Automated Structure Elucidation of Organic Molecules from ^{13}C NMR Spectra Using Genetic Algorithms and Neural Networks

Jens Meiler*,[†] and Martin Will[‡]

Institute of Organic Chemistry, Marie-Curie-Strasse 11, Universität Frankfurt, D - 60439 Frankfurt, Germany, and BASF AG Ludwigshafen, Germany

The automated structure elucidation of organic molecules from experimentally obtained properties is extended by an entirely new approach. A genetic algorithm is implemented that uses molecular constitution structures as individuals. With this approach, the structure of organic molecules can be optimized to meet experimental criteria, if in addition a fast and accurate method for the prediction of the used physical or chemical features is available. This is demonstrated using ^{13}C NMR spectrum as readily obtainable information. ^{13}C NMR chemical shift, intensity, and multiplicity information is available from ^{13}C NMR DEPT spectra. By means of artificial neural networks a fast and accurate method for calculating the ^{13}C NMR spectrum of the generated structures exists. The approach is limited by the size of the constitutional space that has to be searched and by the accuracy of the shift prediction for the unknown substance. The method is implemented and tested successfully for organic molecules with up to 20 non-hydrogen atoms.

INTRODUCTION

Thousands of substances are synthesized every day, and their structures need to be elucidated or validated. Consequently the daily routine work of structure elucidation of molecules produced by organic synthesis, especially by combinatorial synthesis, is still one of the most important demands on chemists. Tools that assist sprectroscopists to elucidate the structure of organic molecules, or are even able to predict the structure of unknowns automatically, are therefore of general importance.

To automatically determine from any source of experimental data the structure of an organic molecule, two steps of data processing are performed: A structure generator creates proposals for the unknown molecular structure. A filter validates these proposals usually by comparing easily derivable properties of the generated molecules with the corresponding experimental values and minimizing the deviation. In theory it is possible to determine the chemical structure of any compound from just one experimental quantity, provided that every compound has its own characteristic experimental quantity and this value is obtainable without any experimental uncertainty. Even if these prerequisites are obtained, two additional requirements are necessary: (1) the experimental value can be calculated from the molecular structure with infinite precision and (2) infinite computational power is available.

Under these conditions all possible structures can be created, the known experimental value can be computed, and a comparison with the experimentally observed value yields an unambiguous answer as to which of the hypothetically proposed structures is the unknown.

However, in practice these requirements are impossible. Even if the experimental parameter differs for every molecule, it can only be measured within experimental uncer-

tainty. If the number of possible structures is large enough and if the error of the property calculation is taken into consideration, "false" positives will be found with smaller deviation of the calculated and the experimental value than the true solution has. Therefore, it is only possible to obtain a hit list of structural proposals ranked according to their similarity to the experimental data. In this hit list the correct solution is provided together with false positives. The introduction of additional experimental data helps to overcome this limitation. A more challenging problem is the infinite number of possible structures. It is impossible to compute an infinite number of proposed structures in a finite period of time. Therefore, the key point is the development of "intelligent" structure generators that include already available experimental data during the generation of structures and create therefore only a finite number of probable structures, each having only a small deviation from the experimentally obtained data. The earlier the comparison of the experimental value with the values calculated for the generated structures is performed and the result is incorporated into the further structure generation process, the more exact the structural space can be defined that has to be searched. This decreases the required computation time.

A frequently used first restriction is the molecular formula. This boundary condition ensures a finite number of possible structures, which then allows a computation of the entire structural space in a finite period of time. The generation of a structural space can be separated into two steps: The generation of all possible constitutions (a constitution formula contains all connectivity but no stereochemical information) and the subsequent generation of all possible stereoisomers for every constitution formula.

Molgen is a powerful structure generator that performs both those steps and creates all possible structures having a given molecular formula.[1,2] A subsequent calculation of a predictable parameter (for example the ^{13}C NMR spectrum) for all these structures and a comparison with the experiment

* Correspondence author phone: 206-543-7228; fax: 206-685-1792; e-mail: jens@jens-meiter.de.
† Universität Frankfurt.
‡ BASF AG Ludwigshafen.

would provide a straightforward approach for automated structure elucidation. However, even for a small number of atoms the computation time increases to an impractically large size.

The CoCon approach by Lindel, Köck, and Junker uses connectivity information from two-dimensional NMR spectroscopy in addition to the molecular formula and so becomes usable for much larger molecules.[3,4] CoCon produces all constitutions that fulfill the introduced connectivity information. However, since CoCon uses only connectivity information, it does not differentiate stereoisomers that may occur in the generated constitutions. Thus Molgen generates a set of all possible constitutions, whereas CoCon reduces this number. In some cases only one constitution fulfills all the connectivity information. However, often CoCon presents a large set of possible constitutions, more than can be validated by hand.

The collection of connectivity information from two-dimensional NMR spectra is time-consuming and difficult to automate. The one-dimensional ^{13}C NMR chemical shift is not only much easier to obtain but also contains diverse constitutional and stereochemical information. Further, artificial neural networks offer a fast and accurate tool for calculating the ^{13}C NMR spectrum of organic compounds.[5] Recently we demonstrated that a combination of CoCon with a subsequent comparison of the experimental and calculated ^{13}C NMR chemical shifts is an effective and efficient possibility to decrease the number of possible constitutions presented by CoCon alone.[6]

A very powerful approach named SpecSolv uses the ^{13}C NMR spectrum in combination with the Specinfo database.[7,8] The molecular constitution formula can be elucidated only from their ^{13}C NMR chemical shifts by a search for similar substructure spectra in the database and reassembling the substructure fragments found.

In contrast to all these approaches, the implementation introduced here uses an entirely new procedure of intelligent structure generation—a genetic algorithm. This allows one to circumvent certain limits and disadvantages of previous approaches: (1) The time-consuming determination of connectivity information for CoCon by two-dimensional NMR spectroscopy is replaced by the much easier and rapidly obtainable chemical shift value. (2) The genetic algorithm is able to use the generated structures immediately as a basis for further optimization process. While Molgen or CoCon generate structural spaces of predefined size and content, the structural space generated by this genetic algorithm is dynamically determined. (3) In contrast to SpecSolv the generation of a structural database by reference to thousands of experimental spectra is no longer necessary and does not limit the searched structural space. The entire space, including all possible structures, can be investigated unaffected by either preferred and neglected regions in a reference database. (4) Additional structure information (including connectivity information from two-dimensional NMR spectroscopy) can be implemented easily as boundary conditions. (5) The generated structures can be ranked by their chemical shift deviation to the target spectrum and not only by a binary quality factor (e.g., in line or not in line with the connectivity information − CoCon).

Since not all structures of the structural space are generated with such an implementation, there can be no guarantee that the correct solution structure is actually created. We will describe one implementation of such a genetic algorithm and discuss its advantages and its limitations. First we will give a brief summary of existing neural networks and genetic algorithms used in context with NMR spectroscopy. The usefulness of the ^{13}C chemical shift values for structure elucidation is reprised.

Methods of artificial intelligence are widespread and accepted methods for data analysis in chemistry and biology.[9] Neural networks have been suggested for more than 10 years as solutions for a wide range of optimization problems. They are intensively used for the prediction of NMR chemical shifts of organic substances, in particular for carbon atoms.[5,10−19] Genetic algorithms are of special interest due to their ability to solve complex optimization problems on complex hyperdimensional surfaces with many local minima. In combination with NMR spectroscopy they are implemented for the assignment and analysis of spectra.[20−25]

Due to its combination of spectral simplicity and the large content of complex chemical information, the ^{13}C NMR chemical shift value is well suited for the storage in databases[7,26] and application in intensive numerical analysis. A triple of three values, chemical shift, intensity, and multiplicity, contains detailed information about the chemical environment of the most common atom in most organic molecules, carbon. Many approaches for the prediction of the chemical shift value and for its use in further analysis are suggested besides these already listed applications involving neural networks. Only a few are mentioned here.[27−30] Moreover the chemical shift plays an important role in the daily routine work of structure elucidation and validation in organic chemistry. Consequently, a fully automated structure elucidation program based exclusively on ^{13}C NMR data is a dream of NMR spectroscopists.

Theory. A genetic algorithm is a method of producing new individual examples from combinations of previous individuals. The algorithm has the same logical structure as inheritance in biological systems and much of the terminology is similar by analogy. So, for example, a genetic algorithm describes the previous examples as "parents" and the combinations produced as "children" or "offspring" or individuals belonging to the next generation. The identity of a particular individual is determined randomly but by a process which is probability-weighted. The probability that an individual will be produced and participate as a parent in a succeeding generation must be defined by some standard. For an optimization process, the suitability of an offspring can be assessed using some "fitness" function. This is a direct analogy to Darwin's evolutionary rules of selection, survival of the fittest. One algorithmic process that mimics biological evolution is described as generating mutation. How these relationships work out for a nonbiological sequence of events, a synthetic calculation occurring entirely within a computer, will now be described in more detail.

The implementation of every genetic algorithm invokes three data processing steps: selection, recombination (cross over) and mutation. For optimizing molecular structures, a genetic code needs to be defined that describes them. Figure 1a) visualizes how a vector of bond states between all (atom−atom) pairs can be defined from the connectivity matrix of a molecule. This vector provides a suitable genetic code for the constitution of an organic molecule. Stereo-
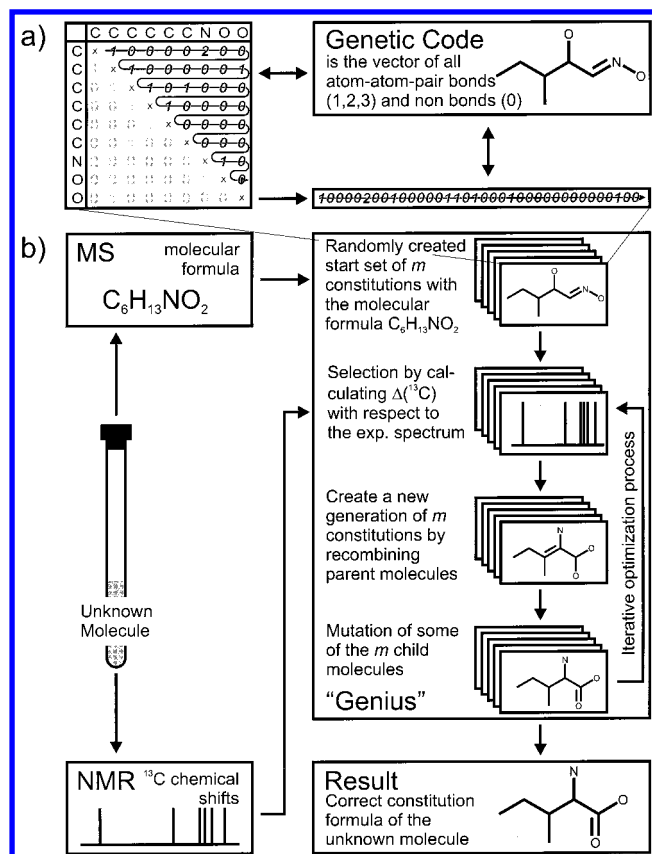
Figure 1. The connectivity matrix of a randomly created constitution with the molecular formula $C_6H_{13}NO_2$ is given. From this connectivity matrix the genetic code is obtained by rearranging the a triangular half matrix into a vector (a). This vector contains now the bond state between all (atom − atom) pairs of the molecule. Part (b) of the figure visualizes the general implemented procedure. The molecular formula (obtained e.g. from mass spectroscopy) is used to generate a random set of m constitutions that fulfill that molecular formula. This set is now valuated by calculating the ^{13}C NMR spectrum and comparing it with the experimental data. The lower the $\Delta(^{13}C)$ value the higher is the probability that a molecule is considered for recombination. A new generation is formed by recombining two parent molecules m times. Optionally some of these m new constitutions can undergo a mutation or l of them can be replaced by the l fittest parent constitutions. This cycle of selection, recombination, and mutation is repeated until $\Delta(^{13}C)$ is minimized.

chemistry is not considered in this implementation, since it was not possible to distinguish between stereoisomers using the selection procedure as discussed below. A set of randomly generated constitutions is taken as the starting parent population. The members of this population satisfy only the molecular formula, which has to be known in advance. Iteratively, the population undergoes the processes of selection, recombination, and mutation to form a child generation which can then be used as the next parent generation (Figure 1b)).

Selection. While recombination and mutation can be implemented independently from boundary conditions, the selection process is affected by using the ^{13}C NMR spectrum as a "fitness function". As mentioned earlier, a fast and exact calculation method for this fitness function is necessary to implement a genetic algorithm. The ^{13}C NMR chemical shift can be determined most efficiently for this purpose using artificial neural networks. Once trained, they are fast and exact. The implementation of neural networks used in the

following approach is described in the literature[5] and is therefore only briefly summarized below.

The spectra can be predicted for all organic substances that contain exclusively C, H, N, O, P, S or the four halogens (F, Cl, Br, I). To obtain the spectrum of a molecule the chemical shift of every carbon atom is successively calculated in an individual run. The environment around the carbon atom of interest is subdivided into six spheres. All atoms inside these spheres are again classified as belonging to one of 28 previously defined atom types. The types consider the atomic number, hybridization, and number of bound hydrogen atoms. The 28 dimensional vector containing the number of atoms of every atom type in a particular sphere is accomplished by two sum parameters holding the number of hydrogen atoms in the sphere (hydrogen is not considered as one of the 28 atom types) and the number of ring closures. This vector contains now 30 numbers and is collected for atoms belonging to one out of six spheres separately. Moreover the information is collected a second time, but only for atoms that belong to a conjugated π-electronic system together with the carbon atom of interest to consider the special influence of such systems on the chemical shift value. Therefore a vector of 360 ($=30\cdot6\cdot2$) numbers describes the carbon atom environment and serves as input for the neural networks. Nine of these 28 atom types describe carbon atoms. For each of the nine types an individual neural network is trained using the overall number of about 1 300 000 chemical shifts out of the Specinfo database.[7] The number of hidden neurons in the single hidden layer varies from 5 to 40, depending on the number of training examples for the carbon atom type. One output neuron calculates the chemical shift. The average deviation of this method is as low as 1.6 ppm relative to an independent database of about 50 000 chemical shifts. Essential advantages of this method are the fast, exact, and database independent shift prediction for all organic molecules. Since spherical description of the carbon atom environment used in the method does not contain stereochemical information, the predicted chemical shift spectrum is the same for all stereoisomers for any particular constitution formula. Consequently the genetic algorithm implemented here can only optimize the molecular constitution relative to the NMR spectrum. Therefore, the genetic code needs also only to define the constitution. If stereochemistry had been considered in estimating the chemical shift, the introduction of stereochemical descriptors in the genetic code would allow the definition of stereochemistry in the structures elucidated.

The chemical shifts of all carbon atoms of a constitution are calculated by the artificial neural networks and sorted by their size. The "fitness" of every single carbon atom i is now the absolute deviation between its experimental and the corresponding calculated chemical shift value: $|\delta^i_{calc}(^{13}C) - \delta^i_{exp}(^{13}C)|$. The fitness of the whole molecular constitution formula is given by the root-mean-square deviation (RMSD) over all N carbon atoms:

$$\Delta(^{13}C) \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\delta^i_{calc}(^{13}C) - \delta_{exp}{}^i(^{13}C))^2}$$

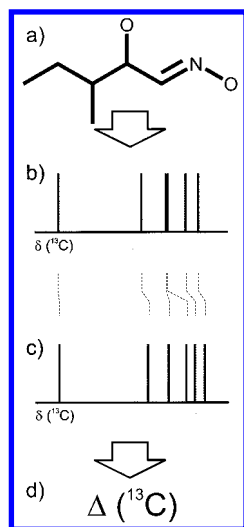(Figure 2). The multiplicity of a signal can be easily incorporated, if experimentally obtained: The absolute

Figure 2. The ^{13}C NMR spectrum for a newly generated constitution (a) is calculated by artificial neural networks (b). By comparing this spectrum with the experimentally obtained spectrum (c) the $\Delta(^{13}C)$ value can be computed as the RMSD of all single deviations (d). The $\Delta(^{13}C)$ is taken as the fitness function in the selection process of the genetic algorithm.

deviation between the experimental and the calculated multiplicity for every carbon atom i $|M^i_{calc} - M^i_{exp}|$ is multiplied by a factor ("multiplicity deviation factor" = MDF (in *ppm*)) and added to the absolute deviation of the chemical shift. The fitness $\Delta(^{13}C)$ becomes now

$$\Delta(^{13}C) \equiv \sqrt{\frac{1}{N}\sum_{i=1}^{N}(|\delta^i_{calc}(^{13}C) - \delta^i_{exp}(^{13}C)| + \mathrm{MDF}\cdot|M^i_{calc} - M^i_{exp}|)^2}$$

The lower the $\Delta(^{13}C)$ value of a structure the higher is its fitness and the higher is its probability of participation in the recombination step of the genetic algorithm.

Recombination. Two molecules from the parent generation are selected to form the child molecule. The smaller the $\Delta(^{13}C)$ value of a molecule the higher is its probability to be considered as parent. This probability for a single molecule j out of a population of m constitutions is given by $p_j = [\Delta_j(^{13}C)]^{-1}/\sum_{i=1}^{m}[\Delta_i(^{13}C)]^{-1}$. After this selection, all possible (atom − atom) pairs in both parents are taken, and the bond type between them is analyzed (0 = non bounded, 1 = single bond, 2 = double bond, or 3 = triple bond). Randomly one out of the two possibilities for every (atom − atom) pair is taken for the newly generated child structure (Figure 3).

Since hydrogen atoms are not explicitly taken into consideration but are added to the free valences afterward, the molecular formula needs to be checked after a new child constitution is generated. If the number of potential hydrogen atoms in the generated constitution is not the same as defined in the target molecular formula, bonds must be added or deleted until this deviation is corrected to zero. For this purpose the same function is used as for mutation (see below). Moreover it is necessary to ensure that a single molecule is formed and not a set of two or three fragments with the correct overall molecular formula but not connected to each other. After both boundary conditions are fulfilled,
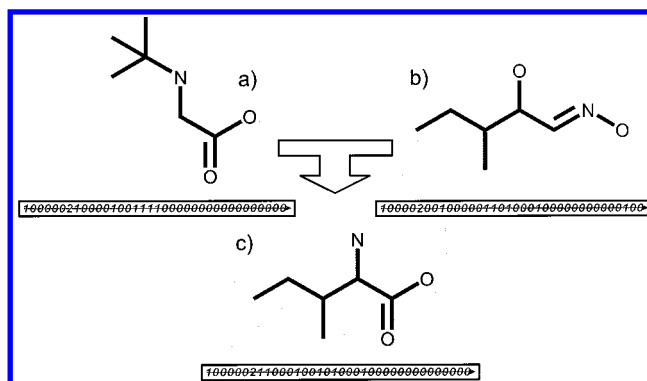

Figure 3. Recombination of two molecules out of the parent generation (a, b) to form a new molecule (c) that becomes a part of the population in the next generation. The gray shaded areas of the parent molecules are linked to form the new molecule. Below each constitution formula the corresponding genetic code of the molecule is given. The vector representing the newly formed child constitution contains at every position exactly one of the both possible values obtained at the corresponding positions in the parent molecule vectors.
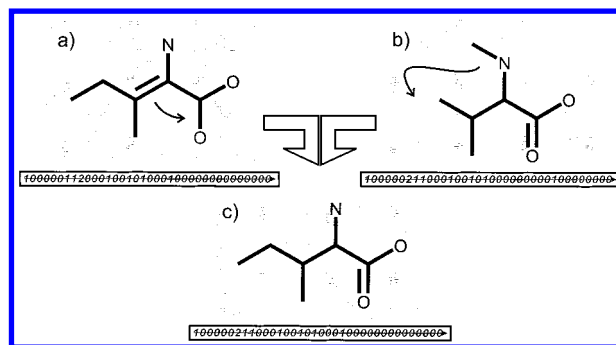

Figure 4. The process of mutation is illustrated on two example constitution (a) and (b). The gray shaded area is again conserved in the mutated constitution (c), while the white marked bond is changed. Below each constitution formula the corresponding genetic code of the molecule is given. The vector changes consequently exactly at two positions.

the new molecule is accepted as a member of the newly formed population.

Mutation. A "mutation" is implemented by modifying simply bonds. Two atoms are randomly selected and a bond is deleted (or the bond type is decreased by one) while, for two other randomly selected atoms, a bond is inserted (or the bond type is increased by one, Figure 4). A deletion is always combined with an insertion, so that the total number of hydrogen atoms remains constant. Also this process has to be controlled so that only one molecule and not a set of fragments is created.

Figure 5 illustrates the formation of a single generation in the form of a simplified flowchart diagram. By repeating the procedure of subsequent recombination and mutation, m molecules for the child generation are created out of the m parent molecules. Optionally the l fittest molecules of the parent generation replace the l worst molecules of the child generation to ensure that the fittest constitutions are not lost. To enable the optimal use of multiprocessor computers n populations can be calculated in a parallel manner without interactions. This procedure takes advantage of the well-known fact that in a genetic algorithm a set of small independent populations converges faster than one large population.
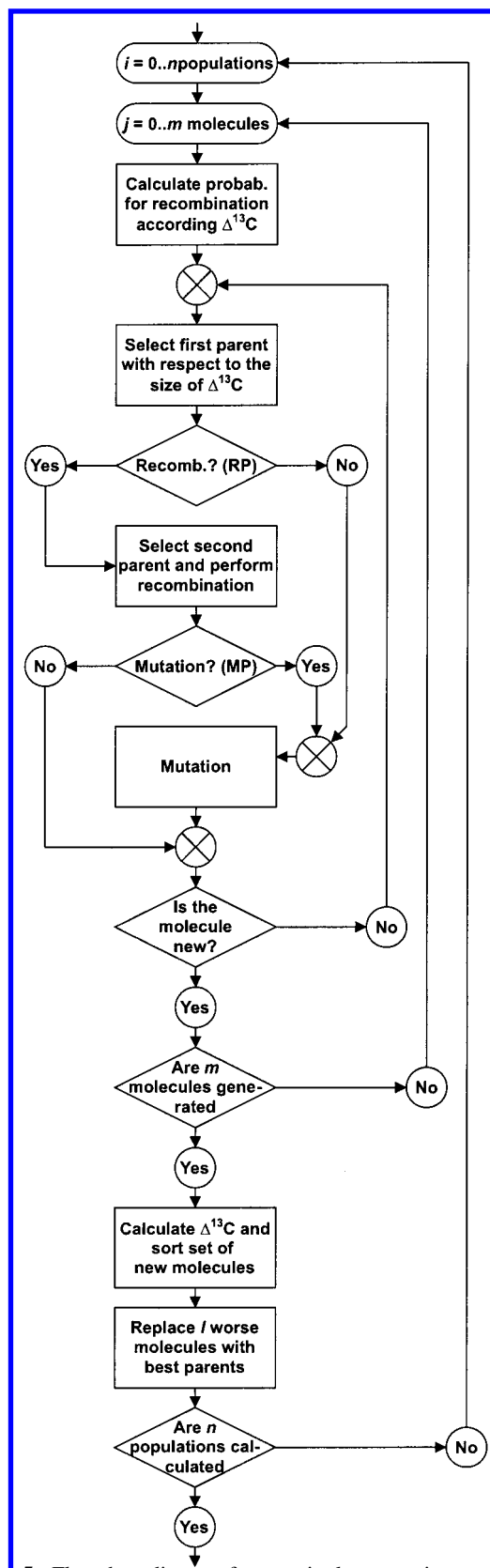
Figure 5. Flowchart diagram for one single generation performed during the genetic algorithm. The formation of n child populations containing m constitutions out of n parent populations is illustrated. The selection is performed by the calculation of the $\Delta(^{13}C)$ values. Recombination and mutation are performed according to the probabilities set with the RP and MP value. To ensure that a new constitutions is calculated one of both processes, recombination or mutation, have to take place. Finally for every generated population the l constitution with largest $\Delta(^{13}C)$ value are replaced by the l fittest molecules out of the parent set of structures.

The recombination probability (RP) and the mutation probability (MP) are parameters systematically varied during the iterative calculation process. RP is the probability that a child is generated by combining two parents (recombination) and not by copying a molecule already in the parent population (no recombination). MP defines the probability that a child generated undergoes a mutation (compare Figure 5). It is well-known that a high mutation rate at the beginning of a genetic algorithm ensures a fast convergence but later high mutation rates are rather counter-productive and simple recombination achieves a better fitting. This fact is also comparable to the evolution of life on the earth: (1) High-intensity UV irradiation caused high mutation rates in the beginning of evolution of life but both UV levels and mutation rates are lower now. (2) Creatures at a low level of evolution frequently reproduce without recombination, whereas creatures on a high level of evolution exclusively reproduce by recombination.

In keeping with the biological analogue, RP and MP are changed during the genetic algorithm. In principle, they could be independently defined for every evolutionary step of the optimization procedure. However, it is sufficient to predefine the RP and MP for certain evolutionary steps and change the values linearly between these points to approach the defined values.

The incorporation of additional information is possible by defining a good and a bad list of fragments that either need to be part of the molecule or are forbidden to use. In the first case, the fragments are incorporated during the initial creation of random molecules and not changed during the further optimization process. In the second case, generated structures that contain forbidden fragments are excluded and not used in the child generation. To avoid a reduction of the genetic pool of a population it is excluded that identical individuals that might be formed during the algorithm are considered for one population more than a single time.

RESULTS AND DISCUSSION

Three experiments are performed to evaluate a genetic algorithm implemented as discussed above: In the first experiment the parameters are optimized, and both the structural space and the generated populations are analyzed for a relatively small molecule. In a second experiment the previously optimized parameters are used to perform a fully automated structure elucidation for a small database consisting of molecules with 9−16 non-hydrogen atoms. In a third class of calculations, the limitations of this method are examined by investigating larger molecules with up to 20 non-hydrogen atoms. In this case an individually optimized setup and the use of additional boundary conditions become necessary.

For reasons of computation time the introduced parameters need to be optimized for a relatively simple example. For the same reason not all possible interactions between the parameters can be analyzed in detail. It is further assumed that the optimized values for these parameters can be later scaled for larger molecules. Moreover the investigation of a small molecule allows a hand analysis of the generated populations resulting in a deeper insight into the operations performed during the optimization. Isoleucine ($C_6H_{13}NO_2$) is chosen because it contains heteroatoms and a double bond. Since it has only nine non-hydrogen atoms and only one

double bond the number of possible constitutions is comparably low (23 946). Therefore the algorithm finds the correct solution in a reasonably short time period. This allows the optimization of parameters and an intensive analysis of the algorithm itself. The total deviation between the experimentally obtained and the neural network calculated chemical shifts is $\Delta(^{13}C) = 1.12$ ppm for isoleucin.

The parameters that need to be optimized are as follows: the size of the population, m; the number of fittest individuals conserved for the next generation, l; the number of parallel calculated populations, n; the multiplicity deviation factor, MDF; the recombination probability, RP; and the mutation probability, MP. The product of m and n defines the size of the genetic pool since it defines the overall number of individuals, whereas l defines the degree of conservative character. It is responsible for the fraction of replaced individuals in each generation. Multiplicity deviation factor MDF weights the influence of the deviation in the chemical shift values with respect to a deviation in the obtained multiplicity. The recombination probability RP and the mutation probability MP select the pathway for generating a new individual and are therefore strongly interacting parameters. One out of the two operations (mutation or recombination) has to be performed in order to generate a individual different from its parent(s) so that to ensure this mutation is forced if no recombination was carried out. RP and MP define which fraction of the newly generated constitutions is obtained by recombination or mutation only and which fraction is obtained by a subsequent application of both operations (Figure 5). Figure 6 summarizes the results of the optimization of all six parameters. Since m and n as well as RP and MP are not independent with respect to each other, experiments were applied to investigate those dependencies.

In a first experiment the size of the population is chosen to be $m = 8, 16, 32, 64, 128$. All other parameters are set to be constant with $l = 0.25 \cdot m$, $n = 1$, MDF = 1 ppm. The mutation probability is 100% during the first four steps and decreases linearly to become 50% between the fifth and the eighth generation. The recombination probability is set to be 0% during the first four generations and increases to become 100% after the eighth generation. For simplicity such a program of RP and MP values will be given in the following notation (MP: $^{0}1.0^{4} => ^{8}0.5^{\infty}$ | RP: $^{0}0.0^{4} => ^{8}1.0^{\infty}$) from now on. To obtain realistic results and avoid the influence of the random start population the average $\Delta(^{13}C)$ value of the best individual in 16 independent test runs with varying randomly generated start populations is computed for all experiments. All runs are stopped after the 16th generation. As visualized in Figure 6a the $\Delta(^{13}C)$ is generally smaller if larger populations are used. Since more molecules are generated, the probability of creating a molecule with a smaller $\Delta(^{13}C)$ increases. However, the generation of more molecules necessarily requires greater computation time. Obviously, a direct comparison of experiments with variable population sizes is not fair.

A realistic picture is given in Figure 6b. In these experiments the number of calculated generations is increased by a factor of 2, 4, 8, and 16 as the number of individuals is decreased from 128 to 64, 32, 16, and 8, respectively. Similarly, the fix points for the MP and RP values are adjusted by these factors. The result of this systematic adjustment is that the overall calculation time as well as the

number of generated structures are the same for all populations and a direct comparison becomes possible. The segment between the second and the ninth generation for the population with 128 individuals is plotted in comparison with the corresponding parts of test runs with a smaller number of individuals. As shown in Figure 6b an optimum size of the population is achieved between 32 and 64 individuals. Although the overall differences in results between the setups are small, the setup with 32 individuals provides the fastest decrease of $\Delta(^{13}C)$ in the first period of the algorithm until the fourth generation. An increased number of subsequent mutations (as mentioned recombination does not take place in this period) has advantages compared to a further increase of the number of parallel calculated individuals. However, if the number of randomly parallel generated structures is too small (below 32 in this case), the starting points for the optimization are badly sampled and the optimization velocity suffers. After the fourth generation the setup using 64 individuals seems to become slightly favored. This is due to the fact that here recombination becomes active, and therefore the number of individuals in a generation plays an increasing role. It defines the size of the "genetic pool" which is incorporated into the recombination process.

In the next experiment (Figure 6c) the number of conserved individuals l is optimized for a setup using $m = 32$ individuals. l is set to be 1, 8, 16, 24, and 31. The overall influence is small. However, an optimum is obtained for $l = 0.25 \cdot m$. A remarkable worse convergence is obtained in the case of 31 conserved individuals. This behavior is plausible in this case due to the small number of changes in the population with each new generation. Therefore the constitutional space searched by the genetic algorithm is very small.

The number of populations calculated parallel, n, is optimized with the constraint of a constant overall calculation time. A setup with $(n|m) = (1|128)$ is compared with $(n|m) = (2|64), (4|32), (8|16),$ and $(16|8)$ in Figure 6d. The optimum is obtained for four parallel calculated populations with 32 individuals each. Only a slight decrease in convergence is obtained going to $(n|m) = (2|64)$. The algorithm is more sensitive for a further decrease in the size of the population m. The "genetic pool" becomes too small in these cases.

The multiplicity deviation factor MDF is optimized in Figure 6e. The optimal value is MDF = 1 ppm for this example. This result is a compromise between the additional usable information coded by the multiplicity, which causes a better convergence compared to MDF = 0 ppm, and the higher complexity of the $\Delta(^{13}C)$ hypersurface, which causes a worse convergence in the case of higher values for MDF.

In the last plot of Figure 6 RP and MP are systematically changed. All combinations of RP and MP equal 0.0 or 1.0 for the two periods between 0..4 generations and 8..16 generations are tested except the case where RP and MP values are 0.0 at the same time (which would be of cause meaningless since mutation is forced if no recombination takes place, compare Figure 5). The test runs are sorted along the left axis by the lowest $\Delta(^{13}C)$ value for the fittest molecule in the population after the 16th generation. Best convergence is obtained for a high RP during the whole run and especially in the second part of the algorithm. A high mutation probability in the second part of the algorithm seems to be counter-productive. The same is true for having
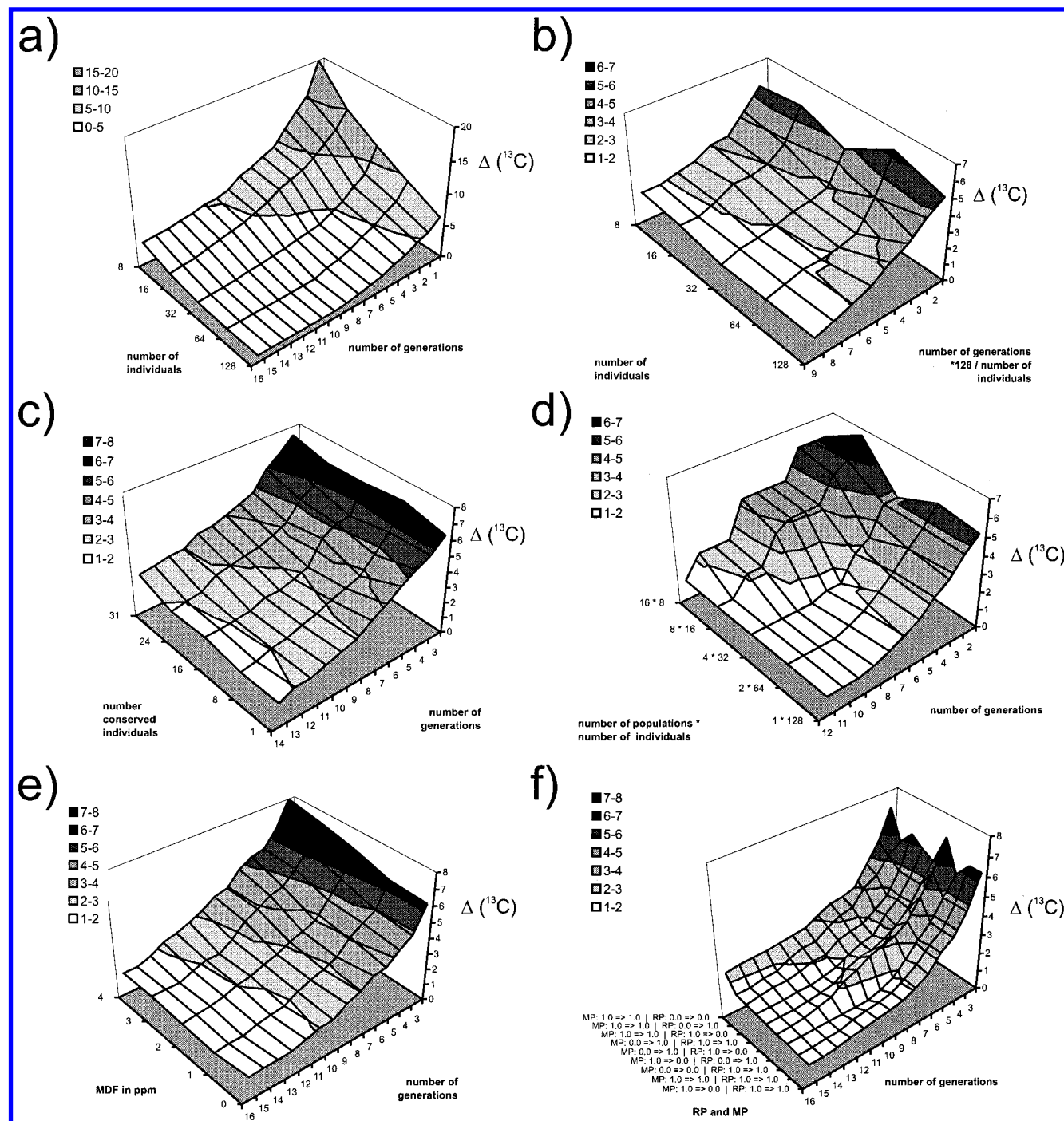
Figure 6. The optimization of the parameters is illustrated for elucidating the isoleucine structure by the implemented genetic algorithm. The $\Delta(^{13}C)$ of the best found solution is always displayed on the z axis. The number of calculated generations is given at the x axis (in the diagram (b) the axis is scaled), and the optimized parameter is displayed at the y axis. Regions on the surfaces coded in the same color are isobars of $\Delta(^{13}C)$. Diagram (a) proves that an increase of the population size m causes a faster convergence of the algorithm due to a higher number of structures generated. Diagram (b) obtains the comparison of differing population sizes m rescaled to an equal number of generated structures for a fair comparison. The results of the optimization of the part of conserved individuals l is optimized in diagram (c). The number of parallel calculated populations n is again optimized with special care of a comparable number of generated structures in experiment (d). The introduced multiplicity deviation factor MDF is investigated in experiment (e), and the influence of mutation probability MP as well as recombination probability RP are visualized in diagram (f).

no recombination at all. However, the differences between the several test runs are again relatively small.

To get a better impression about the decision making processes during the genetic algorithm Figure 7 represents tracks on that isoleucine is formed for a test run with $n = 1$, $m = 32$, $l = 8$, MDF = 1 ppm, and (MP: $^01.0^4 => {}^80.5^\infty$ | RP: $^00.0^4 => {}^81.0^\infty$). In the 10th generation isoleucine itself occurs for the first time. Up to this point the $\Delta(^{13}C)$ values

for all 32 molecules in the population are given. The constitutions that participate in the formation of isoleucine at any point in the algorithm are represented together with the performed mutation and recombination steps. Consistent with the high mutation and low recombination probabilities during the first part of the calculation only mutations take place in this period. A rapid decrease of the $\Delta(^{13}C)$ values is obtained, which can be interpreted as a local minimization of the
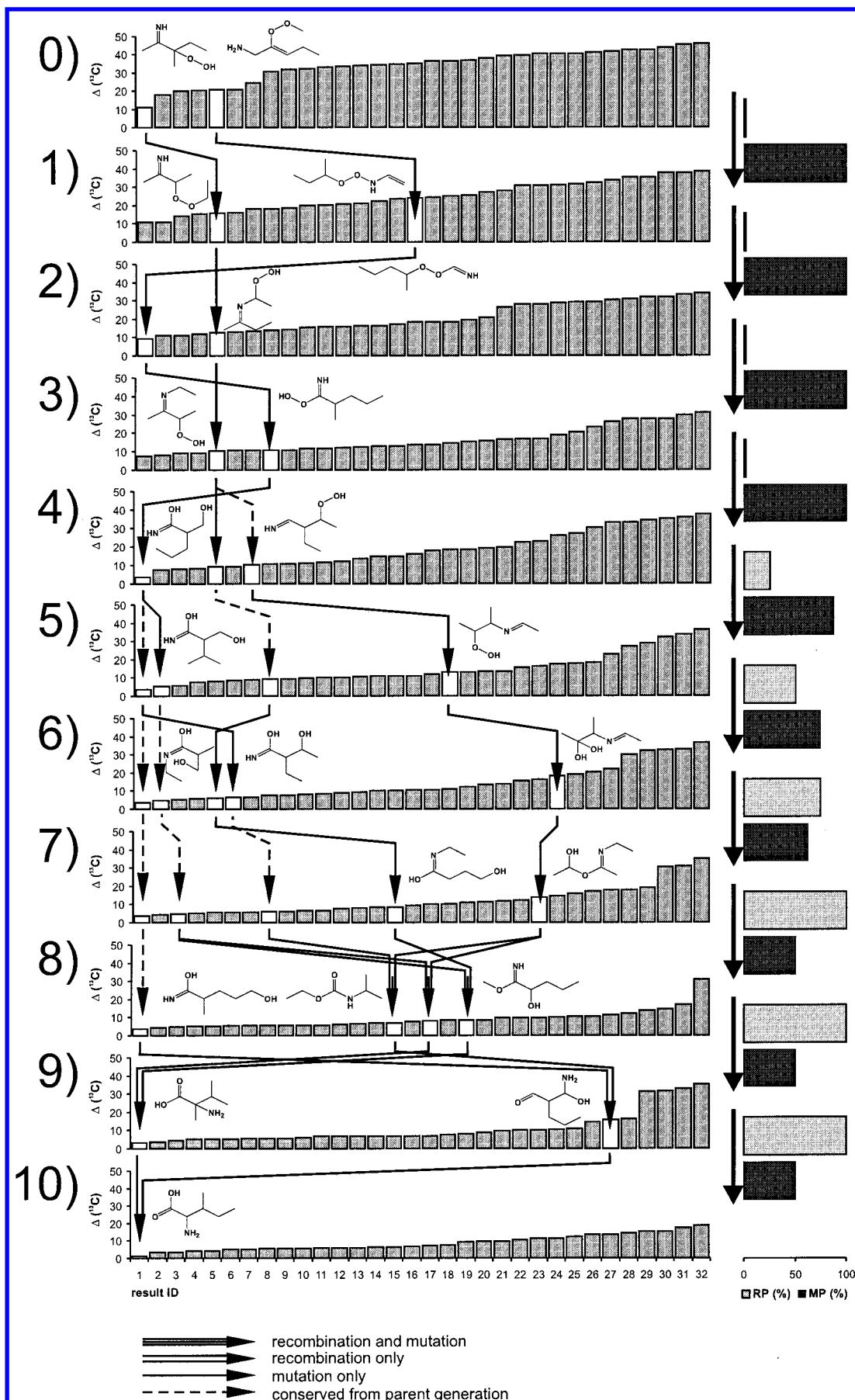
Figure 7. Optimization process for the example molecule of isoleucine. For the 11 performed optimization steps the $\Delta(^{13}C)$ values for all generated constitutions are given. All constitutions that participate in the formation of the correct solution isoleucine in the 11th generation are displayed, and the performed operation is marked by different arrows as indicated in the legend. The right column of the figure illustrates the used recombination and mutation probability values in the single steps.

STRUCTURE ELUCIDATION OF ORGANIC MOLECULES

J. Chem. Inf. Comput. Sci., Vol. 41, No. 6, 2001 **1543**

Table 1. Fully Automated Structure Elucidation for a Database Containing 8·20 Molecules with 9−16 Heavy Atoms

number heavy atoms	9	10	11	12	13	14	15	16
average $\Delta(^{13}C)$ (ppm) for correct solutions[a]	1.23	1.13	1.19	1.00	0.97	1.10	1.18	1.15
number correct solutions[b]	20	20	18	16	14	14	5	4
algorithm stopped with smaller deviation than target[c]	0	0	0	0	0	1	2	4
average calculation time (min)[d]	2	2	13	23	37	51	85	123
average number steps[e]	18	13	82	154	197	258	375	414
generated structures per minute[f]	1952	1733	1605	1708	1375	1293	1133	863

[a] Average $\Delta(^{13}C)$ (ppm) value of the 20 molecules representing the correct solutions. [b] Total number of molecules with correctly determined solutions out of the 20 tested molecules. [c] Number of test runs stopped because a structure with a smaller $\Delta(^{13}C)$ than the correct solution structure was created out of the 20 testes molecules. [d] Total average calculation time in minutes on a PII processor with 450 MHz and 512 MB RAM. [e] Total average number of steps until the algorithm was stopped. [f] Average number of generated and tested structures per minute.

constitutions on the $\Delta(^{13}C)$ hypersurface which can then be used in subsequent iterations as an optimal starting point for the recombination process. With the increase of the RP value the number of successful recombinations increases too. However, the first effectively used recombination for the formation of isoleucine takes place during the generation of the eighth population in this example. During this eighth step recombination is performed in combination with a subsequent mutation. In the next two steps the isoleucine constitution structure is formed by recombination steps without an additional mutation. The average decrease of the $\Delta(^{13}C)$ is more moderate during these later steps. This second part of the optimization can be interpreted as a search through the low $\Delta(^{13}C)$ ranges of the hypersurface for the global minimum. Although constitutions with small $\Delta(^{13}C)$ values play statistically a more important role in the eventual formation of the isoleucine molecular constitution, the influence of some individuals with high $\Delta(^{13}C)$ value is also observed in the process. This behavior is typical for a genetic algorithm.

The second experiment addresses the following two questions: (i) the possibility of automated structure elucidation by this approach and (ii) statistical analysis of a database of molecules is performed. Six groups of 20 molecules containing 9−16 non-hydrogen atoms, respectively, are randomly selected from the Specinfo database.[7] The experimental ^{13}C NMR chemical shifts are used as input for the algorithm. The setup is equivalent for all 160 molecules with $m = 8$, $n = 32$, $l = 8$, MDF = 1 ppm, and (MP: $^{0}1.0^{25} =>$ $^{50}0.0^{\infty}$ | RP: $^{0}0.5^{25} => ^{50}1.0^{\infty}$). The algorithm is repeated until either the right constitution is formed, another constitution with a $\Delta(^{13}C)$ value smaller than the $\Delta(^{13}C)$ value of the correct solution molecule is created (accuracy limit) or a maximum of 500 generations is achieved (time limit).

In the first case the automated structure elucidation is successful, whereas in the second or the third cases the method treated as a failure. If a constitution with a smaller $\Delta(^{13}C)$ value than the correct solution exists. This happens because the ^{13}C chemical shift calculation is not exact enough to determine unambiguously the correct constitution of the unknown. The probability that such constitutions can be found increases with the number of possible structures. It is therefore the first limiting factor for the maximal size of a molecular constitution structure solvable by this algorithm. If a maximum of 500 generations is calculated without the formation of the correct solution, the optimization process is stopped and treated as a failure. The second limiting factor is the time necessary to search the structural space.
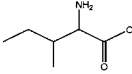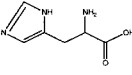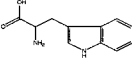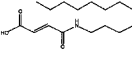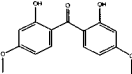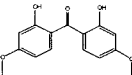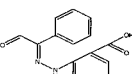
Table 1 summarizes the results of this experiment. The average $\Delta(^{13}C)$ value for the 20 structures is within the statistical deviation constant of about 1.1 ppm. Of 20 molecules tested in each case, the number of correct solutions found decreases slowly from 20 to 14 as the number non-hydrogen atoms increases from 9 to 14 atoms. It decreases rapidly to 5 and 4 corre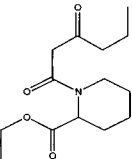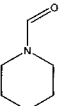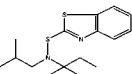ct solutions for 15 and 16 non-hydrogen atoms, respectively. For one out of 20 molecules with 14 non-hydrogen atoms the calculation is stopped for the first time because of a smaller $\Delta(^{13}C)$ value for a generated constitution than the $\Delta(^{13}C)$ value of the correct solution. For 15 and 16 non-hydrogen atoms 2 and 4 calculatons are stopped for this reason (accuracy limit). Both, the average calculation time and the average number of generations performed increase dramatically with an increasing number of non-hydrogen atoms. All calculations are performed on a PII 450 MHz Processor equipped with 512MB memory. The number of generated structures per minute increases more moderately. In comparing the last value with errors typical for other structure generators (e.g. Molgen) it has to be kept in mind that in this method not only the constitutions are generated but also aromatic ring systems must be identified and ^{13}C chemical shift must be calculated. Therefore, the genetic algorithm so implemented is slower compared to Molgen, if only the number of generated constitutions per unit time is compared.

The results prove that an automated elucidation of the constitution is possible for up to 14 non-hydrogen atoms with this setup. However, some points have to be addressed here: For reasons of comparability all calculations are started with the same parameter setup. The drop in the percentage of correct solutions going from 14 to 15 non-hydrogen atoms suggests that a setup optimized for isoleucine (with only nine non-hydrogen atoms) may no longer be optimal for 15 and 16 non-hydrogen atoms. Specifically too small values for n, m, and a maximum of only 500 generations avoid a higher percentage of correct solutions for larger molecules. Also, the inaccuracy in predicting the ^{13}C chemical shift information apparently plays an increasing role for molecules with 15 and more non-hydrogen atoms.

While the first problem can be solved by slightly modifying the setup of the algorithm, in the second case additional information beside the ^{13}C chemical shift is necessary to obtain an unambiguous result. This information can be a list of forbidden substructures (a "bad" list) or a list of substructures to use (a "good" list) in the easiest case. For 20 cases randomly selected form the nonsolved structures the test run is repeated using a modified setup of $l = 16$, $m = 64$, and a bad list of only four fragments (directly bounded heteroatoms: N−O, N−N, O−O, and allenyl fragments: C = C = C). For 17 out of these 20 examples the correct solution is found. However, both limits are present and have

Table 2. Molecular Structures, Parameters and Results Obtained for Some Example Molecules Solved by the Genetic Algorithm Approach

ID	name	molecular formula	numb. heavy atoms	structure	$\Delta(^{13}C)$ (ppm)[a]	number possible structures[b]	n [c]	m [d]	l [e]	MDF (ppm)[f]	MP[g] / RP[h]	excluded or included substructures[i]	numb. steps[j]	calculation time[k] (min)	struct. /time[l] (min^{-1})
1	Isoleucin	$C_6H_{13}NO_2$	9	(structure)	1.12	23,946	1	32	8	1.00	$^0 1.0^4 \Rightarrow {}^8 0.5^\infty$ / $^0 0.0^4 \Rightarrow {}^8 1.0^\infty$	---	11	<1	1408
2	Histidin	$C_6H_9N_3O_2$	11	(structure)	1.63	89,502,542	5	50	25	2.00	$^0 1.0^{10} \Rightarrow {}^{15} 0.5^\infty$ / $^0 0.0^{10} \Rightarrow {}^{15} 1.0^\infty$	For these examples a bad list of fragments was introduced that excludes allens (C=C=C) and any bonds between hetero atoms (X---X)	19	4	1187
3	Trypto-phan	$C_{11}H_{12}N_2O_2$	15	(structure)	1.44	$\approx 36 \cdot 10^9$	1	60	30	2.00	$^0 1.0^{100} \Rightarrow {}^{200} 0.5^\infty$ / $^0 0.0^{100} \Rightarrow {}^{200} 1.0^\infty$		327	20	981
4	-	$C_{16}H_{29}NO_3$	20	(structure)	0.81	$\approx 66 \cdot 10^9$	64	64	16	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$		71	453	642
5	-	$C_{15}H_{14}O_5$	20	(structure)	1.77	?	64	64	16	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$		1113	6704	680
5'	-	$C_{15}H_{14}O_5$	20	(structure)	1.77	?	1	64	16	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$	include 2x (benzene)	38	3	810
6	-	$C_{15}H_{12}N_2O_3$	20	(structure)	1.89	?	16	128	32	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$	include 2x (benzene)	4	18	455
7	-	$C_{14}H_{23}NO_5$	20	(structure)	1.36	?	8	32	8	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$	include (structure)	10	3	853
8	-	$C_{16}H_{24}N_2S_2$	20	(structure)	0.79	?	16	64	16	1.00	$^0 1.0^{25} \Rightarrow {}^{50} 0.0^\infty$ / $^0 0.5^{25} \Rightarrow {}^{50} 1.0^\infty$	include (structure)	302	540	573

[a] $\Delta(^{13}C)$ (ppm) value. [b] Total number of possible constitutions generated by Molgen if available. [c] Number of parallel calculated populations (n). [d] Number of individuals in the populations (m). [e] Number of best ranked (small $\Delta(^{13}C)$ value) individuals in the parent generation that are conserved for the new child generation (**l**). [f] **M**ultiplicity **D**eviation **F**actor defines the penalty added to $\Delta(^{13}C)$ for a wrong multiplicity of a ^{13}C carbon signal in a generated structure. [g] **M**utation **P**robability is the probability that a generated structure undergoes a mutation in a certain part of the algorithm. [h] **R**ecombination **P**robability is the probability that a structure generated by recombination of two individuals of the parent generation and not by just copying and mutating an individual of the parent generation. [i] Included structural fragments that have to be used in all generated structures and excluded structural fragments that are forbidden in all generated structures. [j] Total number of steps until the correct solution was found. [k] Total calculation time in minutes on a PII processor with 450 MHz and 512 MB RAM. [l] Number of generated and tested structures per minute.

to be discussed. The algorithm would run in case of a real unknown fully automatic until it is stopped by hand. Afterward all generated structures with a $\Delta(^{13}C)$ smaller than the average error of the neural network prediction plus the experimental deviation have to be considered as the possible solution to avoid losing the correct solution due to the accuracy limit. Due to the computational time limit, there is no guarantee that the correct solution is within the generated set of constitutions. Therefore this program also does not replace the spectroscopist. It is able to solve a part of routine problems in a fully automated mode. Afterward, the spectroscopist must validate the solutions and concentrate on the cases not unambiguously solved. As discussed below, the algorithm can however still assist solving those more complex and interesting cases.

We will now demonstrate using a few examples the ability of a more individual setup to solve the constitution for

molecules with up to 20 non-hydrogen atoms. Table 2 summarizes the results. Isoleucine is again listed as a first example molecule to enable comparison. The second example, histidine, is much more complex due to the increased number of non-hydrogen atoms and double bond equivalents. More than 89.5 million constitutions are possible for this molecular formula. However, this problem is still solved using a relative simple setup. Unlike all examples previously discussed, a bad list is used here for the first time. If not constrained, the genetic algorithm tends to create molecules that contain bonds between heteroatoms: e.g., N—N, O—O, or O—N. Because of the absence of chemical shift information for such fragments, it is difficult to recognize such structures as false solutions in cases where their overall $\Delta(^{13}C)$ value is small. If these structural fragments can be excluded, an accelerated convergence is obtained. It happens that fragments such as C = C = C are also favored by the

STRUCTURE ELUCIDATION OF ORGANIC MOLECULES

J. Chem. Inf. Comput. Sci., Vol. 41, No. 6, 2001 **1545**

genetic algorithm. The exclusion of this fragment also accelerates the optimization process. For tryptophan with 15 heavy atoms the determination of all possible structures is essentially impossible within a practical period of time. About 36 billion structures exist. The genetic algorithm creates only about 20 000 structures out of this huge number before the correct solution is found. Examples 4 and 5 demonstrate the power of the approach on molecules with 20 non-hydrogen atoms. While the first example has only three double bonds, in the second case the number of double bond equivalents is already nine. Due to this fact, the number of possible structures is much larger, and the problem requires a factor of 15 in computation time. However, computers are becoming faster and time can also be saved by computing parallel populations on parallel processors. A high degree of automation is a significant advantage for the method. By increasing the number of "intelligent" interventions, even molecules with more non-hydrogen atoms might become solvable. This is demonstrated in examples 5′, 6, 7, and 8, where parts of the molecule are defined in advance. In real application such information might be known from the NMR spectrum (e.g., examples 5′ and 6), from the synthesis, or even from the genetic algorithm itself. The letter case is emphasized if one fragment is generated with a high frequency in the process of the genetic algorithm, and the $\Delta(^{13}C)$ values of the corresponding carbon atoms are low. Such a fragments has a high probability of being part of the solution structure. This fragment could then be defined as fixed. Example 5′ differs from example 5 only by the fact that two benzene fragments need to be part of the generated constitutions. The dramatic acceleration of structure elucidation by this small intervention demonstrates how the approach can assist the spectroscopists. All known structural information can be introduced into the initial setup, and the remaining constitutional space can be searched quickly and effectively using the genetic algorithm. Similar essential significant decreases in computational time are observed for the other three example structures.

CONCLUSIONS

A general implementation of a genetic algorithm that uses molecular constitutions as individuals is described. This algorithm is able to optimize a molecular constitution structure to fulfill experimentally observed properties. The ^{13}C NMR spectrum of organic molecules can be accurately determined by experiments and also rapidly predicted by neural network calculation. Consequently the constitution of organic molecules can be optimized relative to an unknown organic sample by combining the genetic algorithm with the neural network spectral prediction. An automated structure elucidation is possible for molecules with up to 14 non-hydrogen atoms. Molecular structures with up to 20 non-hydrogen atoms can be determined using only their ^{13}C NMR chemical shifts by introducing a small list of forbidden fragments. Larger molecular structures become solvable if fragments that need to be in the molecule are known and introduced as a good list. The number of overall possible solution structures is drastically reduced by the inclusion of such known fragments.

The maximal size of the solvable molecule is limited by the size of the structural space accessible (time limit) or by the accuracy of the property determination, either by experiment or by calculation (accuracy limit). Since the ^{13}C NMR chemical shift prediction is the most time-consuming step of the algorithm and is also responsible for the introduction of the calculation error, it is the critical step for both limits. With a fast and accurate chemical shift prediction by neural networks the implementation of such a genetic algorithm becomes possible for the first time.

The described procedures are combined in the program "Genius"[31] that should become a helpful tool for structure elucidation of organic molecules.

REFERENCES AND NOTES

(1) Benecke, C.; Grund, R.; Hohberger, R.; Kerber, A.; Laue, R.; Wieland, T. MOLGEN+, a generator of connectivity isomers and stereoisomers for molecular structure elucidation. *Anal. Chim. Acta* **1995**, *314*, 141−147.

(2) Wieland, T.; Kerber, A.; Laue, R. Principles of the Generation of Constitutional and Configurational Isomers. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 413−419.

(3) Lindel, T.; Junker, J.; Köck, M. COCON: From NMR Correlation Data to Molecular Constitution. *J. Mol. Model.* **1997**, *3*, 364−368.

(4) Köck, M.; Junker, J.; Maier, W.; Will, M.; Lindel, T. A COCON Analysis of Proton-Poor Heterocycles − Application of Carbon Chemical Shift Predictions for the Evaluation of Structural Proposals. *Eur. J. Org. Chem.* **1999**, 579−586.

(5) Meiler, J.; Will, M.; Meusinger, R. Fast Determination of 13C NMR Chemical Shifts Using Artificial Neural Networks. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1169−1176.

(6) Meiler, J.; Köck, M. Structure Elucidation by Automatic Generation and Analysis of Molecule Databases from NMR Connectivity Information Using Substructure Analysis and 13C NMR Chemical Shift Prediction. **2001,** submitted for publication.

(7) Chemical Concepts: Karlsruhe, 2001.

(8) Will, M.; Fachinger, W.; Richert, J. R. Fully Automated Structure Elucidation − A Spectroscopist's Dream Comes True. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 221−227.

(9) Zupan, J.; Gasteiger, J. VCH Verlagsgesellschaft mbH: Weinheim, 1993.

(10) Kvasnicka, V.; Sklenak, S.; Pospichal, J. Application of Recurrent Neural Network in Chemistry. Prediction and Classification of 13C NMR Chemical Shiftsin a Series of Monosubstituted Benzenes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 742−747.

(11) Doucet, J.-P.; Panaye, A.; Feuilleaubois, E.; Ladd, P. Neural networks and carbon-13 NMR shift prediction. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 320−324.

(12) Panaye, A.; Doucet, J.-P.; Fan, B. T.; Feuilleaubois, E.; Azzouzi, S. R. E. Artificial neural network simulation of 13C NMR shifts for methyl-substituted cyclohexanes. *Chemom. Intell. Lab. Syst.* **1994**, *24*, 129−135.

(13) Sklenak, S.; Kvasnicka, V.; Pospichal, J. Prediction of 13C NMR chemical shifts by neural networks in a series of monosubstituted benzenes. *Chem. Pap.* **1994**, *48*, 135−140.

(14) Clouser, D. L.; Jurs, P. C. Simulation of the 13C nuclear magnetic resonance spectra of trisaccharides using multiple linear regression analysis and neural networks. *Carbohydr. Res.* **1995**, *271*, 65−77.

(15) Svozil, D.; Pospichal, J.; Kvasnicka, V. Neural Network Prediction of Carbon-13 NMR Chemical Shifts of Alkanes. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 924−928.

(16) Thomas, S.; Kleinpeter, E. Assignment of the 13C NMR chemical shifts of substituted naphthalenes from charge density with an artificial neural network. *J. Prakt. Chem./Chem.-Ztg.* **1995**, *337*, 504−507.

(17) Clouser, D. L.; Jurs, P. C. Simulation of the 13C Nuclear Magnetic Resonance Spectra of Ribonucleosides Using Multiple Linear Regres-

sion Analysis and Neural Networks. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 168−172.

(18) Ivanciuc, O.; Rabine, J. P.; Cabrol-Bass, D.; Panaye, A.; Doucet, J.-P. 13C NMR Chemical Shift Prediction of sp2 Carbon Atoms in Acyclic Alkenes Using Neural Networks. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 644−653.

(19) Meiler, J.; Meusinger, R.; Will, M. Neural Network Prediction of 13C NMR Chemical Shifts of Substituted Benzenes. *Monatsh. Chem.* **1999**, *130*, 1089−1095.

(20) Pearlman, D. A. Automated detection of problem restraints in NMR data sets using the FINGAR genetic algorithm method. *J. Biomol. NMR* **1999**, *13*, 325−335.

(21) Fisher, B.; Dillon, N.; Carpenter, T.; Hall, L. Design by Genetic Algorithm of a Z Gradient Set for Magnetic Resonance Imagine of the Human Brain. *Measurement Sci. Technol.* **1995**, *6*, 904−909.

(22) Meusinger, R.; Moros, R. In *Software − Entwicklung in der Chemie*; Gasteiger, J., Ed.; Gesellschaft Deutscher Chemiker: Frankfurt am Main, 1995; Vol. 10, pp 209−216.

(23) Pearlman, D. A. FINGAR: A newgenetic algorithm-based method for fitting NMR data. *J. Biomol. NMR* **1996**, *8*, 49−66.

(24) Li, L.; Darden, T. A.; Freeman, S. J.; Furie, B. C.; Furie, B.; Baleja, J. D.; Smith, H.; Hiskey, R. G.; Pedersen, L. G. Refinement of the NMR Solution Structure of the γ-Carboxyglutamic Acid Domain of Coagulation Factor IX Using Molecular Dynamics Simulation with Initial Ca2+ Positions Determined by a Genetic Algorithm. *Biochemistry* **1997**, *36*, 2132−2138.

(25) Choy, W. Y.; Sanctuary, B. C. Using Genetic Algorithms with a Priori Knowledge for Quantitative NMR Signal Analysis. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 685−690.

(26) Robien, W. Das CSEARCH-NMR-Datenbanksystem. *Nachr. Chem. Technol. Lab.* **1998**, *46*, 74−77.

(27) Clerc, J.-T.; Sommerauer, H. A Minicomputer Program Based On Additivity Rules For The Estimation Of 13C NMR Chemical Shifts. *Anal. Chim. Acta* **1977**, *95*, 33−40.

(28) Ebraheem, K. A. K.; Webb, G. A. Semiempirical Calculations of the Chemical Shifts of Nuclei other than Protons. *Prog. NMR Spectrosc.* **1977**, *11*, 149−181.

(29) Bremser, W.; Ernst, L.; Franke, B.; Gerhards, R.; Hardt, A. Verlag Chemie: Weinheim, 1981.

(30) Fürst, A.; Pretsch, E. A computer program for the prediction of 13C NMR chemical shifts of organic compounds. *Anal. Chim. Acta* **1990**, *229*, 17−25.

(31) Meiler, J. www.jens-meiler.de, 2001.