Prediction of ^{13}C NMR Chemical Shifts by Neural Networks in a Series of Monosubstituted Benzenes

[a]Š. SKLENÁK*, [b]V. KVASNIČKA, and [b]J. POSPÍCHAL

[a]*Chemical Faculty, Technical University, CZ-637 00 Brno*

[b]*Department of Mathematics, Faculty of Chemical Technology, Slovak Technical University, SK-812 37 Bratislava*

Received 22 April 1993

Feed-forward back-propagation neural networks with one output neuron were used. This approach was compared with a neural network with four output neurons giving chemical shifts of *ipso*, *ortho*, *meta*, and *para* positions in a series of monosubstituted benzenes. One-output neural networks, each one giving shift for one position of carbon, seem to give better results than the four-output neural network. The work shows that finely tuned simple neural network can work better than the sophisticated neural methods.

In the presented work an application of neural network approach to computation of chemical shifts in ^{13}C NMR spectrum of *ipso*, *ortho*, *meta*, and *para* carbon atoms of monosubstituted benzenes is described. Recently, ^{13}C NMR chemical shifts of monosubstituted benzenes have been studied [1] so that these entities are correlated with respect to the sigma resonance and inductive constants. It was demonstrated in this work that good correlation is obtained only for *para* position while for other three positions (*ipso, ortho,* and *meta*) a poor correlation was achieved. The sigma resonance and inductive constants of functional groups have been successfully predicted by neural networks in our recent work [2], where the functional groups were determined by the same descriptors as the so-called "14-input" (see below). Computation of shifts of NMR spectra is a laborious and complicated problem from the point of view of quantum chemistry. We have tried a neural network [3] as a model for strongly nonlinear relations structure—chemical shifts in ^{13}C NMR spectrum [4—7].

Neural networks are a very useful tool for solving various kinds of problems [8—10], which was applied also in chemistry [11, 12]. This method is helpful for finding relationships between structural — input entities (in our case descriptors that determine structure of molecules) and properties — output entities (here chemical shifts) when the mathematical description of these relationships is not known or is too difficult to use.

The neural networks used most often for solutions of such problems are neural networks with back-propagation strategy [3]. The neural network used in the presented work was composed of input layer, one hidden layer, and output layer (Fig. 1). The number of input neurons is determined by the number of input descriptors. Number of output neurons is given by the number of required output pieces of information. Number of hidden neurons has to be optimized, because there is no general rule determining this number.

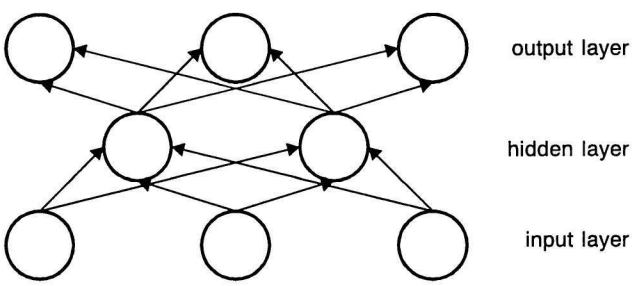

Fig. 1. General structure of three-layered neural network with three input neurons, two hidden neurons, and three output neurons.

The main difficulty with performance of neural networks in solving actual problems is a relevant choice of definition of input and output entities. General rules do not exist, therefore various possibilities must be tried to find out the best one. Here it stands for coding a structure of molecules into a sequence of non-negative integers. Coding of output was not difficult in our case, because the chemical shift is a single real number. Coding of input has been solved by two possibilities, with 14 and 20 input numbers, respectively, as described further.

Our goal in this work was to prove that even single neural network, when finely tuned, can give better results than the sophisticated techniques [4, 5].

*The author to whom the correspondence should be addressed.

THEORETICAL

Definition of Input

Two different ways of definition of input entries were used. The first "14-input" was based on our previous work [4, 5]. Substituent was separated into three levels ascribed to nonhydrogen atoms (Fig. 2):

1. The first level atom is that directly connected with benzene core,

2. the second level atoms are directly connected with the first level atom, and

3. the third level atoms are directly connected with the second level atoms.

Its single entries are described in Table 1. Examples of inputs are displayed in Fig. 3.

The second approach "20-input" was taken from [13] and is based on a connection table. Input was defined as a 20-element vector. Four consequent elements always relate to one nonhydrogen atom of substituent. Firstly atoms of substituent must be numbered. For output to be consistent, arbitrary but unambiguous rules for this numbering must be chosen. Presently used rules were:

1. Numbering follows levels,

2. within one level a lower sequence number is assigned to an atom with higher proton number,

3. lower sequence number is assigned to atom joined with atom from preceding level by a bond with lower multiplicity.

The rules code unambiguously all the functional groups used here. An example of numbering of atoms according to the above specified rules is displayed in Fig. 4.

20-Element input vector is composed of five 4-tuples, each describing one nonhydrogen atom.

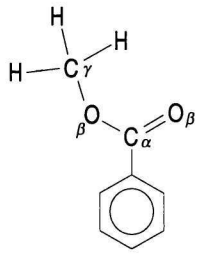

Fig. 2. Division of atoms in substituent into levels is described by attached marks α, β, γ for the first, second, and third level, respectively.

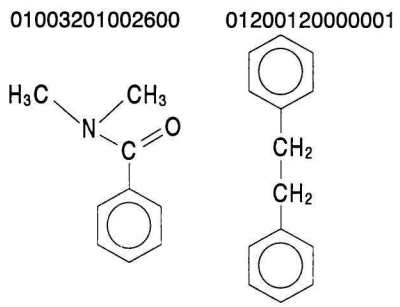

Fig. 3. Description of substituents by "14-input" vectors.

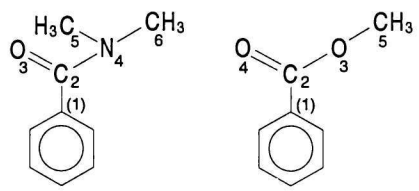

Fig. 4. Example of numbering of atoms of substituents for "20-input".

This presumes that substituent can have at most 5 nonhydrogen atoms. The only exception is phenyl,

Table 1. Fourteen Descriptor Entries of Functional Groups

d_l	Meaning of descriptor
	First-level descriptors
d_1	Number of lone electron pairs on the first level atom
d_2	The main quantum number of the first level atom decreased by 1
d_3	Number of hydrogen atoms attached to the first level atom
d_4	Entry equals 1, when the first level atom is a part of other benzene ring, otherwise it equals 0
	Second-level descriptors
d_5	Number of lone electron pairs on the second level atoms
d_6	Sum of the main quantum numbers (each decreased by 1) of the second level atoms
d_7	Number of hydrogen atoms attached to the second level atoms
d_8	Number of π bonds that connect the first and the second level atoms
d_9	Number of the second level atoms incorporated in benzene cores
	Third-level descriptors
d_{10}	Number of lone electron pairs on the third level atoms
d_{11}	Sum of the main quantum numbers (each decreased by 1) of the third level atoms
d_{12}	Number of hydrogen atoms attached to the third level atoms
d_{13}	Number of π bonds that connect the second and the third level atoms
d_{14}	Number of the third level atoms incorporated in benzene cores

6211 8322 7421 6541 6641

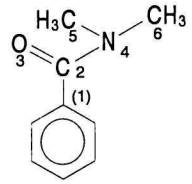

Fig. 5. Description of substituents by "20-input" vectors.

where the whole core is considered as one "atom". The first 4-tuple relates to the atom numbered by 2, the second 4-tuple relates to the atom numbered by 3, and so on.

The first entry of each 4-tuple equals to the proton number of the current atom (phenyl = 99), the second entry is a serial number of the current atom, the third entry is a serial number of the adjacent atom in the lower level, and the fourth entry is a multiplicity of bond between the current atom and the atom in lower level. Examples of these descriptors are displayed in Fig. 5.

From literature [14] 55 monosubstituted benzenes have been chosen and divided into training and testing sets so that the training set (46 compounds) is composed of typical representatives of functional groups whereas the functional groups that are similar to the previous ones are shifted to the testing set (9 compounds).

Definition of Output

Output of a neural network is in general a number from an open interval (0,1). Chemical shifts have a wider range, therefore their transformation into an interval (0,1) was necessary. Instead of (0,1) interval, in the presented approach an interval ⟨0.05,0.95⟩ was used for both linear and sigmoidal transformations. The linear transformation proved to be more successful, therefore only this transformation described by eqn (*1*) was consequently used

$$y = ax + b \qquad (1)$$

where the constants a and b are determined as follows: Since the values of chemical shifts range from 105 to 165, which has to be transformed into interval ⟨0.05,0.95⟩, we get $a = 0.015$ and $b = -1.525$.

Output neurons assigned to shifts of *ipso, ortho, meta,* and *para* carbon atoms were present either in one common neural network, or each position was treated separately in its own network. The number of hidden neurons has been optimized for both approaches. The lowest possible value of the objective function for both training and testing sets was used as a criterion for the above optimization. Ten different starting values of weight and threshold coefficients were tried for each tested number of hidden neurons. For further computations those values were used giving the lowest values of objective function for both sets. This computational procedure was used for both "14-input" and "20-input".

Table 2. Values of Objective Function (Training Set/Testing Set)

	Objective function
"14-input"	
4-output neural network 6 hidden neurons	$7.1 \times 10^{-3}/4.6 \times 10^{-3}$
ipso neural network 6 hidden neurons	$1.0 \times 10^{-5}/1.4 \times 10^{-2}$
ortho neural network 4 hidden neurons	$1.9 \times 10^{-3}/1.2 \times 10^{-3}$
meta neural network 6 hidden neurons	$6.1 \times 10^{-5}/9.2 \times 10^{-5}$
para neural network 5 hidden neurons	$3.4 \times 10^{-4}/1.3 \times 10^{-3}$
"20-input"	
ipso neural network 6 hidden neurons	$1.2 \times 10^{-2}/4.2 \times 10^{-2}$
ortho neural network 5 hidden neurons	$3.0 \times 10^{-2}/3.1 \times 10^{-2}$
meta neural network 6 hidden neurons	$3.8 \times 10^{-4}/1.7 \times 10^{-4}$
para neural network 5 hidden neurons	$2.8 \times 10^{-3}/3.5 \times 10^{-2}$

RESULTS

The value of objective function for both training and testing sets served as a criterion of success in prediction of chemical shifts. The definition of objective function was used as described by *Rumelhart* and *McClelland* [3]. If this value was low for the training set but high for the testing set, the neural network was likely "overtrained". Neural network learned exactly the training set, but could not "generalize" its knowledge on other cases. This happens mostly when the number of hidden neurons is too high. If the value of objective function was high for training set and low for testing set, the neural network was incorrectly trained and the good output for testing set is likely accidental.

Results of "14-Input"

Neural network with four output neurons gave the lowest values of objective function when used with 6 hidden neurons. Neural networks with one output neuron gave the lowest levels of objective function for both training and testing sets with the following numbers of hidden neurons:

1. *ipso* neural network with 6 hidden neurons,
2. *ortho* neural network with 4 hidden neurons,
3. *meta* neural network with 6 hidden neurons, and

4. *para* neural network with 5 hidden neurons.

The obtained results are listed in Table 2.

Results of "20-Input"

Neural network with four output neurons failed to be trained to give value of objective function lower than 7×10^{-3}, for number of hidden neurons from 3 to 14. This type of neural network was therefore skipped from further study.

Neural networks with one output neuron gave the lowest levels of objective function for both training and testing sets with the following numbers of hidden neurons:

1. *ipso* neural network with 6 hidden neurons,
2. *ortho* neural network with 5 hidden neurons,
3. *meta* neural network with 6 hidden neurons, and
4. *para* neural network with 5 hidden neurons.

Values of objective function are listed in Table 2.

Tables 3 and 4 list results of neural network with four output neurons and fourteen input neurons. Each odd row contains computed results whereas each even row contains experimental results taken from literature [14]. The training set is presented in Table 3 while the testing set is given in Table 4. Numbers in the tables are increments $\Delta\delta_c$, given by the equation

$$\delta_c = 128.5 + \Delta\delta_c \qquad (2)$$

Table 3. Example of Computational Results $\Delta\delta_c = \delta_c - 128.5$ of Training Set of Neural Network with Four Output Neurons for "14-Input". Odd Rows Contain Computed Results, Even Rows Contain Literature Results [14]

Substituent	*ipso*	*ortho*	*meta*	*para*
—H	0.2600	0.0467	− 0.0467	0.1400
—H	0.0000	0.0000	0.0000	0.0000
—CH_3	9.3400	0.9333	− 0.0133	− 2.3333
—CH_3	9.2000	0.7000	− 0.1000	− 3.1000
—CH_2CH_3	15.7933	− 0.8800	− 0.0467	− 2.9267
—CH_2CH_3	15.6000	− 0.5000	0.0000	− 2.7000
—$C(CH_3)_3$	21.2533	− 3.3400	− 0.0333	− 3.7533
—$C(CH_3)_3$	22.1000	− 3.4000	− 0.4000	− 3.1000
—$CH_2C(CH_3)_3$	11.3200	0.1933	− 0.4133	− 2.4667
—$CH_2C(CH_3)_3$	10.6000	1.5000	− 1.0000	− 3.1000
—$CH_2CH{=}CH_2$	15.4000	0.3133	− 0.1200	− 2.1467
—$CH_2CH{=}CH_2$	15.3000	0.0000	0.2000	− 2.4000
—CH_2COCH_3	4.9867	2.5133	− 0.3467	− 2.7667
—CH_2COCH_3	6.0000	1.0000	0.2000	− 1.6000
—CH_2NH_2	15.0467	− 0.3267	− 0.0467	− 1.2267
—CH_2NH_2	14.9000	1.4000	− 0.1000	− 1.9000
—CH_2NO_2	2.3467	1.8000	0.9667	0.4600
—CH_2NO_2	2.2000	2.2000	2.2000	1.2000
—CH_2OH	12.8133	− 0.7800	0.0133	0.1600
—CH_2OH	12.4000	− 1.2000	0.2000	− 1.1000
—CH_2SCH_3	9.7200	0.4133	− 0.2733	− 1.5733
—CH_2SCH_3	9.8000	0.4000	− 0.1000	− 1.6000
—CH_2SOCH_3	1.6333	1.5400	0.0400	0.6667
—CH_2SOCH_3	0.8000	1.5000	0.4000	− 0.2000
—$CH_2SO_2CH_3$	− 0.1667	1.7667	0.6733	1.3000
—$CH_2SO_2CH_3$	− 0.1000	2.1000	0.6000	0.6000
—CF_3	2.6133	− 1.0133	0.1867	4.4933
—CF_3	2.5000	− 3.2000	0.3000	3.3000
—CH_2Cl	9.6467	− 0.1200	0.0467	0.4733
—CH_2Cl	9.3000	0.3000	0.2000	0.0000
—CH_2Br	9.4667	0.7533	0.2267	− 0.4267
—CH_2Br	9.5000	0.7000	0.3000	0.2000
—CH_2I	10.5267	0.4267	0.9733	− 0.3067
—CH_2I	10.5000	0.0000	0.0000	− 0.9000
—$C(CH_3){=}CH_2$	13.0133	− 3.0467	0.1600	− 0.7200
—$C(CH_3){=}CH_2$	12.6000	− 3.1000	− 0.4000	− 1.2000
—$CONH_2$	3.9933	− 0.4667	0.1400	4.0067
—$CONH_2$	5.0000	− 1.2000	0.1000	3.4000
—COOH	2.9467	0.1467	0.1067	4.4867
—COOH	2.1000	1.6000	− 0.1000	5.2000
—$COOCH_3$	2.3333	0.6867	0.0600	4.5600
—$COOCH_3$	2.0000	1.2000	− 0.1000	4.3000
—COF	2.4067	1.2133	0.0400	4.8133
—COF	4.3000	1.6000	− 0.7000	5.3000

Table 3 (Continued)

Substituent	*ipso*	*ortho*	*meta*	*para*
—COCl	5.4400	2.5800	− 0.0267	5.4600
—COCl	4.7000	2.7000	0.3000	6.6000
—NH_2	17.8533	− 15.0933	0.9533	− 10.4067
—NH_2	18.2000	− 13.4000	0.8000	− 10.0000
—$NHCH_3$	21.0333	− 15.2667	0.7133	− 11.0533
—$NHCH_3$	21.4000	− 16.2000	0.8000	− 11.6000
—$NHCH_2CH_3$	20.3467	− 15.7533	0.4467	− 10.8733
—$NHCH_2CH_3$	20.0000	− 15.7000	0.7000	− 11.4000
—$NHCOCH_3$	9.3533	− 8.0933	0.4867	− 4.7200
—$NHCOCH_3$	9.7000	− 8.1000	0.2000	− 4.4000
—$NHNH_2$	22.2467	− 15.9333	0.7067	− 10.1933
—$NHNH_2$	22.8000	− 16.5000	0.5000	− 9.6000
—$NH(CH_3)NO$	23.6933	− 9.4133	0.8733	− 1.4867
—$NH(CH_3)NO$	23.7000	− 9.5000	0.8000	− 1.4000
—NCO	5.0533	− 3.8067	1.0933	− 2.6800
—NCO	5.1000	− 3.7000	1.1000	− 2.8000
—NCS	2.9867	− 2.7067	1.3800	− 1.0133
—NCS	3.0000	− 2.7000	1.3000	− 1.0000
—OH	26.6400	− 12.6800	1.2200	− 7.8667
—OH	26.9000	− 12.8000	1.4000	− 7.4000
—OCH_3	31.8400	− 14.6267	1.0400	− 7.8800
—OCH_3	31.4000	− 14.4000	1.0000	− 7.7000
—$OCOCH_3$	22.6733	− 7.0067	0.6933	− 3.0400
—$OCOCH_3$	22.4000	− 7.1000	0.4000	− 3.2000
—SH	2.1333	0.7000	0.6200	− 3.3267
—SH	2.1000	0.7000	0.3000	− 3.2000
—SCH_3	10.0933	− 1.7333	0.3533	− 3.6667
—SCH_3	10.0000	− 1.9000	0.2000	− 3.6000
—$SOCH_3$	17.6600	− 6.1467	0.4733	2.8200
—$SOCH_3$	17.6000	− 5.9000	1.1000	2.4000
—SO_2CH_3	11.6867	− 2.5000	0.2267	3.4267
—SO_2CH_3	12.3000	− 1.4000	0.8000	5.1000
—NO_2	20.3733	− 3.9600	0.4067	6.5600
—NO_2	19.9000	− 4.9000	0.9000	6.1000
—CH_2CH_2Ph	13.2133	0.0733	− 0.0400	− 2.5533
—CH_2CH_2Ph	13.2000	0.0000	− 0.2000	− 2.6000
—$CH(Ph)_2$	15.3000	1.6467	− 0.1533	− 3.2467
—$CH(Ph)_2$	15.3000	0.9000	− 0.3000	− 2.3000
—Ph	13.2400	− 1.1733	0.2933	− 1.0533
—Ph	13.1000	− 1.1000	0.4000	− 1.1000
—COPh	9.0267	1.7667	− 0.0733	2.1267
—COPh	9.3000	1.6000	− 0.3000	3.7000
—CSPh	18.5000	1.1800	0.1000	2.8733
—CSPh	18.7000	1.0000	− 0.6000	2.4000
—NHPh	15.6200	− 10.3667	0.5000	− 10.2000
—NHPh	14.7000	− 10.7000	0.9000	− 10.5000
—OPh	27.2000	11.0867	− 0.6533	− 6.5867
—OPh	27.6000	11.2000	− 0.3000	− 6.9000

where δ_c is the value of chemical shift of the carbon atom in the benzene ring. Values computed using one-output neural networks are not presented.

DISCUSSION

Table 2 presents values of the objective function for both training and testing sets for single neural networks. Better results were obtained using "14-input". This is probably caused by higher "density" of coded input information, even though "14-input" is degenerated when comparing with "20-input". "14-Input" is therefore a better way of characterizing structure for neural networks, "20-approach" is probably too complex to generalize and extract information.

The neural network with four output neurons gave obviously worse results for objective function than the separate one-output-neuron neural networks, when the "14-input" was used.

Ordering the one-output-neuron neural networks according to increasing values of objective function gives the following sequences. The ordering for "14-input" for the training set is *ipso* < *meta* < *para*

Table 4. Example of Computational Results $\Delta\delta_c = \delta_c - 128.5$ of Testing Set of Neural Network with Four Output Neurons for "14-Input". Odd Rows Contain Computed Results, Even Rows Contain Literature Results [14]

Substituent	*ipso*	*ortho*	*meta*	*para*
$-SO_2Cl$	17.6933	− 2.3800	0.3200	7.2600
$-SO_2Cl$	15.6000	− 1.7000	1.2000	6.8000
$-N(CH_3)_2$	23.9400	− 15.7733	0.6200	− 11.2067
$-N(CH_3)_2$	22.5000	− 15.4000	0.9000	− 11.5000
$-CH_2Ph$	13.6067	1.8333	− 0.1267	− 3.0467
$-CH_2Ph$	12.8000	0.5000	0.0000	− 2.3000
$-CH(CH_3)_2$	19.3333	− 2.2200	− 0.0467	− 3.3667
$-CH(CH_3)_2$	20.1000	− 2.0000	0.0000	− 2.5000
$-CH_2N(CH_3)_2$	11.4800	0.2467	− 0.4200	− 2.5733
$-CH_2N(CH_3)_2$	11.1000	0.8000	− 0.2000	− 1.5000
$-CH_2OCH_3$	11.7533	0.1600	− 0.3667	− 2.3133
$-CH_2OCH_3$	11.0000	0.5000	− 0.4000	− 0.5000
$-COCH_3$	5.8867	− 0.7667	0.1400	3.2067
$-COCH_3$	8.9000	0.1000	− 0.1000	4.4000
$-N(CH_2CH_3)_2$	21.0867	−15.8733	0.2533	− 11.4333
$-N(CH_2CH_3)_2$	19.3000	−16.5000	0.6000	− 13.0000
$-CH{=}CH_2$	9.5133	− 3.0000	0.3333	1.3400
$-CH{=}CH_2$	8.9000	− 2.3000	− 0.1000	− 0.8000

< *ortho* and for the testing set *meta* < *ortho* < *para* < *ipso*. Ordering for "20-input" for the training set is *meta* < *para* < *ipso* < *ortho* and for the testing set *meta* < *ortho* < *para* < *ipso*.

These sequences show that the best results were achieved by *meta* neural networks. This is probably caused by the fact that most of the evaluated compounds had a value of chemical shift similar to the value of benzene ring, which is 128.5. The results of computations are in general very good for "14-input" and acceptable for "20-input". In comparison with previous results of the present authors [4, 5], which were obtained for similar training and testing sets, simple neural networks proved to give better results according to the objective function.

In our work we have tried to compute chemical shifts of carbon atoms in ^{13}C NMR spectrum for monosubstituted derivatives of benzene. We have tested two types of neural network, with four outputs for all four positions of shift and four one-output neural networks for each type of shift. The second approach was better. The best results were achieved for *meta* carbon of benzene ring. We have chosen two types of input, "14-input" and "20-input". The first one gave better results. The results support a hope that even simple neural networks can be efficiently used for predicting chemical shifts, when appropriately tuned and provided by information about molecular structure.

REFERENCES

1. Exner, O. and Buděšínský, M., *Collect. Czech. Chem. Commun. 57*, 497 (1992).
2. Kvasnička, V., Sklenák, Š., and Pospíchal, J., *J. Am. Chem. Soc. 115,* 1495 (1993).
3. Rumelhart, D. E. and McClelland, J. L., *Parallel Distributed Processes,* Vol. *1*, *2*. MIT Press, Cambridge, MA, 1986.
4. Kvasnička, V., Sklenák, Š., and Pospíchal, J., *Theochem, J. Mol. Struct. 277*, 87 (1992).
5. Kvasnička, V., Sklenák, Š., and Pospíchal, J., *J. Chem. Inf. Comput. Sci. 32,* 742 (1992).
6. Kvasnička, V., *J. Math. Chem. 6*, 63 (1991).
7. Kvasnička, V., *Chem. Papers 44*, 775 (1990).
8. Anderson, J. A. and Rosenfeld, Z. (Editors), *Neurocomputing: Foundation of the Research.* MIT Press, Cambridge, MA, 1989.
9. Simpson, P. K., *Artificial Neural Systems.* Pergamon, New York, 1990.
10. Guyon, I., *Physics Reports 207*, 215 (1991).
11. Zupan, J. and Gasteiger, J., *Anal. Chim. Acta 248*, 1 (1991).
12. Lacy, M. E., *Tetrahedron Comput. Methodol. 3*, 119 (1990).
13. Elrod, D. W., Maggiora, G. M., and Trenary, R. G., *J. Chem. Inf. Comput. Sci. 30*, 477 (1990).
14. Kalinowski, H.-O., Berger, S., and Braun, S., *^{13}C NMR Spektroskopie.* Thieme Verlag, Stuttgart, 1984.

Translated by J. Pospíchal