ARTICLES

A Scalable Approach to Combinatorial Library Design for Drug Discovery

Puneet Sharma,*[†,§] Srinivasa Salapaka,[‡,§] and Carolyn Beck[†,§]

Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana Champaign, 104 S. Mathews Avenue, Urbana, Illinois 61801, Department of Mechanical Science and Engineering, 1206 W. Green Street, Urbana, Illinois 61801, and Coordinated Science Laboratory, University of Illinois at Urbana Champaign, 1308 W. Main Street, Urbana, Illinois 61801

Received January 23, 2007

In this paper, we propose an algorithm for the design of lead generation libraries required in combinatorial drug discovery. This algorithm addresses *simultaneously* the two key criteria of diversity and representativeness of compounds in the resulting library and is computationally efficient when applied to a large class of lead generation design problems. At the same time, additional constraints on experimental resources are also incorporated in the framework presented in this paper. A computationally efficient scalable algorithm is developed, where the ability of the deterministic annealing algorithm to identify clusters is exploited to truncate computations over the entire data set to computations over individual clusters. An analysis of this algorithm quantifies the tradeoff between the error due to truncation and computational effort. Results applied on test data sets corroborate the analysis and show improvement by factors as large as 10 or more, depending on the data sets.

1. INTRODUCTION

In recent years, combinatorial chemistry techniques have provided some of the most important tools for assisting chemists in the exploration of huge chemical property spaces in search of new pharmaceutical agents. With the advent of these methods, a large number of compounds are accessed and synthesized using basic building blocks. Recent advances in high-throughput screening approaches such as those using micro-/nanoarrays[1] have given further impetus to large-scale investigation of compounds for drug discovery. However, combinatorial libraries often consist of extremely large collections of chemical compounds, typically several millions. The time and cost associated with these experiments makes it practically impossible to synthesize each and every combination from such a library of compounds. To overcome this problem, chemists often work with *virtual combinatorial libraries*, which are essentially combinatorial databases containing an enumeration of all possible structures of a given pharmacophore with all available reactants. They then select a subset of compounds from this virtual library of compounds and use it for physical synthesis and biological target testing. The selection of this subset is based on a complex interplay between various objectives, which is cast as a combinatorial optimization problem. To address this problem, computational optimization tools have been employed to design libraries consisting of subsets of representative compounds which can be synthesized and subsequently tested for relevant properties, such as structural activity, bioaffinity, and aqueous solubility.

At the outset, the problem of designing such a *lead generation library* belongs to the class of combinatorial resource allocation problems, which have been widely studied. These problems are typically cast as multiobjective optimization problems with constraints. They arise in many different areas such as minimum distortion problems in data compression,[2] facility location problems,[3] optimal quadrature rules and discretization of partial differential equations,[4] locational optimization problems in control theory,[5,6] pattern recognition,[7] neural networks,[8] and clustering analysis.[9] These problems are nonconvex and computationally complex. It is well-documented[10] that most of them suffer from many local minima that riddle the cost surface. The feature these combinatorial resource allocation problems share is the goal of an optimal partition of the underlying domain together with an optimal assignment of values from a finite set to each cell of the partition. The distinguishing features of this class of problems come from having different conditions for optimality and constraints, such as the distance metrics used in the definition of *coverage*, the number and types of constraints placed on the resources in the problem formulation, the necessity of computing global versus local optima, and the size and scale of the feasible domain. These differences add further problem-specific challenges to the underlying combinatorial optimization problems.

The lead generation design problem is viewed as a combinatorial resource allocation problem since it requires partitioning the underlying chemical property space spanned by compounds in the virtual library and ascribing a representative compound (i.e., a member of the lead generation

* Corresponding author phone: (217) 244 3364; e-mail: psharma2@uiuc.edu.

[†] Department of Industrial and Enterprise Systems Engineering.

[‡] Department of Mechanical Science and Engineering.

[§] Coordinated Science Laboratory.

10.1021/ci700023y CCC: $40.75
Published on Web 12/06/2007

library) to each cell in the above partition. The main criterion for constructing a lead generation library has traditionally been molecular *diversity*.[11,12] As has been noted in the literature,[13,14] diversity as a sole criterion for selection can yield lead generation libraries with compounds that have limited *druglike* properties, as this selection procedure disproportionately favors outliers or atypical compounds. Though such a library contains maximally diverse compounds, it may be of little significance because of its limited pharmaceutical applications. This drawback can be addressed by designing libraries that are *representative* of all potential compounds and their distribution in the property space.[15−17] Thus, a good lead generation library design will have a manageable number of compounds with adequate diversity so that atypical compounds are duly represented to ensure the synthesis of almost all potential drugs. At the same time, it will be representative of the distribution of compounds in the virtual library. In addition to diversity and representativeness, other requirements may also be considered that further constrain the library design; for example, it may be required that the library contains compounds that exhibit specific druglike properties.[14,18,19]

The specific challenges, in addition to that of combinatorial optimization, that lead generation design poses are (1) designing cost functions that reflect the notions of diversity and representation, (2) designing algorithms that are scalable in order to handle large data sets, and (3) incorporating specific constraints on compounds of lead generation (such as constraints coming from limits on experimental resources).

Different multiobjective optimization methods have been used previously for designing libraries. Stochastic methods such as simulated annealing[20−23] and genetic algorithms[24,25] have been proposed. Most of the stochastic methods attempt to avoid getting stuck in local minima by generating an ensemble of states at every point and assigning a nonzero probability of transition to an “inferior” solution. Multiobjective approaches to the problem are useful for including several different design criteria in the algorithm. Taking into consideration the size of these combinatorial libraries, it becomes essential to consider the scaling issues involved with any algorithm that solves the multiobjective optimization problems. Unfortunately, most of the methods mentioned above do not address the key issues of diversity and representativeness simultaneously, and these algorithms are not scalable and often underperform when data sets are large.

This paper presents an algorithm for lead generation library design (i.e., selecting a subset of compounds from a larger set of virtual compounds) which accounts simultaneously for diversity as well as representativeness by formulating a combinatorial resource allocation problem with constraints. New procedures are presented that address the issues of scalability and various constraints on the compounds in the lead generation library. We present an algorithm based on the concept of deterministic annealing (DA)[26,27] to cater to the specific constraints and demands of combinatorial chemistry. The distinctive feature of the DA algorithm is that it successfully avoids local minima. At the same time, it is typically faster than simulated annealing.[28] We propose a scalable algorithm, which preserves the desirable features of DA and, at the same time, addresses the issue of quadratic complexity.[29]

This paper is organized as follows. In section 2, we provide background information on some of the specifics of combinatorial library design and the key issues that are to be tackled while designing an algorithm. We then provide a mathematical formulation of the library design problem. The underlying approach we employ for the solution of the selection problem, that is, the DA algorithm, is described in section 3.1. We then discuss our extensions and modifications to the DA algorithm in section 3.2. The resulting scalable algorithm, together with the tradeoff bounds, is discussed in section 3.3. Simulation results on a number of test data sets and comparisons with the nonscalable algorithm are presented in section 4. Finally, we conclude the paper by revisiting the key results obtained and identifying future goals.

2. BACKGROUND AND PROBLEM FORMULATION

Library design refers to the process of screening and then selecting a subset of compounds from a given set of similar or distinct compounds (a virtual combinatorial library) for drug discovery.[30] The combinatorial nature of the selection problem makes it impractical to exhaustively enumerate each and every possible subset to obtain the optimal solution. For example, in order to select 30 representative compounds from a set of 1000, we have ${}^{1000}C_{30}$ possibilities, that is, approximately 3×10^{25} different possible combinations from which to choose. This makes the selection (based on enumeration) impractical and calls for efficient algorithms that make the optimal selection under given constraints. The main aim of library design is to reduce the number of compounds for testing without compromising on the diversity, representativeness, and properties such as druglike behavior in the subset chosen. It facilitates the process of drug discovery by replacing the synthesizing and subsequent testing of all compounds by a much smaller set of representative compounds. On the basis of the current state of development in the drug discovery process, library design is broadly classified into two main categories, namely, lead generation and lead optimization. The main purpose of these libraries and their design criteria are discussed below.

2.1. Lead Generation Library Design. The development of a lead generation library usually involves the design and synthesis of a large number of chemical compounds. It is required that these compounds be different from each other. The library containing these diverse compounds is then tested against a host of different biological agents. The main objective in designing such libraries is to obtain structurally diverse compounds so as to be representative of the entire chemical space.

Lead optimization libraries are usually designed at a later stage of the drug discovery process, when it is required to select a subset of compounds that are *similar* to a given lead compound(s). This results in an array of compounds which are structurally and chemically similar to the lead. The criterion of similarity is generally used for designing targeted or focused libraries, with a single therapeutic target in mind.

2.1.1. Issues in Lead Generation Library Design. This paper deals with the problem of designing a library of compounds for the purpose of lead generation. The most common method used to obtain such a library is to maximize the diversity of the overall selection.[11,12] It is based on the

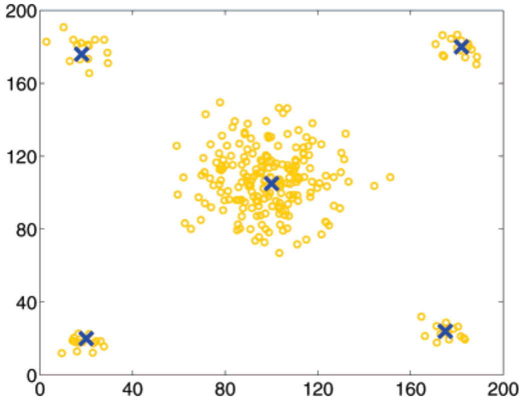

Figure 1. A scenario depiciting the inherent problems with the "diversity" only criterion for lead generation library design.

strategy that the more diverse the set of compounds, the better the chance to obtain a lead compound with desired characteristics. As was noted earlier,[13,14] such a design strategy suffers from an inherent problem which occurs due to the fact that using diversity as the only criterion may result in a set of compounds which are exceptions or singletons. Figure 1 shows such a scenario. Here, the circles represent the distribution of compounds in a descriptor space, while the crosses denote compounds chosen according to the maximum diversity principle. As can be seen from the figure, the cluster in the middle is not adequately represented in the final selection. This selection focuses disproportionately on distant compounds, which may be viewed as exceptions. From a drug discovery point of view, it is desirable for the lead generation library to contain more compounds from the middle cluster (so as to adequately represent all the compounds) or to at least determine how proportionately representative they are in order to make decisions on the amount of experimental resources to devote to these lead compounds. A maximally diverse subset is of little practical significance because of its limited pharmaceutical applications. Hence, the criterion of representativeness should be considered along with diversity.[15,16,31] To address this problem, we present an algorithm which selects a lead generation library that achieves high diversity and at the same time specifies how representative (what percent of all the compounds) each member in the library is. The process involves identifying different representative compound locations in an iterative fashion and is discussed in following sections.

The large size of combinatorial libraries calls for a scalable selection algorithm. We propose a *divide-and-conquer* scheme for the implementation of the algorithm in which the underlying property space of the compounds is recursively and hierarchically partitioned into subdomains that are approximately isolated from each other. The scheme quantifies the degree of isolation or equivalently the extent of interaction between the subdomains after each partitioning step. This interaction term is used in specifying bounds on the deviation from the results had the original DA algorithm been used on the entire property space. The implementation of the selection algorithm on each of these subsets that are smaller in size than their parents in the hierarchy substantially reduces the computational time. The proposed algorithm is easily adaptable to include additional criteria needed in the design of lead generation libraries. In addition to similarity and diversity, other criteria include *confinement*, which quantifies the degree to which the properties of a set of compounds lie between prescribed upper and lower ranges,[32] and maximization of the *activity* of the set of compounds against some predefined targets. Activity is usually measured in terms of the quantitative structure of the given set. The presence of these multiple (and often conflicting) design objectives makes the library design a multiobjective optimization problem with constraints.

2.2. Resource Allocation Problems. As was noted earlier, the lead generation library design problem belongs to the class of combinatorial resource allocation problems. In order to quantify the different criteria used for designing libraries (namely, diversity and similarity), it is required to define appropriate *molecular descriptors* which numerically characterize the various compounds in a chemical space.

2.2.1. Molecular Descriptors. Molecular descriptors are essential for quantifying the properties of different compounds in a chemical space. One-, two-, and three-dimensional descriptors are used to encode the chemical composition, chemical topology, and three-dimensional shape.[33] A large number of descriptors can be used to effectively characterize a chemical space. In practice, many of these descriptors are correlated, and tools from dimensionality reduction (nonlinear mapping and principal component analysis methods[34,35]) are used to extract lower dimensional representations that preserve the *neighborhood behavior*. It has been previously shown that a two-dimensional molecular descriptor space exhibits a proper neighborhood behavior which is essential for characterizing similarity and diversity properties between two compounds.[23] In all of our simulations, we consider a 2D or 3D molecular descriptor space. The Euclidean distance between two points in the space provides a good measure of the degree of similarity between these two compounds. Thus, in this scenario of a 2D descriptor space, the close neighbors of an active compound will also be active. On the other hand, two compounds which are far apart (in terms of the Euclidean distance) can be labeled as diverse. The 2D descriptor space provides a means of quantifying the properties of different compounds (for more details see ref 23).

2.2.2. Problem Formulation. In its prototypical form, the problem of selecting representative compounds for the purpose of library design can be stated as follows.

Given a distribution of N compounds, x_i, in a descriptor space Ω, find the set of M representative compounds, r_j, that solves the following minimization problem:

$$\min_{r_j, 1 \le j \le M} \sum_{i=1}^{N} p(x_i)\{\min_{1 \le j \le M} d(x_i, r_j)\} \tag{1}$$

Here $d(x_i,r_j)$ represents an appropriate *distance* metric between the representative compound r_j and the compound x_i, $p(x_i)$ is the relative weight that can be attached to compound x_i (if all compounds are of equal importance, then the weights $p(x_i) = 1/N$ for each i), and M is typically much smaller than N. Alternatively, this problem can also be formulated as finding an *optimal* partition of the descriptor space Ω into M cells R_j and assigning to each cell R_j a representative compound r_j such that the following cost function is minimized:

$$\sum_{j=1}^{M} \sum_{x_i \in R_j} d(x_i,r_j)\, p(x_i)$$

3. A SCALABLE ALGORITHM FOR MULTIOBJECTIVE COMBINATORIAL LIBRARY DESIGN

As was presented in section 2.2, the lead generation library design problem can be viewed as a combinatorial resource allocation problem with constraints. Due to the large size of such libraries, the solution calls for an algorithm that scales up efficiently with the size of the underlying domain. The DA algorithm[26,27] was introduced to solve the combinatorial resource allocation problem (without addressing the scaling issues). A brief overview of the DA algorithm is presented in section 3.1. Our scalable algorithm is based on the original concepts of the DA algorithm, which we outline here.

3.1. Deterministic Annealing Algorithm. Realistic objective functions in the context of lead library selection problems have unpredictable surfaces with many local minima and, thus, require algorithms designed to avoid these local minima. The DA algorithm is suited for this purpose since it is specifically designed to avoid local minima. This algorithm can be viewed as a modification of another algorithm called Lloyd's algorithm,[2,36] which captures the essence of many other algorithms typically employed for combinatorial resource allocation problems. Lloyd's algorithm is an iterative method which identifies two necessary conditions of the optimal solution and then ensures that, at each iteration, the partition of the domain and the representative compounds satisfy these conditions, which are as follows:

1. Nearest-neighbor condition (Voronoi partitions). The partition of the descriptor space is such that each compound in the descriptor space is associated with the nearest representative compound.

2. Centroid condition. The location of each representative compound, r_j, is determined by the centroid of the jth cell R_j.

In this algorithm, the initial step consists of randomly choosing locations of representative compounds and then successively iterating between the steps: (1) forming Voronoi partitions and (2) moving the representative compounds to respective centroids of cells until the sequence of locations of representative compounds converge. It should be noted that the solution depends substantially on the initial allocation as the locations in the successive iterations are influenced only by *near* points of the domain and are virtually independent of *far* points. As a result, the solutions from this algorithm typically get stuck in local minima.

The DA algorithm diminishes this local influence of domain members by allowing each member $x_i \in \Omega$ to be associated with every representative compound r_j through a weighting parameter $p(r_j|x_i)$. This weighting parameter can also be interpreted as an association probability between r_j and x_i. Thus, this algorithm does away with the hard partitions of Lloyd's algorithm. The DA formulation includes minimizing a *distortion* term:

$$D = \sum_{i=1}^{N} p(x_i) \sum_{j=1}^{M} d(x_{i,} r_j)\, p(r_j|x_i)$$

which is similar to the cost function in eq 1. The term D is referred to as distortion because of its parallels in rate-distortion theory.[37] In addition to the distortion term, it also includes an entropy term (H), given by

$$H = -\sum_{i=1}^{N} p(x_i) \sum_{j=1}^{M} p(r_j|x_i) \log p(r_j|x_i)$$

which measures the randomness of distribution of the associated weights. The entropy is the highest when the distribution of weights over each representative compound is the same [$p(r_j|x_i) = 1/M$] for each x_i, that is, when all x_i values have the same influence over every representative compound. Maximizing the entropy is the key for selecting a maximally diverse subset of compounds. The notion of diversity in a given library of compounds has been previously characterized using concepts from information theory.[38] In these approaches, diversity is related to the information content in a given subset (quantified by Shannon's entropy[39]).

The DA algorithm solves the following multiobjective optimization problem over iterations indexed by k, where

$$\min_{r_j} \min_{p(r_j|x_i)} \underbrace{D - T_k H}_{:=F}$$

T_k is a parameter called *temperature* which tends to zero as k tends to infinity. The cost function F is called *free energy* as this formulation has an analog in statistical physics.[40] Clearly, for large values of T_k, we mainly attempt to maximize the entropy. As T_k is lowered, we trade entropy for the reduction in distortion, and as T_k approaches zero, we minimize D directly to obtain a hard (nonrandom) solution. T_k can be regarded as the Lagrange parameter in this multiobjective optimization problem. Minimizing the free energy term F with respect to the weighting parameter $p(r_j|x_i)$ is straightforward and gives the *Gibbs* distribution

$$p(r_j|x_i) = \frac{e^{-d(x_i,r_j)/T_k}}{Z_i}$$

where

$$Z_i := \sum_j e^{-d(x_i,r_j)/T_k} \qquad (2)$$

Z_i is called the *partition function*. Note that the weighting parameters $p(r_j|x_i)$ are simply radial basis functions, which clearly decrease in value exponentially as r_j and x_i move farther apart. The corresponding minimum of F is obtained by substituting for $p(r_j|x_i)$ from eq 2

$$\hat{F} = -T_k \sum_i p(x_i) \log Z_i \qquad (3)$$

To minimize $\hat{F}$ with respect to the representative compounds $\{r_j\}$, we set the corresponding gradients equal to zero, that is, ($\partial\hat{F}/\partial r_j = 0$); this yields the corresponding implicit equations for the locations of representative compounds

$$r_j = \sum_i p(x_i|r_j)\, x_i, \qquad 1 \le j \le M$$

where

$$p(x_i|r_j) = \frac{p(x_i)\, p(r_j|x_i)}{\sum_k p(x_k)\, p(r_j|x_k)} \qquad (4)$$

Note that $p(x_i|r_j)$ denotes the posterior probability calculated using Bayes's rule and the above equations clearly convey the *centroid* aspect of the solution.

The DA algorithm consists of minimizing $\hat{F}$ with respect to $\{r_j\}$ starting at high values of T_k, and then tracking the minimum of $\hat{F}$ while lowering T_k. The steps at each k are as follows:

1. Fix $\{r_j\}$ and use eq 2 to compute the new weights $\{p(r_j|x_i)\}$
2. Fix $\{p(r_j|x_i)\}$ and use eq 4 to compute the representative compound locations $\{r_j\}$

Representative compound locations r_j together with weighting parameters $p(r_j|x_i)$ define the soft clustering solution to the multiobjective clustering problem. As the temperature T_k is lowered further, these associations give way to hard partitions, finally resulting in a solution where each x_i belongs to one of the clusters with unit weight.

Remark. The *annealing* mechanism in the DA algorithm enables it to successfully overcome the drawbacks of the Lloyd's type iterative process (*K*-means problems[41]), namely, its dependence on initial placement of cluster centers and its inability to avoid local minima. Thus, the DA algorithm results in soft associations at each temperature level and reduces temperature till hard associations are formed (at zero temperature), thereby solving the original problem.

At a fixed value of temperature, each iteration of the DA algorithm is essentially equivalent to the Lloyd's algorithm. In fact, the DA algorithm becomes equivalent to a *K*-means algorithm at asymptotic values of temperature (that is when the temperature variable goes to zero), with the representative compound locations, obtained from the previous iteration, close to the centers of natural cluster. This clustering result could not have been achieved by a K-means algorithm because of an inability to navigate away from local minima. Thus each iteration in DA provides us with a better choice for the representative compound locations, to be used in the next iteration.

3.2. Algorithm Modifications for Constraints on Representative Compounds and Experimental Resources. In the original DA formulation, the individual representative compounds are indistinguishable since each of them carry equal weight. However, capacity constraints often distinguish one representative compound from another in practical situations. In order to account for such capacity constraints, it is necessary to modify the DA algorithm. For the specific problem of library design, one such scenario occurs when we want to address the issue of the representativeness of individual clusters in the final library design. In order to constrain the size of each cluster, it becomes necessary to distinguish between the various locations of representative compounds. Moreover, quantity constraints on the experimental resources also call for a modification of the DA algorithm. In this section, we present two modifications of the original DA algorithm for tackling these issues.

3.2.1. Incorporating Representativeness. The necessity for quantifying and incorporating representativeness is evident, as discussed in section 2.1. We incorporate representativeness while identifying diverse compounds by specifying an additional parameter $\lambda_j, 1 \leq j \leq M$, for each compound in the lead generation library. This parameter λ_j quantifies the extent of representativeness of the compound location r_j; that is, it gives a measure of the number of compounds in the underlying data set that are represented by the compound location r_j. Thus, the resulting library design will have compounds that will cover the entire data set, where the compounds in the library that cover outliers will have low representative weights and the compounds that cover high-volume regular members in the data set will have high representative weights. In this way, the algorithm can be used to identify diverse compounds through locations r_j and at the same time determine how representative each compound in the library is.

This representativeness criterion is incorporated into the algorithm by reinterpreting the DA algorithm. The DA algorithm as presented above assumes that all the representative compounds are identical. However, in the new interpretation, each location r_j can be weighted by λ_j, which translates to the modified partition function in the algorithm:

$$Z_i = \sum_{j=1}^{M} \lambda_j\, \mathrm{e}^{-d(x_i,r_j)/T_k} \qquad (5)$$

The parameters λ_j, $1 \leq j \leq M$, give relative weights (and therefore relative representativeness) of locations r_j, $1 \leq j \leq M$,[5,26] and without a loss of generality can be assumed to sum to 1. In this denotation, $\lambda_j = 0.2$, for example, would mean that the representative compound r_j in the library represents 20% of the compounds in the data set. The modified algorithm solves for locations r_j, which maximize the diversity as before and, in addition, finds the weights λ_j that specify representativeness by solving the following constrained minimization problem:

$$\min_{r_j, 1\leq j\leq M} \sum_i p(x_i)\, \{\min_{1\leq j\leq M} d(x_i,r_j)\}$$

such that

$$\sum_{j=1}^{M} \lambda_j = 1$$

Note that, although the constraints do not seem to occur in the cost function explicitly, they can be interpreted in terms of the DA algorithm where the partition function in eq 2 is modified as in eq 5. These weighting constraints lead to modified Gibbs distribution and free energy terms. Following a procedure similar to that used in the original algorithm, we get

$$p(r_j|x_i) = \frac{\lambda_j\, \mathrm{e}^{-d(x_i,r_j)/T_k}}{\sum_j \lambda_j\, \mathrm{e}^{-d(x_i,r_j)/T_k}}$$

At each value of the temperature, the representative compound location r_j and the weight λ_j are given by

$$r_j = \sum_i p(x_i|r_j)\, x_i$$

$$\lambda_j = \sum_i p(r_j|x_i) p(x_i),\ 1 \le j \le M$$

Thus, we obtain locations for library design r_j and their representative weights λ_j. Experimental results in section 4 demonstrate the efficacy of these algorithms, where the selection is depicted by location r_j in the descriptor space and the corresponding weights λ_j are shown in pie charts.

3.2.2. Incorporating Constraints on Experimental Resources. The cost of chemical compounds and experimental resources is significant and presents one of the main impediments in combinatorial diagnostics and drug synthesis. In fact, recent research in nanoinstrumentation has led to nanoarrays which bring together the many capabilities crucial for rapid and high-volume research and production—including design, prototyping, assembly, testing, and reliable replication at the specificity of biomolecules.[1,42] One of the main advantages of this approach is its economy, since very small amounts of biological and chemical agents are needed, thereby reducing the cost considerably. Still, different compounds require different experimental supplies which are typically available in limited quantities. In this section, we include these constraints into our algorithm for lead generation, as described below.

The library is classified into q types corresponding to experimental supplies required by the compounds for testing. We incorporate the supply constraints into the algorithm by translating them into direct constraints on each of the representative compounds. For example, the jth representative compound can avail only an amount of the nth resource equal to W_{jn}. This type of a constraint is generally referred to as a multicapacity constraint.[5]

The modified optimization problem then is given by

$$\min_{r_j} D = \sum_n \sum_i p_n(x_i{}^n) \sum_{j=1}^{M} d(x_i{}^n, r_j)\, p(r_j|x_i{}^n) \qquad (6)$$

such that

$$\lambda_{jn} = W_{jn}, \qquad 1 \le j \le M, \qquad 1 \le n \le q \qquad (7)$$

where $p_n(x_i{}^n)$ is the weight of the compound location $x_i{}^n$, which requires the nth type of experimental supply, and W_{jn} is the amount of the nth supply that the jth representative compound can avail.

We proceed along the same lines as the DA algorithm by defining the entropy term by

$$H = -\sum_n \sum_i p_n(x_i{}^n) \sum_{j=1}^{M} p(r_{j|}x_i{}^n) \log p(r_j|x_i{}^n)$$

and minimizing the corresponding free energy, given by $F = D - T_k H$. This procedure yields the Gibbs distribution of the form

$$p(r_j|x^n) = \frac{\lambda_{jn}\, \mathrm{e}^{-d(x^n, r_j)/T_k}}{\sum_k \lambda_{kn}\, \mathrm{e}^{-d(x^n, r_k)/T_k}}$$

Adding the constraints to this equation, we derive the new Lagrangian given by

$$F' = -1/T \sum_n \sum_i \log(\sum_j \sum_n \lambda_{jn}\, \mathrm{e}^{-d(x_i{}^n, r_j)/T})\, p_n(x_i{}^n) + \sum_j \sum_n q_{jn}(\lambda_{jn} - W_{jn})$$

where q_{jn}, $1 \le n \le q$ and $1 \le j \le M$, are Lagrange multipliers. Finally, the optimal locations r_j of representative compounds for the lead generation library under experimental constraints are obtained by setting $\partial F'/\partial r_j = 0$. This gives the following set of equations

$$r_j = \frac{\sum_n p_n(x^n)\, p(r_j|x^n)\, x^n dx^n}{\sum_n p_n(x^n)\, p(r_j|x^n)\, dx^n}$$

which has the centroidal aspect in which averages are also taken about experimental supplies.

3.3. A Scalable Algorithm. As noted earlier, one of the major problems with combinatorial optimization algorithms is that of scalability, that is, the number of computations scales up exponentially with an increase in the amount of data. In the DA algorithm, the computational complexity can be addressed in two steps—first, by reducing the number of iterations and, second, by reducing the number of computations at every iteration. The DA algorithm, as described earlier, exploits the phase transition feature[26] in its process to decrease the number of iterations (in fact, in the DA algorithm, typically the temperature variable is decreased exponentially, which results in few iterations). The number of computations per iteration in the DA algorithm is $O(M^2N)$, where M is the number of representative compounds and N is the number of compounds in the underlying data set. In this section, we present an algorithm that requires fewer computations per iteration. This amendment becomes necessary in the context of the selection problem in combinatorial chemistry, as the sizes of the initial data set are so large that the DA is typically too slow to be practical and often fails to handle the computational complexity.

We exploit the features inherent in the DA algorithm to reduce the number of computations in each iteration. We use the fact that, for a given temperature, the farther an individual data point is from a cluster, the lower is its influence on the cluster. This is evident from the form of the association probabilities $p(r_j|x_i)$ in eq 2. That is, if two clusters are far apart, then they have very little interaction between them. Thus, if we ignore the effect of a separated cluster on the remaining data points, the resulting error will not be significant. Here, note that ignoring the effects of separated regions (i.e., groups of clusters) on one another will result in a considerable reduction in the number of computations since the points that constitute a separated region will not contribute to the distortion and entropy computations for the remaining points. Thus, the identification of separated regions in a data set enables us to process the algorithm much more quickly for large data sets. This savings on the number of computations increases as the temperature decreases since the number of separated regions, which are now smaller, increases as the temperature decreases.

The first step required to identify separated regions is to define and quantify interaction that exists between the various clusters. The next step is to group together sets of clusters among which significant interaction exists, but which have significantly few interactions with other clusters. Once groups of *separate* clusters are identified, the final step is to modify the DA algorithm such that it ignores the effects of separate groups on one another. This modification significantly reduces the number of computations to be performed by the algorithm.

3.3.1. Cluster Interaction and Separation. In order to characterize the interaction between different clusters, it is necessary to consider the mechanism of cluster identification during the process of the DA algorithm. As the temperature (T_k) is reduced after every iteration, the system undergoes a series of *phase transitions* (see ref 26 for details). The phase transitions refer to the expansion of the set of *distinct* locations of representative compounds at some critical values of temperature. We partition the underlying data set into clusters by associating each distinct location to all the points in the data set that are closest; that is, we form the *Voronoi* partition using the nearest neighbor condition. In this way, we achieve finer partitions (smaller clusters), as the temperature decreases. More precisely, at high temperatures that are above a precomputable critical value, all the representative compounds are located at the centroid of the entire underlying data set; therefore, there is only one distinct location for the representative compounds. As the temperature is decreased, successive phase transitions occur, which lead to a greater number of distinct locations for representative compounds and consequently finer clusters are formed. This provides us with a tool to control the number of clusters we want in our final selection. It is shown[26] that a cluster R_i splits at a critical temperature T_c when twice the maximum eigenvalue of the posterior covariance matrix, defined by $C_{x|rj} = \sum_i p(x_i)\ p(x_i|r_j)(x_i - r_j)(x_i - r_j)^T$, becomes greater than the temperature value, that is, when $T_c \leq 2\lambda_{\max}[C_{x|ri}]$. This is exploited in the DA algorithm to reduce the number of iterations by jumping from one critical temperature to the next without a significant loss in performance.

In the DA algorithm, the location of a representative compound is primarily determined by the data points (compound locations) near to it since far-away points exert little influence, especially at low temperatures. The association probabilities $p(r_j|x_i)$ determine the level of interaction between the cluster R_j and the data point x_i. This interaction level decays exponentially with the increase in the *distance* between r_j and x_i. The total interaction exerted by all the data points in a given space determines the relative weight of each cluster

$$p(r_j) := \sum_{i}^{N} p(x_i,r_j) = \sum_{i}^{N} p(r_j|x_i)\ p(x_i) \qquad (8)$$

where $p(r_j)$ denotes the weight of cluster R_j.

We define the level of interaction that data points in cluster R_i exert on cluster R_j by

$$\epsilon_{ji} = \sum_{x \in R_i} p(r_j|x)\ p(x) \qquad (9)$$

The higher this value is, the more interaction that exists between clusters R_i and R_j. This gives us an effective way to characterize the interaction between various clusters in a data set. In a probabilistic framework, this interaction value can also be interpreted as the probability of transition from R_i to R_j.

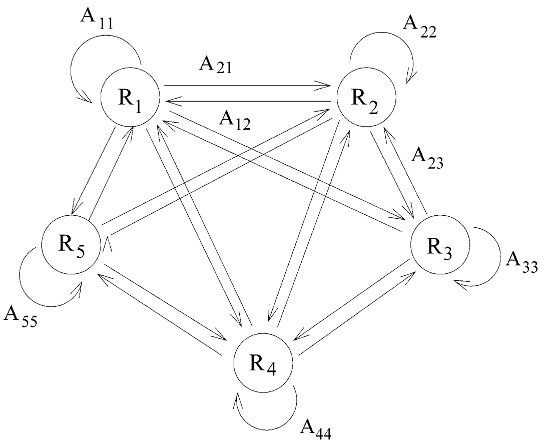

Figure 2. Markov chain.

Consider the $m \times m$ matrix ($m \leq M$)

$$A = \begin{pmatrix} \sum_{x \in R_1} p(r_1|x)p(x) & .. & \sum_{x \in R_m} p(r_1|x)p(x) \\ \sum_{x \in R_1} p(r_2|x)p(x) & .. & \sum_{x \in R_m} p(r_2|x)p(x) \\ \vdots & \ddots & \vdots \\ \sum_{x \in R_1} p(r_m|x)p(x) & .. & \sum_{x \in R_m} p(r_m|x)p(x) \end{pmatrix}$$

In a probabilistic framework, this matrix can be considered a finite dimensional Markov operator, with the term $A_{j,i}$ denoting the transition probability from region R_i to R_j. Figure 2 shows the transition probabilities of the associated Markov chain. The higher the transition probability, the greater is the amount of interaction between the two regions. Once the transition matrix is formed, the next step is to identify regions, that is, groups of clusters, which are separate from the rest of the data. The separation is characterized by a quantity which we denote by ϵ. We say a cluster (R_j) is ϵ-separate if the level of its interaction with each of the other clusters ($A_{j,i}$; $i = 1, 2, ..., n$; $i \neq j$) is less than ϵ. Note that the value ϵ determines the number of separate regions formed, if any, which in turn decides the error in distortion due to the proposed algorithm with respect to that of the original DA algorithm.

Alternative ways to identify invariant regions in the underlying space, which we have explored but do not present here, can be motivated by concepts from graph theory. One can consider the Markov matrix as a undirected weighted graph, with each entry A_{ij} representing the weight between node i and node j. Tools from the spectral partitioning of graphs[43−45] can be used to bipartition (or multipartition) this resulting graph into invariant subgraphs. In order to get balanced subgraphs, partitioning techniques such as normalized cuts[46] can be used on this weighted graph. However, algorithms for the recursive bipartitioning of graphs result in successive relaxations of the problem, thereby resulting in cumulative errors. Efficient algorithms for multiway partitions have not been exhaustively studied in the literature.

3.3.2. Tradeoff between Error in Representative Compound Location and Computation Time. As was discussed in section 3.3, the greater the number of separate regions we use, the smaller the computation time for the modified algorithm. At the same time, a greater number of separate regions results in a higher deviation in the distortion term of the proposed algorithm from the original DA algorithm. This tradeoff between a reduction in computation time and an increase in distortion error is systematically addressed below. We introduce the following notation on a subset V of the domain Ω for a representative compound r_j:

$$G_j(V) := \sum_{x_i \in V} x_i p(x_i)\, p(r_j|x_i) \tag{10}$$

$$H_j(V) := \sum_{x_i \in V} p(x_i)\, p(r_j|x_i) \tag{11}$$

Then, from the DA algorithm, the location of the representative compound (r_j) is determined by

$$r_j = \frac{G_j(\Omega)}{H_j(\Omega)} \tag{12}$$

Since the cluster Ω_j is separated from all the other clusters, the representative compound location r'_j will be determined in the modified algorithm by

$$r'_j = \frac{\sum_{x_i \in \Omega_j} x_i p(x_i)\, p(r_j|x_i)}{\sum_{x_i \in \Omega_j} p(x_i)\, p(r_j|x_i)} = \frac{G_j(\Omega_j)}{H_j(\Omega_j)} \tag{13}$$

We obtain the component-wise difference between r_j and r'_j by subtracting terms. Note that the operations in eq 14, 16, and 18 are component-wise operations. On simplifying the component-wise terms, we have

$$|r_j - r'_j| \leqslant \frac{\max[G_j(\Omega_j^c)\, H_j(\Omega_j), G_j(\Omega_j)\, H_j(\Omega_j^c)]}{H_j(\Omega)\, H_j(\Omega_j)} \tag{14}$$

where $\Omega_j^c = \Omega[\backslash]\Omega_j$. Now note that

$$G_j(\Omega_j^c) \le \Big(\sum_{x_i \in \Omega_j^c} x_i\Big)\, H_j(\Omega_j^c) = NM_j^c\, H_j(\Omega_j^c) \tag{15}$$

where N is the cardinality of Ω and $M_j^c = 1/N \sum_{x_i \in \Omega_j^c} x_i$. We have assumed that $x \geqslant 0$ without any loss of generality since the resource allocation problem definition is independent of translation or scaling factors. Thus

$$|r_j - r'_j| \leqslant \frac{\max[NM_j^c\, H_j(\Omega_j), G_j(\Omega_j)]\, H_j(\Omega_j^c)}{H_j(\Omega)\, H_j(\Omega_j)} \tag{16}$$

$$= \max\left[NM_j^c, \frac{G_j(\Omega_j)}{H_j(\Omega_j)}\right]\left[\frac{H_j(\Omega_j^c)}{H_j(\Omega)}\right] \tag{17}$$

Then, dividing through by N and $M = 1/N \sum_{x_i\ \in \Omega} x_i$ gives

$$\frac{|r_j - r'_j|}{MN} \leqslant \max\left[\frac{M_j^c}{M}, \frac{M_j}{M}\right]\eta_j \tag{18}$$

where

$$\eta_j = \frac{\sum_{k \neq j} \epsilon_{kj}}{\sum_k \epsilon_{kj}}$$

and ϵ_{kj} is the level of interaction between cluster Ω_j and Ω_k as defined in eq 9.

For a given data set, the quantities M, M_j, and M_j^c are known a priori. For the error in representative compound location $|r_j - r'_j|/M$ to be less than a given value δ_j (where $\delta_j > 0$), we must choose η_j such that

$$\eta_j \le \frac{\delta_j}{N \max\left[\frac{M_j^c}{M}, \frac{M_j}{M}\right]} \tag{19}$$

3.3.3. Algorithm. The modified algorithm is as follows:

- Step 1: Initiate the DA algorithm and determine representative compound locations together with the association probabilities.
- Step 2: When a split occurs (phase transition), identify individual clusters and use the association weights [$p(r_j|x)$] to construct the transition matrix.
- Step 3: Use the transition matrix to identify separated clusters, and group them together to form separated regions. Ω_k will be separated from Ω_j if the entries $A_{j,k}$ and $A_{k,j}$ are less than a chosen ϵ_{jk}.
- Step 4: Apply the DA algorithm to each region, neglecting the effect of separate regions on one another.
- Step 5: Stop if the stopping criterion [such as number of representative compounds (M), or computation time etc.] is met; otherwise, go to step 2.

Identification of the separate regions in the underlying data provides us with a tool to efficiently scale the DA algorithm. The number of computations at a given iteration is proportional to $\sum_{k=1}^{s} M_k^2 N_k$, where N_k is the number of compounds and M_k is the number of clusters in the kth group. For the original DA algorithm, at any iteration, the number of computations is M^2N where $N = \sum_{k=1}^{s} N_k$. Thus, the scalable algorithm saves computations at each iteration. This difference becomes bigger as temperature decreases since corresponding values of N_k decrease. That is, the scalable algorithm effectively runs s parallel DA algorithms, each with a relatively smaller number of data points N_k. Moreover, as the temperature decreases, the scalable algorithm runs an increasing number of parallel DA algorithms on progressively shrinking subsets, which results in increasing savings on computations.

4. SIMULATION RESULTS

To demonstrate the advantages of the proposed algorithm, we have applied it on a number of data sets. We show that the algorithm we have proposed simultaneously addresses the key issues of diversity and representativeness in clustering the descriptor space. At the same time, it also addresses scalability issues, which are imperative for huge data sets associated with the drug discovery process.

4.1. Case 1: Design for Diversity and Representativeness. As a first step, a fictitious data set was created to present the "proof of concept" for the proposed optimization

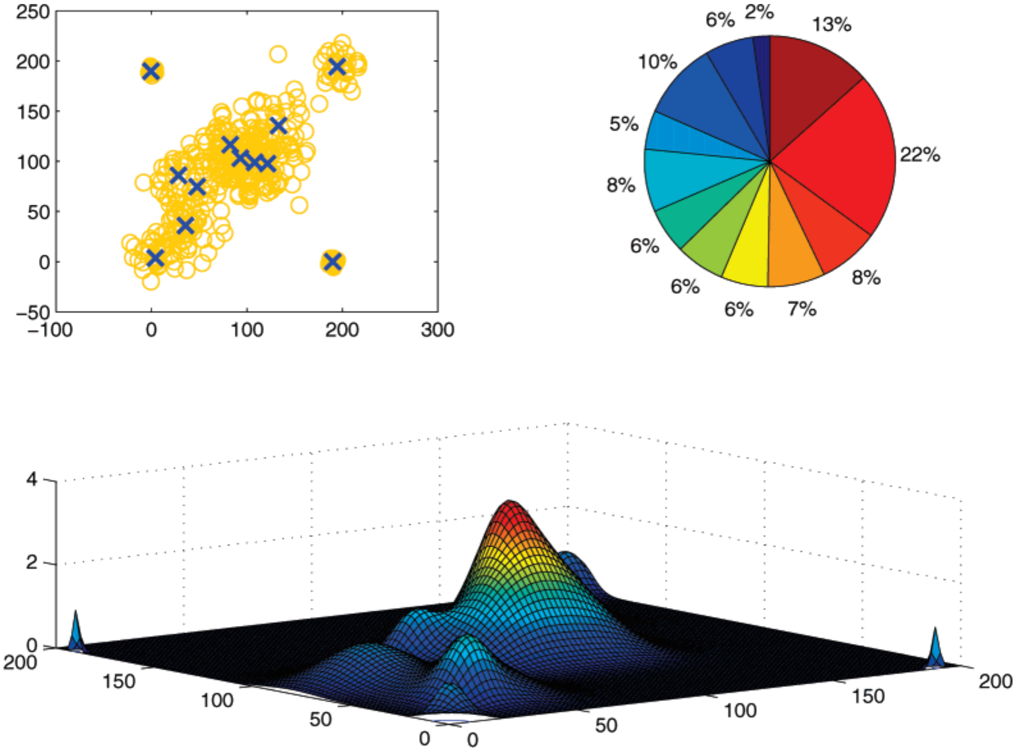

Figure 3. Simulation results for data set 1: (a) The locations x_i, $1 \le i \le 200$, of compounds (circles) and r_j, $1 \le j \le 10$, of representative compounds (crosses) in the 2D descriptor space. (b) The weights λ_j associated with different locations of representative compounds. (c) The given weight distribution $p(x_i)$ of the different compounds in the data set.

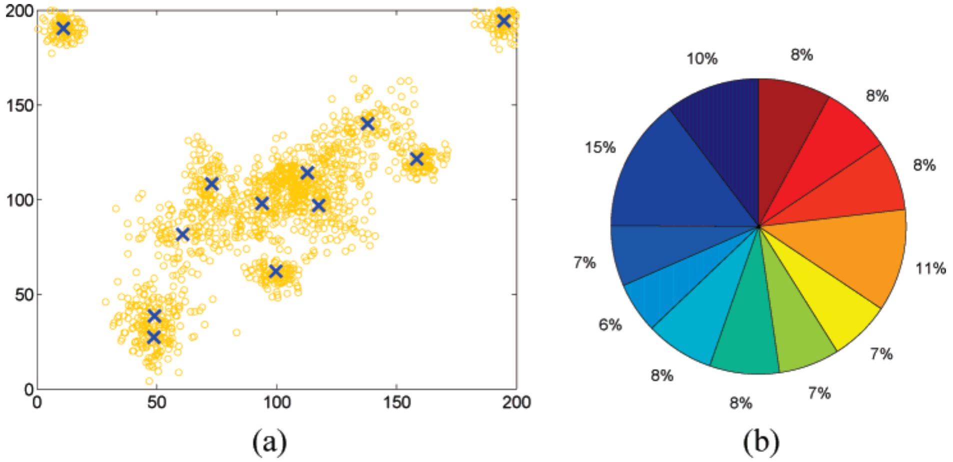

Figure 4. (a) Locations x_i of compounds (circles) and representative compounds r_j, $1 \le j \le 12$ (crosses). (b) Weights (λ_j) associated with different locations of representative compounds.

algorithm. The data set was specifically designed to simultaneously address the issue of diversity and representativeness in the lead generation library design. This data set consists of few points that are outliers, while most of the points are in a single cluster. Simulations were carried out in MATLAB. The results for data set 1 are shown in Figure 3.

The pie chart in Figure 3 shows the relative weight of each representative compound. Representative compounds with a large weight signify that more compounds are chosen from that area. As was required, the algorithm gave higher weights at locations which had larger numbers of similar compounds. Thus, different weights at each representative compound location address the issue of representativeness in the library. At the same time, it should be noted that the key issue of diversity is not compromised. This is due to the fact that the algorithm inherently recognizes the *natural* clusters in the population.

As is seen from the figure, the algorithm identifies all cluster locations. The two cluster locations which were quite diverse from the rest of the compounds are also identified, albeit with a smaller weight. As can be seen from the accompanying pie chart, the outlier cluster was assigned a weight of 2%, while the central lump was assigned a significant weight of 22% .

Results on another data set are presented in Figure 4. In this case too, the data set was specifically designed to simultaneously address the issue of diversity and representativeness in the design. As with the previous data, the algorithm automatically gave higher weights at locations which had larger numbers of similar compounds. The outlier clusters in this case contained appreciable numbers of data points. As a result, the outlier clusters were also identified,

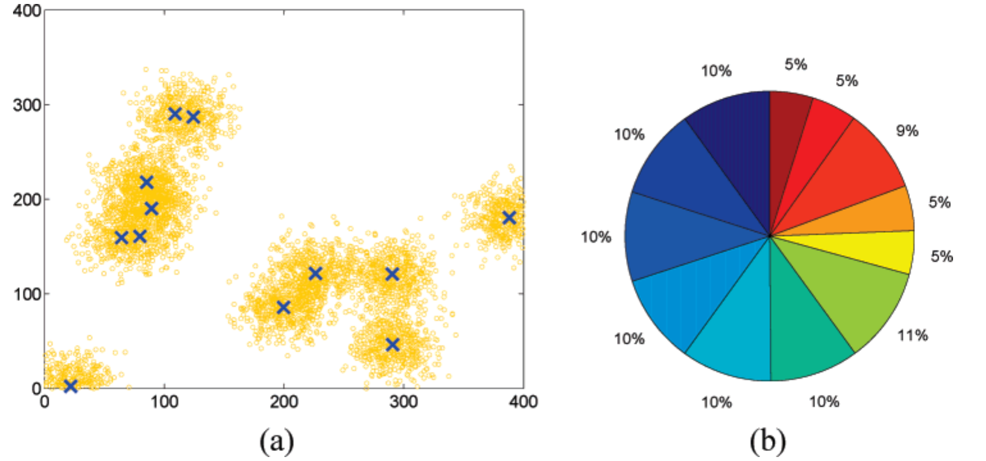

Figure 5. (a) Locations x_i, $1 \le i \le 5000$, of compounds (circles) and r_j, $1 \le j \le 12$, of representative compounds (crosses) in the 2D descriptor space determined from a nonscalable algorithm. (b) Relative weights λ_j associated with different locations of representative compounds.

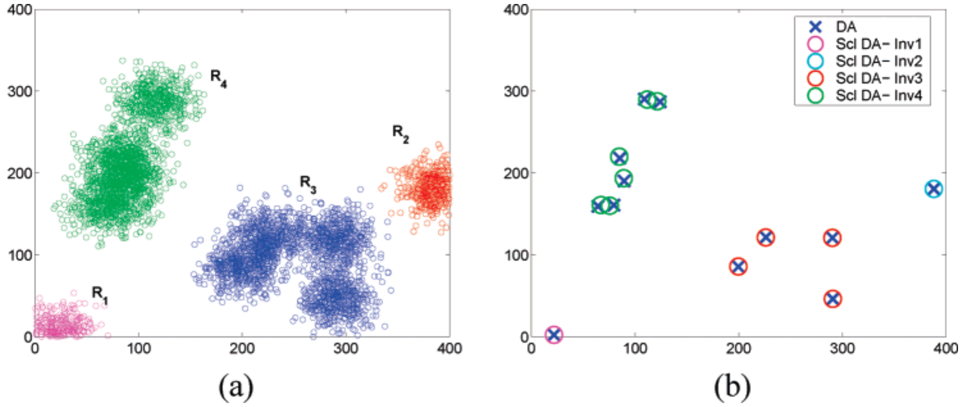

Figure 6. (a) Separated regions R_1, R_2, R_3, and R_4 (denoted by different colors) as determined by the proposed algorithm. (b) Comparison of representative compound locations r_j and r'_j obtained from two algorithms.

but with a smaller weight than that of the central clusters (as indicated by the pie chart in Figure 4b). Each cluster is appropriately represented in the final selection.

Another feature of this algorithm is the flexibility it provides in dealing with *unique* (i.e., far-away) clusters. By properly assigning the relative importance of clusters a priori, we can choose whether to include or reject these unique clusters in the library design. Though an immediate rejection of unique clusters compromises the diversity of the library, there can be scenarios where the properties of these unique cluster compounds are totally undesired in the lead generation library. Thus, our approach gives us a means to deal with such scenarios effectively.

4.2. Case 2: Scalability and Computation Time. Taking into consideration the huge size of combinatorial libraries, it becomes essential to consider the scaling issues involved with any clustering algorithm. As was pointed out earlier, the identification of separate regions in the lower dimensional descriptor space speeds up the algorithm (as it requires a lesser amount of computation) and at the same time allows for larger data sets to be clustered. In order to demonstrate this fact, the algorithm was tested on a host of synthesized data sets. For the purpose of simulation, these data sets were created in the following manner.

The first set was obtained by identifying 10 random locations in a square region of size 400 × 400. These locations were then chosen as the cluster centers. Next, the size of each of these clusters was chosen, and all points in the cluster were generated by a normal distribution of randomly chosen variance. A total of 5000 points comprised this data set. All the points were assigned equal weights [i.e., $p(x_i) = 1/N$ for all $x_i \in \Omega$]. Figure 5 shows the data set and the representative compound locations obtained by the original DA algorithm. The crosses denote the representative compound locations (r_j), and the pie chart gives the relative weight of each representative compound (λ_j).

Table 1. Comparison between Nonscalable and Proposed Algorithms

algorithm	distortion	computation time (sec)
nonscalable DA	300.80	129.41
proposed algorithm	316.51	21.53

Our proposed algorithm starts with one representative compound at the centroid of the data set. As the temperature is reduced, the clusters are split, and separated regions are determined at each such split. Figure 6a shows the four separate regions identified by the algorithm (as described in section 3.3.1) at the instant when 12 representative compound locations have been identified. Figure 6b offers a comparison between the two algorithms. Here, the crosses represent the representative compound locations (r_j) determined by the non-scalable DA algorithm.[47] The circles represent the locations (r'_j) determined by the modified algorithm that we have proposed. Note that the data set was partitioned into

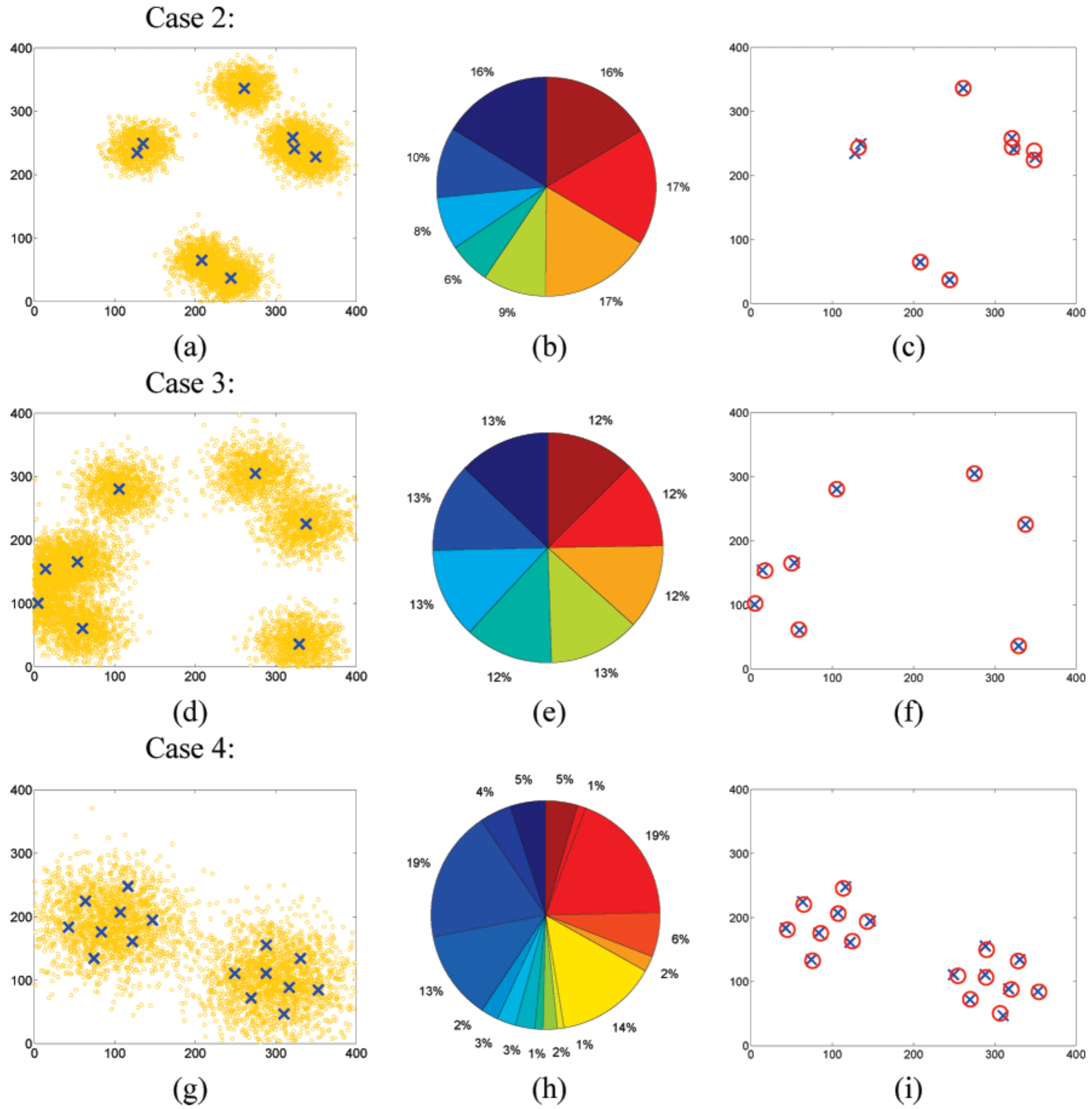

Figure 7. (a, d, g) Simulated data set with locations x_i of compounds (circles) and representative compound locations r_j (crosses) determined by a nonscalable algorithm. (b, e, h) Relative weights λ_j of representative compound. (c, f, i) Comparison of representative locations r_j and r_j' obtained from two algorithms.

Table 2. Distortion and Computation Times for Different Data Sets

case	algorithm	distortion	computation time (sec)
case 2	nonscalable DA	290.06	44.19
	proposed algorithm	302.98	11.98
case 3	nonscalable DA	672.31	60.43
	proposed algorithm	717.52	39.77
case 4	non-scalable DA	808.83	127.05
	proposed algorithm	848.79	41.85

four separated regions (represented by the four colors of the circles). As can be seen from the figure, there is not much difference between the locations obtained by the two algorithms. The main advantage of the modified algorithm is in terms of the computation time of the algorithm and its ability to handle larger data sets. The results from the two algorithms have been presented in Table 1. As can be seen from Table 1, the proposed scalable algorithm uses just one-sixth of the time taken by the original (nonscalable) algorithm and results in only a 5.2% increase in distortion; this was obtained for $\epsilon = 0.005$. Both the algorithms were terminated when the number of representative compounds reached 12. It should be noted here that, in case of the modified algorithm, the computation time can be further reduced (by changing ϵ), but at the expense of error in distortion.

4.2.1. Further Examples. The scalable algorithm was applied on a number of different data sets. Results for three such cases have been presented in Figure 7. The data set in case 2 is comprised of six randomly chosen cluster centers with 1000 points each. All the points were assigned equal weights [i.e., $p(x_i) = 1/N$ for all $x_i \in \Omega$]. Figure 7a shows the data set and the eight representative compound locations obtained by the proposed scalable algorithm. The pie chart in Figure 7b gives the relative weight of each representative compound location. Figure 7c offers a comparison between the two algorithms. As in case 2, the data set in case 3 is comprised of eight randomly chosen cluster locations with 1000 points each. Both of the algorithms were executed till they identified eight representative compound locations in the underlying data set. The clustering results are shown in Figure 7a−c. The data set in case 4 is comprised of two

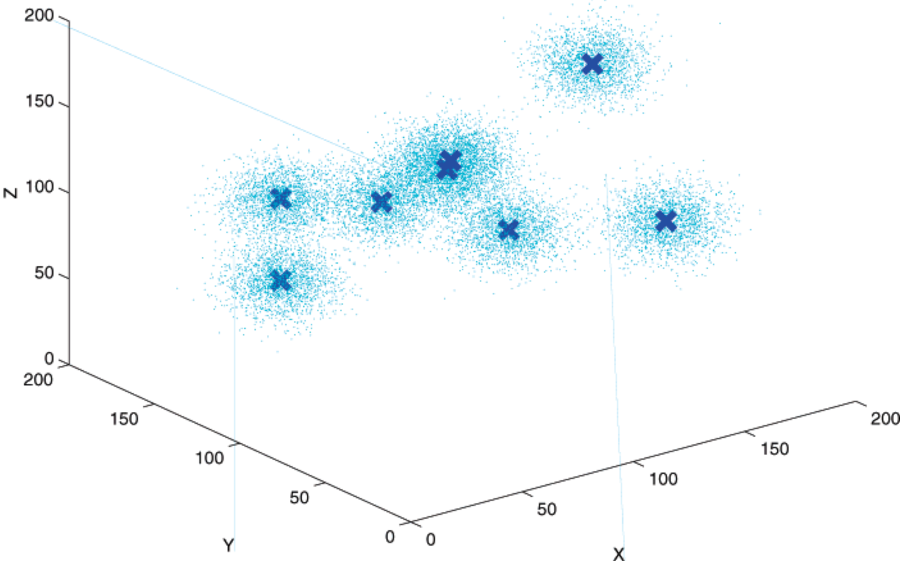

Figure 8. Clustering of a 3D data set. Locations x_i of compounds (dots) and representative compounds r_j (crosses).

Table 3. Comparison between Nonscalable and Proposed Algorithms

algorithm	distortion	computation time (sec)
nonscalable DA	161.5	166.4
proposed algorithm	172.1	54.2

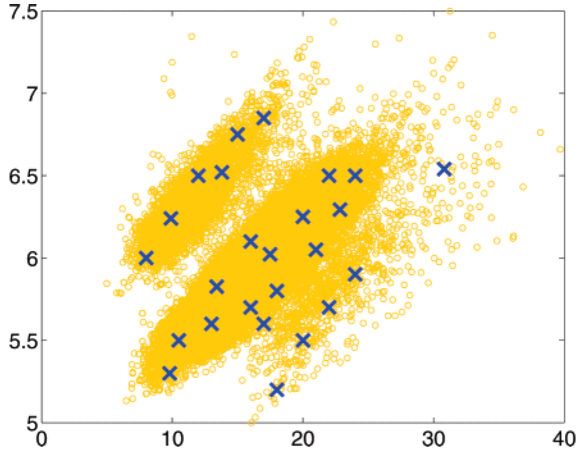

Figure 9. Choosing 24 representative locations from the drug discovery data set.

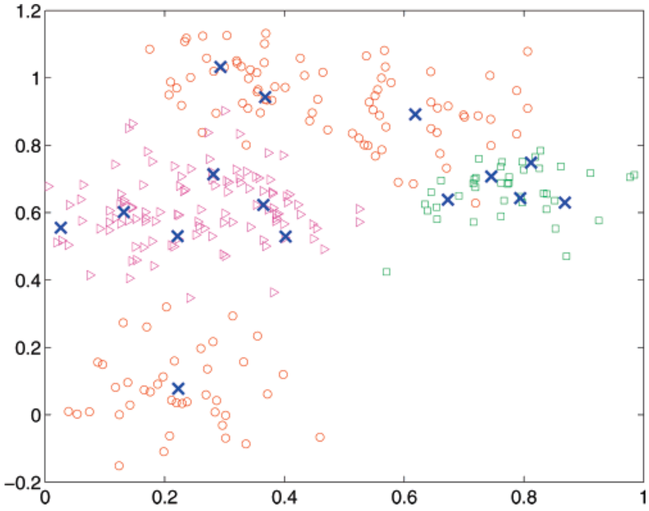

Figure 10. Simulation results with constraints on experimental resources.

cluster centers with 2000 points each. Both of the algorithms were executed till they identified 16 representative compound locations in the underlying data set. As was pointed out earlier, the main advantage of the modified algorithm is in terms of the computation time and scalability. Results for the three cases (from the two algorithms) have been presented in Table 2. It should be noted that both of the algorithms were terminated after a specific number of representative compound locations had been identified. The proposed algorithm took far less computation time when compared to the nonscalable algorithm while maintaining less than 5% error in distortion.

The proposed algorithm has also been applied on higher dimensional data sets. Results for one such data set are presented in Figure 8. The data set was created by identifying eight random locations in a region of size 200 × 200 × 200. These locations were then chosen as the cluster centers. Next, the sizes of each of these clusters were chosen, and all points in the clusters were generated by normal distributions of randomly chosen variance. A total of 16 000 points comprised this data set. All of the points were assigned equal weights [i.e., $p(x_i) = 1/N$ for all $x_i \in \Omega$]. Figure 8 shows the clustering results obtained by the proposed scalable algorithm. Here, the crosses represent the representative compound locations (r_j) determined by our algorithm. The main advantage of the modified algorithm is in terms of the computation time of the algorithm and its ability to tackle larger data sets. The results from the two algorithms have been presented in Table 3. As can be seen from Table 3, our proposed scalable algorithm uses just 54.2 s (as compared to 166.4 used by the nonscalable algorithm) and results in only a 6.5% increase in distortion. It should be noted that both of the algorithms were terminated when the number of representative compounds reached eight. In the case of the modified algorithm, the computation time can be further reduced at the expense of error in distortion.

4.3. Case 3: Drug Discovery Data Set. This data set was a modified version of the test library set.[48] Each of the 50 000

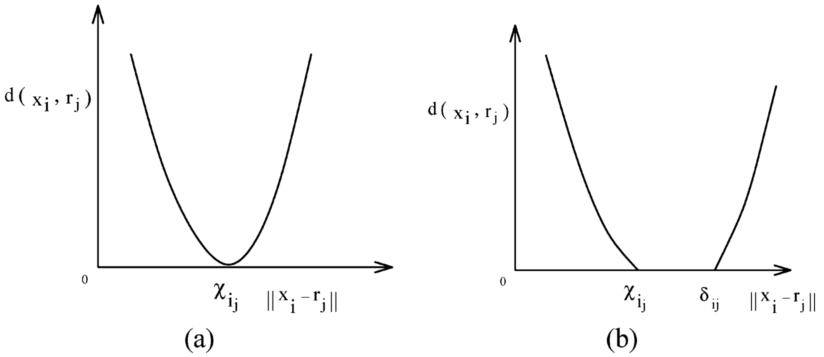

Figure 11. (a) Modified "distance" metric for exclusion of specific regions. (b) Modified "distance" metric for exclusion and inclusion of specific regions.

members in this set were represented by 47 descriptors, which included topological, geometric, hybrid, constitutional, and electronic descriptors. These molecular descriptors were computed using the Chemistry Development Kit (CDK) Descriptor Calculator.[49,50] As has been shown previously, these descriptors are well-suited for diversity- and similarity-based searching as they exhibit a neighborhood behavior.[51] This 47-dimensional data was then normalized and projected to a two-dimensional space. The projection was carried out using principal component analysis on the higher dimensional data. Simulations were carried out on this two-dimensional data set. We applied our proposed scalable algorithm to identify 25 representative compound locations from this data set. The results are shown in Figure 9.

As was demonstrated in the previous section, the proposed algorithm gave higher weights at locations which had larger numbers of similar compounds. Compounds which are maximally diverse from all the others are identified with a very small weight. It should be noted that the nonscalable version of the algorithm could not handle the number of computations for this data set (we ran both the scalable and the original DA algorithm using MATLAB on a 1.5 GHz Intel Centrino processor with 512 MB of RAM) due to the combinatorial nature of the problem.

4.4. Case 4: Additional Constraints on Representative Compounds. As was discussed in section 3, the multiobjective framework of the proposed algorithm allows us to incorporate additional constraints in the selection problem. In this section, we have addressed two such constraints, namely, the experimental resources constraint and the exclusion/inclusion constraint. Results on simulated data sets have also been presented, wherein the proposed algorithm was adapted to cater to these additional constraints in the selection criterion.

4.4.1. Constraints on Experimental Resources. A data set was created to demonstrate the manner in which the algorithm presented in section 3.2.2 accounts for the constraints placed on experimental resources. Different compounds require different experimental supplies which are typically available in limited quantities. As was discussed in section 3.2.2, in addition to diversity and representativeness, we include these constraints on experimental supplies into our algorithm for selecting compounds. In this data set, the library has been divided into three classes on the basis of the experimental supplies required by the compounds for testing, as shown in Figure 10 by different symbols. It contains a total of 280 compounds with 120 of the first class (denoted by circles), 40 of the second class (denoted by squares), and 120 of the third class (denoted by triangles). We incorporate experimental supply constraints into the algorithm by translating them into direct constraints on each of the representative compounds (eq 7). With these experimental supply constraints in place, the algorithm was used to select 15 representative compound locations (r_j) in this data set with capacities (W_{jn}) fixed for each class of resource. The crosses in Figure 10 represent the selection from the algorithm in the wake of the capacity constraints for different types of compounds. As can be seen from the selection, the algorithm successfully addressed the key issues of diversity and representativeness together with the constraints that were placed due to experimental resources. To the authors' best knowledge, this is the first algorithm which specifically addresses this type of constraint on experimental resources in a multiobjective optimization framework.

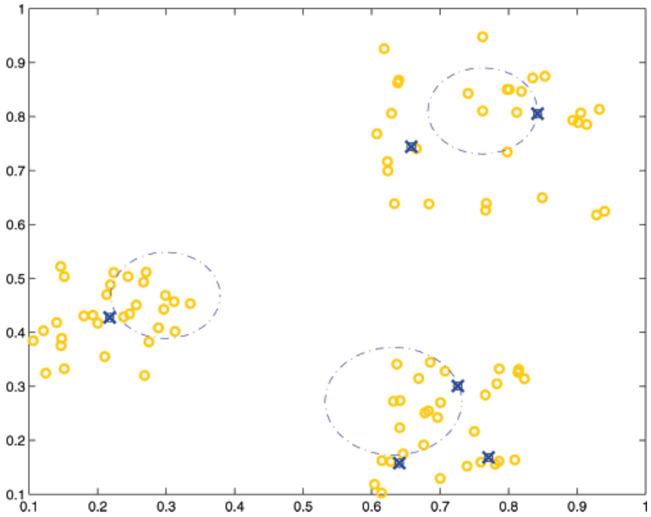

Figure 12. Simulation results with exclusion constraint. The locations x_i, $1 \leq i \leq 90$, of compounds (circles) and r_j, $1 \leq j \leq 6$, of representative compounds (crosses) in the 2D descriptor space. Blue circles represent undesirable properties.

4.4.2. Constraints on Exclusion and Inclusion of Certain Properties. In the process of selecting a subset of compounds, there may arise scenarios where we would like to inhibit selection of compounds exhibiting properties within certain prespecified ranges. This constraint can be easily incorporated in the cost function by modifying the distance metric used in the problem formulation.

Consider a case in a 2D data set where each point x_i has an associated radius (denoted by χ_{ij}). The selection problem is fundamentally the same, but with the added constraint that all of the selected points (r_j) must be at least a distance of χ_{ij} removed from x_i. The proposed algorithm can be modified to solve this problem by defining the distance function as follows:

$$d(x_i,r_j) = (\|x_i - r_j\| - \chi_{ij})^2 \tag{20}$$

Such a distance function penalizes any selection (r_j) which is in close proximity to the data points. This is depicted in Figure 11a. For the purpose of simulation, a data set was created with points at 90 locations (x_i, $i = 1, ..., 90$). The blue circle around the locations x_i denotes the region in the property space that is to be avoided by the selection algorithm. The objective was to select six representative compounds from this data set such that the criterion of diversity and representativeness is optimally addressed in the selected subset. The selected locations are represented by blue crosses.

From Figure 12, note that the algorithm identifies the six clusters under the constraint and that none of the cluster centers are located in the undesirable property space (denoted by blue circles). The same analysis can be extended to higher dimensional descriptor spaces, thereby making it possible to exclude ranges of certain properties (e.g., molecular weight and solubility). Another distance metric is shown in Figure 11b. As can be seen from the figure, it favors representative compounds within a certain radius around a particular compound. The flat region in the figure denotes this radius. Anything outside this region is penalized by this choice of distance metric. Depending on the scenarios encountered, the distance metric can be suitably modified to address further such exclusion and inclusion constraints.

5. CONCLUSIONS

In this paper, we proposed an algorithm for the design of lead generation libraries. The problem was formulated in a constrained multiobjective optimization setting and posed as a resource allocation problem with multiple constraints. As a result, we successfully tackled the key issues of diversity and representativeness of compounds in the resulting library. Another distinguishing feature of the algorithm is its scalability, thus making it computationally efficient as compared to other such optimization techniques. We characterized the level of interaction between various clusters and used it to divide the clustering problem with huge data size into manageable subproblems with small sizes. This resulted in significant improvements in the computation time and enabled the algorithm to be used on larger-sized data sets. The tradeoff between computation effort and error due to truncation is also characterized, thereby giving an option to the end-user.

Currently, we are working on ways to formulate an optimization problem where the computational expense forms a part of the total cost. This gives a way to prescribe how the subsets need to be formed for a given tradeoff between the computational expense and the deviation from the solution that acts on the whole set (that is, which does not subdivide the data and therefore does not save computational time). The main principle that the framework relies on is the maximum entropy principle (MEP) developed extensively by Jaynes.[52,53] The problem is viewed in a stochastic setting, and MEP proves to be a very useful tool for formulating the optimization problem and finding an algorithm that solves it. Formulating the lead generation library design problem in such a manner has enabled automation of the *divide-and-conquer rule*, with predetermined bounds.

ACKNOWLEDGMENT

This work is partially supported by NSF ITR Collaborative Research Grant ECS-0426831 and a NSF ECS 0449310 CAR grant.

REFERENCES AND NOTES

(1) Demers, L.; Cioppa, G. D. Drug Discovery: Nanotechnology to Advance Discovery R&D. *Genet. Eng. News* **2003**, *23*.

(2) Gersho, A.; Gray, R. *Vector Quantization and Signal Compression*, 1st ed.; Kluwer: Boston, Massachusetts, 1991.

(3) Drezner, Z. *Facility Location: A Survey of Applications and Methods*; Springer-Verlag: New York, 1995; Springer Series in Operations Research.

(4) Du, Q.; Faber, V.; Gunzburger, M. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Rev.* **1999**, *41*, 637−676.

(5) Salapaka, S.; Khalak, A. Constraints on Locational Optimization Problems. In *Proceedings of the IEEE Control and Decision Conference*; IEEE: Los Alamitos, CA, 2003.

(6) Cortés, J.; Martínez, S.; Karatas, T.; Bullo, F. Coverage Control for Mobile Sensing Networks: Variations on a Theme. *Proceedings of the Mediterranean Conference on Control and Automation*; Mediterranean Control Association: Notre Dame, IN, 2002.

(7) Therrien, C. *Decision, Estimation and Classification: An Introduction to Pattern Recognition and related topics*, 1st ed.; Wiley: New York, 1989; Vol. 14.

(8) Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall: Englewoods Cliffs, NJ, 1998.

(9) Hartigan, J. *Clustering Algorithms*; Wiley: New York, 1975.

(10) Gray, R.; Karnin, E. Multiple local minima in vector quantizers. *IEEE Trans. Inf. Theory* **1982**, *IT-28*, 256−361.

(11) Willett, P. Computational tools for the analysis of molecular diversity. *Perspect. Drug Discovery Des.* **1997**, *7/8*, 1−11.

(12) Blaney, J.; Martin, E. Computational approaches for combinatorial library design and molecular diversity analysis. *Curr. Opin. Chem. Biol.* **1997**, *1*, 54−59.

(13) Rassokhin, D. N.; Agrafiotis, D. K. Kolmogorov-Smirnov statistic and its applications in library design. *J. Mol. Graphics Modell.* **2000**, *18*, 370−384.

(14) Lipinski, C. A.; Lomabardo, F.; Dominy, B. W.; Feeny, P. J. Experimental and Computational Approaches to estimate solubility and permeability in drug discovery and development setting. *Adv. Drug Delivery Rev.* **1997**, *23*, 2−25.

(15) Higgs, R. E.; Bemis, K. G.; Watson, I. A.; Wikel, J. H. Experimental designs for selecting molecules from large chemical databases. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 861−870.

(16) Snarey, M.; Terrett, N.; Willet, P.; Wilton, D. J. Comparison of algorithms for dissimilaritybased compound selection. *J. Mol. Graphics Modell.* **1997**, *15*, 372−385.

(17) Mount, J.; Ruppert, J.; Welch, W.; Jain, A. N. IcePick: A flexible surface-based system for molecular diversity. *J. Med. Chem.* **1999**, *42*, 60−66.

(18) Sheridan, R. P. The most common chemical replacements in drug-like compounds. *J. Chem. Inf. Comput. Sci.* **2002**, *2*, 103−108.

(19) Wang, J.; Ramnarayan, K. Toward designing drug-like libraries: a novel computational approach for prediction of important structural features. *J. Comb. Chem.* **1999**, *1*, 52−533.

(20) Waldman, M.; Li, H.; Hassan, M. Novel algorithms for the optimization od molecular diversity of combinatorial libraries. *J. Mol. Graphics Modell.* **2000**, *18*, 412−426.

(21) Good, A. C.; Lewis, R. A. New methodology for profiling combinatorial libraries and screening sets: cleaning up the design process with HARPcik. *J. Med. Chem.* **1997**, *40*, 3226−3236.

(22) Agrafiotis, D. K. Stochastic Algorithms for maximizing molecular diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841−851.

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

(24) Sheridan, R. P.; San Feliciano, S. G.; Kearsley, S. K. Designing targeted libraries with Genetic Algorithms. *J. Mol. Graphics Modell.* **2000**, *18*, 320−333.

(25) Brown, R. D.; Martin, Y. C. Designing combinatorial library mixtures using Genetic Algorithms. *J. Med. Chem.* **1997**, *40*, 2304−2313.

(26) Rose, K. Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems. *Proc. IEEE* **1998**, *86*, 2210−39.

(27) Rose, K. Constrained Clustering as an Optimization Method. *IEEE Trans. Pattern Anal. Machine Intell.* **1993**, *15*, 785−794.

(28) Kirkpatrick, S.; Gelatt, C.; Vechhi, M. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671−680.

(29) Agrafiotis, D. K.; Lobanov, V. S.; Salemme, F. R. Combinatorial Informatics in the postgenomics era. *Nat. Rev.* **2002**, *1*, 337−346.

(30) Gordon, E.; Barrett, R.; Dower, W.; Fodor, S.; Gallop, M. Applications of Combinatorial Technologies to Drug Discovery. 2. Combinatorial Organic Synthesis, Library Screening Strategies, and Future Directions. *J. Med. Chem.* **1994**, *37*, 1385−1401.

(31) Clark, R. D. OptiSim: An Extended Dissimilarity Selection Method for Finding Diverse Representative Subsets. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1181−1188.

(32) Agrafiotis, D.; Lobanov, V. S. Ultrafast Algorithm for Designing Focussed Combinatorial Arrays. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1030−1038.

(33) Livingston, D. J. The characterization of molecular structures using molecular properties: A survey. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 195−209.

(34) Schlkopf, B.; Smola, A. J.; Mller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299−1319.

(35) Jolliffe, I. T. *Principal Component Analysis*; Springer-Verlag: New York, 2002; Springer Series in Statistics.

(36) Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129−137.

(37) Berger, T. *Rate Distortion Theory: A Mathematical Basis for Data Compression*; Prentice Hall: Englewood Cliffs, NJ, 1971.

(38) Lin, S. K. Molecular Diversity Assesment: logarithmic relations of information and species diversity and logarithmic relations of entropy and indistinguishability after rejection of Gibbs paradox of entropy mixing. *Molecules* **1996**, *1*, 57−67.

(39) Shannon, C. E.; Weaver, W. *The mathematical theory of communication*; University of Illinois Press: Urbana, Illinois, 1949.

(40) Rose, K. Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett.* **1990**, *65*, 945−948.

(41) MacQueen, J. B. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, 1967.

(42) Lynch, M.; Mosher, C.; Huff, J.; Nettikadan, S.; Johnson, J.; Henderson, E. Functional protein nanoarrays for biomarker profiling. *Proteomics* **2004**, *4*, 1695−702.

(43) Fiedler, M. Algebraic connectivity of graphs. *Czech. Math. J.* **1973**, *23*, 298−305.

(44) Fiedler, M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. J.* **1975**, *25*, 619−633.

(45) Hendrickson, B.; Leland, M. *Multidimensional Spectral Load Balancing*; Sandia National Laboratories: Albuquerque, NM, 1993.

(46) Shi, J.; Malik, J. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Machine Intell.* **2000**, *22*, 888−905.

(47) Sharma, P.; Salapaka, S.; Beck, C. A Deterministic Annealing Approach to Combinatorial Library Design for Drug Discovery. In *Proceedings of the American Control Conference*; IEEE: Los Alamitos, CA, 2005.

(48) McMaster HTS Lab Competition. HTS Data Mining and Docking Competition. http://hts.mcmaster.ca/Downloads/82BFBEB4−F2A4−4934-B6A8−804CAD8E25A0.html (accessed June 2006).

(49) Guha, R. Chemistry Development Kit (CDK) Descriptor Calculator GUI (v. 0.46). http://cheminfo.informatics.indiana.edu/rguha/code/java/cdkdesc.html (accessed October 2006).

(50) Steinbeck, C.; Hoppe, C.; Kuhn, S.; Floris, M.; Guha, R.; Willighagen, E. L. Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source JAVA Library for Chemo and Bioinformatics. *Curr. Pharm. Des.* **2006**, *12*, 2110−2120.

(51) Patterson, D. E.; Cramer, R. D.; Ferguson, A. M.; Clark, R. D.; Weinberger, L. E. Neighborhood Behavior: A Useful Concept for Validation of “Molecular Diversity” Descriptors. *J. Med. Chem.* **1996**, *39*, 3049−3059.

(52) Jaynes, E. T. Information Theory and Statistical Mechanics. *Phys. Rev.* **1957**, *106*, 620−630.

(53) Jaynes, E. T. Information Theory and Statistical Mechanics II. *Phys. Rev.* **1957**, *108*, 171−190.

CI700023Y