De Novo Drug Design Using Multiobjective Evolutionary Graphs

Christos A. Nicolaou,*[,†,‡] Joannis Apostolakis,[§] and Costas S. Pattichis[†]

Computer Science Department, University of Cyprus, 75 Kallipoleos Street, CY-1678 Nicosia, Cyprus, Noesis Chemoinformatics, Metochiou 66, CY-1599 Nicosia, Cyprus, and Ludwig Maximilian University, Amalienstrasse 17, D-80333 Munich, Germany

Received August 31, 2008

Drug discovery and development is a complex, lengthy process, and failure of a candidate molecule can occur as a result of a combination of reasons, such as poor pharmacokinetics, lack of efficacy, or toxicity. Successful drug candidates necessarily represent a compromise between the numerous, sometimes competing objectives so that the benefits to patients outweigh potential drawbacks and risks. De novo drug design involves searching an immense space of feasible, druglike molecules to select those with the highest chances of becoming drugs using computational technology. Traditionally, de novo design has focused on designing molecules satisfying a single objective, such as similarity to a known ligand or an interaction score, and ignored the presence of the multiple objectives required for druglike behavior. Recently, methods have appeared in the literature that attempt to design molecules satisfying multiple predefined objectives and thereby produce candidate solutions with a higher chance of serving as viable drug leads. This paper describes the Multiobjective Evolutionary Graph Algorithm (MEGA), a new multiobjective optimization de novo design algorithmic framework that can be used to design structurally diverse molecules satisfying one or more objectives. The algorithm combines evolutionary techniques with graph-theory to directly manipulate graphs and perform an efficient global search for promising solutions. In the Experimental Section we present results from the application of MEGA for designing molecules that selectively bind to a known pharmaceutical target using the ChillScore interaction score family. The primary constraints applied to the design are based on the identified structure of the protein target and a known ligand currently marketed as a drug. A detailed explanation of the key elements of the specific implementation of the algorithm is given, including the methods for obtaining molecular building blocks, evolving the chemical graphs, and scoring the designed molecules. Our findings demonstrate that MEGA can produce structurally diverse candidate molecules representing a wide range of compromises of the supplied constraints and thus can be used as an "idea generator" to support expert chemists assigned with the task of molecular design.

INTRODUCTION

Drug discovery focuses on identifying molecules that selectively bind and interact with specific biological receptors and cause a certain desired behavior. The ability to interact is controlled by the molecular structure of the drug and namely by its complementarity to the targeted receptor site. The complementarity refers not only to shape but also to electrostatic and hydrophobicity properties of the receptor site.[1−6] However, not all potent binding molecules are suitable as drugs. In order to be truly effective within a living organism a molecule must satisfy several additional properties, including pharmacokinetics, pharmacodynamics, solubility, and toxicity. These properties depend on the way a drug behaves "in vivo" (i.e., in the living organism to be treated) and how well it reaches the region of the target without binding nonselectively to other receptors.[7] An additional equally important requirement drugs need to satisfy is synthetic feasibility. The presence of these requirements turn drug discovery into a multiobjective problem in which any candidate solution needs to fulfill multiple objectives concurrently. Given that molecular graphs define the structure or at least the conformational space of a given molecule, drug design can be thought of as an optimal graph design problem that seeks to identify compounds that show a number of favorable properties, such as potent and specific binding, favorable pharmacokinetics properties, low toxicity, and synthetic feasibility.

Computational de novo drug design involves searching an immense space of feasible, druglike molecules to select those with the highest chances of becoming drugs.[2] Modeled after traditional experimental procedures, which typically follow a sequential optimization paradigm, most de novo design research has been ignoring the multiobjective nature of the problem and focused on the optimization of one molecular property at a time.[8] Typically the property serving as the primary objective has been similar to a known ligand or an interaction score with a target receptor. Multiobjective optimization (MOOP) methods introduce a new approach for optimization that is founded on compromises and tradeoffs among the various, potentially conflicting objectives. A typical example of a problem with conflicting objectives from drug discovery is the design of chemical structures with increased potency to a specific target and no toxic effects. The aim of MOOP methods is to discover a set of satisfactory

* Corresponding author e-mail: cnicolaou@cs.ucy.ac.cy.
[†] University of Cyprus.
[‡] Noesis Chemoinformatics.
[§] Ludwig Maximilian University.

10.1021/ci800308h CCC: $40.75
Published on Web 01/26/2009

compromise solutions and, through them, the globally optimal solution(s) by optimizing numerous dependent properties simultaneously.[9] A major benefit of MOOP methods is that local optima corresponding to one objective can be avoided by consideration of all the objectives simultaneously, thereby escaping single objective dead-ends and leading to a more efficient overall process. Encouraged by successful applications of this technology in a number of scientific fields[9] and the initial successes of related research in the drug discovery field[4,10] we propose a general algorithmic framework for designing chemical structures (i.e., molecular graphs) that satisfy several pharmaceutically important requirements. The approach we have chosen to follow combines multiobjective evolutionary algorithms with local search techniques and expands evolutionary algorithms by incorporating novel features, such as self-adaptation capabilities. The resulting hybrid algorithms are customized for graph design and optimization and are enhanced with problem specific knowledge.

The remainder of this paper is organized as follows. The next section presents the requirements set for the proposed system followed by a section that reviews related methodology and background information. The proposed methodology is then described in detail. Next, the experimental settings and the results from the application of the method on designing compounds is described followed by a discussion of our findings. The final section of our paper contains our conclusions and future work plans.

DE NOVO DESIGN SYSTEM REQUIREMENTS

In a multiobjective problem setting multiple equivalent solutions representing different compromises among the objectives are possible. Although in many applications the presence of multiple solutions may be considered a problem, and therefore methods for selecting a priori the single 'best' solution are often employed, in drug discovery the availability of several diverse solutions that can serve as leads is generally preferred. Based on this requirement a truly multiobjective de novo design system must be able to produce multiple, diverse solutions and enable users to choose a posteriori from a variety of candidates. Considering the combinatorial nature of the problem, the system must also employ a powerful search strategy in order to detect the best possible solutions within a reasonable amount of time. The strategy must be able to handle complex, nonuniform search spaces since the presence of multiple conflicting objectives point to the potential presence of multiple solutions at different regions of the space. The system must also be able to take advantage of existing pharmaceutically relevant knowledge to streamline the search process and achieve better performance. Such knowledge may be supplied in the form of implemented objective functions or rules to be used for rejecting low quality candidates. Not only to facilitate human expert understanding of the internal operations but also to avoid information loss and misleading results the system must represent candidate solutions using a molecular graph data structure. Special emphasis must also be placed on system flexibility, to enable easy choice of objectives, and on performance and scalability issues to ensure practical usefulness.

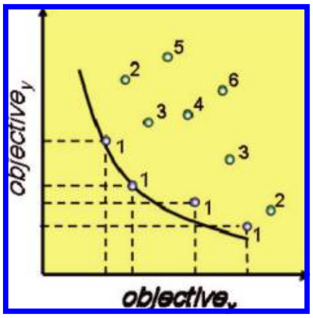


Figure 1. In a multiobjective problem several equivalent, nondominated solutions may exist representing compromises among the different objectives. Typically solutions to the problem are ranked according to the number of other solutions dominating them, i.e. solutions that are better in all objectives. Nondominated solutions are labeled with '1'. The curved line represents the Pareto-front. Note that both objectives in the example shown should be minimized.

RELATED WORK

The research proposed in this paper follows a hybrid approach, combining methods from the fields of multiobjective optimization, evolutionary algorithms/graphs, and de novo molecular design and appropriately introducing innovative steps where needed. In this section we describe the current status of related research placing special emphasis on attempts to utilize ideas across the various fields.

Multiobjective Optimization. Optimization problems can be classified according to the number of criteria their objective function encodes. Single-objective problems encode a single criterion while multiobjective problems encode multiple. Accordingly, optimization techniques have been designed to solve single-objective and multiobjective problems. Single-objective optimization (SOOP) methods explore the feasible search space to find the optimum, the single best solution, to a one-objective function. In contrast, in multiobjective settings it is often the case that no single best solution can be found that outperforms all others in every criterion, especially if the objectives involved are conflicting.[7] In such cases, there are multiple solutions representing various possible compromises among the objectives.[4] These solutions, known as nondominated solutions, are characterized by the fact that there are no other solutions that are better than them in all of the objectives considered. The set of nondominated solutions is also known as the Pareto-front or the tradeoff surface. Figure 1 illustrates the concept of nondominated solutions and the Pareto-front. Approximating the Pareto-front is a challenging task since multiobjective problems are often characterized by complex, nonuniform search spaces with various local optima.

There are several classification schemes for MOOP algorithms. Based on the quantity of the solutions produced MOOP methods can be divided into those producing single 'best' solutions and those that attempt to map the entire Pareto-front and thus provide a number of solutions of equal rank. A second classification scheme is based on the user interaction specifics. According to this scheme the three distinct types of MOOP algorithms are the a priori methods that take into account the preferences of the user before the optimization process is conducted, the progressive methods that require user interaction to guide the search, and the a posteriori methods that require user intervention only after

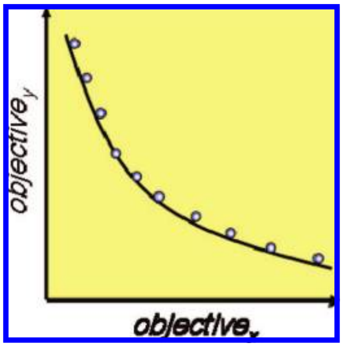


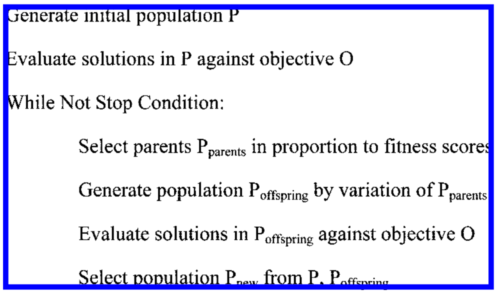
Figure 2. The solution set (blue dots) produced by a Pareto-based multiobjective optimization method must converge to the true Pareto-front (continuous curved line) and cover it effectively. Note that both objectives in the example shown should be minimized.

the optimization process is completed.[11] Typically, a posteriori methods tend to produce the Pareto-front of equivalent solutions and enable the user to choose based on their specific requirements.

A common MOOP method is the weighted-sum-of-objective-functions method which essentially associates a weight with each objective function and takes the weighted sum of the objectives to be the new composite objective function.[9] The problem is thus transformed from multi- to single-objective, and any method capable of solving single optimization problems may be used to solve it. Despite its simplicity and appeal the method faces several challenges. The first is choosing the objective weights since it may not be clear how objectives should be ranked. The second problem relates to the loss of solutions due to the transformation of the problem to a simpler form. The use of a single-objective method limits the number of solutions that are retrieved to practically one when there exist a multitude of equally good solutions with potentially different characteristics. A final criticism is that the weighted-sum-of-objective-functions method is limited in its ability to find solutions to problems with nonconvex solution surfaces.[9] Consequently, even multiple runs of the method with varying objective weights may be unable to approximate the true Pareto-front.

Pareto-based methods aim to identify multiple solutions representing various compromises among the objectives considered. Typically, the solutions are presented to the user to choose according to the a posteriori paradigm. By optimizing numerous properties simultaneously Pareto methods manage to avoid the pitfalls associated with methods combining multiple objectives into a single one.[4] The challenge facing Pareto-based methods is to produce a set of well-dispersed solutions representing effectively the true optimal front[12] as shown in Figure 2. The problem is exacerbated by the complexity and size of the solution space especially in cases with several objectives. Typical problems faced by Pareto-based methods include failure to converge to the true Pareto-front and lack of effective coverage of the front. The latter is often due to the so-called genetic drift phenomenon which leads to a distribution of solutions that is not representative of the true front since it favors certain regions and under-represents others.[4]

Evolutionary Algorithms. In the field of optimization research, when confronted with complex, unknown search landscapes, it is common to utilize algorithms inspired by nature.[13] Such algorithms, known as Evolutionary Algorithms (EA) (see Figure 3), are known to have excellent capabilities for global search even when little or no information is available about the underlying functions defining the feasible search space and only a target function of the optimization problem to guide its search is available.[14] Any EA designed to solve a particular problem must have a genetic representation for potential solutions to the problem, a way to create an initial population of potential solutions, a scoring mechanism to rate solutions in term of their fitness, a method to select the subset of parents, and evolutionary operators that generate offspring from parents.[13] Additionally, the EA needs to be supplied with values for various parameters it uses i.e. population size, evolutionary operators probabilities, etc.



```
Generate initial population P
Evaluate solutions in P against objective O
While Not Stop Condition:
        Select parents $P_{parents}$ in proportion to fitness scores
        Generate population $P_{offspring}$ by variation of $P_{parents}$
        Evaluate solutions in $P_{offspring}$ against objective O
        Select population $P_{new}$ from P, $P_{offspring}$
```

Figure 3. General scheme for evolutionary algorithms.

There are several variations of EA algorithms that differ mainly in the representation of the solutions and the genetic operators used to generate new candidate solutions by varying representatives from the current population. The representation of individuals in an EA is a task of crucial importance since these mechanisms influence the ability to represent accurately problem-specific individuals and measure their fitness. In addition, the mechanisms are instrumental to the processes of evolution since genetic operators act on the chromosome structure. It is important to note the potential consequences of encoding and decoding problem solutions to algorithmic representations. Algorithmic representations are the genotypes of EAs on which evolutionary operations occur. Just as in nature, genotypes need to be expressed into phenotypes for biological operations to take place; in EAs algorithmic representations are decoded into solutions of the problem at hand. Consequently, the ability of the genotype to capture and represent the specifics of the problem and provide a suitable environment for phenotype evolution and optimization is essential. Simple genotype structures may be convenient to manipulate, but in the case of complex solutions they are subject to information loss and require a more elaborate encoding/decoding mechanism.

Genetic Algorithms (GA) use a linear representation of individuals, specifically a string of characters, traditionally consisting of only 0 and 1. GAs utilize the evolutionary operators of selection, random variation, and mating, through mutation and recombination/crossover on the population of individuals, e.g. the string chromosomes. Recombination is achieved via the process of crossover where one point—or more—in the genotype of the parents is selected, and the genotypes are split at that point and exchanged between the parents. A round of random mutations may also be applied on the parents or the offspring from the recombination process often using a simple mutation operator to invert a randomly chosen bit of an individual. The linear structure of the chromosome and the limited availability of genes/

building blocks is an issue to be confronted when using GAs especially when dealing with problems where solutions are naturally represented as more complex data structures.[13] Several extensions to GAs that keep the linear nature of the chromosome but use a richer set of genes, e.g. nonbinary or real-valued, face a similar set of issues.

Genetic Programming (GP) is an extension of the GA that uses a procedural or functional representation for solutions, typically tree-based.[15] In a typical setting GP uses a directed acyclic graph with functions as nodes and terminals as leaves to encode and optimize computer programs using EA principles.[14] Note that except for the solution representation GP is identical to standard GA. Recombination is implemented as subtree crossover between two parents, while mutation selects a random node of the tree and alters the node accordingly depending on its type.[15]

Graph representations of individuals have also been used in combination with EA methodology. Graph chromosome encoding may consist of the graph and edge tables only,[16] a mechanism for serializing the information in the tables,[17] or graph data structures that enable manipulation of node and edge objects.[18] In all cases, appropriate mutation and crossover operations need to be used to operate on the graph chromosome representation. Crossover operations involve the exchange of subgraphs, while mutation may alter nodes, edges, or subgraphs.

Multiobjective Evolutionary Algorithms (MOEA). EAs have also been used extensively for multiobjective problems with several multiobjective optimization EAs (MOEA) cited in the literature.[4,9,10,12] MOEAs are particularly attractive since their population-based approach enables the simultaneous search of multiple search space regions and thus the identification of numerous Pareto-solutions in a single run. Additionally, since EAs impose no constraints on the morphology of the search space, they are suitable for complex, multimodal surfaces such as the ones typically produced by MOOP problems.[4]

In MOEAs the selection step of the solutions involves fitness assessment of each individual solution to all objectives, establishment of a domination relation between all pairs of individuals, Pareto ranking, and selection based on the rank. Additionally, modern MOEAs often use niching (to preserve solution diversity) and elitism (to record all nondominated solutions discovered in different rounds of execution). The latter components aim to assist MOEAs to converge quicker and map the entire Pareto-front in a single run.[9]

De Novo Design. DND methods face the task of effectively exploring a chemical search space estimated to be on the order of $10^{60}-10^{100}$.[19] Such space cannot possibly be fully enumerated, and so powerful search methods need to be applied to detect the best possible solutions in a limited amount of time. Pro_Ligand[20] uses a Depth-First Search method to incrementally design ligands fitting a model derived from a well defined target receptor site or a collection of highly similar actives. The method constructs the ligands using standard chemical rules by matching fragments with the model components. In later publications Pro_Ligand[21] was complemented by a postprocessing GA-driven module that further evolved the designed compounds. Chemical Genesis[22] used 3D molecular fragments to design molecules using a single-objective evolutionary algorithm. The method recognized the need to accommodate multiple objectives and therefore used a composite objective function, combining both receptor and ligand-based objectives. Structure evolution took place by directly operating on the chemical graph via mutation and crossover with a preference on the former.

Several EA-driven applications have been using linear or simple tree representations of molecules. Most of these approaches use molecular fragments as building blocks. Perturbation of the chemical structure is achieved by actions such as addition, deletion, substitution, or exchange of whole fragments. Douguet et al.[23] used a pool of 3D fragments which were combined in a linear fashion to form a chromosome string and a composite fitness function taking into account both receptor and ligand-based constraints. TOPAS[24] used 2D building blocks derived from known drug molecules and an EA method to design molecules similar to a target chemical structure. TOPAS as well as Flux[25,26] used RECAP analysis[27] to generate the fragment building blocks and kept information about the type of bonds at each attachment point. Candidate compounds were then evolved via operations taking into account chemical synthetic rules. Nachbar[28] proposed a ligand-driven DND algorithm that uses a treelike structure for molecule representation. Rings were represented using special pointer nodes to link appropriate tree branches. The method uses a single objective evolutionary approach employing both mutation and a constrained crossover version paying special attention not to make or break rings.

A different category of DND methods exploits the SMILES chemical language[29] to represent molecules. Weininger proposed an evolutionary method that used single atoms and bonds as building blocks to design molecules satisfying any single given objective function,[30] while Douguet et al.[31] chose fragments for building blocks. More recently, Lameijer et al.[32] describe a DND application that uses both atoms and fragments to construct molecules although its mutations are solely atom-based. Interestingly, this application requires the human user to serve as a fitness function to the otherwise evolutionary-based algorithm.

Globus[18] introduced an evolutionary algorithm using labeled, cyclic graphs to represent individuals. Evolution takes place using exclusively crossover through a process of fragmenting individuals in two and combining the parts from the different parents. COG[3] also uses a graph representation of chromosomes, but it uses both fragments and atoms/bonds as building blocks and has an extended set of genetic operators that includes both mutations and crossover that enables crossing over rings. An added novelty to COG is the use of a multiobjective fitness score mechanism based on Pareto-ranking. Individual fitness is calculated using quantitative structure−property relationship models.[3] Brown et al. have more recently published a similar method that optimizes molecules directly in property space, allowing multiple molecular properties to be optimized simultaneously.[33]

Overall, a wide variety of solution representation schemata, molecule synthesis engines, compound evaluation criteria, and search methods have been applied to de novo design. While no standard de novo design methodology has emerged, most of the recent approaches use some form of an EA to search the chemical space and impose various limitations on compound design by choosing a simpler representation schema for molecular graphs. Similarly, perturbation operators encoded are often restricted to exclusive use of fragments as building blocks and constrained versions of mutation and crossover. These limitations reduce the feasible search space to solutions that, for example, can be formed using certain fragments and synthesis rules.[26] Although such reduction in the search space risks missing potentially interesting solutions, it may also be advantageous, provided that the search space is reduced in a meaningful way since it enables a more thorough exploration of the remaining space. Additionally, the majority of current DND methods ignores the multiobjective nature of the problem and uses a SOOP approach capable only of finding solutions from a limited region of the Pareto-front. A final point that needs to be raised is the limited use of domain-specific knowledge in DND applications although some efforts have been made to ensure that the solutions generated are valid through the use of building-blocks derived from drugs and standard synthetic rules for molecule generation. More extensive reviews on the topic can be found in refs 2, 34, and 35 for the older methodologies.

METHODS

The Multiobjective Evolutionary Graph Algorithm (MEGA) framework combines evolutionary techniques with graph data structures to directly manipulate graphs and perform a global search for promising solutions. Additionally, MEGA can incorporate problem-specific knowledge and local search heuristics and techniques, to improve performance and scalability.

MEGA initiates with the supply of a set of molecular building blocks, the implemented objectives to be used for scoring the graphs and a set of attributes controlling mutation and crossover methods and probabilities, selection method, hard filters for solution elimination, etc. Optionally, a set of molecules to be used as the initial population may be supplied as well. The supplied data are used to initiate internal data structures, for example to create graph-based chromosomes representing the molecules and to construct a list of building block objects to use in subsequent steps. Next the algorithm applies the objectives on the initial population to obtain a list of scores for each individual. The list of scores may be used for the elimination of solutions with values outside the range allowed by the corresponding active hard filters. In the next step, the list of scores is subjected to a Pareto-ranking procedure as described in ref 36. According to this procedure the rank of an individual is set to the number of individuals that dominate it incremented by 1, thus nondominated individuals are assigned rank order 1 (see Figure 1). At this phase the algorithm proceeds to calculate a MultiObjective Fitness (MOFit) score for each individual. There are two ways of calculating such a score, controlled by user preferences: the first simply uses a linear transformation function that assigns a higher score to solutions with low Pareto-rank. This method operates exclusively on phenotypes, i.e. in solution space. The second method invokes a niching mechanism that performs diversity analysis of the population via clustering of the genotypes, i.e. the chemical structures, and subsequently prepares a two-valued MOFit score that consists of both the linear transformation of the Pareto-rank and the cluster assignment of the individual. An additional optional step at the users' disposal is the application of elitism which creates and maintains an external archive of Pareto-solutions found during all previous iterations. If elitism is enabled, then the archive of Pareto-solutions is merged with the current population before the MOFit calculation step to form an extended population. Recalculation of the Pareto-rank and diversity analysis are performed on the extended set to calculate the MOFit score of the solutions. The nondominated solutions of the extended population are then stored in the Pareto-archive. Following, MEGA checks for the termination conditions, typically if the number of preset maximum allowed iterations has been reached; if satisfied the process terminates. However, if this is not the case, then the process moves to select the parent subset population. Parent selection is performed using one of the "best", "roulette", or "tournament" methods on the MOFit scores of the solutions. The "best" method simply selects the subset of solutions with the highest transformed Pareto-rank score, whereas the "roulette" method selects solutions via a probabilistic mechanism that assigns higher selection probability to solutions with higher transformed Pareto-rank. The "tournament" method picks random pairs of solutions and selects the one with the highest transformed Pareto-rank score. If the niching mechanism is not enabled, then the chosen parent selection method is applied once on the entire set of candidate solutions to generate the parent subpopulation. If the niching mechanism is enabled, and thus the MOFit scores consist of the transformed Pareto-rank and the cluster assignment of the individual, then the selection methods are applied on the clusters rather than the entire population. The process picks one solution from each cluster starting from the most populous cluster and proceeding to clusters containing the fewest compounds. The process traverses the set of clusters until the number of parents is selected. The parents are then subjected to mutation and crossover according to the probabilities indicated by the user. The new population is formed by merging the original population and the newly produced mutants and crossover children. The process then iterates, and the new population is subjected to fitness calculation against all objectives, hard filtering and Pareto-ranking. Following, MEGA proceeds to reduce the new population to the user defined population size using a "roulette"-like method. The method is essentially identical to the "roulette" parent selection method described previously except that it assigns a higher selection probability to the worst performing solutions. Best performing solutions, i.e. nondominated solutions, have a selection probability of zero. In the special case where the number of best performing solutions exceed the user defined population size an adequate number is randomly selected and marked as "excess" solutions. If elitism is enabled, then these solutions are treated as normal members of the population in the next steps of the algorithm. However, if elitism is not enabled, then these solutions are removed from the current population prior to the parent selection step. Figure 4 summarizes the MEGA framework process. While MEGA has been designed to search for solutions compromising multiple objectives it can also be used in a SOOP mode simply by eliminating the Pareto-rank step and replacing the transformed Pareto-rank score with the transformed single-objective score in all following steps. Correspondingly, the diversity score is then used in the same way as in the standard MOOP case.

The major components of MEGA are described in detail below:

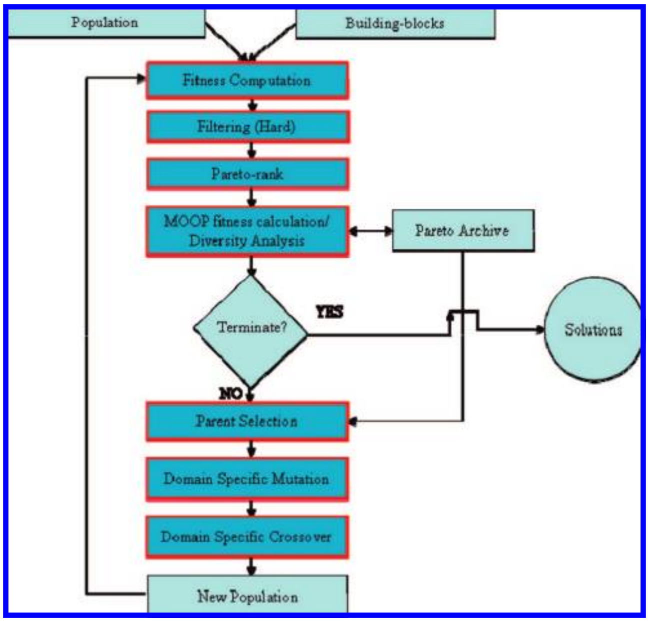


Figure 4. The MEGA algorithm framework. Note the Pareto-archive component storing an elite population of solutions at each generation.

Graph-Based Chromosomes. MEGA uses graph-based chromosomes to avoid the information loss associated with the encoding of more complex structures into simpler ones. In addition to the molecular graph, MEGA chromosomes contain information that can be used during evolutionary design including details related to their synthesis. Specifically, chromosomes contain information about their parent structure(s) and the operation that produced them including the fragments used (if any), the synthetic rule (if any), the type of operation (mutation or crossover), etc.

Building Block Generation. MEGA uses atoms, bonds, and fragments as building blocks. For the purposes of this implementation we use a substructure mining tool[37] able to extract fragments from graphs in a variety of ways including frequent subgraph mining[38] and the RECAP chemical bond type identification and cleaving technique.[27] This tool facilitates the preparation of a large pool of building blocks that contain information about their attachment points and the type of bond cleaved at each attachment point prior to evolutionary design. The resulting fragments are profiled using available knowledge on the molecules that contain them, and weights, reflecting their privileged or not status, are recorded. Essentially, the weight of a specific fragment is incremented by one for each molecule containing it that has a favorable biological profile, e.g. is a drug, or is "active" against the target of interest and decremented by one for each molecule containing it that is considered unfavorable. The weights are exploited in later steps of the algorithm in two ways: first, to increase the chances of the highly weighted fragments to take part in the formation of the initial population and, second, to favor the selection of privileged fragments for use during evolutionary steps, and especially mutation, described in detail in a later section. The RECAP utility of the tool is also used during evolutionary operations as explained in a following section.

Objective Encoding-Scorers. Several methods for assessing the fitness of a molecule have been prepared. The methods fall in three main categories:

(a) Binding Affinity Scorers. We have chosen to use the docking program Glamdock[5] recently developed by Tietze and Apostolakis. We have developed pyChill, a python wrapper for Glamdock to enable tight integration into our de novo design system and facilitate the encoding of docking related objectives, i.e. objectives based on the predicted binding affinity of a designed molecule to a target protein. The designed molecules are docked into the binding site of the corresponding protein, and the interaction score of the best solution is used as an objective function. Settings for docking correspond to the slow settings described in ref 5. The ChillScore[5] is used to score interactions. This integration allows us to easily prepare a binding affinity scorer for a chosen target protein and produce fitness scores of our designed molecules via an interactive process in real time.

(b) Molecular Similarity Scorers. Our system can use molecular similarity as a distinct objective when one of our goals is to produce molecules that resemble (or are quite different from) a known ligand. The method used for this purpose relies on the calculation of a variation of the atom-type descriptor vectors proposed in ref 39 that includes ring membership information of the calculated atom types. Similarity calculations are performed using the Tanimoto measure.[40]

(c) Chemical Structure Scorers. Often, selected classes of drug molecules tend to obey some simple rules easily calculable from the chemical graph of a molecule. An example of this type of rules is the widely known oral-bioavailability Rule-of-Five described by Lipinski et al.[41] In order to exploit such rules and generate fitness scores for the molecular structure we have encoded scorers measuring simple molecular structure properties such as the number of rotatable bonds, the number of hydrogen bond donors and acceptors, and the molecular weight.

In addition to their use to guide the optimization process, objective scorers may also be used as hard-filters, to remove solutions with fitness values outside a predefined allowed range provided by the user. Objectives used in this manner are typically referred to as secondary, while objectives used to guide optimization are considered primary. In a typical scenario chemical structure scorers may be used as secondary objectives to filter out solutions not conforming to the Rule-of-Five[41] and constrain the search space.

Niching Mechanism. The MEGA framework uses a unique niching mechanism to preserve graph chromosome diversity and ensure that a variety of different promising subgraphs (scaffolds/chemotypes) survive long enough in the evolution cycle to contribute to the solution search. The technique is based on diversity analysis at the genotype level, i.e. the graph chromosomes. In the current implementation we have used the Wards agglomerative clustering technique[42] and our variation of atom-type descriptors.[39] The resulting Wards cluster tree is processed with the Kelley cluster level selection method[42] to identify the main solution clusters. Typically 4−50 clusters are produced at each population depending on the size and diversity of the population, the type and number of objectives to be optimized, the type of evolutionary operations performed, etc. We have found no significant trend in the number of clusters as the algorithm goes through the populations. The clusters obtained are ranked based on their size, and solutions are labeled with the characteristics of the cluster they belong to. The results from clustering are used in the preparation of a special solution score, the MOFit, that has two values, the linear transformation of the Pareto-rank of the solution and the cluster assignment, to allow the sampling of diverse populations. Care is exercised to accommodate the likely presence of singleton and under-represented clusters often found when the population size is small or particularly diverse. Such clusters may cause problems during selection since, for example, the "tournament" method requires at least two members in each cluster to function. To avoid this type of problem MEGA implements appropriate rules, such as allowing only simple selection from singleton clusters.

Elitism-Pareto-Archiving. Elitism, a mechanism designed specifically to preserve good nondominated solutions from getting lost,[43] is implemented in MEGA. The mechanism uses an archive of Pareto-solutions of unlimited size where all nondominated solutions found over all iterations are stored. In effect, MEGA merges the Pareto archive with the current population before the MOFit calculation step and uses this larger set as the current population. The algorithm then processes the new current population set via Pareto-ranking, diversity analysis, and MOFit calculation as described above. The Pareto-archive is then reset to contain only the Pareto-solutions of this new set. Note that the unlimited size of the Pareto-archive allows the storage and preservation of a number of solutions exceeding the user-defined population size. The algorithm is a result of observations made during initial runs of MEGA where some promising nondominated solutions were lost due to limitations related to population size.

Graph-Specific Evolutionary Operations. MEGA evolves solutions through a set of operations at the level of single bonds and atoms as well as on fragments. The processes can be divided into two main categories, those inspired by mutation and those inspired by crossover. The mutation inspired processes include modifications of the atom and bond types and insertion and removal of single atoms and bonds as well as insertion, removal, and exchange of fragments. For fragment insertion, an attachment point is first chosen, and a fragment from the weighted fragment collection is chosen and attached. For the fragment removal and exchange operations RECAP is used to break the molecule in two disconnected parts and either remove or replace one of them with a fragment from the fragment collection. Note that fragment weights influence the probability of selection of a fragment for the insertion and exchange operations. Also note that users can optionally restrict the exchange fragment operation to building blocks with attachment points of compatible RECAP bond types in order to increase synthetic feasibility chances. Crossover takes place using two operations. The first identifies and cleaves a RECAP-type bond in each of two parents and recombines the resulting fragments to generate offspring. In a manner similar to the exchange fragment operation described above, this type of crossover can also be restricted to breaking specific bond types and combining fragments with compatible bond types in order to produce chemical designs with higher chances of being synthesizable. The second uses a methodology initially proposed in ref 18 and later used in ref 3 as well. A bond is selected randomly in each of two parents and removed. If the bond is part of a ring system, then additional appropriate bonds are also cleaved to obtain two fragments. The resulting fragments are then recombined to generate offspring. Note that a check and repair or discard mechanism is applied to ensure that the resulting offspring are valid molecules with respect to valences. Briefly, in its current implementation the mechanism identifies atoms with valence problems and attempts to repair them by either removing hydrogens attached to the atom or by downgrading atom bonds to a lower order, i.e. converts a double bond to single or a triple to double. If such action is not possible or sufficient to fix the problem, then the offspring is discarded.

Exploitation of Available and Generated Knowledge. In an effort to improve search efficiency MEGA incorporates simple heuristics that can be used to exploit existing pharmacologically relevant knowledge that has accumulated during past efforts to discover drugs. The heuristics involve the usage of the weights associated with the fragment building blocks provided and result in favoring those with a privileged status, i.e. increased weight. Additionally, knowledge of the current status of the Pareto-front is also monitored and exploited to self-adapt certain attributes controlling the optimization process. In the current implementation MEGA uses the results of the diversity analysis to identify under-represented regions in the Pareto-front. Based on this new knowledge as well as the information about the generation history contained within each chromosome MEGA favors fragments that were used previously to produce under-represented solutions and thus increases the chance of generating new solutions from specific regions of the space.

RESULTS

The validation of MEGA included tests on both single and multiobjective de novo design problems. An account of the tests performed and the results obtained follows. All runs were performed on a computer equipped with an Intel Core 2 Duo 3 GHz processor and 2GB of memory. Since single-objective tests were only used for testing algorithm correctness and finding reasonable settings for the MOOP runs, they are only briefly described.

Data Sets. Two data sets were used during the MEGA tests performed. Data set 1, a set of Estrogen Receptor (ER) ligands,[44] contains 53 compounds. Data set 2, obtained from PubChem,[45] is an ER-alpha inhibitor assay data set with the label Bioassay 713 - HTS for Estrogen Receptor-alpha Coactivator Binding inhibitors. Data set 2 contains 439 compounds for which an activity label is also available from PubChem.

Building Blocks. A single collection of 2363 building blocks was used for all the tests performed. The collection was prepared in advance of the tests and was supplied by the user during run time. The building blocks were obtained via fragmentation of data sets 1 and 2 described above with the substructure mining tool provided by ref 37 described previously. Building block weights were assigned according to the activity label of the molecules containing it. If the molecule was in data set 1 or was labeled active in data set 2, then the building block weight was incremented by 1. If the molecule containing a building block was labeled inactive in data set 2, then the weight was decremented by 1.

SOOP Application. MEGA has been initially applied to a series of single-objective de novo design problems for testing purposes. Both ligand-driven and receptor-driven tests

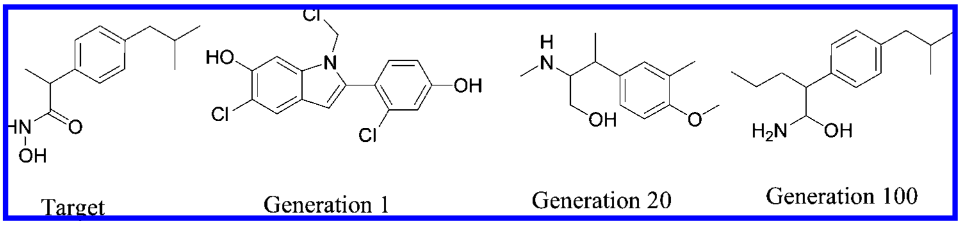


Figure 5. Successive evolution of a chemical structure toward a known molecule, in this case ibuproxam. The molecules shown were the best performers (most similar) in generations 1, 20, and 100. At generation 250 the best molecule was identical to the query molecule in descriptor space. The size of the population used in the specific run was 50, and the selection mechanism was roulette.

were performed to test the ability of the algorithm to design molecules meeting single criteria that can be relatively easier to assess. The tests involved designing molecules with high similarity to a known ligand or binding affinity to a given receptor. We applied similar experimental settings for the two tests. Analytically, we used population sizes 10, 20, 50, and 100 and all types of parent selection mechanisms encoded, i.e. best, roulette, tournament. Runs were performed with the diversity-based niching mechanism enabled or off. Three types of evolutionary operation combinations were tested with mutation only, crossover only, and both mutation and crossover applied. Mutation probability was set at 0.25 and crossover at 1.0. Runs were performed for each settings combination. Results were assessed after 20, 50, 100, 500, and 1000 iterations for the ligand-driven tests and 20, 50, and 100 iterations for the receptor-driven tests. For each test case the initial population was selected from a user-defined data set.

In the similarity-driven tests the single objective function used was similarity to ibuproxam, a molecule with a relatively simple chemical structure (see Figure 5). Note that this test case mimics the capabilities of several modern, ligand-driven DND algorithms including ones in refs 18, 24, and 26. The initial population was selected randomly from data set 1, and the 2363 building block collection was used. Since the building blocks were not related in any way with the specific similarity objective, the weights were disabled, i.e. all fragments were set to weight one. Results indicated that the MEGA algorithm meets this similarity objective. The results also demonstrated that the effect of population size is significant with larger population sizes consistently producing better results both in number of iterations required to converge and the similarity of the final products to the query molecule. Overall, the roulette selection mechanism using both mutation and crossover provided the best set of results. Time requirements for the execution of the runs were limited to less than 2 min for the lengthiest of the runs, i.e. a run with population 100, 1000 iterations, mutation, crossover, and diversity analysis enabled.

A second single-objective test aimed to design molecules based on predicted binding affinity to the ER-alpha receptor. Note that this test case corresponds to several of the original DND approaches.[2] The initial population was sampled in a quasi-random approach from data set 2. The method used atom-type descriptors[39] and the Tanimoto measure[40] to calculate the similarity of all compounds to tamoxifen and then selected randomly from the subset of compounds having similarity values less than 0.4 to tamoxifen. This guaranteed that no member of the initial population would be similar to a molecule known to bind strongly to the target receptor. The 2363 fragment collection was used; however, weights were enabled this time. Time requirements for the execution of the runs were substantially higher than in the similarity-based objective tests. A typical run with population 50, 100 iterations, mutation, crossover, and diversity analysis enabled took approximately 20 h. Results indicate that MEGA can generate solutions comparing favorably with the known solutions, i.e. the drugs designed for the specific receptors, in this case tamoxifen. This conclusion refers only to the docking scores obtained for the new compounds in comparison to tamoxifen. Figure 6 presents a designed structure with a high predicted binding affinity score to ER-alpha. Note the obvious nondruglike characteristics and potential structural issues of the design caused by focusing the exploration of the space exclusively to interaction score. This outcome exemplifies the risks associated with excessive training of the optimization process in combination with single-objective optimization. Failure to take into account any other objectives to guide the optimization, or as hard filters, may lead to overfit individuals that suffer from overspecialization to the objective under consideration.

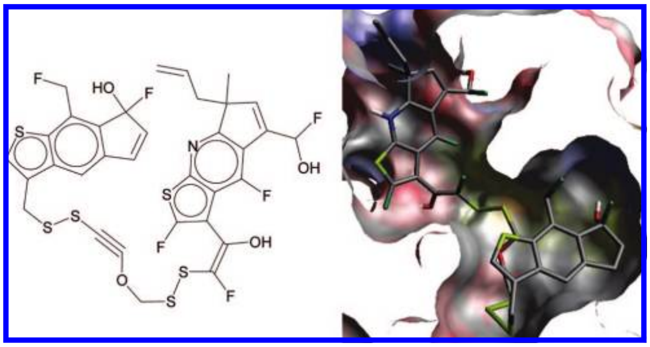


Figure 6. Results from a SOOP de novo design run with the only objective being binding score to ER-alpha. The depicted design has a significantly better score than tamoxifen, the commercial drug targeting ER-alpha. This extreme design has been selected to showcase the potential problems with optimization runs that ignore the presence of multiple objectives.

Overall, the results of MEGA for the single-objective tests demonstrate the ability of the algorithm to explore the chemical space given a clear objective to use for solution scoring. Furthermore, our tests demonstrated the importance of the population size used which consistently had a substantial effect on the quality of the solutions produced. This is probably true because the size of the population provides a greater, more diverse sample of the global search space to the algorithm to start with and a breeding environment more representative of the global search space. The number of iterations was also of substantial importance as well as the parent selection mechanism, with “roulette” typically performing better than the “best” or “tournament” selection methods. However, the latter settings could not

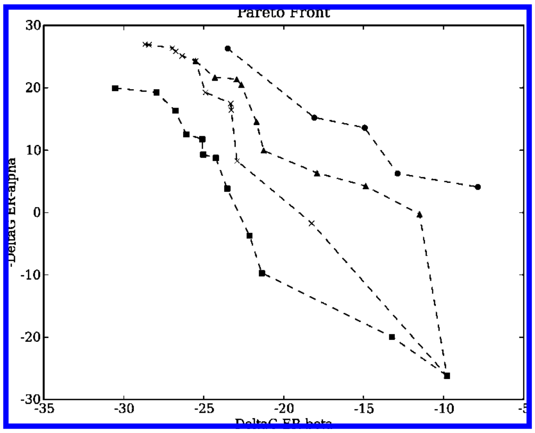


Figure 7. Pareto-front approximations with population 10 at generations 1, 20, 50, and 100 (lines with circles, triangles, x's, and squares, respectively). The objective function results were transformed for clarity to depict two minimization objectives, i.e. the ER-alpha score is inverted. Later generations tend to "move" the approximated Pareto-front toward the ideal-point (bottom left). The run shown uses the diversity-based niching mechanism but not elitism, and thus the number of Pareto-solutions is limited by the user defined population size.

compensate for the effect of a small population size. It was also observed that the use of the diversity mechanism did not really affect the quality of the single best solution produced. Finally, our findings highlighted the problem of overfitting, associated with excessive training and focus of the search process on a single objective.

MOOP Application. In a more challenging test case, appropriate for testing the abilities of MEGA, the algorithm implementations have been used to design molecules exhibiting selectivity between two target receptors. Specifically, the goal in these experiments is to discover molecules with high binding affinity toward a known target receptor site and low binding activity to another receptor. For this specific test case we used ER-beta (pdb code: 2fsz1) as the "positive" target and ER-alpha (pdb code: 1xpc), a closely related target, as a "negative" target. Additionally, a third objective imposed was similar to a known ER-alpha ligand, tamoxifen. This additional objective, encoded via the similarity scorer mechanism described previously, was applied as a hard filter to constrain the chemical structures designed to those exhibiting Tanimoto similarity to tamoxifen greater than 0.4. The similarity objective has been applied in order to favor the design of more familiar chemical structures which could facilitate validation of the testing results. A collection of chemical structure scorers, e.g. molecular weight, hydrogen-bond donors and acceptors, and number of rotatable bonds, were also used. The latter scorers, set to values generally in line to the Rule-of-Five[41] for oral bioavailability, were not used during the optimization process; rather they were applied as hard filters in each generation to remove potentially problematic designs from further consideration. In the case where the application of the hard filters resulted in fewer solutions than required by the algorithm for parent selection the filters were ignored. This only proves necessary in very few cases during the initial iterations of the algorithm. The experimental settings for the MOOP tests were similar to those of the SOOP tests performed previously. The initial population was sampled as described previously, in a quasi-random fashion from data set 2, and included populations ranging between 10 and 100. The 2363 building block collection with weights was used. The number of iterations ranged from 10 to 100. All selection mechanisms encoded were tested. Furthermore, runs were performed with and without the diversity mechanism for niching and elitism. The MOOP selectivity tests carry a heavy computational burden due to the multiple, repeated docking experiments performed at each generation. Indicatively, a MEGA run with the settings described above and population 20, 100 iterations, mutation, crossover, diversity analysis, and elitism enabled took approximately 40 h.

Apart from the solutions removed during the hard filtering step all other designs were kept. This allowed the algorithm to search for the entire Pareto-front compromising the targeted objectives. As a result we obtained solutions ranging from one extreme, i.e. high binding affinity to both receptors, to the other extreme, i.e. reduced binding affinity to both receptors. In between the two extremes several compromising solutions were obtained approximating the ideal point (Figure 7). As expected, runs with niching resulted in solutions with more diverse genotypes, i.e. chemical structures, than runs with no use of diversity. The density of the Pareto-front, i.e. the number of Pareto-solutions in between the two extremes, was heavily influenced by the use of niching and/or elitism. In runs with no diversity or elitism enabled, and runs with just diversity enabled, the size of the final Pareto-front was limited by the user-defined population size. Note that despite this limitation, the final Pareto-front produced may slightly exceed the population size depending on the number of nondominated solutions produced in the final population since MEGA retains all best performing solutions until after the termination conditions are checked. Figure 7 shows an example of a final Pareto-front produced by a MEGA run with population size 10 and the niching mechanism enabled, consisting of 12 nondominated solutions. In contrast, in runs using elitism where no such limitations exist due to the storage of the entire set of nondominated solutions found over all iterations in the Pareto-archive, a larger number of nondominated solutions, exceeding the user-defined population size, was produced. Typically, when the user-defined population size is too small, for example 10 or 20, the number of nondominated solutions in the Pareto-archive reaches the user-defined population size more quickly than when the population size is larger. The nondominated solutions were subsequently used in conjunction with the current generation population by the niching mechanism and the parent selection methods as described in the algorithmic section of this paper. The final Pareto-front approximation presented to the user was that stored in the Pareto-archive, if elitism was enabled, or the nondominated solutions of the final population. A run with a population of 20, 50 iterations using mutation, crossover, diversity analysis, and roulette selection produced a Pareto-front consisting of 11 nondominated solutions. A run with identical settings but employing elitism resulted in a Pareto-front with 23 solutions.

The solutions in the final Pareto-front provide a global view of the possible solution space to the user and enable him/her to make their selections a posteriori based on their preferences and goals. Although all the solutions of the generated Pareto-front satisfy the multiple hard filters applied and represent different compromises of the two objectives used to guide the optimization, i.e. docking scores toward ER-beta and ER-alpha, the interesting region of the Pareto-

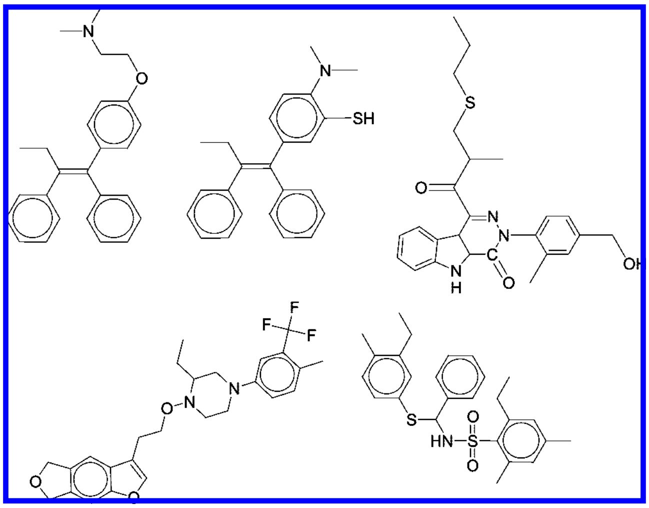


Figure 8. Tamoxifen (upper left) and a group of 4 chemical structures designed by MEGA during a run performed for the MOOP validation tests. The 4 structures are representative of the section of the Pareto-front where solutions exhibit increased docking affinity to ER-beta than ER-alpha and are sorted according to predicted selectivity potential.

front for the problem under consideration is found where solutions exhibit high docking affinity to ER-beta and substantially lower affinity to ER-alpha. For the test run described above, with elitism enabled and a resulting Pareto-front of 23 solutions, the interesting region contained 14 structures. Several of these structures have reasonable, interesting chemical designs and may serve as idea generators for further development following expert validation, while others are less promising due to reasons such as relatively poorer selectivity performance or potential structural issues. Further analysis via visual inspection and Tanimoto similarity calculations revealed that the 14 structures belong to 4 structural groups. Figure 8 presents tamoxifen and a group of 4 chemical structures, one from each of the identified structural groups. Each of the structures shown is characterized by the highest predicted selectivity potential within its group. Note the diversity of the structures, a result of the diversity analysis performed during the MEGA fitness calculation step. In order to obtain such range of solutions with traditional, single-objective methods multiple runs with different settings favoring one or the other objective at a different ratio each time could be attempted. However, even an exhaustive set of experiments with different objective weights would fail to discover solutions in certain solution surfaces.[9] Moreover, single-objective optimization runs require caution to avoid overtraining and generation of overfit individuals. MEGA, through the use of multiple objectives, can restrict the search space and prevent the survival of overspecialized solutions. Figure 9 presents one of the structures, the most similar to tamoxifen, docked in the ER-beta and ER-alpha pockets.

Once again the results obtained stressed the positive effect of a large population size. As a general rule, runs with larger population sizes consistently outperformed runs with smaller population sizes, all other parameters being equal. This observation held even for relatively small population size differences, e.g. 20 vs 25. The observations on the effect of the number of iterations and the selection mechanism used were similar to those made for the single-objective tests.

The use of the niching mechanism in combination with elitism provided a larger number of solutions covering a more extended range of compromises. In comparison, the application of MEGA without niching and elitism resulted not only in less dense Pareto-fronts as expected but also in Pareto-fronts of comparatively limited span. Interestingly, for small enough population sizes, a similar effect was sometimes obtained when only using the diversity-based niching mechanism. Further analysis indicated that the niching mechanism on its own sometimes fails to meet its goals, i.e. preserve solution diversity at the genotype level, despite the diversity analysis and sampling of the nondominated solutions it performs. The mechanism clearly works initially in identifying a diverse set of nondominated solutions, but as more and more points on the Pareto-front are produced, the algorithm necessarily drops some of the nondominated solutions because of the fixed population size allowed. This behavior may sometimes lead to dropping solutions important enough to cause loss of large sections of the Pareto-approximation surface and thus reduce diversity. However, on average, the niching mechanism proved to be effective in preventing genetic drift and domination conditions which appeared significantly more frequently when MEGA was used with no niching. It is worth noting that niching, as implemented in MEGA, enables the preservation of diversity both in genotype and phenotype space as can be seen by the span of the Pareto-approximations shown in Figure 7 and the structurally diverse solutions presented in Figure 8. Elitism, through the use of the Pareto-archive, managed to preserve diversity and produce comparatively better Pareto-solution sets in each execution run. Figure 10 presents and compares various Pareto-front approximations.

Further, our series of tests showed that MEGA with niching and elitism enabled consistently produces similar Pareto-approximations in different runs with the same

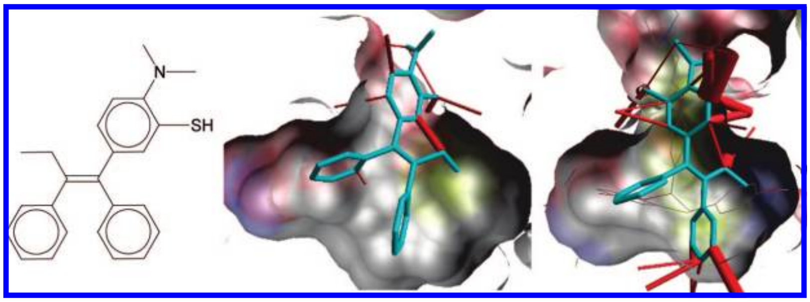


Figure 9. Sample chemical structure proposed by the algorithm. Left: The sample design in 2D format. Center: The sample docked in ER-beta. Right: The sample design docked in ER-alpha. Note that the proposed design causes several collisions (red barrels) in ER-alpha.

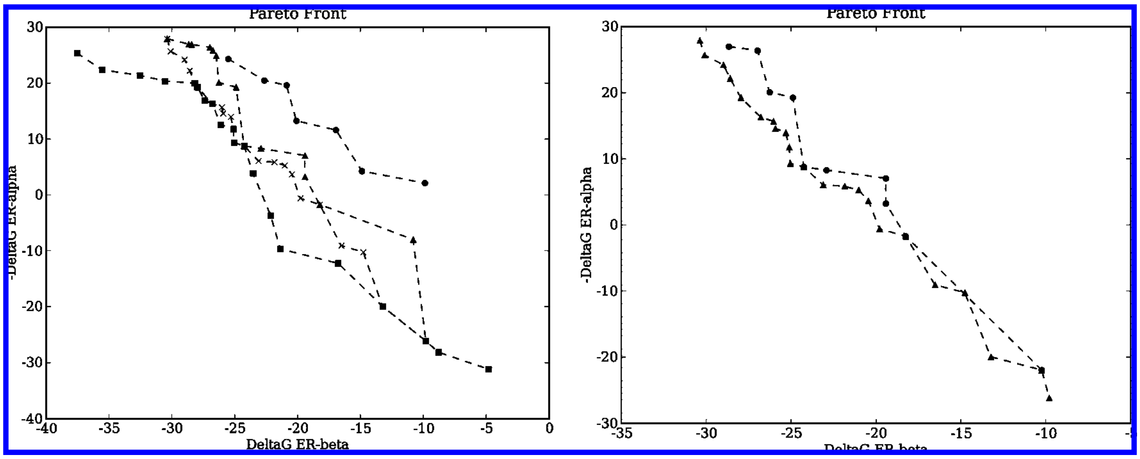


Figure 10. Left: Evolution of Pareto-front approximations obtained with elitism at iterations 1, 10, 20, and 50 (lines with circles, triangles, x's, and squares, respectively). Later generations have a more advanced (closer to the ideal point) and dense front approximation. Right: A comparison between the final Pareto-front approximations obtained with and without elitism. Note that the front with elitism (line with triangles) contains more Pareto-solutions and has a larger spread than the front without elitism (line with circles).

parameters (similarity here refers to the density and span of the generated Pareto-approximations). On the contrary, when no niching and no elitism are used, results show a much greater variation. Results obtained with runs of the algorithm using just the niching mechanism, although more consistent on average than the results of the algorithm used without the niching and elitism mechanisms, still showed considerable deviation occasionally. Overall, smaller populations tend to show greater variation than larger populations possibly explained by the inability of small populations to cover effectively a representative sample of the solution space. The set of building blocks used by the algorithm is also very important for the obtained results. Experimentation with smaller and/or unweighted building block sets, all other parameters kept equal, showed that the resulting Pareto-approximations had a more limited span and were less advanced compared to the fronts produced by runs using the 2363 weighted building block collection derived from compounds tested on the biological targets of interest. A similar conclusion was reached for the importance of the data sets used for the selection of the initial populations although to a lesser extent. These observations indicate that the sources of building blocks and the data sets used to select the initial population represent a simple, yet effective, possibility for the user to control the region of the chemical space for the search.

MEGA also proved able to cope with situations where one of the primary objectives is easier to achieve than the other. In the specific example, identifying solutions with reduced binding affinity to ER-alpha is substantially simpler than designing solutions with increased binding affinity to ER-beta. If left unattended, the uneven nature of the objectives can lead to the discovery of more solutions satisfying the easier objective, a more intense exploration of the corresponding section of the search space which in turn may result to genetic drift and potential domination conditions. Owing to its hard-filtering capabilities MEGA enabled the elimination of solutions overfit to the simpler objective, while the usage of the niching mechanism ensured that no domination conditions from solutions found in any specific region of the search space could take place. As a result, the progress of the search process as indicated by the Pareto-sets produced in successive iterations of the specific MOOP study has been even, spanning an increasingly larger range of the search space (Figures 7 and 10).

Overall the specific test case, which constitutes one of the main motivating factors for this research, showed that the MEGA algorithmic framework can be of use to the drug discovery process. The effect of various evolutionary algorithm parameters has been investigated and partially elucidated, and the contributions that both a chemical structure-based niching mechanism and an elite population, to preserve solution loss, have been demonstrated. We believe that the latter findings, i.e. the usefulness of appropriate chemical structure-based niching mechanism and an elite population in multiobjective problem settings, are particularly important.

CONCLUSIONS

Modern drug discovery process typically emphasizes potency and underestimates additional molecular properties in the early stages of lead identification and optimization.[1] Indeed, one of the common causes for lead compounds to fail in the later stages of drug discovery is the lack of adequate consideration of multiple objectives (e.g., ADMET properties) at the early stage of optimization of candidate compounds.[4] Current de novo design approaches also focus on optimizing a single property, typically similar to a known ligand or docking affinity to a receptor. In this paper we propose the use of MEGA, a custom de novo design algorithmic framework capable of optimizing several pharmaceutical objectives simultaneously.

The research presented introduces a new class of hybrid multiobjective algorithms that use graphs for solution representation and exploit available knowledge through the use of privileged fragment building blocks. Furthermore, it proposes the use of a custom niching mechanism, specifically designed to preserve chemotype diversity in the population of solutions, and elitism, to avoid loss of promising solutions. As a general framework MEGA has the ability to accommodate numerous objectives as primary, to guide the optimization process, or secondary, as hard filters, to limit the search space. Accordingly, the current MEGA implementation has been equipped with mechanisms for the assessment of receptor, ligand, and chemical structure related objectives. The results obtained indicate that the use of niching and elitism, as implemented and applied, contribute to improved performance of the proposed system. As a simple case study, we have applied our de novo design system to the significant and challenging problem of target affinity selectivity and obtained several interesting designs predicted to have a significantly higher binding affinity to ERb than the similar ERa.

FUTURE WORK

Our future work plans include both research on algorithmic enhancements as well as problem-specific applications of the system. Our immediate next step will focus on developing and integrating additional objective functions, especially ADME-Tox criteria, to guide search and optimization away from problematic chemical structures. Research in this direction will need to address issues with high-dimensional problems and Pareto-frontiers. Such problems pose additional challenges compared to low-dimensional problems in terms of the identification of a good Pareto set approximation, the computational resources required, and the choice of an appropriate solution subset from a set of alternative solutions by the end user.[46] We plan to exploit current research findings on the usage of dimensionality reduction techniques to decrease the number of objectives,[46,47] to develop methods to assess the usefulness of each objective function in conjunction with other objectives, and to exploit potential objective orthogonality and redundancy issues. A related effort will concentrate on adding further domain knowledge exploitation capabilities to MEGA in the form of rules and local search techniques. Of special interest and priority is the issue of synthetic feasibility of the chemical structures proposed by the algorithm. Although MEGA includes some elements that may favor designs with higher synthesizability potential, such as the storage of attachment point bond types and the exploitation of this information during evolutionary design, efforts in this direction are still inadequate. As a result the algorithm may produce designs with obvious structural problems, and thus expert validation of the final outcome of the program is essential. Currently, we are exploring the addition of a virtual synthesis engine that will utilize available domain knowledge encoded in the form of synthetic rules. An additional direction of research will investigate the parallelization of the algorithm implementation to enable the application of the system on large scale distributed systems (e.g., the grid) and handling a considerably larger number of objectives and search spaces. On the application front, effort will be placed on applying the system as part of a pharmaceutical project through collaboration with expert partners a development that will enable actual experimental laboratory validation of our results.

ACKNOWLEDGMENT

The authors are grateful to Christos Kannas for help with substructure mining tasks and Dr. Emmanuel Mikros and Dr. Anna Tsantili-Kakoulidou for helpful discussions and their expert opinions. We also wish to thank Simon Tietze for setting up the proteins for the docking experiments.

REFERENCES AND NOTES

(1) Ekins, S.; Boulanger, B.; Swaan, P. W.; Hupcey, M. A. Z. Towards a New Age of Virtual ADME/TOX and Multidimensional Drug Discovery. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 381–401.

(2) Schneider, G.; Fechner, U. Computer-based de novo design of druglike molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649–663.

(3) Brown, N.; McKay, B.; Gilardoni, F.; Gasteiger, J. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1079–1087.

(4) Nicolaou, C. A.; Brown, N.; Pattichis, C. S. Molecular optimization using computational multiobjective methods. *Curr. Opin. Drug Discovery Dev.* **2007**, *10*, 316–324.

(5) Tietze, S.; Apostolakis, J. GlamDock: development and validation of a new docking tool on several thousand protein-ligand complexes. *J. Chem. Inf. Model.* **2007**, *47*, 1657–1672.

(6) Marshall, G. R. Introduction to Chemoinformatics in Drug Discovery - A Personal View. In *Chemoinformatics in Drug Discovery*; Oprea, T., Ed.; Wiley-VCH: Weinheim, Germany, 2004; pp 1−22.

(7) Baringhaus, K.-H.; Matter, H. Efficient strategies for lead optimization by simultaneously addressing affinity, selectivity and pharmacokinetic parameters. In *Chemoinformatics in Drug Discovery*; Oprea, T., Ed.; Wiley-VCH: Weinheim, Germany, 2004; pp 333−379.

(8) Xu, J.; Hagler, A. Chemoinformatics and drug discovery. *Molecules* **2002**, *7*, 566–600.

(9) Colette, Y.; Siarry, P. *Multiobjective Optimization: Principles and Case Studies*; Springer-Verlag: Berlin, Germany, 2004.

(10) Handl, J.; Kell, D. B.; Knowles, J. Multiobjective Optimization in Bioinformatics and Computational Biology. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2007**, *4*, 279–292.

(11) van Veldhuizen, D. A.; Lamont, G. B. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evol. Comput.* **2000**, *8*, 125–147.

(12) Zitzler, E.; Laumanns, M.; Bleuler, S. A Tutorial on Evolutionary Multiobjective Optimization. In *Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems*; Gandibleux, X., Sevaux, M., Sörensen, K., T'kindt, V., Eds.; Springer: Berlin, 2004; Vol. 535, pp 3−37.

(13) Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolutions Programs*; Springer-Verlag: New York, 1992.

(14) Weise, T. *Global Optimization Algorithms - Theory and Application.*http://www.it-weise.de/ (accessed August 8, 2008).

(15) Poli, R.; Langdon, W. B.; McPhee, N. F. *A field guide to genetic programming [Online]; Lulu:com*: U.K., 2008. http://www.gp-field-guide.org.uk (accessed July 27, 2008).

(16) Eloranta, T.; Makinen, E. TimGA: A Genetic Algorithm for Drawing Undirected Graphs. *Divulgaciones Matematicas* **2001**, *9*, 155–171.

(17) Miller, J. F.; Job, D.; Vassilev, V. K. Principles in the Evolutionary Design of Digital Circuits -- Part I. *J. Genet. Program. Evolvable Mach.* **2000**, *1*, 8–35.

(18) Globus, A.; Lawton, J.; Wipke, W. T. Automatic Molecular design using evolutionary algorithms. *Nanotechnology* **1999**, *10*, 290–299.

(19) Bohacek, R. S.; Martin, C.; Guida, W. C. The Art and Practice of Structure-Based Drug Design: A Molecular Modelling Approach. *Med. Res. Rev.* **1996**, *16*, 3–50.

(20) Clark, D. E.; Frenkel, D.; Levy, S. A.; Li, J.; Murray, C. W.; Robson, B.; Waszkowycz, B.; Westhead, D. R. PRO LIGAND: an approach to *de novo* molecular design. 1. Application to the design of organic molecules. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 13–32.

(21) Westhead, D. R.; Clark, D. E.; Frenkel, D.; Li, J.; Murray, C. W.; Robson, B.; Waszkowycz, B. PRO_LIGAND: An approach to de novo molecular design. 3. A genetic algorithm for structure refinement. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 139–148.

(22) 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.

(23) Douguet, D.; Munier-Lehmann, H.; Labesse, G.; Pochet, S. LEA3D: A Computer-Aided Ligand Design for Structure-Based Drug Design. *J. Med. Chem.* **2005**, *48*, 2457–2468.

(24) Schneider, G.; Lee, M.-L.; Stahl, M.; Schneider, P. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 487–494.

(25) Fechner, U.; Schneider, G. Flux (1): A Virtual Synthesis Scheme for Fragment-Based De Novo Design. *J. Chem. Inf. Model.* **2005**, *46*, 699–707.

(26) Fechner, U.; Schneider, G. Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design. *J. Chem. Inf. Model.* **2007**, *47*, 656–667.

(27) Lewell, X. O.; Budd, D. B.; Watson, S. P.; Hann, M. M. RECAP - Retrosynthetic Combinatorial Analysis Procedure: a powerful new technique for identifying privileged molecular fragments with useful

applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511–522.

(28) Nachbar, R. B. Molecular evolution: automated manipulation of hierarchical chemical topology and its application to average molecular structures. *Genet. Programming Evolvable Mach.* **2000**, *1*, 57–94.

(29) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

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

(31) 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.

(32) Lameijer, E.-W.; Kok, J. N.; Baeck, T.; Ijzerman, A. P. The Molecule Evoluator. An Interactive Evolutionary Algorithm for the Design of Druglike Molecules. *J. Chem. Inf. Model.* **2006**, *46*, 545–552.

(33) Brown, N.; McKay, B.; Gasteiger, J. A novel workflow for the inverse QSPR problem using multiobjective optimization. *J. Comput.-Aided Mol. Des.* **2006**, *20*, 333–341.

(34) Lameijer, E. W.; Baeck, T.; Kok, J. N.; Ijzerman, A. P. Evolutionary algorithms in drug design. *Nat. Comput.* **2005**, *4*, 177–243.

(35) Clark, D. E.; Westhead, D. R. Evolutionary algorithms in computer-aided molecular design. *J. Comput.-Aided Mol. Des.* **1996**, *10*, 337–358.

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

(37) Noesis Chemoinformatics, Ltd. Nicosia, Cyprus. http://www.noesis-informatics.com (accessed Feb 5, 2008).

(38) Nicolaou, C. A.; Pattichis, C. S. Molecular Substructure Mining Approaches for Computer-Aided Drug Discovery: A Review. In *Proceedings of ITAB*; Ioannina, Greece, October 26−28, 2006.

(39) Kearsley, S. K.; Sallamack, S.; Fluder, E. M.; Andose, J. D.; Mosley, R. T.; Sheridan, R. P. Chemical Similarity Using Physiochemical Property Descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 118–127.

(40) Willet, P.; Barnard, J. M.; Downs, G. M. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *39*, 983–996.

(41) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability. Drug Discovery and Development Settings. *Adv. Drug Discovery Rev.* **1997**, *23*, 3–25.

(42) Wild, D. J.; Blankley, C. J. Comparison of 2D fingerprint types and hierarchy level selection methods for structural grouping using wards clustering. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 155–162.

(43) Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271.

(44) Stahl, M.; Rarey, M. Detailed Analysis of Scoring Functions for Virtual Screening. *J. Med. Chem.* **2001**, *44*, 1035–1042.

(45) Wheeler, D. L.; Barrett, T.; Benson, D. A.; Bryant, S. H.; Canese, K.; Chetvernin, V.; Church, D. M.; DiCuccio, M.; Edgar, R.; Federhen, S.; Geer, L. Y.; Helmberg, W.; Kapustin, Y.; Kenton, D. L.; Khovayko, O.; Lipman, D. J.; Madden, T. L.; Maglott, D. R.; Ostell, J.; Pruitt, K. D.; Schuler, G. D.; Schriml, L. M.; Sequeira, E.; Sherry, S. T.; Sirotkin, K.; Souvorov, A.; Starchenko, G.; Suzek, T. O.; Tatusov, R.; Tatusova, T. A.; Wagner, L.; Yaschenko, E. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **2006**, *34*, D173–D180.

(46) Brockhoff, D.; Saxena, D. K.; Deb, K.; Zitzler, E. On Handling a Large Number of Objectives A Posteriori and During Optimization. In *Multi-Objective Problem Solving from Nature: From Concepts to Applications*; Knowles, J., Corne, D., Deb, K. Eds.; Springer: Berlin, 2007; pp 377−403.

(47) Deb, K.; Saxena, D. Searching For Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. In *Proceedings of the World Congress on Computational Intelligence*; IEEE Press: Vancouver, Canada, 2006; pp 3352−3360.

CI800308H