to fullfil $n \leq c \leq n^2$.

procedure DIAG (i, n, c: integer);
  var $r_{\min}$, $r_{\max}$, r, n', c': integer;
begin
  $r_{\min}$:= $\min\{r : r * (n - n \,\mathbf{mod}\, r) + (n - n \,\mathbf{mod}\, r)^2 \geq c\}$;
  $r_{\max}$:= $\min\{\text{trunc } (0.5 + \text{sqrt}\,(c - n + 0.25)), R_{i-1}\}$;
  for r:= $r_{\min}$ **to** $r_{\max}$ **do begin**
    n':= $n - r$;
    c':= $c - r^2$;
    R_i:= r;
    if $n' = c'$ **then** output R supplied by n' "1-es"
    else DIAG $i+1, n', c'$
  end
end;

Note 1. When $c \geq n^2/2$ the computation of $r_{\min}$ can be simplified to

$$r_{\min} := \lceil 0,5 * (n + \text{ sqrt}\,(2 * c - n^2)) \rceil;$$

since the argument of the celling $\lceil \ \rceil$ solves (3.5) with equality.

Note 2. The output of DlAG of the form

$$R_1+1,\dots,R_i+1,\underbrace{2,\dots,2}_{n'},\ \underbrace{1,\dots,1}_{R_0+2-n'-i}$$

gives the degree sequence of a tree with the value of branching b we started with.

IV. Computational experiments. Actually the presented algorithm realizes an extensive search in a subset of N^n and its complexity is not polynomial in any way. In spite of this sample computations show its unexpectably efficient behaviour in the majority of cases. This can be explained by the existence of many different solutions of (3.1) for smaller c, which causes most of recursive tracks to finish with success. On the other hand, for greater c, when the number of solutions decreases, a lot of further work is eliminated by the constrained selection of greater values of r in the first steps. Table 1 presents results of computations performed on an $R32$ computer. For each n seven values of c were selected according to $c \approx kn^2/8$, $k = 1,2,\dots,7$ approximately. For each pair (n,c) all the solutions

$$r_1^{(i)} \geq \dots \geq r_{k_i}^{(i)} \geq 1 \qquad i=1,\dots,s$$

of (3.1) were generated and the following quantities were recorded:

s	— # of solutions
cls	— total # of DIAG calls
$Mk = \frac{1}{s}\sum_{i=1}^{s} k_i$	— averge solution length
$Mcls = cls/s$	— average # of calls per 1 solution generated
$\rho = \frac{1}{cls}\sum_{i=1}^{s} k_i$	— averge # of components r_i generated per 1 call
T	— amount of CPU time used (in seconds)

It should be noted that for greater n the algorithm seems to behave more efficiently in average, which can be observed by the increase of the ρ proportion.

The situation when only one solution of (3.1) is requested was also examined. In this case the program terminates after generating the first solution

$$r_1(c) \geq \dots \geq r_{k_c}^{(c)} \geq 1$$

for particular n and c (or after finding that no solution exists), recording the number of DIAG calls $= cls\,(n,c)$. Two strategies were examined: **I** — select r as small as possible at each step (**for** $r := r_{min}$ **to** $r_{max}\dots$), **II** — select the greatest possible r (**for** $r := r_{\max}$ **downto** $r_{\min}\dots$). For each n in the experiment c took all possible values, i.e. $c = n, n+2, n+4, \dots, n^2$. Denote by S_n the set of those values of c for which solution of (3.1) exists and by U_n — the set of c for which (3.1) is unsolvable. Table 2 contains the following data that allow us to compare the efficiency of both strategies in solvable cases:

$s = |S_n|$ — # of solvable cases

$Mcls = \dfrac{1}{s} \sum_{c \in S_n} cls(n,c)$ — average # of DIAG calls

$Maxcls = \max_{c \in S_n} cls(n,c)$ — maximal # of DIAG calls

$Mk = \dfrac{1}{s} \sum_{c \in S_n} k_c$ — average solution length

$\sigma = \dfrac{1}{s} \sum_{c \in S_n} \frac{cls(n,c)}{k_c}$ — average # of DIAG calls per 1 solution component r_i generated

n	c	s	cls	Mk	$Mcls$	ρ	T
25	547	0	1	—	—	—	0.0008
	469	0	1	—	—	—	0.0008
	391	0	2	—	—	—	0.0016
	313	2	8	3.0	4.0	0.8	0.0063
	235	4	13	6.0	3.3	1.8	0.0103
	157	21	53	7.0	2.5	2.8	0.0424
	79	27	88	11.4	3.3	3.5	0.0705
50	2186	0	1	—	—	—	0.0008
	1874	1	3	3.0	3.0	1.0	0.0024
	1562	2	7	5.5	3.5	1.6	0.0056
	1250	8	27	6.4	3.4	1.9	0.0216
	936	33	106	8.4	3.2	2.6	0.0845
	624	252	779	9.5	3.1	3.1	0.6230
	312	1149	4275	14.3	3.7	3.9	3.4200
100	8750	0	1	—	—	—	0.0008
	7500	1	3	6.0	3.0	2.0	0.0024
	6250	9	21	5.9	2.3	2.5	0.0168
	5000	101	364	10.0	3.6	3.0	0.2913
	3750	1030	3295	11.8	3.2	3.7	2.6360
	2500	13538	47287	13.8	3.5	3.9	37.8300
	1250	282549	1139250	18.6	4.0	4.6	911.5000
250	54686	0	1	—	—	—	0.0008
	46874	98	302	10.3	3.1	3.3	0.2418
	39062	2079	8385	16.8	4.0	4.2	6.7080
	31250	50373	206147	18.3	4.1	4.5	164.9200
	23436	1550275	6933214	21.6	4.5	4.8	5546.5800

Tab. 1

n	s	$strat.$	$Mcls$	$Maxcls$	Mk	σ
25	193	**I**	5.3	14	5.7	1.1
		II	2.5	5	9.8	0.4
50	872	**I**	8.1	29	6.9	1.4
		II	2.9	6	18.0	0.3
100	3881	**I**	11.4	65	7.9	1.7
		II	3.3	8	34.3	0.2
250	26753	**I**	17.4	307	8.3	2.4
		II	3.6	8	83.5	0.1
500	112337	**I**	24.4	728	10.3	3.1
		II	3.8	7	166.0	0.06
1000	464693	**I**	—	—	—	—
		II	3.9	8	331.3	0.04

Tab. 2

It can be clearly seen that the first strategy is much worse than the second one. However, the worst cases for strategy **I** appear rarely, for single values of c: for all n examined less than $n/2$ calls of DIAG suffice to generate a solution in nearly 99 % of solvable cases. Finally, table 3 documents the extremely good performance of the algorithm in unsolvable cases. Obviously, this is independent of the strategy chosen. Table 3 contains

$u = |U_n|$ — # of unsolvable cases

$Mcls = \frac{1}{u} \sum_{c \in U_n} cls(n,c)$ — average # of DIAG calls

$Maxcls = \max_{c \in U_n} cls(n,c)$ — maximal # of DIAG calls

n	u	$Mcls$	$Maxcls$
25	108	1.2	4
50	354	1.3	9
100	1070	1.3	8
250	4373	1.3	11
500	12414	1.3	10
1000	34808	1.2	10

Tab. 3

V. Acknowledgnment. The author is greatly indebted to Prof. R. S. Ingarden, Prof. E. Ruch and Dr. M. M. Sysło for their valuable remarks.

REFERENCES

[1] C. J. Colburn, *Graph Generation*, Research Report CS–77–37, Dept. Computer Sci. Univ. Waterllo, 1977.

[2] D. M. Cvetkovij, M. Doob, H. Sachs, *Spectra of Graphs–Theory and Applications*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1980.

[3] D. M. Cvetković, I. Gutman, *Note on branching*, Croat. Chem. Acta, **49** (1977), 115–121.

[4] L. Lovász, J. Pelikán, *On the Eigenvalues of Trees*, Period. Math. Hungar. **3** (1973), 175–182.

[5] A. W. Marshall, I. Olkin, *Inequalites: Theory of Majorization and its Applications*, Academic Press, 1979.

[6] E. Ruch, *The Diagram Lattice as Structural Principle*, Theor. Chim. Acta (Berlin) **38** (1975), 167–183.

[7] E. Ruch, I. Gutman. *The branching extent of graphs*, J. Combin. Inform. System Sci. **4** (1979), 285–295.

[8] M. M. Syło, *On tree and unicyclic realizations of degree seyuences*, Demonstatio Math. **15** (1982), 1071–1076.

Institute of Physics
Nicolas Copernicus University
Grudziadzka 5/7
87-100 Torun, Poland

(Received 24 04 1985)