A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules

Nathan Brown,*,† Ben McKay,† François Gilardoni,† and Johann Gasteiger‡

Avantium Technologies B.V., P.O. Box 2915, 1000 CX, Amsterdam, The Netherlands, and
Computer-Chemie-Centrum and the Institute for Organic Chemistry, University of Erlangen-Nürnberg,
Nägelsbachstrasse 25, D-91052, Erlangen, Germany

Received December 15, 2003

In this paper we propose a novel graph-based genetic algorithm for the evolution of novel molecular graphs from a predefined set of elements or molecular fragments with an external objective function. A brief overview of existing genetic algorithm approaches in molecular design is provided followed by a description of our approach. The paper continues to suggest a novel application of this program to the multiobjective evolution of median molecules that are structurally representative of a set of objective molecules. We conclude with a summary of our initial results along with a discussion of a variety of improvements and applications of our approach.

1. INTRODUCTION

Computer-aided molecular design (CAMD) has been an active area of research for a number of years[1,2] with a substantial amount of this research being directed at evolving novel structures (de novo design) and often-applying genetic search procedures.[3−6] The most common approaches in this area of CAMD research are the development of fragment-positioning[7,8] and molecular growth[9,10] methods for the design of ligand candidates, although the constraint of the size and structure of the ligands that are evolved significantly reduces the search space of these problems. However, the research conducted in the area of the more general de novo design of molecules from elements or structural fragments using genetic algorithms (GAs) is less well-defined and yet very important in instances when the search for a target molecule is not constrained by a binding pocket, which is typically the case for catalyst, reagent, and additive optimization in chemical process development.

A new approach of evolving novel molecular graphs by perturbing the graph-based chromosomes directly using both existing and novel graph-based genetic operators is presented in this paper. An overview of the algorithm and its data structures is provided, followed by a more substantial description of the genetic mutation and crossover operators that have been developed.

An application of our GA is then proposed, that of evolving median molecules from a set of objective molecules by applying multiobjective optimization techniques followed by an overview of our initial experiments and results. In this context, median molecules are the set of ‘in-between structures’ that are designed such that they exhibit characteristics of the objective molecules to some measurable degree. The paper concludes with a discussion of improvements to our current software and some possible applications of our approach in chemoinformatics.

2. GENETIC ALGORITHMS AND COMPUTER-AIDED MOLECULAR DESIGN

Genetic algorithms (GAs) are applied widely in discovering globally optimal solutions to optimization problem instances and particularly to problems where no efficient deterministic algorithm is available.[11] The simple GA operates on binary strings, which encode candidate solutions in the search space of interest and perturbs these strings with computational analogues of natural recombination and mutation. Many different configurations of the GA have been applied to solving problems in the field of chemoinformatics (see ref 12 for a review).

The genetic programming (GP) algorithm is similar in concept to the GA approach; however, the chromosomes are represented as trees rather than the fixed-length strings of the simple GA.[13] The tree representation of GP permits the chromosomes to be both extensible and contractible, through crossover and mutation, a characteristic that is not present in the standard GA, although approaches have been suggested to achieve this.[14,15]

The tree-based representation of GP is the most-often applied technique for evolving molecular graphs, with two particular approaches being apparent in the method of encoding molecular structures as trees. The first of these approaches generalizes molecular fragments as the set of allele values that genes may take. This generalization obviates the need for complex crossover operators and chromosome repair strategies since cycles are collapsed into single gene nodes in a similar approach to the reduced graph[16] and feature tree[17] techniques. The second approach of representing cyclic graphs as trees uses a special leaf node that points to another node in the graph, basically a hyperlink node; therefore, all of the structural information is preserved in the tree allowing the molecule to be expressed as a graph.

Venkatasubramanian et al.[18] applied the first type of GP approach in evolving novel polymeric structures from a set of molecular fragments. The reported experiments indicate that this approach is very effective at evolving solutions; however, the chromosome representation in this work is effectively string-based or at the very most trees with limited

* Corresponding author phone: +31-20-586-8019; e-mail: nathan.brown@avantium.com.

† Avantium Technologies B.V.

‡ University of Erlangen-Nürnberg.

10.1021/ci034290p CCC: $27.50 © 2004 American Chemical Society
Published on Web 05/05/2004

branching. The same type of approach has been reported in a number of papers for a similar application.[19,20]

Nachbar[21] applied the second type of GP encoding strategy to the design of novel molecular graphs by encoding the topological structure of molecules as trees. The crossover operator was constrained not to make or break cycles; special mutation operators were defined to control this. This means that any node or edge that is part of a cycle cannot be exchanged in part between chromosomes, considerably restricting the operator. Additionally, hydrogen atoms are explicitly represented as leaf nodes within the tree. Although this representation allows cyclic graphs to be properly encoded as tree-based chromosomes, the position of the nodes that encode the cycles appears somewhat arbitrary and could easily suffer from side effects from the application of genetic operators. The paper reports that the algorithm has been applied in evolving the average chemical structure of two molecules from their average descriptor vector with the intention of generating a structure that has similar biological activity to both structures. This is somewhat similar to the application reported in section 6 of this paper to evolving sets of median molecules.

Although both tree-based representations have been demonstrated to be effective at evolving molecular graphs, they are often limited to either evolving relatively simple types of molecules or, for cyclic structures, employing complex and potentially disruptive genetic operators and repair strategies. It is evident that the molecular graph (or fragment graph) itself can be applied directly as the genotype in a GA approach, although new genetic operators would be required to perturb these types of chromosomes. However, even though the graph itself is intuitively suitable as a chromosome representation, we are aware of only two reported implementations of this method.

The patent held by Weininger[22] describes a method of evolving molecules using a genetic search technique. The crossover operator of this approach takes two parents and generates a single child chromosome. However, it has been noted[23] that the crossover operator can result in disconnected graphs, although only in situations where the fitness function can be calculated from disconnected structures. In the crossover operator reported by Weininger, bonds are removed from the parent molecules according to a ‘digestion rate’, and the resulting fragments are then copied into the child chromosomes according to a ‘dominance rate’. The method was reported to be effective at evolving to a given target molecule and for application to novel ligand design using CoMFA (Comparative Molecular Field Analysis).

Globus et al.[23] proposed a graph-based GA to evolve molecular graphs from individual elements. The crossover operator devised by Globus et al. was shown to be very effective at exchanging genetic material between chromosome graphs with relatively minimal disruption to the genetic material, and we have adapted this type of crossover operator for our program—a detailed description of which is provided in section 5.1 of this paper.

The encoding of molecular graphs as trees is fraught with issues, requiring either generalized molecular fragments or special node types to implicitly encode cyclic structures. The former approach requires a fragment library to be defined, which may not be possible in all situations. The latter representation complicates crossover operations since apparently simple genetic exchanges will tend to have side effects as a result of gene adjacency not being preserved in the tree. The encoding of any cyclic graphs as trees using this approach will always result in nodes, which are adjacent in the graph, not being adjacent in the tree.

The chromosome representations applied by Weininger and Globus et al. are perhaps the most flexible and powerful approaches yet reported in the evolution of novel graph structures since both algorithms operate directly on the graphs. We have developed a new program, Compound Generator (CoG), which uses an evolutionary algorithm approach, but where the chromosomes are themselves graphs, with no constraint on the size and complexity of the graphs other than those of the normal valence bond model. The program also represents the molecules internally as metagraphs where the nodes may simply be single atoms or existing substructures of interest such as functional groups or common scaffolds. This approach permits the focus of the evolution to be more tightly controlled than is the case where atoms map directly to nodes. Additionally, we have designed novel genetic crossover and mutation operators that perform on these graphs that allow the genetic search process a larger degree of freedom in perturbing the structures. The following sections provide a detailed overview of our algorithm and, in particular, its genetic operators.

3. A GRAPH-BASED GENETIC ALGORITHM FOR EVOLVING MOLECULES

Topological molecular graphs are simply a collection of atoms (abstracted as nodes) and the bonds (abstracted as edges) that specify the relationships between those atoms. However, it is a simple process to identify molecular fragments that tend to occur frequently in molecules or in specific data sets, such as functional groups, ring systems, and common scaffolds. Therefore, our program has been designed to build molecules from both simple atoms and also from defined substructures, allowing the user to relax or constrain the search space, as required by specific problem instances. CoG accepts an input library of molecular fragments, which are the set of possible allele values that the nodes of the chromosomes may exhibit. These structures can be a collection of simple atoms or complex substructures, providing that free valences are available to allow connection to other fragments. The library of permitted molecular fragments is passed to CoG at runtime in the Tripos Mol2 file format.[24] The available valences of the fragments are perceived by one of three methods: hydrogen-depletion, R-group perception, or simply from a list of standard valences for each atom specified in an external plain text file.

Since CoG allows fragments to be defined as the genes of the chromosomes, it is possible that two different atoms in one fragment can be connected to two different atoms in another fragment, respectively. This means that the CoG chromosomes can be explicit instances of multigraphs where multiple edges are incident with the same nodes.[25] Chemical structures are instances of implicit multigraphs in that a double bond is generalized as a single edge. Therefore, the chromosomes in this program are not molecular graphs per say, rather they are meta-graphs that describe the relationships between a set of subgraphs (Figure 1). However, in situations where the structural fragments are simply individual elements, this distinction is practically irrelevant.

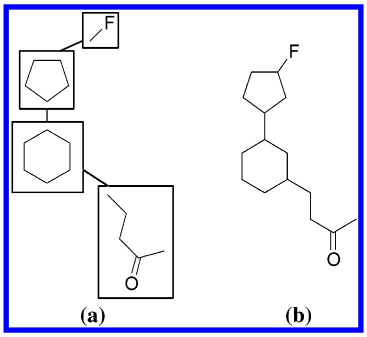

Figure 1. The structure encoding method used within CoG for (a) the chromosome and (b) the expressed molecule.

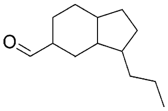

Figure 2. The source molecule used to demonstrate node and edge mutation operators in Figures 3 and 4, respectively.

The flexibility in the types of building blocks that may be used by CoG allows the user to determine for themselves whether they limit their search space to a set of molecular fragments or provide a set of atoms resulting in not only a much larger search space but also much more freedom in the exploration of chemical space.

4. GRAPH-BASED MUTATION OPERATORS

In the simple GA, mutation is achieved by randomly flipping the bits in a population of chromosomes according to some, generally low, probability. In the context of the graph-based GA this is analogous to swapping an existing fragment node with a new fragment node, such that the connector profile of the new fragment is compatible with the existing one; the connector profile is simply a vector of required free valences. However, the evolution of graphs is concerned with determining not only a suitable element or fragment type for a node but also the connecting edge types, quantities, and connecting position in the case of multiatom fragments. Therefore, to allow the GA the ability to extensively manipulate the graph-based chromosomes, it is essential to define a substantially richer and more complex set of mutation operators. Here, we consider the sets of node and edge mutation separately.

4.1. Node Mutation. An important requirement for evolving novel molecules is the ability for the chromosome representations to vary in size, in terms of both nodes and edges, during the evolution process. Four approaches to changing the size of a graph in terms of nodes through mutation have been implemented in our program: append, prune, insert, and delete. All of the node mutation operators defined here are illustrated in Figure 3 perturbing the molecule given in Figure 2. The first two mutation operators are simple in that a single leaf node and its incident edge is either added or removed, respectively, from the graph being mutated. In the case of appending a new node, a source and target connector atom is selected at random from each set of connector atoms within the nodes that have available valences. The type of the connecting edge is determined randomly from those that are valid for the source and target connector atoms.

The latter two operators are somewhat more complex but obviously related to the former two operators, respectively.

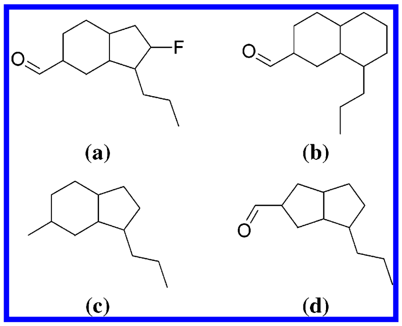

Figure 3. Examples of the node mutation operators (a) append, (b) insert, (c) prune, and (d) delete being performed on the source molecule in Figure 2.

The insert mutation operator selects a single edge in the graph and cuts it in half, resulting in two hanging edges, referred to as the hanging edge set. A list of candidate fragments that can accommodate these hanging edges is then generated from the list of all fragments in the library. Finally, the two hanging edges are connected to the connector atoms within the new node. In situations where there are no candidate fragments that fit the current hanging edge set, the hanging edges are gradually reduced in valence until there is at least one fragment in the candidate fragment list. Should this not be possible, the process restarts by considering an alternative chromosome edge.

The delete mutation operator identifies a single node at random and deletes it leaving a number of hanging edges, which can be anything from one edge to the total number of edges in the chromosome. An attempt is then made to reconnect the nodes with hanging edges with the intention of minimizing the disruption to the chromosome. The hanging edges are sorted in order of valence with successive pairs being reconnected in the chromosome; one edge is deleted, while the other is reconnected to it at that deletion point. In situations where the two edges are of different valence, one of the bond types is selected at random that allows reconnection. If a single edge remains in the list after this process, it is randomly either reconnected to any other existing node or deleted from the chromosome.

The final node mutation operator randomly selects a node to mutate and generates its connector profile. The list of available fragments that satisfy this connector profile is then calculated, with one of those fragments selected at random. The selected fragment then replaces the existing fragment, reconnecting the hanging edges as required.

The mutation operators described above can be summarized as being for both nodes and edges as one of a set of three perturbation effects: addition, deletion, and alteration. It is necessary in all cases to ensure that the mutated chromosome is a connected graph, which is achieved at random should any of the above mutations result in disconnection.

4.2. Edge Mutation. The second general class of graph mutation operator can be defined as those that, in some way, perturb the set of edges of a chromosome. The edge mutation operators defined in this graph-based GA are as follows: add, delete, and substitution. These edge mutation operators are shown in Figure 4 perturbing the molecule in Figure 2. By allowing the GA to add new edges to the chromosomes it is possible for cyclic structures to be generated through mutation. The type of a new edge is selected randomly from a list of edges that satisfy the free valences of the source and

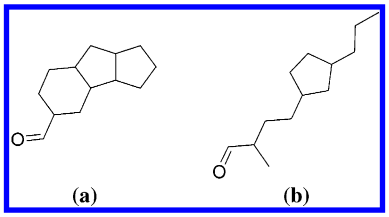

Figure 4. Examples of the edge mutation operators (a) add and (b) delete being performed on the source molecule in Figure 2.

target connector atoms, also selected at random, within the nodes.

A new edge can be added to an existing graph so long as available valences within the fragment nodes permit this. Similarly, when selecting an edge to delete, this can only be accomplished when there are at least as many edges as nodes and only from the subset of edges that, once removed, will result in a connected graph. Last, the substitution mutation operator replaces an existing edge with a different edge type within the constraints of the available valences in the fragment nodes.

The mutation operators defined here are by no means an exhaustive list of perturbations it is possible to make to a particular graph, but they do represent the typical requirements in this problem domain.

5. GRAPH-BASED CROSSOVER OPERATORS

The definition of the crossover operator is perhaps the most important component of any GA. When considering string-based chromosomes, it is trivial to define a common point on both chromosomes around which to exchange the genetic material of the parents—although repair procedures are often required in certain representations and issues regarding gene adjacency when expressed as the phenotype also require consideration. Similarly, when considering tree chromosomes, it is trivial to select subtrees to exchange between parents since it is a property of trees that they can be split into two subgraphs by the removal of any edge. In graphs with cycles, however, the removal of a single edge will not necessarily result in two disconnected graphs, significantly complicating the process since it is difficult to determine what approach should be used to disconnect the graphs. Additionally, the parent chromosomes will often be very different in size and connectivity. These issues result in it being quite difficult to define a reliable and general-purpose method of exchanging the genetic information and reconnecting the subgraphs.

5.1. Multipoint Crossover. The graph-based equivalent of the multipoint crossover is defined here as the removal of edges from each parent graph until each graph consists of two disconnected subgraphs but employing a strategy of edge removal that attempts to minimize disruption to the overall topology of the graph. This approach is essentially the one proposed by Globus et al.[23] At least one of the subgraphs of each parent is then exchanged between the parents, resulting in two intermediate child chromosomes that contain aspects of both parents but are disconnected. The intermediate child chromosomes are reconnected using the information that has been retained regarding where edges were present prior to the disconnection process performed on the parents. If, after exhausting the deleted edges, either of the child chromosomes are still disconnected, they are reconnected at random by adding new edges as required. Figure 5 illustrates a typical multipoint crossover operation.

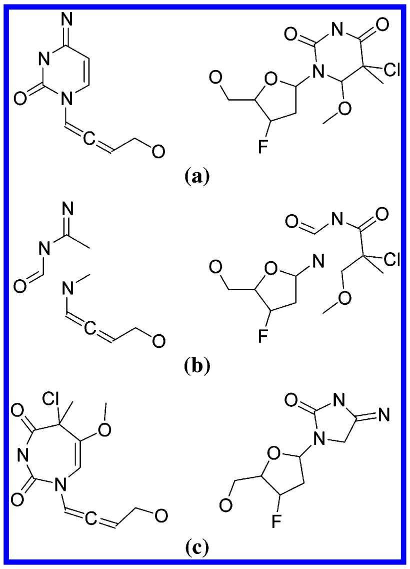

Figure 5. Example of the multipoint crossover with (a) the parent molecules, (b) the disconnected substructures of each parent, and (c) the exchanged and reconnected child molecules.

The approach of disconnecting a molecule into two subgraphs, reported by Globus et al., proceeds by the deletion of a single edge from the graph at random, recording the incident nodes of that edge and then continuing to delete edges at random from the list of edges that form the shortest path between the initial incident nodes until the graph is disconnected. This approach will break cycles whenever an edge is selected that is a member of a cycle; in extreme cases the process could lead to all cycles being broken. Additionally, isolated nodes can occur quite frequently if applying this algorithm as above; although this can be avoided by not allowing edges that are incident with the original nodes to be removed unless they are the only edges that can be considered. A limitation of the approach is that there is no apparent method of controlling the resultant size of the child chromosomes since it is difficult to control the disconnection of the parents and, therefore, the sizes of the subgraphs.

5.2. Subgraph Crossover. This operator is based on existing work in the pattern recognition field,[26] which was subsequently adapted for use as a crossover operator in a molecular graph matching GA.[27] Here, the operator has been further adapted for application to graph-based chromosomes. A connected subgraph is induced, but not removed, at random from each parent chromosome. The two subgraphs are then recombined using the same approach as for the multipoint crossover in an attempt to retain the general topology of the two subgraphs. This process is then repeated for the second child by selecting different edge-induced subgraphs. The selection of an edge-induced subgraph begins by selecting a single edge at random from the graph and storing this edge in the list of selected edges. The next edge is selected from the list of edges that are incident with the current edge and stored in the list of selected edges. The process continues by selecting edges randomly from the list of edges that are incident with any edge in the selected edge list that has not already been selected until an arbitrary termination condition is met, typically this is once approximately half of the overall graph has been induced. A

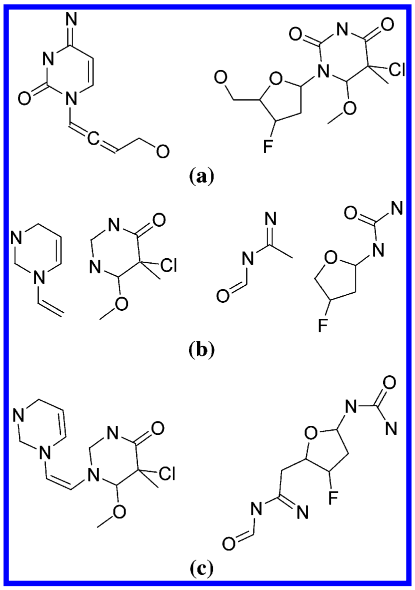

Figure 6. Example of the subgraph crossover with (a) the parent molecules, (b) the disconnected substructures of each parent, and (c) the exchanged and reconnected child molecules.

typical example of a subgraph crossover operation is shown in Figure 6.

One beneficial aspect of the graph-based chromosome representation is that the relative positions of the subgraphs do not need to be perceived in order to create a valid chromosome. There is no need, as with binary string chromosomes, to split each chromosome into two distinct halves, rather, the approach for graph-based chromosomes can take random subgraphs from each parent and recombine the subgraphs using the information that is known concerning how they were connected in the parents.

In standard binary string chromosomes, both the length of the chromosomes and the connectivity of the genes are fixed, whereas in graph-based chromosomes, the number of genes, the allele values of the genes, and the adjacency of the genes are all being evolved.

The crossover operators described above provide a powerful and varied set of methods for manipulating the genetic material that is within the chromosome graphs while also, in large part, minimizing the disruption that will tend to be introduced by such procedures.

6. MULTIOBJECTIVE EVOLUTION OF GENERALIZED MEDIAN MOLECULES

6.1. Multiobjective Evolution of Median Molecules. The concept of a median graph was first proposed in the graph theory field by Jiang et al.[28] as a method of generating a single graph that is, to some measurable degree, representative of a given set of graphs. Although the concept is trivial, the automated generation of a graph with these properties is nontrivial.

Jiang et al. applied a string-based GA to evolve a median graph by minimizing the total number of graph-edits that are required to change the evolved graph to each of the objective graphs; the graph-edit distance.[29] The fitness of each chromosome can then be calculated as the sum of the graph-edit distances from the objective graphs. Therefore, the intention is to evolve a graph, which is as close to each of the objectives as possible.

Nachbar[21] proposed a somewhat similar approach to the median graph, that of evolving an average chemical structure of two objective structures. Rather than using the sum of edit distances approach of Jiang et al., Nachbar evolved toward an average descriptor vector, calculated from the descriptor vectors of the objective structures. However, the average of the descriptor vectors, as Nachbar notes, does not necessarily have a representative point in chemical space. This is particularly apparent when the average descriptor vector, generated from two integer vectors, results in a real-valued vector, since the structure will be impossible to realize since there is not physical embodiment of a real value when considering integer descriptors such as counts or bit strings.

The sum of edit distances approach applied by Jiang et al. has limitations in that the resultant graphs can often be dominated by one of the objectives, thereby resulting in a single solution that will often be more representative of one of the objectives than the other. The approach of Nachbar, however, assumes that a structure exists for a calculated point in the descriptor space and if even if it does exist represents an average of the objectives. Therefore, to avoid the limitations that are inherent in both these approaches for evolving representative graphs, we employ a multiobjective optimization approach to evolve sets of nondominated solutions. To differentiate between the median graphs and average chemical structures proposed by Jiang et al. and Nachbar, respectively, we refer to our evolved structures as median molecules—since they are more similar in concept to the median graph.

6.2. Multiobjective Optimization. The multiobjective GA (MOGA) applies a Pareto ranking scheme to the evolution of candidate solutions that are not dominated in all objectives by any other chromosome in a population.[30] Essentially, points are Pareto ranked according to the number of solutions that dominate them in all objectives being considered. The Pareto ranking method has been applied to a number of multiobjective optimization problems in chemoinformatics with significant success.[31−33] An inherent characteristic of the Pareto ranking scheme is that it will tend to evolve a set of equally valid solutions, the nondominated set, which is particularly beneficial since one run of the program can provide multiple solutions of interest.

Here, the median molecule concept is applied to the generation of a set of median molecules, which are representative of a given set of input molecular graphs. In these experiments CoG is applied as the GA engine, while the molecular similarities are calculated with a molecular fingerprint generation and analysis program developed internally, called Fingal. A feature was added to Fingal to automatically calculate the Tanimoto similarity of each candidate molecule for a number of objective molecules and then calculate the Pareto ranking which is then passed as input to CoG for evolving the next generation of median molecules.

6.3. Evolving Median Molecules from Two Objectives. To test the effectiveness of the CoG program in evolving median molecules that are in some way representative of the objective molecules, our initial experiments reported here evolve novel structures with two similar and two diverse objective molecules, respectively, selected from the NCI AIDS database.[34] The two pairs of objective molecules were selected from the 11-member subset of confirmed active pyrimidine nucleosides. The first two molecules (Figure 7)

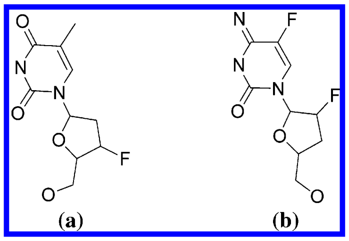

Figure 7. Two similar molecules used as objectives in the first experiment selected from the NCI AIDS pyrimidine nucleosides subset.

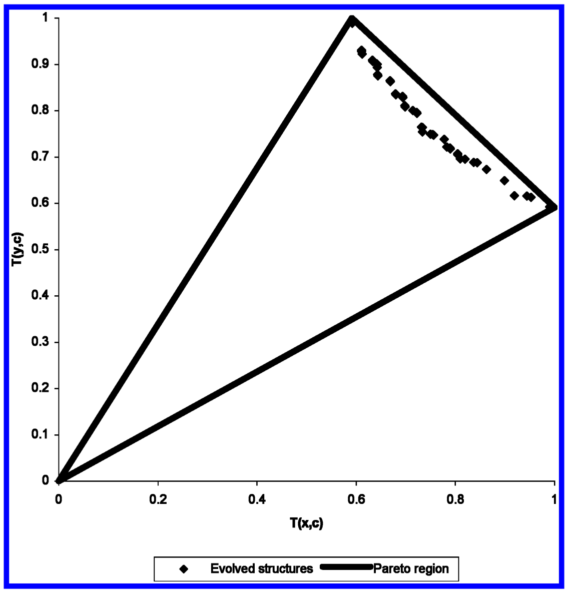

Figure 8. The evolved nondominated structures (*c*) plotted against the similarities to the (*x*) first and (*y*) second similar objectives of the first experiment along with the theoretical Pareto region.

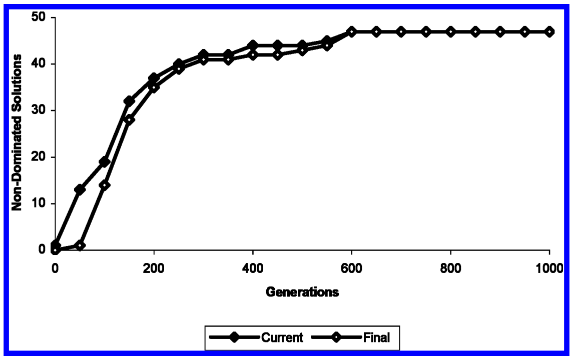

Figure 9. The size of the current Pareto frontier at 50-generation intervals and the proportion of each population that is also in the final Pareto frontier after 1000 generations from the first experiment.

were selected to be very similar to each other in terms of the fingerprints generated by Fingal, while the second pair of molecules (Figure 11) was selected to be structurally diverse. The Tanimoto similarities from the Fingal fingerprints for the similar and diverse pairs of molecules are 0.591 and 0.087, respectively.

The CoG program was then used to evolve novel molecules by maximizing the similarity of the evolved molecules to the objective molecules of each subset and applying the Pareto ranking method to identify the nondominated solutions in each generation. Each experiment was executed 5 times, with 1000 generations in each run, with the same data to demonstrate the reproducibility of the process. Both experiments applied single atoms in the list of available fragments since we did not feel it necessary to constrain the search space with defined substructural fragments in this example. The types of these atoms were derived from the elements present in the objective molecules: these were C, N, O, and F for the first experiment using similar molecules, while Cl was added to this list for the second experiment with diverse objectives. The first experiment, evolving with two similar molecules, used a population size of 200 for each run, while the second experiment, with two diverse structures, used a population size of 500. All other parameters remained constant between experiments: an equal probability of either a crossover or mutation operation being performed; within these sets, each crossover and mutation operator, respectively, had equal weighting. The similarity between the candidate molecules and the objective molecules were all calculated using the Fingal program with all paths being hashed into the 4096 length fingerprints as single bit patterns. These relatively strict parameters were used to ensure highly accurate reporting of molecular similarities since the GA tends to take advantage of imperfections, such as bit collisions, with less rigorous fingerprint generation parameters. The typical runtimes of the population initialization, evaluation, and evolution procedures, respectively, are

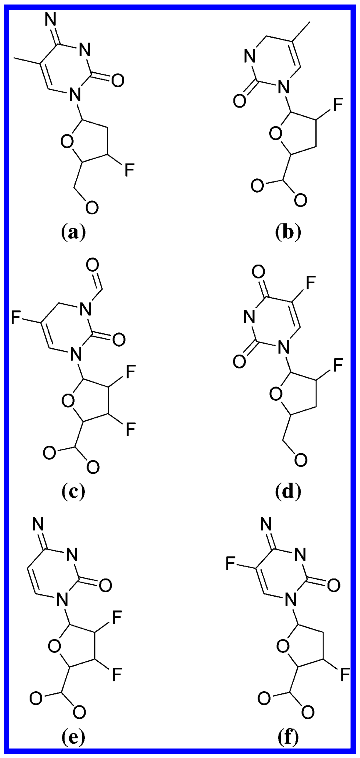

Figure 10. A six-member representative, diverse subset of the evolved median molecules from the first experiment with similar molecular objectives selected using the DBCS algorithm in Fingal.

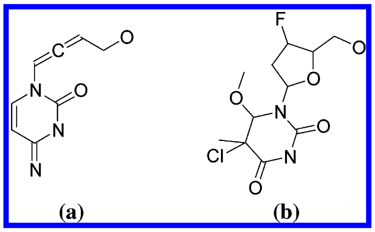

Figure 11. The two diverse molecules used as objectives in the second experiment selected from the NCI AIDS pyrimidine nucleosides subset.

Table 1. Typical Runtimes for Initializing the Random Populations (CoG), Evaluating the Current Population (Fingal), and Evolving a New Population (CoG)[a]

	initialization	evaluation	evolution
similar molecules	0.237	1.893	0.312
diverse molecules	0.475	3.879	0.629

[a] Population sizes for the similar and diverse objectives are 200 and 500, respectively. All results are the average time in seconds from 5 separate runs.

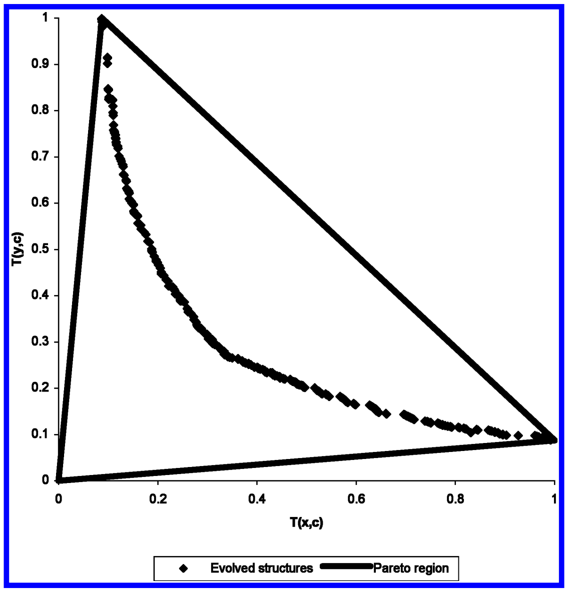

Figure 12. The evolved nondominated structures (*c*) plotted against the Tanimoto similarities to the (*x*) first and (*y*) second diverse objectives of the second experiment along with the theoretical Pareto region.

provided in Table 1; each of the reported runs was executed on a 500 MHz Intel Pentium III CPU.

Our first experiment, considering two similar objective molecules, typically evolved between 40 and 50 nondominated solutions. The graph in Figure 8 plots each of the nondominated solutions from one of the runs; the remaining runs provided very similar sets of structures. For reasons of brevity we have provided a representative six-member subset of molecules selected using a dissimilarity-based compound selection (DBCS) algorithm from a single run (Figure 10), where it may be observed that all of the evolved structures in this subset are significantly similar to the objective molecules. Additionally, the evolved median molecule whose similarity to both objectives is most equal is given in Figure 15a. The Tanimoto similarities of all seven of these evolved median molecules compared with the two objectives are also provided in Table 2. Additionally, the plot of the nondominated solutions in 50-generation intervals is provided in Figure 9. This experiment highlights that, even between two very similar molecules, there are a substantial number of additional molecules that lie between them.

Conversely, the second experiment considers two significantly diverse molecules as the objective structures (Figure 11). Each of the runs in this case resulted in the evolution of between 300 and 350 nondominated molecules on the Pareto frontier between the objectives. Perhaps unsurprisingly Figure 12 illustrates that the evolved Pareto frontier is substantially less close to the theoretical upper-bound than in the previous experiment, although this does not indicate that the GA had prematurely converged on that set of solutions. Indeed, each of the runs evolved to a similar Pareto frontier, suggesting that the GA is quite robust in discovering either the actual upper-bound of the frontier or one that is quite close to that frontier.

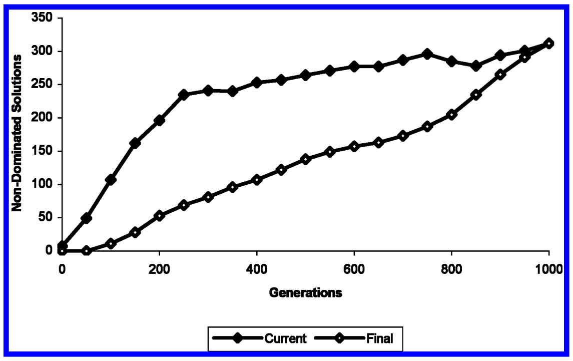

Figure 13. The size of the current Pareto frontier at 50-generation intervals and the proportion of each population that is also in the final Pareto frontier after 1000 generations from the second experiment.

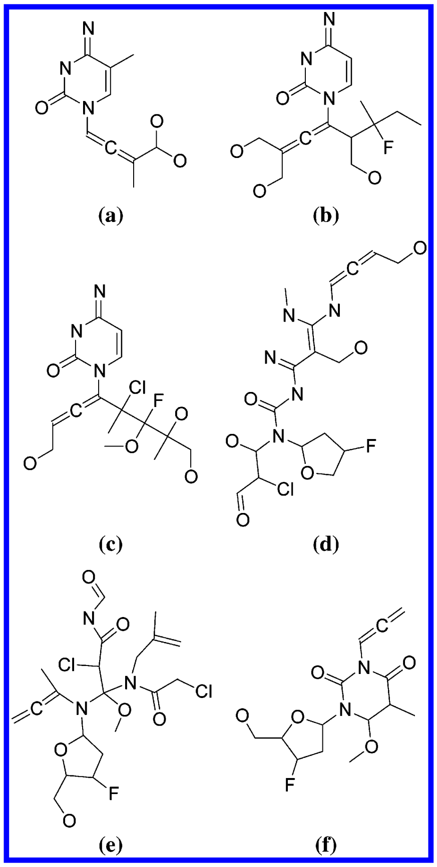

Figure 14. A six-member representative, diverse subset of the evolved median molecules from the second experiment with diverse molecular objectives selected using the DBCS algorithm in Fingal.

As for the first experiment, a diverse six-member subset has been selected from the nondominated set of one of the runs (Figure 14), along with the most balanced median molecule (Figure 15b) and the Tanimoto similarities of all seven structures to the objectives. Additionally, the plot of the nondominated solutions in 50-generation intervals is provided in Figure 13.

It is impossible to conclude with certainty that our approach has evolved molecules on the actual Pareto frontier

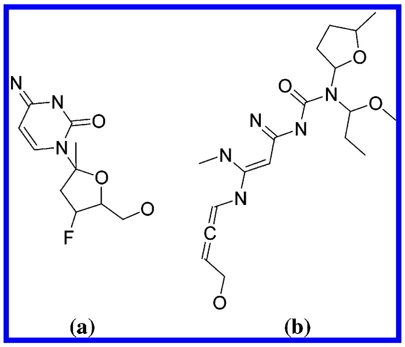

Figure 15. The median molecules evolved from the (a) similar objectives and (b) the diverse objectives that are most balanced in similarity to each of the objectives.

Table 2. Tanimoto Similarities of the Evolved Median Molecule that Is Most Balanced in Similarities to the Objectives and the Six-Member Diverse Subset from the Similar Objectives Experiment

	objective 1	objective 2
median	0.749	0.749
A	0.789	0.719
B	0.733	0.755
C	0.732	0.764
D	0.714	0.801
E	0.694	0.828
F	0.679	0.834

between the two objectives in each experiment without further investigation and, most likely, an alternative molecular descriptor generation method. However, given that in both experiments a similar frontier was evolved in each run, it can be said with a high level of confidence that this approach, at the very least, evolves sets of structures that are very close to this frontier.

The results of these experiments demonstrate that, using our approach, it is possible to automatically generate a substantial number of molecules that are discernibly similar to the target molecules in each case. Additionally, when the objective molecules are very dissimilar, a significantly higher number of nondominated points in chemical space are evolved than when similar objectives are considered. However, this is not to suggest that even when the objectives are very similar, as in our first experiment, that there is not still a substantial number of viable points in chemical space between the structures.

7. CONCLUSIONS

We have presented the CoG program as a generic procedure to evolve novel topological molecules from a set of user-defined atoms, molecular fragments, or a combination of both using a graph-based genetic algorithm and an external program for the calculation of the objective function. Furthermore, we have proposed an application of this software, that of generating a set of median molecules by the novel application of multiobjective optimization techniques, which allows the evolution of multiple molecules on the Pareto frontier between a set of objective molecules.

In this paper, we have applied only fingerprints as the method of describing molecules and calculating their similarity to the objectives since molecular fingerprints are relatively inexpensive to generate and compare. Alternative and richer descriptor generation and molecular comparison software could easily be applied with our software since the evaluation of the evolved molecules is performed externally to the evolution process itself. However, a significant drawback to using these alternate descriptor generation and similarity calculation programs is that these descriptors tend to be substantially more computationally intensive. In addition, the generation of a minimized 3D structure will often be a necessary precursor. A parallel implementation is one option that we are currently investigating to allow the descriptor generation routines to be partitioned and distributed to additional CPUs, possibly using a Grid-type architecture[35] allowing a simple scale-up to take advantage of available Grid resources should this be necessary.

Our initial experiments reported here demonstrate that the CoG program can rapidly and reliably evolve a set of nondominated solutions on the Pareto frontier between two objective molecules. This controlled enumeration of chemical space between two known objectives could feasibly have many applications in the field of chemoinformatics.

We have reported the evolution of median molecules from only two similar and two diverse objective molecules, respectively. Our experiments demonstrate that it is necessary, particular with pairs of diverse molecules, to intelligently reduce the set of nondominated solutions in an attempt to stop this set of solutions overwhelming the population. Indeed, this is the only limitation of our approach that we have observed in its extension to larger numbers of objective molecules, whether they are diverse or not. Therefore, a crucial next step is to design a niching method[36] that operates by niching the set of nondominated solutions to maintain a smaller but still diverse set; this is different to most niching strategies where the intention is to maintain diversity in the overall population. The chosen strategy will likely involve a hybrid niching approach employing the radius niching of the nondominated set, which is often used in MOGA studies, as a first tier screening stage. The second tier of niching would be a more rigorous, and computationally expensive, similarity calculation of these structures. The second stage is necessary since it cannot be determined that two structures are similar to each other only if their Tanimoto similarities to the objectives are similar.

Currently, cluster centroids are often represented as the single structure that exists in the cluster that is closest to the real centroid of the cluster. However, this artificial centroid may not necessarily best represent all of the molecules within the cluster. Applying CoG in these situations may result in a point in chemical space that is closer to the actual centroid than simply the nearest extant structure. Furthermore, it is likely that this technique could be applied in automatically generating novel molecules that lie within a particular cluster given a set of cluster points as objectives. This set of evolved median molecules could then be used in virtual screening to complement existing combinatorial library approaches. An alternative application is to automatically generate these structures as a rational method of virtual library design or for the enrichment of large corporate libraries by enumerating certain types of molecules that are under-represented in these libraries.

A further option is to evolve pseudomolecules that are more similar to all of the objective molecules, possibly by relaxing the valence bond model, as a more generalized molecular consolidation method than existing chemical hyperstructure approaches.[37] We envisage that this process

Table 3. Tanimoto Similarities of the Evolved Median Molecule that Is Most Balanced in Similarities to the Objectives and the Six-Member Diverse Subset from the Diverse Objectives Experiment

	objective 1	objective 2
median	0.302	0.307
A	0.879	0.105
B	0.516	0.201
C	0.377	0.254
D	0.233	0.414
E	0.130	0.683
F	0.113	0.755

could be applied as a screening method by creating generic graphs to represent clusters in large compound databases, thereby significantly reducing expensive structure-by-structure comparisons.

The inverse quantitative structure−activity/property relationship (Inverse QSAR/QSPR) problem could feasibly be another application of CoG, in which molecules are generated based on predictions from an existing QSAR/QSPR model. Indeed, evolutionary molecular design approaches have been applied to the Inverse QSAR/QSPR problem.[38]

We are currently investigating the applicability of our approach with more objectives than have been used in this initial study to observe whether, given a set of objectives, it is possible to sensibly enumerate molecules in chemical space between these points. We are also considering a number of methods of farming out descriptor calculations to additional processors. These extensions will open up our approach to the more generalized and application-bound problems described above.

ACKNOWLEDGMENT

This research has been supported by a Marie Curie Fellowship of the European Community program ‘Exploring leads in combinatorial catalysis for novel clean pharmaceutical/fine chemical processes’ under contract number HPMI-CT-2001-00108.

REFERENCES AND NOTES

(1) Willis, R. C. 2001: A Dock Odyssey. *Modern Drug Discovery* **2001**, *4*(9), 30−32.

(2) Joseph-McCarthy, D. An Overview of *In Silico* Design and Screening: Toward Efficient Drug Discovery. *Current Drug Discovery* **2002**, March, 20−23.

(3) Glen, R. C.; Payne, A. W. R. A Genetic Algorithm for the Automated Generation of Molecules within Constraints. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 181−202.

(4) Devillers, J. Designing Molecules with Specific Properties from Intercommunicating Hybrid systems. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1061−1066.

(5) Goh, G. K.-M.; Foster, J. A. Evolving Molecules for Drug Design Using Genetic Algorithms via Molecular Trees. In *GECCO 2000: Proceedings of the Genetic and Evolutionary Computation Conference*; Whitley, D., Goldberg, D. E., Cantu-Paz, E., Spector, L., Parmee, I., Beyer, H.-G., Eds.; Morgan-Kaufmann: San Francisco, CA, 2000; pp 27−33.

(6) Pegg, S. C.-H.; Haresco, J. J.; Kuntz, I. D. A Genetic Algorithm for Structure-Based *De Novo* Design. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 911−933.

(7) Böhm, H.-J. The Computer Program LUDI: A New Method for the *De Novo* Design of Enzyme Inhibitors. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 61−78.

(8) Gillet, V.; Johnson, A. P.; Mata, P.; Sike, S.; Williams, P. SPROUT: A Program for Structure Generation. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 127−153.

(9) Nishibata, Y.; Itai, A. Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. *Tetrahedron* **1991**, *47*, 8985−8990.

(10) Rotstein, S. H.; Murcko, M. A. GenStar: A method for *de novo* drug design. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 23−43.

(11) Goldberg, D. E. *Genetic Algorithms in Search, Optimisation and Machine Learning*; Addison-Wesley: Reading, MA, 1989.

(12) *Evolutionary Algorithms in Computer-Aided Molecular Design;* Clark, D. E., Ed.; Wiley-VCH: Weinheim, 2000.

(13) Koza, J. R. *Genetic Programming*; MIT Press: Cambridge, MA, 1992.

(14) Harvey, I. Species Adaptation Genetic Algorithms: A Basis for a Continuing SAGA. In *Proceedings of the First European Conference on Artificial Life: Toward a Practice of Autonomous Systems*; Varela, F. J., Bourgine, P., Eds.; MIT Press/Bradford Books: Cambridge, MA, 1992; pp 346−354.

(15) Ferreira, C. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Systems* **2001**, *13*, 78−129.

(16) Gillet, V. J.; Downs, G. M.; Ling, A. (B.); Lynch, M. F.; Venkataram, P.; Wood, J. V.; Dethlefsen, W. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 8. Reduced Chemical Graphs, and their Application in Generic Chemical Structure Retrieval. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 126−137.

(17) Rarey, M.; Dixon, J. S. Feature Trees: A New Molecular Similarity Measure based on Tree Matching. *J. Comput.-Aided Mol. Des.* **1998**, *12*, 471−490.

(18) Venkatasubramanian, V.; Chan, K.; Caruthers, J. M. Evolutionary Design of Molecules with Desired Properties using the Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 188−195.

(19) Douguet, D.; Thoreau, E.; Grassy, G. A Genetic Algorithm for the Automated Generation of Small Organic Molecules: Drug Design using an Evolutionary Algorithm. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 449−466.

(20) Kamphausen, S.; Höltge, N.; Wirsching, F.; Morys-Wortmann, C.; Riester, D.; Goetz, R.; Thürk, M.; Schwienhorst, A. Genetic Algorithm for the Design of Molecules with Desired Properties. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 551−567.

(21) Nachbar, R. B. Molecular Evolution: Automated Manipulation of Hierarchical Chemical Topology and Its Application to Average Molecular Structures. *Genetic Programming Evolvable Machines* **2000**, *1*, 57−94.

(22) Weininger, D. *Method and Apparatus for Designing Molecules with Desired Properties by Evolving Successive Populations*. U.S. Patent No. 5,434,796, 1995.

(23) Globus, A.; Lawton, J.; Wipke, W. T. Automatic Molecular Design Using Evolutionary Algorithms. *Nanotechnology* **1999**, *10*, 290−299.

(24) The Tripos Mol2 File Format is available from Tripos Inc. at http://www.tripos.com/.

(25) Diestel, R. *Graph Theory*, 2nd ed.; Springer-Verlag: New York, 2000.

(26) Cross, A. D. J.; Wilson, R. C.; Hancock, E. R. Inexact Graph Matching using Genetic Search. *Pattern Recognit.* **1997**, *30*, 953−970.

(27) Brown, N. Generation and Display of Activity-Weighted Chemical Hyperstructures, Ph.D. Thesis, The University of Sheffield, UK, 2002.

(28) Jiang, X.; Münger, A.; Bunke, H. On Median Graphs: Properties, Algorithms, and Applications. *IEEE Trans. Patt. Anal. Mach. Intell.* **2001**, *23*, 1144−1151.

(29) Shapiro, L. G.; Haralick, R. M. Structural Descriptions and Inexact Matching. *IEEE Trans. Patt. Anal. Mach. Intell.* **1981**, *3*, 504−519.

(30) Fonseca, C. M.; Fleming, P. J. Genetic Algorithms for Multiobjective Optimisation: Formulation, Discussion and Generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*; Forrest, S., Ed.; Morgan Kaufmann: San Mateo, CA, 1993; pp 416−423.

(31) Handschuh, S.; Wagener, M.; Gasteiger, J. Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 220−232.

(32) Agrafiotis, D. K. Multiobjective Optimisation of Combinatorial Libraries. *IBM J. Res. Dev.* **2001**, *45*, 545−566.

(33) Wright, T.; Gillet, V. J.; Green, D. V. S.; Pickett, S. D. Optimizing the Size and Configuration of Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 381−390.

(34) The NCI AIDS database available from http://dtp.nci.nih.gov.

(35) *The Grid: Blueprint for a New Computing Infrastructure*; Foster, I., Kesselman, C., Eds.; Morgan Kaufmann: San Francisco, CA, 1999.

(36) Horn, J.; Nafpliotis, N.; Goldberg, D. E. A Niched Pareto Genetic Algorithm for Multiobjective Optimisation. In *Proceedings of the First IEEE Conference on Evolutionary Computation*; IEEE: New York, 1994; pp 82−87.

(37) Brown, N.; Lewis, R. A.; Willett, P.; Wilton, D. J. Generation and Display of Activity-Weighted Chemical Hyperstructures. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 288−297.

(38) de Julián-Ortiz, J. V. Virtual Darwinian Drug Design: QSAR Inverse Problem, Virtual Combinatorial Chemistry, and Computational Screening. *Comb. Chem. High Throughput Screening* **2001**, *4*, 295−310.

CI034290P