

# Fast Evaluation of Chemical Distance by Simulated-Annealing Algorithm

Jiří Pospíchal\* and Vladimír Kvasnička

Department of Mathematics, Slovak Technical University, 81237 Bratislava, Slovakia (Czechoslovakia)

Received January 26, 1993\*

The concept of chemical distance belongs among basic formal tools of mathematical models of organic chemistry. In particular, it serves as a proper measure of similarity (or dissimilarity) between two molecules represented by molecular graphs. The principle of minimal chemical distance, introduced by Dugundji and Ugi, is a formal counterpart of the well-known principle of minimal structural change widely used in organic chemistry in designing the reaction mechanism of chemical reaction  $\text{educt} \Rightarrow \text{product}$ . Applications of standard combinatorial methods to evaluation of chemical distances are ineffective for larger molecular graphs. A simulated-annealing algorithm is developed that locates suboptimal chemical distances closely (or even precisely) to their exact values. The most effective operator (transforming a vertex-matching permutation onto another feasible matching) is the Fontain's reordering operator that reflects to some extent bonding conditions in both molecular graphs. An augmented simulated-annealing algorithm is formulated that optimizes a pool of mutually interacting (via the exchange operator) permutations.

## INTRODUCTION

In the model of constitutional chemistry introduced by Dugundji and Ugi<sup>1</sup> (see also refs 2–10), the concept of chemical distance plays a crucial role. It serves as the proper mathematical tool to “measure” similarity (or dissimilarity) between two molecules represented by molecular graphs.<sup>11–15</sup> Moreover, chemical distance is very important as the main driving force for construction of reaction networks<sup>3,6,8,16–19</sup> composed of intermediates that are constituents of a reaction mechanism of the reaction  $\text{educt} \Rightarrow \text{product}$ .

Let  $G_1$  and  $G_2$  be two molecular graphs composed of the same number  $N$  of vertices – atoms (we say that  $G_1$  and  $G_2$  are stoichiometric). The corresponding adjacency matrices (or BE matrices<sup>1</sup> in the terminology of Dugundji and Ugi) will be denoted by  $A_1 = (A_{ij})$  and  $A_2 = (A'_{ij})$ , respectively. Let  $P = (p_1, p_2, \dots, p_N)$  be a permutation of  $N$  objects  $\{1, 2, \dots, N\}$ . The chemical distance between  $G_1$  and  $G_2$  may be defined as a minimum value of the  $L_1$  norm of a “difference” of adjacency matrices for all possible permutations  $P$ ,

$$CD(G_1, G_2) = \min_P \sum_{i,j=1}^N |A_{ij} - A'_{p_i p_j}| \quad (1)$$

We have to note that if  $G_1$  and  $G_2$  are molecular graphs (their vertices are labeled by atomic symbols), then the permutation  $P$  relates vertices of  $G_1$  only to those vertices of  $G_2$  that are evaluated by the same atomic symbol. The chemical distance  $CD(G_1, G_2)$  corresponds to the minimum number of edges – bonds that should be simultaneously canceled and created in  $G_1$  in order to “transform” this graph into another one isomorphic to  $G_2$ . Graph- and set-theoretically, we may say that chemical distance  $CD(G_1, G_2)$  is the minimum value of the symmetric difference of the edge sets of  $G_1$  and  $G_2$  regarding all possible indexings (realized by permutations  $P$ ). Thus indexings of  $G_1$  and  $G_2$  match such that the intersection of the edge sets (corresponding to the maximal common subgraph) has the maximum number of edges. This means that the chemical distance is given by the number of edges in

the graphs  $G_1$  and  $G_2$  that are not in the maximum common subgraph. The above interpretation of definition 1 initiates its alternative pure graph-theoretical determination of chemical distance as follows:<sup>2,10</sup>

$$CD(G_1, G_2) = |G_1| + |G_2| - 2|G_1 \cap G_2| \quad (2)$$

where  $|G|$  denotes the cardinality of the edge set of graph  $G$  and  $G_1 \cap G_2$  is a maximum common subgraph (not induced) of  $G_1$  and  $G_2$ . Both of these definitions 1 and 2 are fully equivalent; it should be emphasized that the concept of maximal common subgraph  $G_1 \cap G_2$  has been used as a noninduced subgraph.<sup>20,24</sup> In the opposite case, assuming that  $G_1 \cap G_2$  forms an induced common subgraph, the algebraic definition 1 is not equivalent to the graph-theoretical definition 2. It has been demonstrated<sup>2,4</sup> that the chemical distance defined either by (1) or (2) is the so-called metrics; it is nonnegative (zero only for isomorphic graphs) and symmetric and satisfies the triangular inequality.

In general, evaluation of chemical distance belongs to the so-called *NP*-complete problems;<sup>21</sup> its correct evaluation needs  $N!$  matchings of vertices between graphs  $G_1$  and  $G_2$ . Definition 1 may be used<sup>7</sup> for the backtrack searching algorithm<sup>22</sup> accelerated by making use of the branch-and-bound conditions which stop the construction of a searching tree if the current value of the chemical distance is greater than its best value already achieved. This simple and straightforward method for the evaluation of chemical distance is applicable only for molecular graphs composed of at most 9–10 vertices. A more sophisticated algorithm suggested by McGregor<sup>23</sup> (see also ref 7), which uses edge matchings instead of vertex matchings, represents substantial progress in the methodology of correct evaluation of the chemical distance. Unfortunately, similarly as in the previous method, for larger graphs or for graphs with all vertices of the same valence, its efficiency is also very small. It seems that the best algorithm known for the correct evaluation of chemical distance is the recently used method<sup>20,25</sup> based on the process of looking for maximum cliques in the product  $\hat{G}_1 \times \hat{G}_2$ , where  $\hat{G}_1$  and  $\hat{G}_2$  are the line graphs<sup>20,24</sup> assigned to the original molecular graphs  $G_1$  and  $G_2$ , respectively. Though this method is much more powerful

\* Abstract published in *Advance ACS Abstracts*, September 15, 1993.

than the previous two methods, it has also a "magic" number of vertices (usually 12–14) after which its efficiency strongly decreases. This is caused by the fact that the method is based on the process of looking for the maximum cliques, the number of which for greater graphs may grow enormously. Finally, two different methods for the evaluation of chemical distance have been elaborated in the laboratory of Ivar Ugi. The first one<sup>26</sup> uses methods known in operational research, e.g. branch-and-bound acceleration and perturbation calculations; the other<sup>27,28</sup> is based on a relaxation procedure for canonical indexing of chemical structures, in which the basic common substructures are located and then the chemical distance is combinatorially evaluated for substantially reduced molecular graphs. As Fontain<sup>29,30</sup> recently mentioned, both these approaches require an appreciable amount of computing time for larger molecules.

In summary, all the methods discussed above are applicable only for smaller molecular graphs. Their application to larger molecular graphs composed of 20–30 vertices engenders enormous consumption of computer time (combinatorial catastrophe). Consequently, application of the concept of chemical distance as a driving force for the construction of reaction networks with large educt and product molecular graphs calls for a different type of chemical distance-evaluating algorithm, faster than the combinatorial algorithms discussed above.

Recently, Fontain<sup>29,30</sup> has applied a genetic algorithm<sup>31</sup> for fast suboptimal evaluation of chemical distance between large molecular graphs. He has demonstrated that a modified version of this approach is able to evaluate chemical distance much faster than conventional combinatorial methods. The used modification of the genetic algorithm introduces the so-called reordering operator, one which reflects to some extent bonding conditions in both compared molecular graphs. Inclusion of this operator to the genetic algorithm substantially improves the resulting chemical distances, making them closer to their exact values.

The purpose of the present paper is to apply the *simulated-annealing algorithm*<sup>32,33</sup> for the fast suboptimal evaluation of chemical distances. The simulated annealing has been successfully used<sup>32a,b</sup> for solution of the famous combinatorial task—the traveling salesman problem.<sup>34</sup> Its effectiveness and simple implementation are the main reason why it serves as a benchmark for new algorithms that are suggested for solution of the traveling salesman problem. Recently, the simulated annealing was used in chemistry as a powerful optimization tool for finding the global conformation-energy minimum of macromolecules.<sup>35</sup> The simulated annealing shares many common features with the genetic algorithm. Both methods use as basic entities a linear string of numbers to be optimized and operators that are transforming the string onto another feasible string. However, their optimization strategies are of quite different nature. Whereas the genetic algorithm uses a pool of strings and the main driving optimization force is the crossover operator (an exchange of substrings between two strings), the simulated annealing employs only one string and the driving optimization force is Metropolis' algorithm<sup>32</sup> based on physical considerations using an analogue of Boltzmann distribution. We shall demonstrate that the simulated annealing enlarged by Fontain's reordering operator provides a very effective method for the fast evaluation of chemical distance between large molecular graphs. The results are close to (or equal) exact values of chemical distances. Comparison of the proposed algorithm with the genetic algorithm allows conclusion that the simulated annealing, when

applied to the evaluation of chemical distance, is more effective than the genetic algorithm. Its implementation is very simple, and it does not pose extreme demands on the amount of computer high-speed memory.

The simulated-annealing algorithm can be generalized in a form called *augmented simulated annealing*, where, instead of the single permutation, a pool composed of a few permutations is optimized. The permutations interact mutually through an *exchange operator*, which is realized in the manner known in genetical algorithm by the crossover operator. For two permutations this operator exchanges some segments of indices and ensures also that the resulting objects are genuine permutations. We have observed that the augmented simulated annealing is more effective in finding correct minima of chemical distance than its original version with a "pool" composed of one permutation.

## SIMULATED-ANNEALING ALGORITHM

The simulated-annealing algorithm<sup>32,33</sup> is a technique which has recently been getting significant attention as an effective approach suitable for optimization of very large scale (e.g. combinatorial) problems. Its practicability has been illustrated by finding a suboptimal solution of the famous traveling salesman problem.<sup>32</sup> This problem is of great practical importance for our purposes since it solves a combinatorial problem similar to the evaluation of chemical distance. The traveling salesman problem requires a permutation of  $N$  objects (cities) that minimizes the cyclical itinerary for a traveling salesman who must visit each of  $N$  cities in turn. For the evaluation of chemical distance between two molecular graphs (see previous chapter), one has to find also a permutation of  $N$  objects (vertices of graphs) which corresponds to a minimum of the  $L_1$  norm of the difference of adjacency matrices or to their maximal common subgraph.

The concept of simulated annealing has been derived from physics,<sup>33</sup> where it is used to describe the process of eliminating lattice defects in crystals by heating, followed by slow cooling to low temperature. In the present context a lattice defect corresponds to the incorrect matching between two molecular graphs. So the essence of the simulated-annealing algorithm is slow cooling, allowing ample time for redistribution of the atoms as they lose mobility. This is the technical definition of annealing, and it is essential for ensuring that a low energy state will be achieved. Although the analogy is not perfect, there is a high probability that its applications will provide suboptimal solutions of combinatorial problems that are closely related to correct solutions.

The basic entity that should be optimized by the simulated annealing is a permutation of  $N$  objects

$$P = (p_1, p_2, \dots, p_N) \quad (3)$$

We assign to this permutation  $P$  the current chemical distance (a counterpart of the objective function known in optimization methods) determined by the right-hand side of (1)

$$CD(P) = \sum_{\substack{i,j=1 \\ (i \neq j)}}^N |A_{ij} - A'_{p_i p_j}| \quad (4)$$

It means that the exact chemical distance  $CD(G_1, G_2)$  between two molecular graphs  $G_1$  and  $G_2$  is equal to the minimum of  $CD(P)$  over all  $(N!)$  permutations from the  $N$ th-order symmetric group. The permutation  $P$  is transformed onto

another permutation by an operator  $O$

$$O : P = (p_1, p_2, \dots, p_N) \rightarrow P' = (p'_1, p'_2, \dots, p'_N) \quad (5)$$

shortly  $P' = O(P)$ .

In our forthcoming considerations we shall use four types of operators that are specified by a proper selection of two or three indices, called the *parameters* of the operator.

**Transposing Operator.** Let two entries of permutation  $P$  indexed by  $i_1$  and  $i_2$ , where  $1 \leq i_1 < i_2 \leq N$ , be mutually transposed; then we get new permutation  $P'$ ,

$$P = (p_1, \dots, p_{i_1-1}, p_{i_1}, p_{i_1+1}, \dots, p_{i_2-1}, p_{i_2}, p_{i_2+1}, \dots, p_N)$$

$$P' = (p_1, \dots, p_{i_1-1}, p_{i_2}, p_{i_1+1}, \dots, p_{i_2-1}, p_{i_1}, p_{i_2+1}, \dots, p_N)$$

**Reordering operator** was initially introduced in the framework of a genetic algorithm by Fontain.<sup>29,30</sup> In contrast to the other three operators which do not reflect in any way the structure of graphs  $G_1$  and  $G_2$ , the reordering operator reflects to some extent their structure. Let, for an index  $1 \leq i \leq N$  and for the permutation  $P$ ,  $\Gamma_1(i)$  and  $\Gamma_2(p_i)$  be the vertex subsets composed of vertices adjacent to the vertex  $i$  in  $G_1$  and to the vertex  $p_i$  in  $G_2$ , respectively. Then, the permutation  $P$  is modified such that the indices from  $\Gamma_1(i)$  are randomly mapped onto the indices from  $\Gamma_2(p_i)$ . If these subsets have different cardinalities, then the mapping contains the number of entries equal to the smaller cardinality. The resulting permutation  $P'$  is formed from the original  $P$  such that transpositions of entries  $p_a$  and  $p_c = b$  in the permutation are successively done for each entry  $a \rightarrow b$  of the mapping  $\Gamma_1(i) \rightarrow \Gamma_2(p_i)$ .

The work of the reordering operator will be illustrated by the following simple example. Let us consider a permutation composed of 10 entries,  $P = (3, 1, 6, 4, 2, 10, 7, 5, 9, 8)$ . Let, for the index  $i = 3$ , the vertex subsets be determined by  $\Gamma_1(3) = \{1, 2, 4\}$  and  $\Gamma_2(6) = \{5, 7, 9, 10\}$ . These subsets determine those atoms that are connected with atoms 3 and 6 in the molecular graph  $G_1$  and  $G_2$ , respectively. The vertex subset  $\Gamma_1(3)$  is randomly mapped onto the vertex subset  $\Gamma_2(6)$  as follows:  $1 \rightarrow 7$ ,  $2 \rightarrow 10$ , and  $4 \rightarrow 5$ . This means that by applying the first mapping  $a = 1 \rightarrow b = 7$  the permutation  $P$  is transformed so that its entries  $p_a = 3$  and  $p_c = 7$  are transposed, similarly to the other two remaining mappings,

$$\text{mapping } 1 \rightarrow 7: \quad P = (3, 1, 6, 4, 2, 10, 7, 5, 9, 8)$$

$$\text{mapping } 2 \rightarrow 10: \quad P = (7, 1, 6, 4, 2, 10, 3, 5, 9, 8)$$

$$\text{mapping } 4 \rightarrow 5: \quad P = (7, 10, 6, 4, 2, 1, 3, 5, 9, 8)$$

$$\text{result:} \quad P' = (7, 10, 6, 5, 2, 1, 3, 4, 9, 8)$$

**Transporting Operator.** A segment of permutation  $P$  (specified by indices  $i_1$  and  $i_2$ ) is excised and then inserted into a new position (specified by  $i_3$ ), where the indices are restricted by  $1 \leq i_1 < i_2 < i_3 \leq N$ ,

$$P = (p_1, \dots, p_{i_1-1}, p_{i_1}, \dots, p_{i_2}, p_{i_2+1}, \dots, p_{i_3}, p_{i_3+1}, \dots, p_N)$$

$$P' = (p_1, \dots, p_{i_1-1}, p_{i_2+1}, \dots, p_{i_3}, p_{i_1}, \dots, p_{i_2}, p_{i_3+1}, \dots, p_N)$$

Theoretically, there exists also the second possibility, namely,  $i_3 < i_1$  (i.e. a segment is now moved to an insertion position on its left-hand side), but it may be demonstrated by simple

considerations that this alternative possibility is equivalent to the first process.

**Reversing Operator.** A segment of perturbation  $P$  specified by indices  $i_1$  and  $i_2$ , where  $1 \leq i_1 < i_2 \leq N$ , is reversed

$$P = (p_1, \dots, p_{i_1-1}, p_{i_1}, p_{i_1+1}, \dots, p_{i_2-1}, p_{i_2}, p_{i_2+1}, \dots, p_N)$$

$$P' = (p_1, \dots, p_{i_1-1}, p_{i_2}, p_{i_2-1}, \dots, p_{i_1+1}, p_{i_1}, p_{i_2+1}, \dots, p_N)$$

The current chemical distance  $CD(P)$  determined by (4) is formally interpreted as an energy of the physical system with a structure described by the permutation  $P$ . An ensemble of such systems in thermal equilibrium is simulated. As a parameter  $T$  (a formal analog of temperature) is decreasing, the system is frozen out to a state with a low value of the current chemical distance (energy). To achieve an equilibrium state, the Metropolis method<sup>32</sup> is used. For the randomly selected operator  $O$  (and its randomly selected parameters), the permutation  $P$  is transformed to another permutation  $P'$ ,  $P' = O(P)$ . The resulting permutation  $P'$  is accepted with probability

$$\Pr(P \rightarrow P') =$$

$$\begin{cases} 1 & \text{for } CD(P') - CD(P) \leq 0 \\ \exp\left(-\frac{CD(P') - CD(P)}{T}\right) & \text{for } CD(P') - CD(P) > 0 \end{cases} \quad (6)$$

Notice that if  $CD(P') < CD(P)$ , the exponential formula would give a probability greater than unity. In such cases the probability of the acceptance of a new permutation  $P'$  is automatically set to unity; that is, the old permutation  $P$  is always replaced by the new permutation  $P'$ . We have to emphasize that all four operators are repeatedly applied with the same probability; each of them is used with the same preference as the other ones.

The annealing schedule requires some experimentation. The procedure is initialized by random generation of the permutation  $P$ . A starting value of the parameter  $T$  must be suitably chosen. In our applications  $T_{\max} = 3-5$  worked well. We proceed downward in multiplicative steps amounting to the 10% decrease in  $T$ . Each value of  $T$  is kept fixed for  $100N$  applications of operators (randomly applied with the same priority and with randomly generated parameters) or for 10N successful transformations (according to Metropolis method). The whole optimization process is terminated if a current value of  $T$  becomes smaller than a prescribed value  $T_{\min}$  (for example, 0.01) or if for a current value of  $T$  (greater than  $T_{\min}$ ) no transformation has been successful. The implementation of this annealing schedule is coded in pseudo-Pascal.

#### Algorithm Simulated\_Annealing.

```

1  permutation P is randomly generated;
2  T:=T_max; success:=1;
3  WHILE (T>T_min)^(success>0) DO
4  BEGIN counter:=0; success:=0;
5      WHILE (counter<counter_max)^(success<success_max) DO
6          BEGIN counter:=counter+1;
7              select randomly operator o and its parameters;
8              P':=O(P);
9              IF CD(P')-CD(P)<0 THEN
10                 Pr:=1 ELSE Pr:=exp(-(CD(P')-CD(P))/T);
11                 IF random<Pr THEN
12                     BEGIN P:=P'; success:=success+1 END;
13             END;
14         T:=alpha*T;
15     END;
```

The temperature  $T$  ranged by  $T_{\min} \leq T \leq T_{\max}$ , where we set  $T_{\max} = 3-5$  and  $T_{\min} = 0.01$ . The parameter  $\alpha$ , which we set  $\alpha = 0.90$ , multiplicatively decreases the temperature  $T$  (see line 12). It means that the decreasing parameter  $\alpha$  is kept fixed in the course of the whole annealing process. This is indeed disadvantageous for smaller values of temperature  $T$  when the optimized system is relatively invariant with respect to applications of operators. In this case the temperature may be decreased much faster than at the beginning of simulated annealing. We therefore recommend an updating of the parameter  $\alpha$  such that for each outer cycle (line 3) it is updated by  $\alpha := \alpha_{\max}(\beta)^n$ , where  $0 \ll \beta < 1$  and the nonnegative integer  $n$  corresponds to a number of repeats of the outer cycle. The algorithm started with  $\alpha = \alpha_{\max}$  (for  $n = 0$ ), the next value of  $\alpha$  is  $\alpha_{\max}(\beta)$  (for  $n = 1$ ), etc.

Integer variables *counter* and *success* enumerate all and successful applications of transformation operators, respectively, their upper bounds are determined by  $\text{counter}_{\max} = 100N$  and  $\text{success}_{\max} = 10N$ . The statement placed in line 7 means that one of four operators together with random specification of its parameters (indices) is randomly selected. All four operators are of equal opportunity (that is of the same probability). The Metropolis selection algorithm is implemented in lines 9 and 10. The new permutation  $P'$  is incorporated in the optimization process (i.e.  $P := P'$ ) if a random number  $0 < \text{random} < 1$  is smaller than the probability  $\text{Pr}$ .

#### AUGMENTED SIMULATED-ANNEALING ALGORITHM

The original version of the simulated annealing uses (see previous section) only single permutation which is optimized according to the Metropolis algorithm. This idea may be generalized such that instead of the single permutation a pool of permutations is optimized. If we use only the standard four operators (transposing, transporting, reversing, and reordering), then nothing new may happen. The pool permutations are fully independent, unable to mutually interact; the used four operators act only on isolated permutations. Therefore, in order to introduce an interaction among pool permutations, it is necessary to introduce a new type of operator, which acts on a pair of permutations. This pair-interaction operator  $O$  creates from two permutations  $P$  and  $Q$  new permutations  $P'$  and  $Q'$ , formally  $(P', Q') = O(P, Q)$ , where the question of replacing initial permutations in the pool by created permutations is solved in the standard way of simulated annealing, that is, by the Metropolis algorithm.<sup>32</sup> In our forthcoming considerations the pair-interaction operator will be called the exchange operator. It reflects a fact that some exchange of information between permutations  $P$  and  $Q$  is realized in the course of creation of new permutations  $P'$  and  $Q'$ . We have tried to use many types of the exchange operator; the best experience we have obtained with the so-called partially matched crossover. This operator was initially introduced for purposes of genetical-algorithm approach to the traveling salesman problem by Goldberg and Lingle<sup>36</sup> (see also ref 31).

**Exchange Operator.** Let us select for permutations  $P$  and  $Q$  two indices  $1 \leq i_1 < i_2 \leq N$ ; they specify a segment

$$\begin{aligned} P &= (p_1, \dots, p_{i_1-1}, p_{i_1}, \dots, p_{i_2}, p_{i_2+1}, \dots, p_N) \\ Q &= (q_1, \dots, q_{i_1-1}, q_{i_1}, \dots, q_{i_2}, q_{i_2+1}, \dots, q_N) \end{aligned} \quad (7)$$

Exchanging the segments between permutations, we obtain

$$\begin{aligned} \hat{P} &= (p_1, \dots, p_{i_1-1}, q_{i_1}, \dots, q_{i_2}, p_{i_2+1}, \dots, p_N) \\ \hat{Q} &= (q_1, \dots, q_{i_1-1}, p_{i_1}, \dots, p_{i_2}, q_{i_2+1}, \dots, q_N) \end{aligned} \quad (8)$$

Unfortunately, these new objects  $\hat{P}$  and  $\hat{Q}$  may not be permutations at all. It may happen that some integer appears more than once; therefore it is necessary to apply a repairing process (called the partial matching) which transforms the objects  $\hat{P}$  and  $\hat{Q}$  to other objects  $P'$  and  $Q'$ , respectively, that are permutations. Let us define a mapping  $f_{PQ}$  such that entries of the sequence in  $P$  are mapped onto entries of the sequence in  $Q$

$$f_{PQ}: p_i \rightarrow q_i, \text{ for } i_1 \leq i \leq i_2 \quad (9)$$

The second mapping  $f_{QP}$ , determined as an inverse of  $f_{PQ}$ ,  $f_{QP} = f_{PQ}^{-1}$  is used for transformation (if necessary) of entries  $p_i$  outside the segment in  $\hat{P}$ . If an entry  $p_i$  does not belong to the definition domain of  $f_{PQ}$ , then the entry  $p_i$  remains unchanged. In the opposite case the entry  $p_i$  is changed by the last value of iteration scheme  $x_{k+1} = f_{QP}(x_k)$ , where  $x_0 = p_i$ , which still remains within the definition domain of mapping  $f_{QP}$ . The same approach is used for the repairing process of entries outside of the segment in  $\hat{Q}$ , but now the mapping  $f_{PQ}$  is employed.

For illustration, let us study the exchange operator, on the following two permutations

$$P = (3, 4, 5, 1, 2, 6, 10, 8, 9, 7)$$

$$Q = (1, 2, 6, 10, 8, 3, 4, 5, 7, 9)$$

The segment is determined by  $i_1 = 4$  and  $i_2 = 8$ ; then  $\hat{P}$  and  $\hat{Q}$  are formed from  $P$  and  $Q$  by exchanging the segments between them

$$\hat{P} = (3, 4, 5, 10, 8, 3, 4, 5, 9, 7)$$

$$\hat{Q} = (1, 2, 6, 1, 2, 6, 10, 8, 7, 9)$$

We see that  $\hat{P}$  and  $\hat{Q}$  are not genuine permutations, in both of them some integers appear more than once (e.g. 3, 4, 5 in  $\hat{P}$  and 1, 2, 6 in  $\hat{Q}$ ). In order to remove this drawback of  $\hat{P}$  and  $\hat{Q}$ , we have to construct the mappings  $f_{QP}$  and  $f_{PQ}$ , respectively

$$\begin{aligned} f_{QP} &= \begin{pmatrix} 10 & 8 & 3 & 4 & 5 \\ 1 & 2 & 6 & 10 & 8 \end{pmatrix}, \\ f_{PQ} &= \begin{pmatrix} 1 & 2 & 6 & 10 & 8 \\ 10 & 8 & 3 & 4 & 5 \end{pmatrix} \end{aligned}$$

For instance, the first three entries of  $\hat{P}$  should be repaired by the mapping  $f_{QP}$  as follows:  $3 \rightarrow 6$ ,  $4 \rightarrow 10 \rightarrow 1$ ,  $5 \rightarrow 8 \rightarrow 2$ . Then, the resulting permutation  $P'$  is

$$P' = (6, 1, 2, 10, 8, 3, 4, 5, 9, 7)$$

Similarly, the mapping  $f_{PQ}$  repairs the first three entries of  $\hat{Q}$ ,  $1 \rightarrow 10 \rightarrow 4$ ,  $2 \rightarrow 8 \rightarrow 5$ ,  $6 \rightarrow 3$ , so that we get

$$Q' = (4, 5, 3, 1, 2, 6, 10, 8, 7, 9)$$

The permutations  $P'$  and  $Q'$  represent the result of applying the exchange operator onto the permutations  $P$  and  $Q$ .

An algorithmic realization of the augmented simulated annealing may be done in a slightly modified version of the standard simulated annealing. Its implementation in the pseudo-Pascal form is presented as the following algorithm.

## Algorithm Augmented\_Simulated\_Annealing.

```

1  A pool of permutations  $P, Q, \dots, R$  is randomly generated;
2   $T := T_{\max}$ ;  $success := 1$ ;
3  WHILE ( $T > T_{\min}$ )  $\wedge$  ( $success > 0$ ) DO
4    BEGIN  $counter := 0$ ;  $success := 0$ ;
5    WHILE ( $counter < counter_{\max}$ )  $\wedge$  ( $success < success_{\max}$ ) DO
6      BEGIN  $counter := counter + 1$ ;
7        select randomly operator  $o$  and its parameters;
8        IF (operator  $o$  is exchange) THEN
9          BEGIN select randomly permutations  $P$  and  $Q$ ;
10             ( $P', Q'$ ) =  $o(P, Q)$ ;
11             END ELSE
12             BEGIN select randomly a permutation  $P$ ;
13                 $P' := o(P)$ ;
14             END;
15             IF  $CD(P') - CD(P) < 0$  THEN
16                $Pr := 1$  ELSE  $Pr := \exp(-(CD(P') - CD(P))/T)$ ;
17             IF  $random < Pr$  THEN
18               BEGIN  $P := P'$ ;  $success := success + 1$  END;
19             IF  $o$  is exchange operator THEN
20               BEGIN IF  $CD(Q') - CD(Q) < 0$  THEN
21                   $Pr := 1$  ELSE  $Pr := \exp(-(CD(Q') - CD(Q))/T)$ ;
22                  IF  $random < Pr$  THEN
23                    BEGIN  $Q := Q'$ ;  $success := success + 1$  END;
24                  END;
25             END;
26              $T := \alpha * T$ ;
27       END;
28     END;

```

In the first line a pool of randomly generated permutations is formed; its cardinality  $M$  (number of permutations) is kept fixed in the course of the whole algorithm. If the pool contains only single permutation, then the run of algorithm is fully equivalent to the algorithm of standard simulated annealing presented in the previous section. The upper bounds  $counter_{\max}$  and  $success_{\max}$  of integer variables  $counter$  and  $success$ , respectively, are determined by  $counter_{\max} = 100NM$  and  $success_{\max} = 10NM$ ; that is, the upper bounds from the algorithm Simulated Annealing are multiplied by the number of pool permutations. A probability of selection of the exchange operator (line 7) should be, say, one-tenth of the probabilities of other operators. In the opposite case, if the probability of application of the exchange operator is comparable with probabilities of other operators, the exchange operator strongly couples pool permutations such that all are quickly equalized with the resulting chemical distance often very far from the correct value.

## APPLICATIONS

The evaluation of chemical distance between molecular graphs composed of a moderate number of vertices – atoms (up to 10–12) by the (augmented) simulated-annealing algorithm is much slower than its evaluation by sophisticated combinatorial methods. In particular, the combinatorial method based on the process of looking for maximum cliques in the graph product of the corresponding line graphs<sup>20,25</sup> offers a very effective approach to evaluation of chemical distances between smaller molecular graphs. However, increasing the number of vertices to 20–30 makes this algorithm as blatantly ineffective as all other combinatorial algorithms for the evaluation of chemical distance. The combinatorial “barrier” stands in the way of applications for larger molecular graphs. Partial acceleration and sophistication of combinatorial algorithms could not surmount this barrier. It means that the main field of applications of noncombinatorial algorithms for the evaluation of chemical distance can be found in larger molecular graphs composed of a few tens of vertices.

The simulated-annealing algorithm, in standard as well as in augmented version, has been extensively tested by randomly generated molecular graphs (with prescribed numbers of atoms and bonds). For simplicity, the atoms – vertices are labeled by the same symbol (say C – carbon). If atoms are labeled by different symbols, then the simulated-annealing algorithm operators should be realized independently for different classes of atoms labeled by the same symbol. This immediately follows from the fact that when (1) is being optimized, only those permutations are permitted which map the vertices of the first graph onto vertices of the second graph such that the pairs of mapped vertices are labeled by the same symbol.

Since for greater graphs (composed of more than 20 vertices) the combinatorial methods are unable to give exact chemical distances in real CPU time, the second graph in our calculations has been created from the first graph by moving  $p$  randomly selected bonds – edges to unoccupied positions. Then, the chemical distance  $CD(G_1, G_2)$  is bound from above by<sup>4</sup>

$$CD(G_1, G_2) \leq 2p \quad (10)$$

where the relation “smaller than” is satisfied mainly if  $G_1$  or  $G_2$  has a nontrivial automorphism. This means that we are able simply to evaluate an exact upper bound of chemical distance  $CD(G_1, G_2)$  for each pair of randomly generated molecular graphs which differ in the positions of  $p$  bonds. In most cases, the actual distance between two graphs randomly generated by moving  $p$  edges to unoccupied positions will achieve the upper bound  $2p$ . Consequently, in the following performance runs of the algorithm, whenever the computed distance  $CD(P)$  is less than or equal to the upper bound of  $2p$ , we shall treat the computed distance as correct, even though in some cases the actual distance will be slightly less than the computed distance. Our numerical experiences may be summarized as follows:

The standard version of the simulated-annealing algorithm (one permutation is optimized) offers frequently suboptimal chemical distances which are greater than correct values. This happens more frequently for larger molecular graphs. For graphs composed of more than 15 vertices the suboptimal value is much more probable than the correct one; see Table I. Moreover, this fact of increasing value of the resulting chemical distances is strongly amplified by the number  $p$  of shifted edges, i.e. by the increased upper bound of the exact chemical distance. We have tried successfully to remove this serious shortcoming of the standard simulated annealing by its generalization to the augmented version, where a pool of permutations (composed usually of 10–20 permutations) is optimized. As a result, we have observed that the resulting chemical distances are correctly ranged by (10). Unfortunately, for molecular graphs composed of more than 30 vertices and mainly when the number of shifts  $p$  has been greater than 10 (i.e. the chemical distance is estimated by 20) the frequency of occurrence of suboptimal results is increasing, even for the augmented simulated annealing; see Table II. Fortunately, this undesirable feature of the augmented version does not show strongly for graphs with a smaller density of edges (connected molecular graphs containing few cycles) and the smaller number of shifts  $p$ , e.g.  $p \leq 7$ , which is just the case of real chemical molecular graphs when a reaction network is constructed.

The simulated-annealing chemical distances have been also compared with those obtained by the genetic algorithm; see Table III. To compare these two methods properly, we have carefully selected the probabilities of applications<sup>31</sup> of all five single operators as follows:  $P(\text{transposing}) = 0.01$ ,  $P(\text{reor-})$

**Table I.** Comparison of Chemical Distances Calculated by Standard and Augmented Simulated-Annealing Algorithms<sup>a</sup>

no. of vertices	no. of edges	no. of shifted edges	standard version	augmented version
15	15	0	0 (100)	0 (100)
	15	10	20 (90)	20 (100)
	20	0	0 (100)	0 (100)
	20	10	20 (90)	20 (100)
	25	0	0 (100)	0 (100)
	25	10	20 (90)	20 (100)
20	20	0	0 (100)	0 (100)
	20	10	20 (80)	20 (100)
	25	0	0 (100)	0 (100)
	25	10	20 (70)	20 (100)
	30	0	0 (100)	0 (100)
	30	10	20 (70)	20 (100)
25	25	0	0 (100)	0 (100)
	25	10	20 (70)	20 (100)
	30	0	0 (100)	0 (100)
	30	10	20 (60)	20 (100)
	35	0	0 (100)	0 (100)
	35	10	20 (60)	20 (100)
30	30	0	0 (100)	0 (100)
	30	10	20 (70)	20 (90)
	35	0	0 (100)	0 (100)
	35	10	20 (50)	20 (90)
	40	0	0 (100)	0 (100)
	40	10	20 (50)	20 (90)

<sup>a</sup> Numbers in parentheses correspond to the percentage from ten trials of the achievement of the chemical distance smaller than or equal to  $2p$  ( $p$  is the number of shifted edges).

**Table II.** Resulting Chemical Distances Calculated by the Augmented Simulated-Annealing Algorithm<sup>a</sup>

no. of vertices	no. of edges	chemical distance for given no. of shifted edges					
		0	1	2	3	4	5
10	10	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
	11	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
	12	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
	13	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
	14	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
	15	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (100)
20	20	0 (100)	2 (100)	4 (100)	6 (100)	8 (100)	10 (90)
	21	0 (100)	2 (100)	4 (100)	6 (90)	8 (90)	10 (80)
	22	0 (100)	2 (100)	4 (100)	6 (90)	8 (90)	10 (80)
	23	0 (100)	2 (90)	4 (90)	6 (80)	8 (80)	10 (80)
	24	0 (100)	2 (90)	4 (90)	6 (80)	8 (70)	10 (70)
	25	0 (100)	2 (90)	4 (80)	6 (80)	8 (70)	10 (60)
30	30	0 (100)	2 (100)	4 (100)	6 (90)	8 (90)	10 (90)
	31	0 (100)	2 (100)	4 (90)	6 (90)	8 (80)	10 (80)
	32	0 (100)	2 (90)	4 (90)	6 (90)	8 (80)	10 (70)
	33	0 (100)	2 (90)	4 (90)	6 (70)	8 (70)	10 (70)
	34	0 (100)	2 (80)	4 (80)	6 (70)	8 (70)	10 (60)
	35	0 (100)	2 (80)	4 (70)	6 (70)	8 (70)	10 (50)
40	40	0 (100)	2 (90)	4 (90)	6 (80)	8 (70)	10 (60)
	41	0 (100)	2 (80)	4 (80)	6 (70)	8 (70)	10 (60)
	42	0 (100)	2 (80)	4 (70)	6 (70)	8 (60)	10 (50)
	43	0 (90)	2 (80)	4 (70)	6 (60)	8 (50)	10 (50)
	44	0 (90)	2 (80)	4 (60)	6 (60)	8 (50)	10 (50)
	45	0 (90)	2 (80)	4 (60)	6 (50)	8 (50)	10 (40)

<sup>a</sup> See footnote in Table I.

dering) = 0.05,  $P(\text{transporting}) = 0.01$ ,  $P(\text{reversing}) = 0.01$ , and  $P(\text{exchange}) = 0.7$ . The population of chromosomes (pool of permutations) is composed of  $N = 400$  permutations. The strategy of the genetic algorithm has been organized in a way described by Goldberg in his textbook.<sup>31</sup> A new pool of permutations is formed from the previous one such that for all  $N/2$  pairs of randomly selected permutations by the "roulette wheel" method<sup>31</sup> the crossover operator is applied and the resulting permutations are undertaken transformations

**Table III.** Chemical Distances Produced by Genetical Algorithm<sup>a</sup>

no. of vertices	no. of edges	chemical distance for given no. of shifted edges					
		0	1	2	3	4	5
10	10	0 (100)	2 (90)	4 (90)	6 (80)	8 (70)	10 (60)
	11	0 (100)	2 (90)	4 (80)	6 (80)	8 (60)	10 (60)
	12	0 (90)	2 (90)	4 (80)	6 (60)	8 (60)	10 (50)
	13	0 (90)	2 (80)	4 (80)	6 (60)	8 (50)	10 (50)
	14	0 (80)	2 (80)	4 (70)	6 (60)	8 (50)	10 (50)
	15	0 (70)	2 (70)	4 (60)	6 (60)	8 (50)	10 (40)
20	20	0 (70)	2 (70)	4 (60)	6 (50)	8 (40)	10 (40)
	21	0 (70)	2 (60)	4 (60)	6 (50)	8 (40)	10 (40)
	22	0 (70)	2 (60)	4 (50)	6 (40)	8 (40)	10 (40)
	23	0 (70)	2 (50)	4 (40)	6 (40)	8 (40)	10 (30)
	24	0 (60)	2 (40)	4 (40)	6 (40)	8 (30)	10 (30)
	25	0 (60)	2 (40)	4 (40)	6 (30)	8 (30)	10 (30)
30	30	0 (50)	2 (50)	4 (50)	6 (40)	8 (30)	10 (30)
	31	0 (50)	2 (50)	4 (40)	6 (40)	8 (30)	10 (30)
	32	0 (50)	2 (40)	4 (40)	6 (30)	8 (30)	10 (30)
	33	0 (50)	2 (40)	4 (30)	6 (30)	8 (30)	10 (30)
	34	0 (40)	2 (40)	4 (30)	6 (20)	8 (20)	10 (20)
	35	0 (40)	2 (30)	4 (30)	6 (20)	8 (20)	10 (20)
40	40	0 (50)	2 (50)	4 (40)	6 (30)	8 (30)	10 (30)
	41	0 (50)	2 (40)	4 (40)	6 (30)	8 (30)	10 (30)
	42	0 (40)	2 (40)	4 (30)	6 (30)	8 (30)	10 (20)
	43	0 (40)	2 (30)	4 (30)	6 (20)	8 (20)	10 (20)
	44	0 (30)	2 (20)	4 (30)	6 (20)	8 (20)	10 (10)
	45	0 (20)	2 (20)	4 (20)	6 (20)	8 (20)	10 (10)

<sup>a</sup> See footnote in Table I.

by randomly selected single-permutation operators. The results of Table III are for the larger graphs plagued similarly as for the standard version of simulated annealing, by the relatively high frequency of occurrence of suboptimal values of chemical distances. We have tried to rectify this fact by increasing the pool size (number of pool permutations) up to 1000, but without a positive result.

In summary, the method of simulated annealing represents in its augmented version a useful algorithm for fast evaluation of chemical distances between relatively great molecular graphs (up to 30–40 atoms). For graphs which have a relatively small density of edges (containing few cycles) and which are not very dissimilar from the standpoint of chemical metrics (with the number of edge shifts going from 0 to 5), it provides chemical distances that are ranged by (10); i.e. they are closely related to their exact values. According to our numerical experiences, when the results obtained by simulated annealing and genetical algorithm have been compared, an occurrence of incorrect chemical distances (i.e. those ones that are not ranged by the inequality (10)) is of lower frequency for the augmented simulated annealing than for the genetic algorithm.

## DISCUSSION

In general, application of the simulated annealing (similarly for the genetic algorithm) to the evaluation of chemical distances between molecular graphs is not a standard numerical task. This method contains inherently some parameters (e.g. starting temperature of annealing, probabilities of application of operators, pool size, etc.) that may very dramatically change the resulting value of chemical distance depending on the size of the molecular graph, the density of edges, and an exact value of chemical distance. Therefore, when we are going to apply the simulated annealing for the evaluation of chemical distances, it is necessary to perform some experimentation with proper adjusting of its parameters to the actual problem. Graph-theoretically, the simulated annealing may be successfully used not only for the evaluation of chemical distances



between relatively great graphs but also as a noncombinatorial algorithm for finding isomorphism (then chemical distance is equal to zero), common subgraphs, and many other problems. In the last paragraph of the previous section a comparison between simulated annealing and genetical algorithm has been presented. It does not mean that the simulated annealing is definitely more advantageous than the genetic algorithm. Since both algorithms on first sight are very similar (but they also contain concepts that could not be simply related), a comparable numerical efficiency might be expected provided proper selection of their basic parameters was made.

## REFERENCES AND NOTES

- (1) (a) Dugundji, J.; Ugi, I. An Algebraic Model of Constitutional Chemistry as a Basis for Chemical Computer Programs. *Top. Curr. Chem.* **1973**, *39*, 19–64. (b) Jochum, C.; Gasteiger, J.; Ugi, I.; Dugundji, J. The Principle of Minimal Chemical Distance and the Principle of Minimal Structural Change. *Z. Naturforsch.* **1982**, *37B*, 1205–1215. (c) Ugi, I.; Wochner, M.; Fontain, E.; Bauer, J.; Gruber, B.; Karl, R. Chemical Similarity, Chemical Distance, and Computer Assisted Formalized Reasoning by Analogy. In *Concepts and Applications of Molecular Similarity*; Johnson, M. A., Maggiora, G. M., Eds.; Wiley: New York, 1990; pp 239–288.
- (2) Baláz, V.; Koča, J.; Kvasnička, V.; Sekanina, M. A Metric for Graphs. *Cas. Pěst. Mat.* **1986**, *111*, 431–433.
- (3) Koča, J.; Kratochvíl, M.; Kvasnička, V.; Matyska, L.; Pospíchal, J. *A Synthon Model of Organic Chemistry and Synthesis Design*. Lecture Notes in Chemistry 51; Springer Verlag: Berlin, 1989.
- (4) Baláz, V.; Kvasnička, V.; Pospíchal, J. Dual Approach for Edge Distance Between Graphs. *Cas. Pěst. Mat.* **1989**, *114*, 155–159.
- (5) Kvasnička, V.; Pospíchal, J. Two Metrics for a Graph-theoretical Model of Organic Chemistry. *J. Math. Chem.* **1989**, *3*, 161–191.
- (6) Kvasnička, V.; Pospíchal, J. Chemical and Reaction Metrics for Graph-theoretical Model of Organic Chemistry. *J. Mol. Struct. (THEOCHEM)* **1991**, *227*, 17–41.
- (7) Pospíchal, J.; Kvasnička, V.; Gutman, I. Chemical Distance for Molecules Whose Classical Structural Formula is not Unique. *Collect. Sci. Pap. Faculty Sci. Kragujevac* **1991**, *12*, 109–125.
- (8) Kvasnička, V.; Pospíchal, J.; Baláz, V. Reaction and Chemical Distances and Reaction Graphs. *Theor. Chim. Acta* **1991**, *79*, 65–79.
- (9) Baláz, V.; Kvasnička, V.; Pospíchal, J. Two Metrics in a Graph Theory Modeling of Organic Chemistry. *Discrete Appl. Math.* **1992**, *35*, 1–19.
- (10) Johnson, M. A. Relating Metrics, Lines and Variables Defined on the Space of Graphs. In *Graph Theory and its Applications to Algorithms and Computer Science*; Alavi, Y., Chartrand, G., Lesniak, L., Wall, C., Eds.; Wiley: New York, 1985; pp 457–470.
- (11) Balaban, A. T., Ed. *Chemical Applications of Graph Theory*; Academic Press: London, 1976.
- (12) Gutman, I.; Polanski, O. *Mathematical Concepts in Organic Chemistry*; Springer: Berlin, 1986.
- (13) King, R. B., Ed. *Chemical Applications of Topology and Graph Theory*; Elsevier: Amsterdam, 1983.
- (14) Trinajstić, N. *Chemical Graph Theory*; CRC Press: Boca Raton, FL, 1983.
- (15) Trinajstić, N., Ed. *Mathematics and Computational Concepts in Chemistry*; Ellis Horwood: Chichester, U.K., 1986.
- (16) Fontain, E.; Bauer, J.; Ugi, I. Computer-Assisted Bilateral Generation of Reaction Networks from Educs and Products. *Chem. Lett.* **1987**, 37–40.
- (17) Fontain, E.; Reitsam, K. The Generation of Reaction Networks with Rain. 1. The Reaction Generator. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 96–101.
- (18) Kvasnička, V.; Pospíchal, J. Graph-theoretical Interpretation of Ugi's Concept of the Reaction Network. *J. Math. Chem.* **1990**, *5*, 309–322.
- (19) Pospíchal, J.; Kvasnička, V. Reaction Graphs and a Construction of Reaction Networks. *Theor. Chim. Acta* **1990**, *76*, 423–435.
- (20) Kvasnička, V.; Pospíchal, J. Maximal Common Subgraphs of Molecular Graphs. *Rep. Mol. Theory* **1990**, *1*, 99–106.
- (21) Garey, M. R.; Johnson, D. S. *Computers and Intractability—A Guide to the Theory of NP-Completeness*; Freeman: San Francisco, 1979.
- (22) Lawler, E. L. Branch and Bound Method. *J. Oper. Res. Soc. Am.* **1966**, *14*, 217–235.
- (23) McGregor, J. J. Backtrack Search Algorithm and the Maximal Common Subgraph Problem. *Software Pract. Exp.* **1982**, *12*, 23–34.
- (24) Harary, F. *Graph Theory*; Addison-Wesley: Reading, MA, 1969.
- (25) Tsai, C.-C.; Nicholson, V.; Johnson, M.; Naim, M. A Subgraph Isomorphism Theorem for Molecular Graphs. *Graph Theory and Topology in Chemistry*; Elsevier: Amsterdam, 1987; pp 226–230.
- (26) Jochum, C.; Gasteiger, J.; Ugi, I. The Principle of Minimum Chemical Distance (PMCD). *Angew. Chem., Int. Ed. Engl.* **1980**, *19*, 495–505.
- (27) Wochner, M.; Brandt, J.; von Scholey, A.; Ugi, I. Chemical Similarity, Chemical Distance and Its Exact Determination. *Chimia* **1988**, *42*, 217–225.
- (28) Ugi, I.; Fontain, E.; Bauer, J. Transparent Formal Methods for Reducing the Combinatorial Wealth of Conceivable Solutions to a Chemical Problem—Computer Assisted Elucidation of Complex Reaction Mechanisms. *Anal. Chim. Acta* **1990**, *235*, 155–161.
- (29) Fontain, E. The Problem of Atom-to-atom Mapping. An Application of Genetic Algorithms. *Anal. Chim. Acta* **1992**, *265*, 227–232.
- (30) Fontain, E. Application of Genetic Algorithms in the Field of Constitutional Similarity. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 748–752.
- (31) Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Reading, MA, 1989.
- (32) (a) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equation of State Calculations for Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. (b) Kirkpatrick, S.; Gelatt, C. D., Jr.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. (c) Cerný, V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *J. Opt. Theory Appl.* **1985**, *45*, 41–51.
- (33) van Laarhoven, P. M. J.; Aarts, E. H. L. *Simulated Annealing: Theory and Applications*. Reidel: Dordrecht, The Netherlands, 1987.
- (34) Lin, S.; Kernighan, B. W. An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Oper. Res.* **1973**, *21*, 498–516.
- (35) Snow, M. E. Powerful Simulated Annealing Algorithm Locates Global Minimum of Protein-Folding Potentials from Multiple Starting Conformations. *J. Comput. Chem.* **1992**, *13*, 597–584.
- (36) Goldberg, D. E.; Lingle, R. Alleles, Loci and the Traveling Salesman Problem. *Proceedings of an International Conference on Genetic Algorithms and Their Application*; Grefenstette, J. J., Ed.; Carnegie-Mellon University Press: Pittsburgh, PA, 1985; pp 154–159.