Combinatorial Library Design Using a Multiobjective Genetic Algorithm

Valerie J. Gillet,*,† Wael Khatib,†,‡ Peter Willett,† Peter J. Fleming,§ and Darren V. S. Green||

Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom, Department of Automatic Control and Systems Engineering, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom, and GlaxoSmithKline, Gunnels Wood Road, Stevenage, SG1 2NY, United Kingdom

Received July 12, 2001

Early results from screening combinatorial libraries have been disappointing with libraries either failing to deliver the improved hit rates that were expected or resulting in hits with characteristics that make them undesirable as lead compounds. Consequently, the focus in library design has shifted toward designing libraries that are optimized on multiple properties simultaneously, for example, diversity and druglike physicochemical properties. Here we describe the program MoSELECT that is based on a multiobjective genetic algorithm and which is able to suggest a family of solutions to multiobjective library design where all the solutions are equally valid and each represents a different compromise between the objectives. MoSELECT also allows the relationships between the different objectives to be explored with competing objectives easily identified. The library designer can then make an informed choice on which solution(s) to explore. Various performance characteristics of MoSELECT are reported based on a number of different combinatorial libraries.

INTRODUCTION

The development of combinatorial chemistry techniques during the past decade has revolutionized the processes involved in the discovery of novel bioactive compounds in the pharmaceutical and agrochemical industries.[1,2] Initially the focus in combinatorial library design was on selecting diverse sets of compounds on the assumption that maximizing diversity would result in a broad coverage of bioactivity space[3] and hence would maximize the chances of finding hits. However, early results from combinatorial libraries were disappointing[4,5] with libraries either failing to deliver the improved hit rates that were expected or resulting in hits that did not have "druglike" characteristics. Thus, it is now evident that diversity alone is an insufficient criterion for library design and other factors should also be taken into account. For example, the physicochemical properties of the molecules that determine effects such as ADME[6] are important as well as other factors such as cost and availability of reactants. Consequently, the focus in combinatorial library design has now shifted toward designing libraries based on a number of properties simultaneously.[4,7−10]

The techniques that have been developed for designing combinatorial libraries can be divided into reactant-based and product-based methods. In reactant-based methods,[3,11] optimized subsets of reactants are selected on the assumption that when they are combined combinatorially they will approximate to an optimized set of products. However, evidence suggests that more diverse libraries can be achieved by performing the design in product space.[12−14] The product-based approaches are computationally demanding[9] and are typically implemented via an optimization technique such as a genetic algorithm[7,15−17] or simulated annealing.[8,9,18,19]

The SELECT program[7] is an example of a product-based approach to library design in which combinatorial subsets are selected from a fully enumerated virtual library using a genetic algorithm (GA). SELECT takes as input a virtual library together with molecular descriptors that have been calculated for each molecule within the library. Initially, SELECT was developed to optimize a single objective, namely, the diversity of the combinatorial subset using a distance-based diversity index. Each chromosome of the GA represents a combinatorial subset of the virtual library encoded as lists of reactants selected from each reactant pool. The GA begins with a population of individuals that are initialized with random values and thus represent randomly selected combinatorial subsets. A chromosome is scored by enumerating the combinatorial subset it represents and then measuring the diversity of the subset via a fitness function, as shown by

$$f(n) = \text{diversity}$$

Typically diversity is measured as the sum-of-pairwise dissimilarities calculated using the cosine coefficient and Daylight fingerprints, although other diversity indices and other descriptors can also be used.[13] The population is then sorted according to fitness. Next, the GA enters an iterative phase where individuals are chosen for reproduction using roulette wheel parent selection, reproduction takes place via mutation or crossover, the newly created individuals are scored and inserted into the population replacing the worst individuals, and the population is resorted. The iterations continue until convergence is reached. The number of

* Corresponding author phone: +44 1142 222 652; e-mail: v.gillet@sheffield.ac.uk.

† Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield.

‡ Current address: Icosystem Corporation, 545 Concord Ave., Cambridge, MA 02138.

§ Department of Automatic Control and Systems Engineering, University of Sheffield.

|| GlaxoSmithKline.

10.1021/ci010375j CCC: $22.00
Published on Web 01/31/2002

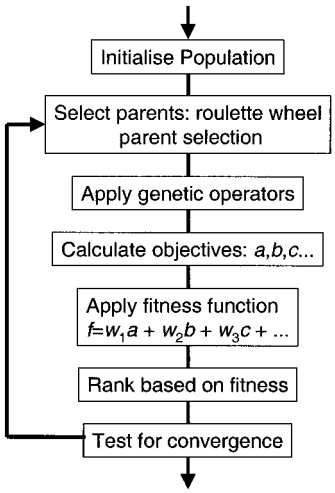

Figure 1. Overview of the GA implemented within SELECT.

chromosomes selected for reproduction is determined by the replacement rate; for example, a typical value is 10%. Convergence occurs when there has been no change in the fitness of the best individual for a user-specified number of iterations. Most of the parameters of SELECT are configured via an input file, for example, population size, the relative rates of crossover versus mutation, and the replacement rate. SELECT has been used to demonstrate the benefits of performing product-based library design over reactant-based design.[12,13]

Traditionally optimization techniques, such as genetic algorithms and simulated annealing, have tended to deal with a single optimization criterion or objective, i.e., the minimization or maximization of a single measure or quantity. However, most practical search and optimization applications are characterized by the existence of many objectives against which a final search result is measured. For example, as already described, in the library design context such objectives would typically include diversity, some measure of druglikeness, and cost. However, optimal performance in one objective often implies unacceptably low performance in one or more of the other objectives. For example, libraries designed on diversity alone have a tendency to contain molecules that exist in nondruglike regions of chemistry space, e.g., molecules with high molecular weights.[4] Thus, there is a need for compromise and the search for solutions that offer acceptable performance in all objectives, even though they may be suboptimal in the single objective sense. One way of achieving a compromise is to combine objectives via a weighted-sum fitness function. For example, SELECT has been extended to perform multiobjective optimization in product space where other properties of the libraries can be optimized simultaneously with diversity, such as the physicochemical property profiles of the libraries. The fitness function now has the form

$$f(n) = w_1(\text{diversity}) + w_2(\text{property1}) + w_3(\text{property2}) + ...$$

where the weights (w_1, w_2, w_3, etc.) are user defined and the properties (property1, property2, etc.) can include physicochemical property profiles such as molecular weight profile or other calculable properties such as cost. Typically, each objective is normalized before being combined. An outline of the GA implemented in SELECT is shown in Figure 1.

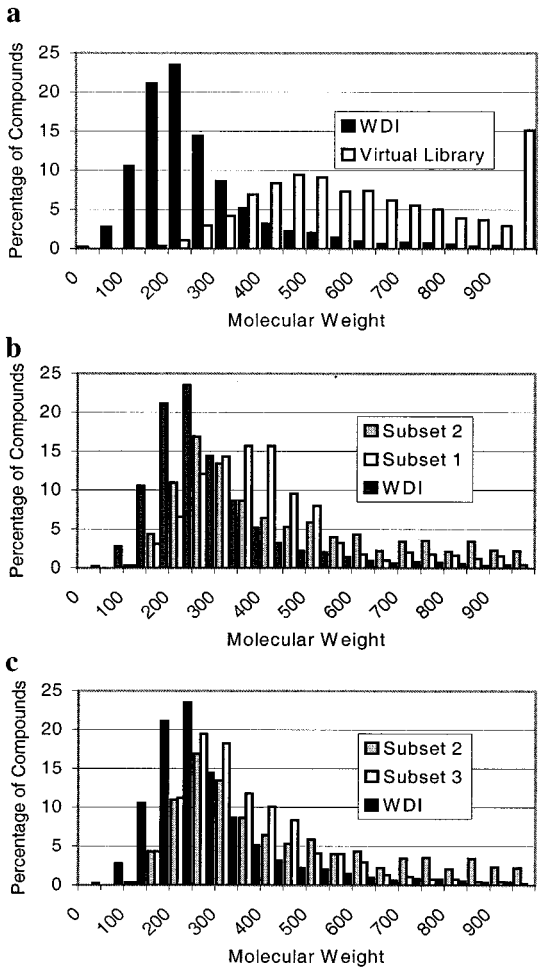

Figure 2. (a) Molecular weight profile of the 10K virtual amide library (white) is shown superimposed on the molecular weight profile found in WDI (black). (b) Molecular weight profiles of amide libraries designed using SELECT are superimposed on the molecule weight profile found in WDI (black). Subset 1 (white) is optimized on diversity alone, whereas subset 2 (gray) is optimized on molecular weight profile and diversity simultaneously. The relative diversities of subset 1 and subset 2 are 0.596 and 0.582, respectively. It can be seen that an improved molecular weight profile is achieved at the expense of a small change in diversity. (c) A further improvement in the molecule weight profile is achieved by assigning penalties to the various bins that reflect their relative importance, subset 3.

The benefits of performing multiobjective optimization in library design are illustrated in Figure 2 for an amide library. SELECT was configured to design 30 × 30 combinatorial subsets from a two-component amide library consisting of 100 amines and 100 carboxylic acids (representing a virtual library of 10K amides). The molecular weight profile of the virtual library is shown in Figure 2a. Two runs of SELECT were performed. In the first, the combinatorial subset (subset 1) was optimized on diversity alone, where diversity was measured as the normalized sum-of-pairwise dissimilarities using Daylight fingerprints and the cosine coefficient.[12] The maximum diversity achieved was 0.596. In the second run, the subset (subset 2) was optimized on both diversity and molecular weight profile. Specifically, the aim was to maximize diversity, using the same diversity measure as before, while simultaneously minimizing the root-mean-square deviation (rmsd) between the molecular weight profile of the library and the molecular weight profile found in the

World Drug Index (WDI).[20] The weighted-sum fitness function was specified as

$$f(n) = w_1(1 - D) + w_2\Delta\mathrm{MW}$$

where D is diversity, included in the fitness function as $1 - D$ so that the term $w_1(1 - D)$ is minimized; ΔMW is the normalized rmsd between the two profiles; w_1 and w_2 are set to 1.0 so that the objectives are weighted equally. The diversity of subset 2 is 0.582. Figure 2b shows the molecular weight profiles of subset 1 (white) and subset 2 (gray) superimposed on the molecular weight profile found in WDI (black). It can be seen that overall a more druglike molecular weight profile is achieved for subset 2 at the expense of a relatively small change in diversity. The increase in occupancy of the higher molecular weight bins in subset 2 relative to subset 1 is due to the characteristics of the virtual library itself (there are many high molecular weight compounds in the virtual library as seen in Figure 2a) and the fact that all bins are equally weighted in the rmsd calculation. The occupancy of the bins representing high molecular weights can be reduced by adopting a strategy similar to that described by Brown et al.,[9] where penalties are assigned to individual bins to reflect their relative importance, as shown in Figure 2c for subset 3.

A similar weighted-sum approach has been used in several other programs for library design,[8−10] and there are many other examples in computational chemistry where multiple selection criteria are combined through the use of a weighted-sum fitness function. For example, in the GOLD program for flexible docking the fitness function of its GA involves weighted components that reflect contributions due to hydrogen bonding, steric interactions, and the internal energy of the ligand.[21]

The advantage of combining multiple objectives via a weighted-sum fitness function is that a single compromise solution is produced. However, there are many limitations to such an approach. These are summarized as follows:

(a) The definition of the fitness function can be difficult especially with noncommensurable objectives; for example, in library design it is not obvious how diversity should be combined with cost.

(b) The setting of the weights is nonintuitive; for example, in the SELECT program several trial-and-error experiments may be required to choose appropriate weights.[22]

(c) The fitness function determines the regions of the search space that are explored, and combining objectives via weights can result in some regions being obscured.

(d) The progress of the search or optimization process is not easy to follow since there are many objectives to monitor simultaneously.

(e) The objectives may be coupled, thus implying conflict and competition, which can make it more difficult for the optimization process to achieve reasonable or acceptable results.

(f) A single solution is found which is typically one among a family of solutions that are all equivalent in terms of the overall fitness, although they may have different values of the individual objectives. For example, consider a two-objective problem where the fitness function is defined as

$$f(n) = w_1x + w_2y$$

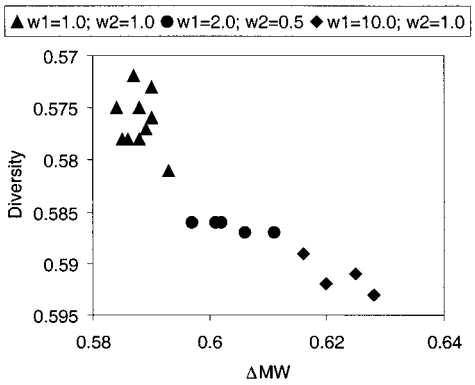

Figure 3. SELECT configured to choose libraries optimized on two objectives simultaneously, namely, diversity and molecular weight profile. The effect of varying the relative weights of the objectives is shown by the three distinct clusters of solutions.

where x and y are hypothetical objectives and w_1 and w_2 are both set to unity. The solution $x = 0.4$, $y = 0.5$ has the same fitness (0.9) as the potential solution $x = 0.5$, $y = 0.4$, and thus both solutions can be considered as equivalent; typically, however, only one of them will be found.

Some of these limitations are illustrated in Figure 3, which shows the results of 20 runs of SELECT for the amide library design problem described previously. The libraries are optimized on diversity and molecular weight profile simultaneously. The y-axis is reversed so that diversity decreases with distance from the origin, and the aim is to find a solution that is as close to the origin as possible. The triangles show the results found when both weights, w_1 and w_2, are unity (10 runs). These points appear to cluster in the top left-hand corner of the graph, favoring low (good) values of ΔMW with relatively poor values for diversity. The average value of the weighted-sum fitness function is 1.163 (standard deviation 0.004). The small variation in values is due to the stochastic nature of the GA. Increasing the relative importance of diversity by adjusting the weights to $w_1 = 2$ and $w_2 = 0.5$ results in a cluster of solutions with improved diversity but at the expense of higher values of ΔMW, shown by the circles (five runs). The diamonds show results obtained for $w_1 = 10$ and $w_2 = 1$, respectively (five runs), with the points shifted further in favor of diversity at the expense of the molecular weight profiles of the libraries. Each of the solutions found represents a different compromise between the two objectives, and all are equally valid. Thus, finding an acceptable solution using a weighted-sum fitness function may require that many runs are performed using different weights to ensure that adequate coverage of the search space is achieved.

MULTIOBJECTIVE OPTIMIZATION

As seen above, multiobjective optimization problems tend to be characterized by a family of alternative solutions that are all considered equivalent in the absence of additional information. Multiple solutions arise even in the simplest case of two competing objectives, and in general, as the number of objectives increases the problem of finding a satisfactory compromise solution rapidly becomes increasingly complex. Generally, a hypersurface exists in the search space that represents a continuum of solutions where all the solutions are seen as equivalent and they all represent compromises or tradeoffs between the various objectives.

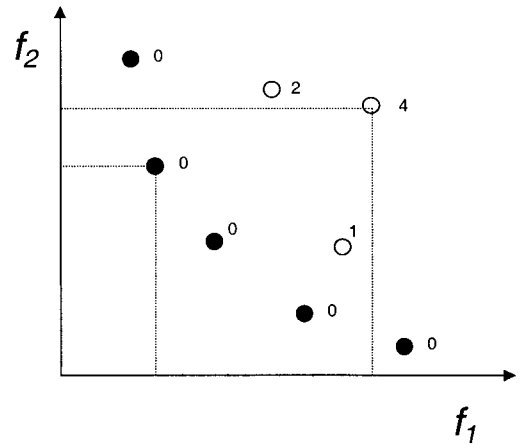

Figure 4. Potential solutions in a two-objective (f_1 and f_2) problem. The circles represent pairs of values that reflect the two objectives. The solid circles are nondominated solutions and fall on the Pareto frontier. Dominated solutions are shown as unfilled circles. In MOGA, individuals are ranked according to the number of times they are dominated; thus the nondominated solutions are given rank zero and the dominated solutions are given ranks as shown.

Conventional optimization techniques such as gradient-based and simplex-based methods, and also less conventional ones such as simulated annealing, are difficult to extend to the *true* multiobjective case, because they are not designed with multiple solutions in mind. In practice, multiobjective problems have to be reformulated as a single objective prior to optimization, leading to the production of a single solution per run of the optimizer. Evolutionary algorithms, however, operate with a population of individuals and are thus well suited to search for multiple solutions in parallel; hence they can be readily adapted to deal with multiobjective search and optimization. Fonseca and Fleming give a thorough survey of various approaches to multiobjective optimization[23] and have developed a multiobjective evolutionary framework called MOGA (MultiObjective Genetic Algorithm).[24]

In MOGA, multiple objectives are handled independently without summation and without the need for normalization. The method attempts to map out the hypersurface in the search space where all the solutions are seen as equivalent. The continuum of points on the hypersurface is referred to as a frontier or surface. The actual solutions are called nondominated or Pareto solutions. (Pareto was a French mathematician who dealt with this issue toward the end of the nineteenth century.) In multiobjective optimization, a set of nondominated solutions is sought rather than a single solution. A nondominated solution is one where an improvement in one objective results in a deterioration in one or more of the other objectives when compared with the other solutions in the population. Thus, one solution dominates another if it is either equivalent or, better, in all the objectives and, strictly, it is better in at least one objective. In MOGA, the ranking of the population is based on dominance (also known as Pareto ranking) instead of ranking based on fitness that is used in a standard GA. Pareto ranking allows the population to map out the Pareto frontier or tradeoff surface by evolving multiple nondominated solutions simultaneously.

Consider a problem with two objectives f_1 and f_2 where the aim is to minimize both of the objectives, as illustrated in Figure 4. The graph represents a number of potential solutions to the problem with each point representing a pair of values that reflect the two objectives. If lines parallel to the axes are drawn from each point, a solution is nondominated if the square area bounded by the two lines and the axes does not contain any other points. The nondominated or Pareto solutions are represented by the solid circles. In MOGA, individuals are ranked according to the number of times they are dominated; thus the nondominated solutions are given rank zero and the dominated solutions (unfilled circles) are given ranks as shown.

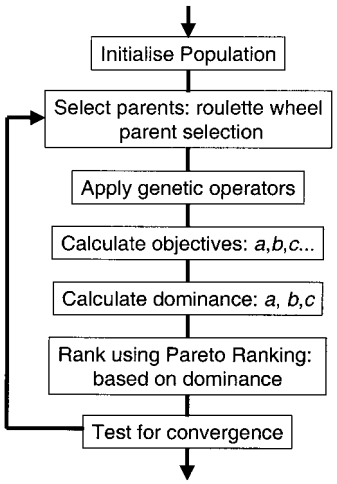

Figure 5. Basic structure of the MOGA implemented in MoSELECT.

Pareto frontiers have been used in many applications of multiobjective optimization; see, for example, the review article by Coello Coello.[25] The only application to date in chemoinformatics of which we are aware is the work of Handschuh et al.,[26] who have used Pareto optimization in the GA they developed for the flexible superposition of three-dimensional structures. Their method finds the maximum common substructures (MCSS) between two molecules. The search for the MCSS involves two criteria: the number of atoms in the substructure and the fit of the matching atoms. These are conflicting criteria since a larger MCSS will by definition have a larger deviation in the coordinates of the superimposed atoms when the larger MCSS is a superset of the smaller. Rather than attempting to combine the different criteria into a single weighted-sum fitness function, a set of Pareto solutions is obtained at the end of each run whereby an optimal geometric fit is found for each possible size of MCSS.

We have adopted the MOGA approach in a new development of SELECT, called MoSELECT (MultiObjective SELECT),[27,28] for the design of combinatorial libraries optimized on multiple objectives. MoSELECT overcomes many of the limitations of the weighted-sum approach used in SELECT. In the following sections we begin by describing the modifications made to SELECT to produce MoSELECT. We then describe a series of experiments that have been performed to test the operation of MoSELECT and to compare its performance with SELECT. The experiments have been performed on a number of different combinatorial libraries and library design objectives.

MOSELECT

The algorithm used in MoSELECT is outlined in Figure 5. The definition of chromosomes and the reproduction

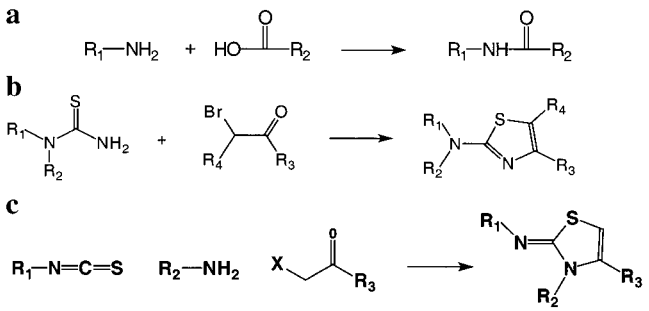

Figure 6. (a) Amide library. (b) 2-Aminothiazole library. (c) Thiazoline-2-imine library.

operators are the same as used in SELECT; however, in MoSELECT, individuals are chosen for reproduction using Pareto ranking based on the values of the individual objectives rather than the weighted-sum fitness function implemented in SELECT. Each time a new chromosome is produced the values of all of the objectives are calculated and stored independently with the chromosome. After each iteration of the MOGA the rank of each chromosome is calculated as the number of chromosomes in the population by which it is dominated; see Figure 4. Thus, nondominated individuals are assigned rank zero, individuals that are dominated by one other chromosome are assigned rank one, and so on. Roulette wheel parent selection is employed to bias parent selection toward the "best" members of the population. Each chromosome is assigned a segment of the roulette wheel with segment size determined by rank so that individuals are selected for reproduction with a probability that is inversely proportional to rank (or dominance).

In SELECT, performance is monitored by the progress of the fittest chromosome and the program converges when there is negligible change in this. In MoSELECT, however, there is no longer a single value assigned to an individual; hence the convergence test used in SELECT cannot be applied. Thus, in the initial experiments both SELECT and MoSELECT are run for a fixed number of iterations to allow a comparison to be made. Potential convergence criteria that could be applied in MoSELECT are investigated in later experiments.

EXPERIMENTAL DETAILS

The combinatorial libraries used in the experiments are shown in Figure 6. They are a two-component amide library, a two-component 2-aminothiazole library, and a three-component thiazoline-2-imine library. The amide library represents a virtual library of 10K compounds formed by the coupling of 100 amines and 100 carboxylic acids, extracted at random from the SPRESI database.[29] The 2-aminothiazole virtual library of 12 850 products consists of 74 α-bromoketones coupled with 170 thioureas. In this case, reactants for each pool were extracted from the Available Chemicals Directory (ACD)[30] and filtered using the ADEPT software[31] (reactants having molecular weight greater than 300 and more than 8 rotatable bonds were removed, and a series of substructure searches were performed to remove reactants that contain undesirable substructural fragments). The thiazoline-2-imine library consists of 70 092 virtual products constructed from 12 isothiocyanates, 99 amines, and 59 haloketones, again extracted at random from SPRESI.

Each virtual library was enumerated, and various properties were calculated for the product molecules comprising each library (namely, 1024-bit Daylight fingerprints, molecular weight (MW), number of rotatable bonds (RB), number of hydrogen bond donors (HBD), and number of hydrogen bond acceptors (HBA)). Except where otherwise stated, diversity was calculated as the sum of pairwise dissimilarities using the cosine coefficient.[7] In the examples presented here the virtual libraries are enumerated upfront and descriptors are calculated for all products prior to running MoSELECT. However, the MOGA technique is not restricted to the descriptors used here nor is it restricted to prior enumeration: the technique can also be applied when libraries are enumerated and descriptors are calculated on the fly.

THE AMIDE LIBRARY

The first experiment was designed to investigate the overall performance of MoSELECT. The aim was to select 30 × 30 combinatorial subsets from the 10K amide virtual library using two objectives, namely, diversity and molecular weight profile. As in the libraries illustrated in Figures 2 and 3, the aim was to maximize diversity while minimizing the rmsd between the molecular weight profile of the library and the molecular weight profile found in WDI. MoSELECT was run for 5000 iterations with a population size of 50. The progress of the search is shown in Figure 7a. As in Figure 3 the *y*-axis is reversed so that the direction of improvement for both objectives is toward the bottom left-hand corner of the graphs. In each of the graphs the Pareto frontier, i.e., the set of nondominated individuals in the current population, is represented by circles with the crosses representing the dominated individuals. The top left-hand graph shows the initial population, with the remaining graphs showing progress at 100, 500, 1000, 2500, and 5000 iterations. The improvement in multiobjective space is demonstrated by the advance of the Pareto frontier toward the origin, especially during the first 1000 iterations. Negligible change in the position of the Pareto frontier is seen over the next 4000 generations. As the search progresses, the number of solutions that are nondominated increases from 4 in the initial population to 17 in the final population. Figure 7b gives an expanded view of the final results after 5000 iterations, where it can be seen that the result of the search is a family of equivalent solutions on the Pareto frontier that span a range of values in each objective. The competing nature of the two objectives is clearly seen with increasing diversity leading to solutions with less favorable molecular weight profiles, and vice versa; thus the relationship between the two objectives is apparent from a single run of MoSELECT.

COMPARISON OF SELECT AND MOSELECT

The next set of experiments was designed to examine the robustness of MoSELECT and to compare its performance with SELECT. The MoSELECT run described in the previous experiment was repeated 10 times, and the family of nondominated solutions found at the end of each run was noted. These results were then compared with the 20 runs of SELECT already reported in Figure 3, for different relative weights attributed to diversity and molecular weight profile in the weighted-sum fitness function. Finally, SELECT was configured to optimize each objective separately in order to

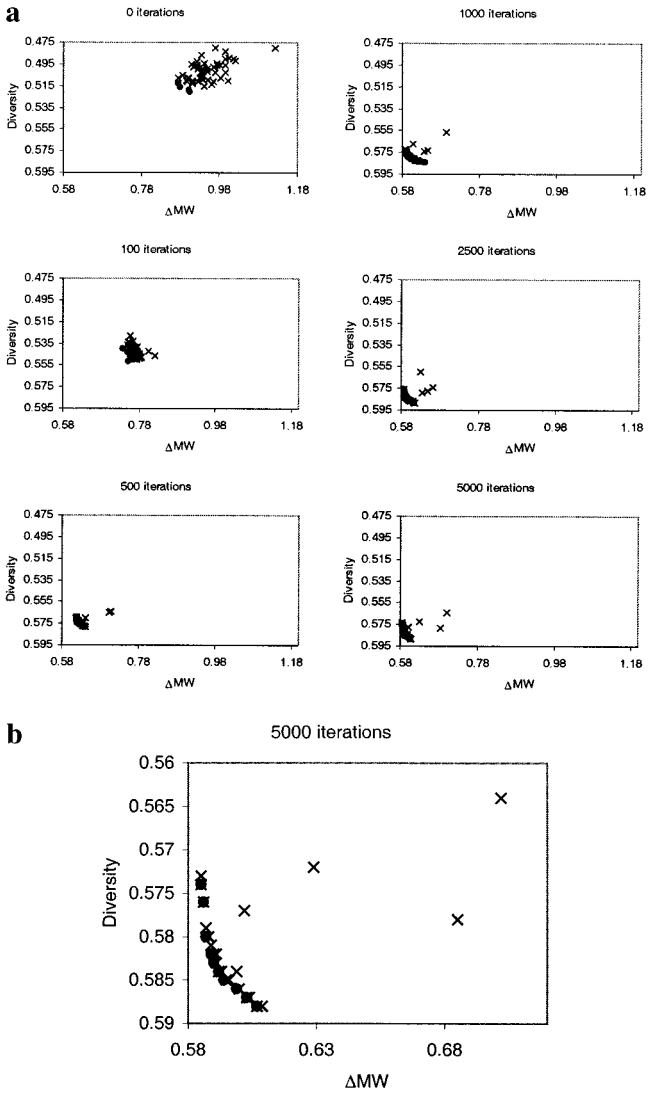

Figure 7. (a) Progress of a MoSELECT run for the optimization of an amide library using two objectives, namely, diversity and molecular weight (MW). The nondominated solutions are shown as filled circles with the dominated solutions shown as crosses. (b) The final graph of (a) is shown expanded.

find optimized values of each objective independently. The values found over 10 runs were an average of 0.593 for diversity (standard deviation 0.002) and 0.585 for ΔMW (standard deviation 0.002).

The results of the comparison are shown in Figure 8. The nondominated solutions found in the 10 MoSELECT runs are shown as crosses. The even spread of solutions indicates that the Pareto frontier has been mapped efficiently. The dashed lines show the optimum values achieved when diversity and molecular weight profile are optimized independently, and it can be seen that the MoSELECT runs also include solutions at these extremes.

The results for the runs of SELECT shown in Figure 3 are also included in Figure 8 as the solid triangles, circles, and diamonds. It can be seen that one run of SELECT will produce a single solution that typically lies somewhere on the Pareto frontier of a MoSELECT run; i.e., the solution is likely to be close to one of those within a family of solutions produced by MoSELECT. However, in general, each time SELECT is run, a different member of the family will be found and, as described in the Introduction, it is usually

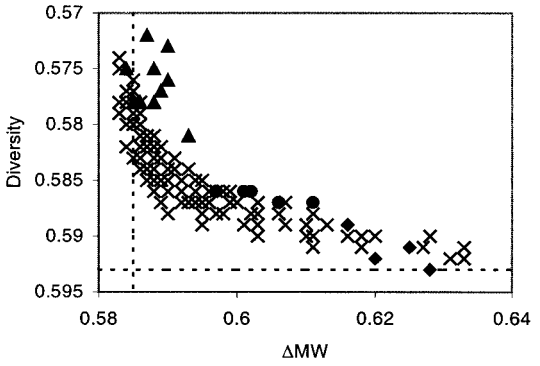

Figure 8. Distribution of nondominated solutions for the design of 30 × 30 amide subset optimized on diversity and molecular weight profile found over 10 runs of MoSELECT shown as crosses. The results found in the SELECT runs shown in Figure 3 are superimposed on the MoSELECT results. The dashed lines show the optimum values achieved when diversity and molecular weight profile are optimized independently.

necessary to vary the relative weights of the two objectives in order to find an acceptable solution. This has the effect of mapping out the Pareto surface by performing multiple runs, whereas MoSELECT will produce solutions that span the entire Pareto frontier in a single run. There is some variation in the quality of the solutions found by MoSELECT due to the stochastic nature of a MOGA; however, even the worst family of solutions found by MoSELECT contains individuals that dominate many of the SELECT solutions. Furthermore, there are no significant overheads associated with adopting Pareto ranking with the MoSELECT runs taking on average 27 min compared with 31 min for the SELECT runs (SGI R10K workstation running at 195 MHz). The average number of solutions found for the MoSELECT runs is 31 (standard deviation 12), out of a population of 50, which contrasts with the single solution found in a run of SELECT. Once a family of solutions has been found, the user can then browse through them and choose one that is acceptable based both on the objectives used in the search and on other criteria, for example, availability of reactants.

CONVERGENCE CRITERIA IN MOSELECT

As mentioned previously, SELECT is usually run with a convergence criterion that is used to terminate the search. Convergence is reached when no change is seen in the fitness function of the best individual solution over 250 iterations (measured at 50 iteration intervals). MoSELECT, however, aims to identify a family of nondominated solutions, all of which are equally valid but which may have different values for the objectives, There is no longer a single summed value assigned to a potential solution; thus the convergence criterion used in SELECT is inappropriate for MoSELECT.

The next experiments were designed to investigate the effect of two different convergence criteria that have been implemented in MoSELECT. The first is an adaptation of that used in SELECT and attempts to monitor the progress of the Pareto frontier, rather than the single best solution monitored in SELECT. Once the initial population has been created, a copy of the nondominated set is maintained. The search then proceeds for some given number of iterations, for example 50, after which the current nondominated set is compared with the previously stored nondominated set. If

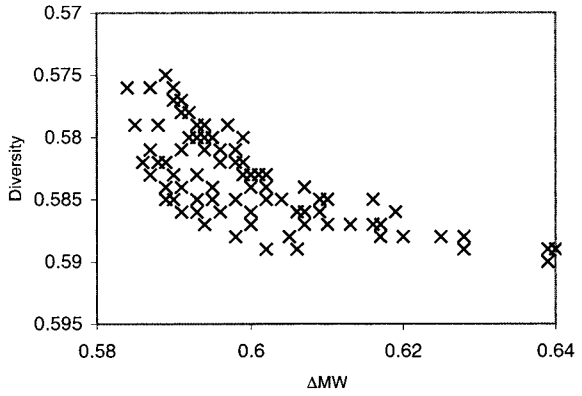

Figure 9. Distribution of nondominated solutions for the design of 30 × 30 amide subset optimized on diversity and molecular weight profile found over 10 runs of MoSELECT with convergence criterion defined as no change in the progress of the Pareto frontier.

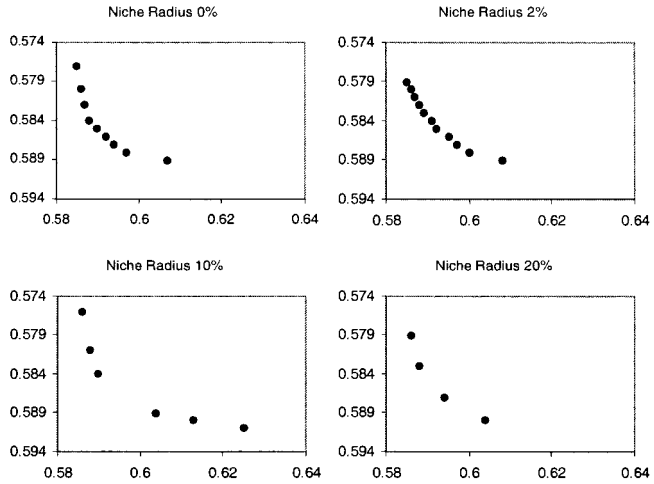

Figure 10. Effect of niche induction on the amide library. As the niche radius is increased, the number of nondominated solutions is decreased but the spread of the solutions is maintained.

none of the chromosomes of the previous nondominated set are dominated by the current nondominated set, then the Pareto frontier is said to be unchanged over the 50 iterations. The previous nondominated set is replaced by the current nondominated set, and the search continues for another cycle of 50. If the Pareto frontier is unchanged over a total of 250 iterations, then the search is terminated.

The distribution of Pareto frontiers over 10 runs of MoSELECT with this convergence criterion in place is shown in Figure 9. The distribution is similar to that shown in Figure 8 when no convergence criterion is applied; however, there does appear to be some loss of coverage of the extreme values and the spread of frontiers is broader, indicating some loss of robustness of the algorithm. Despite the small loss of coverage, the use of such a convergence criterion can be advantageous since the results are achieved for a significantly reduced number of GA cycles. The mean number of iterations to convergence for MoSELECT is 1715 (standard deviation 525), compared to the 5000 iterations used in Figure 8. This can be compared with a mean of 1245 iterations (standard deviation 291) over 10 SELECT runs for the same problem with the convergence criterion described above. It should be noted that, although the average number of iterations to convergence is greater for MoSELECT than SELECT, a single MoSELECT run produces an entire family of equivalent solutions whereas one run of SELECT produces a single solution only. The large standard deviations indicate a high degree of variability in the number of iterations required to reach convergence in both algorithms.

The second convergence criterion that was investigated involves calculating the percentage of nondominated solutions in the Pareto set as the search progresses. As already seen in the earlier experiments, the number of nondominated solutions in the population tends to increase throughout the run (in the experiment illustrated in Figure 7 the number of nondominated solutions increased from 4 to 17 out of a total of 50 individuals); hence it was felt that this might provide a method of testing for convergence. The aim was to determine a suitable threshold such that once the percentage of nondominated solutions in the population was above the threshold the search would be terminated. This method, however, did not prove to be effective since there was no clear trend to indicate what a valid threshold should be. Thus, it was concluded that if a test for convergence is desirable for efficiency reasons, then the first convergence criterion is more reliable, that is, monitoring the progress of the Pareto frontier.

NICHE INDUCTION

One potential problem associated with GAs and with MOGAs is that of genetic drift or speciation. This is where the algorithm tends to "drift" toward areas where there are clusters of closely matched solutions and leaves other areas not well mapped out or not explored at all. The effect of speciation can be reduced by employing the technique of niche induction. In niche induction, the density of solutions within hypervolumes of either the decision or the objective variable spaces is restricted.

In MoSELECT, the objective space was used to attempt to spread the distribution of solutions over the Pareto frontier. After each iteration, the Pareto frontier is identified and each solution on the frontier is compared with all others in order to establish proximity in the various objectives. This is implemented as an order-dependent process where the first solution encountered is positioned at the center of a hypervolume, or niche. If the (absolute) difference in the objectives of the next solution and the objectives of any solution that already forms the center of a niche is within a given threshold, for all objectives, the fitness (or dominance) of the current solution is penalized; otherwise it forms the center of a new niche. The threshold is also known as the *niche radius.* This process is repeated for all solutions on the Pareto frontier. The niche radius is set dynamically throughout a run and is given as a percentage of the range of values that exist for each objective on the current Pareto frontier. Figure 10 shows the final Pareto frontiers of a series of MoSELECT runs for the amide library using different niche radii. The first graph shows the solutions found when no niching is performed; the subsequent graphs show the solutions found as the niche radius is increased progressively up to 20%. The loss of resolution as the niche radius is increased is evident, and a niche radius of 10% provides a good compromise for this library design problem having a relatively small number of evenly spread solutions.

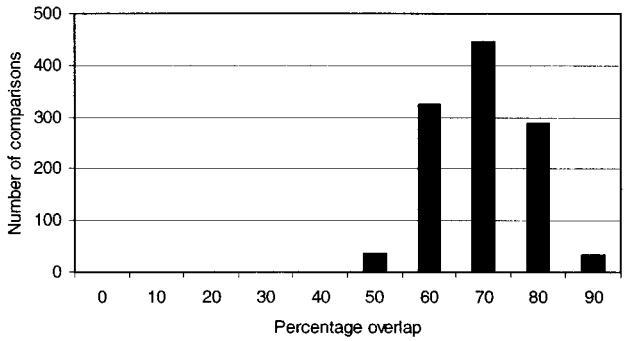

Figure 11. Distribution of pairwise overlaps among the 48 solution libraries found in a MoSELECT run.

CLUSTERING THE SOLUTIONS

Niche induction has been applied as the search progresses in an attempt to increase the efficiency of the search by reducing the number of potential solutions to explore and to increase the effectiveness of the search by preventing speciation. It could also be used as a way of clustering the final Pareto set according to their spread in objective space. An alternative way of clustering the solutions is according to their similarities in terms of the actual molecules contained in the libraries, or the degree of overlap of the libraries. This is illustrated in Figure 11 for the amide 30 × 30 subsets where a MoSELECT run with a population of 50 resulted in a final Pareto set consisting of 48 solution libraries. A pairwise overlap matrix was constructed for the 48 libraries, where the overlap between each pair of libraries was calculated as the number of molecules in common between the libraries divided by the library size. The distribution of the resulting 1128 ($N(N-1)/2$) overlap values is shown.

In principle, it is possible to implement niche induction based on library comparisons during the search process itself; however, comparing the libraries represented by chromosomes is more computationally demanding than merely comparing the values of the objectives. Thus, the efficiency of the search process would be compromised. An alternative computationally more efficient approach would be to compare the overlap of the selected reactants.

INCREASING THE NUMBER OF OBJECTIVES

So far, the performance of MoSELECT has been investigated for library designs based on just two objectives, which is the simplest possible case of multiobjective optimization. The following experiments investigated the effect of increasing the number of objectives. The same amide library was used with the number of objectives increased to five, namely, diversity and profiles of the following properties: molecular weight (MW), occurrence of rotatable bonds (RB), occurrence of hydrogen bond donors (HBD), and occurrence of hydrogen bond acceptors (HBA) (the SMARTS definitions for RB, HBD, and HBA are given in Table 1). The aim was to minimize the difference in the distribution of each property with respect to its distribution in WDI while simultaneously maximizing diversity.

When there are more than two objectives, it is no longer possible to show the tradeoffs between all the objectives in a simple two-dimensional graph. Instead, tradeoffs between multiple objectives can be illustrated using parallel graphs. Figure 12 shows a parallel graph representation of the Pareto

Table 1. SMARTS Definitions for Rotatable Bonds (RB), Hydrogen Bond Donors (HBD), and Hydrogen Bond Acceptors (HBA)

property	SMARTS
RB	[!$(*#*)&!D1]-&!@[!$(*#*)&!D1]
HBD	[!#6;!H0]
HBA	[$([!#6;+0]);!$([F,Cl,Br,I]);!$([o,s,nX3]);!$([Nv5,Pv5,Sv4,Sv6])]

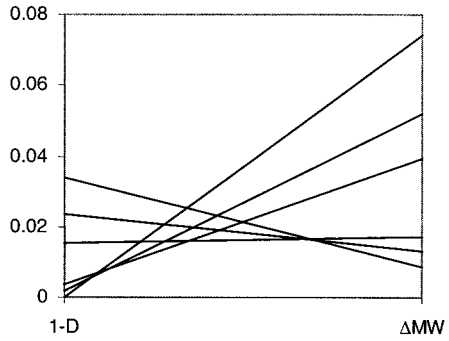

Figure 12. Parallel graph representation of the two-objective problem illustrated in Figure 10 at 10% niching.

frontier for the two-objective problem illustrated in Figure 10 at 10% niching. The horizontal axis represents the two objectives, namely, molecular weight profile and diversity, and the vertical axis represents the values of each objective. Diversity is represented as its complement ($1 - D$) so that the direction of improvement in both objectives is toward zero on the *y*-axis. The two objectives have been scaled to allow them to be plotted on the same graph. Scaling was achieved by finding best and worst values for each objective independently using SELECT. For example, the best value of diversity is found by configuring SELECT to maximize diversity, whereas the worst value is found by configuring SELECT to minimize diversity. The best and worst values of diversity for the amide library are 0.593 and 0.363, respectively. Thus zero on the *y*-axis represents the best value that can be achieved when an objective is optimized independently. (This method of scaling is used in all subsequent parallel graphs.) Each continuous line in the graph in Figure 12 represents one solution on the final Pareto frontier. The competing nature of the objectives is shown clearly by the crossing lines. In this example, it can be seen that close to the best values possible in each objective are achievable; however, they are not achievable simultaneously in a single solution, so an optimum value of diversity corresponds to a suboptimal value in molecular weight profile, and vice versa.

Increasing the number of objectives over which a library is optimized results in an increase in the size of the search space that MoSELECT should explore, and hence a larger population is required to ensure that the search space is well covered. An initial run of the amide library over five objectives with a population size of 200 showed that there was a tendency for speciation to occur. This is illustrated in Figure 13a, which shows solutions on the Pareto frontier, in a parallel graph representation, after 5000 iterations of MoSELECT. A total of 188 nondominated solutions were found; however, these were clustered in a relative small part of the search space.

Figure 13b shows the same library design problem run with niching, with the niche radius set at 30%. Only the

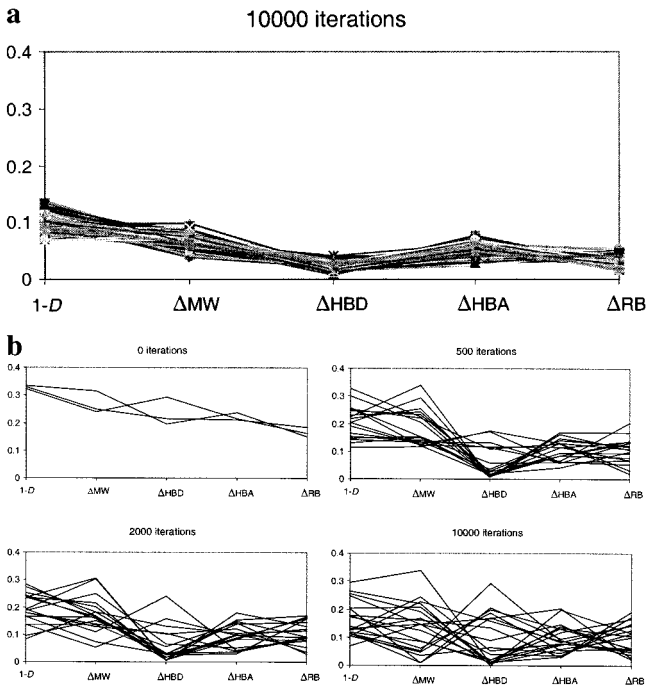

Figure 13. (a) Results of a MoSELECT run for the optimization of 30 × 30 amide library on five objectives, namely, molecular weight profile (MW), hydrogen bond donor profile (HBD), hydrogen bond acceptor profile (HBA), rotatable bond profile (RB), and diversity. The population size is 200 and MoSELECT is run without niching. (b) Progress of a MoSELECT run for the optimization of a 30 × 30 amide library on five objectives, namely, diversity ($1 - D$); molecular weight profile (ΔMW), hydrogen bond donor profile (ΔHBD), hydrogen bond acceptor profile (ΔHBA), and rotatable bond profile (ΔRB). Results are shown at 0, 500, 2000, and 10 000 iterations. The population size is 200 and the niche radius is 30%.

solutions on the Pareto frontier are shown in each of these graphs for clarity. It can be seen that as the search progresses the solutions drift in the direction of multiobjective improvement, i.e., to lower values on the vertical axis, indicating lower values of the objectives. Also, the number of nondominated solutions tends to increase. A much smaller number of solutions is found (24) after 5000 iterations compared to the run without niching; however, these solutions represent a much greater range of values in each of the objectives and they indicate that much more of the search space has been explored. The relatively large niche radius is appropriate because of the large search space and because the niche radius is applied to all objectives simultaneously. Competition between the objectives is evident, for example, between the profiles of molecular weight and hydrogen bond donors (ΔHBD) and between hydrogen bond donors (ΔHBD) and acceptors (ΔHBA), as shown by the crossing lines in the graph. Near-optimum values are achievable for all of the objectives; however, these are not achievable simultaneously in a single solution and a compromise should be sought. The relationship between all pairs of objectives could be examined by reordering the objectives on the horizontal axis. Where there is no competition between objectives, i.e., improvement in one corresponds to improvement in another, these would be shown by parallel lines and it would not be necessary to include both within the search process.

Increasing the population size leads to a consequent increase in search times with MoSELECT taking approximately ~97 min with population size 200 for 5000 iterations at 30% niching.

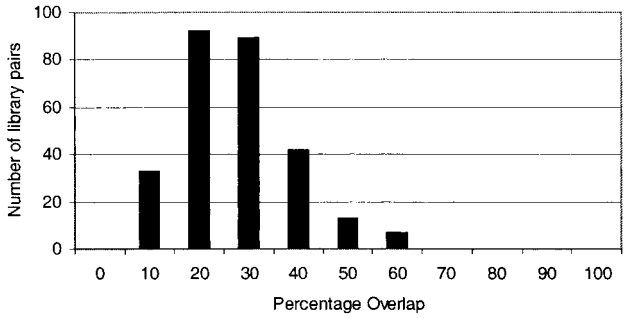

Figure 14. Overlap of 24 amide libraries shown above.

Figure 14 shows the overlap plot for the amide libraries optimized on five objectives. When this is compared with the plot shown in Figure 11, it can be seen that there is less overlap between the libraries and this indicates that forcing the solutions to be separated in this five-objective space seems to correspond to a separation in terms of the molecules shared between the different solution libraries.

THE 2-AMINOTHIAZOLE LIBRARY

The 2-aminothiazole library consists of 74 × 170 virtual products. The library design experiments involved the optimization of six objectives including cost, diversity, and profiles of molecular weight, hydrogen bond donors, hydrogen bond acceptors, and rotatable bonds. Reactant costs in price per gram were extracted from the Available Chemicals Directory.[30] When the cost for a particular reactant was not available, it was assigned a cost equal to the most expensive reactant in the corresponding reactant pool. Diversity was measured using a cell-based method.[32] The Cerius[2] (version 4.5) default set of topological parameters and physicochemical properties was calculated for each of the molecules with the descriptor space being reduced to three dimensions using principal components analysis.[33] The resulting three-dimensional space was partitioned into a series of 1134 cells (9 × 14 × 9), and each molecule was then mapped to a cell. The virtual library of 12 580 products occupies a total of 364 cells. The aim of the runs was to select 15 × 30 combinatorial subsets (450 compounds) with maximum cell coverage, minimum cost, and druglike distributions of the various physicochemical properties.

First, SELECT was run to maximize diversity alone in order to find the maximum cell coverage achievable without considering the additional objectives. The maximum coverage found was 282 cells for a library of cost $149,141. SELECT was then run to find a library with minimum cost, this library cost $1,485 and occupies 68 cells. Thus it can be seen that diversity is in competition with cost, with maximum diversity corresponding to a high cost and a minimum cost library corresponding to low diversity. MoSELECT was then run to optimize cell coverage simultaneously with cost. The solutions found on the Pareto frontier are shown in Figure 15. The dashed lines show the maximum diversity and minimum cost achievable when each objective is optimized independently. In this example, it is likely that a compromise would be sought between diversity and cost. The extreme values for diversity and cost are not found.

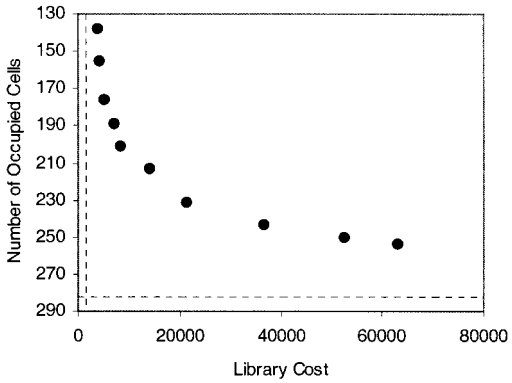

Figure 15. Optimizing cell based diversity simultaneously with cost for 15 × 30 subsets selected from the 74 × 170 2-aminothiazole libraries. MoSELECT was run with a population of 50 and a niche radius of 10%.

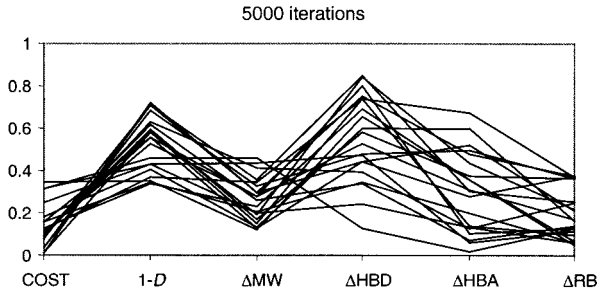

Figure 16. Optimizing 15 × 30 2-aminothiazole subsets based on six objectives: minimum cost, maximum cell coverage (represented as $1 - D$), and profiles of molecular weight, hydrogen bond donors, hydrogen bond acceptors, and rotatable bonds. MoSELECT was run with a population of 200 and a niche radius of 30%.

The parallel graph in Figure 16 shows the results of running MoSELECT using all six objectives simultaneously. Solutions are found with near-optimum cost and hydrogen bond acceptor and rotatable bond profiles; however, optimum diversity is further compromised by the inclusion of additional objectives, relative to Figure 15. Some relationships between the objectives are apparent: diversity competes with both cost and molecular weight profile; molecular weight profile also appears to be in competition with hydrogen bond donor profile. The relationship between donors and acceptors is less evident in this multiobjective case. Searching over the entire Pareto surface results in solutions that have extreme values over many of the objectives; however, it is likely that a compromise solution would represent the best combinatorial library to synthesize. MoSELECT allows the library designer to choose a solution that has acceptable values over as many of the objectives as possible.

INCREASING THE NUMBER OF COMPONENTS IN THE LIBRARY

The three-component thiazoline-2-imine library of size 12 × 99 × 59 (70 092 virtual products) was used to investigate further performance aspects. MoSELECT was configured to design 3 × 20 × 10 libraries optimized on the five objectives used in the amide experiment and to run for 5000 iterations with a population size of 200 at 30% niching. The Pareto frontier, shown in Figure 17, consists of 20 individuals found after 5000 iterations. Here, near-optimum values are achievable for each of the objectives, although these are clearly not achievable simultaneously. Competition is evident between some of the objectives, for example, between diversity and molecular weight profile, between donor and acceptor profiles, and between molecular weight and donor profiles. The variation in donor profiles is relatively large compared to the other objectives as is the range of diversity values, showing that these objectives are strongly affected by the other objectives included in the experiment. Again, it is likely that a compromise solution would represent the best solution in terms of all of the objectives.

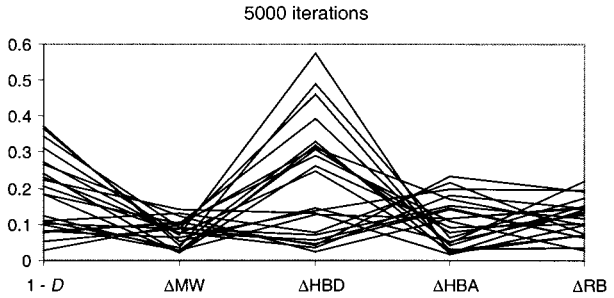

Figure 17. Pareto frontiers after 5000 iterations of MoSELECT configured to select 3 × 20 × 10 combinatorial subsets from the 12 × 99 × 59 thiazoline-2-imine library. MoSELECT is run with a population size of 200 and a niche radius of 30%.

When the same library design was run without niching, the solutions were concentrated in a localized region of the search space, as was observed with the amide library, confirming that it is necessary to use niching if a wide spread of solutions is required.

CONCLUSIONS

MoSELECT is a new approach to combinatorial library design based on optimizing multiple objectives simultaneously. MoSELECT overcomes many of the limitations of the weighted-sum approach used in SELECT and results in a family of solutions, all of which are equally valid in terms of overall fitness. The library designer can then make an informed choice on which solution(s) to explore rather than proceeding with the single solution generated by SELECT which may lie anywhere on the Pareto frontier. MoSELECT maps out the Pareto frontier, which allows the relationships between the different objectives to be explored with competing objectives easily identified. There are no significant overheads in terms of computing time for adopting Pareto ranking, and a single run of MoSELECT takes approximately the same time as a run of SELECT but with the advantage of finding a whole family of solutions. A number of different combinatorial libraries have been used to study various performance characteristics of MoSELECT.

The work reported here represents our initial attempts to explore the use of Pareto ranking for multiobjective optimization problems in chemistry. Our initial results suggest that the approach has considerable potential in our chosen domain of combinatorial library design. More recent work explores the design of focused libraries, where the approach has been shown to be equally effective.[34] Future work will investigate the possibility of interacting with the search process so that the relationships between objectives are explored during the search. This will allow the user to observe which objectives are relatively hard to improve, which are more easily optimized, and which objectives are in competition. The search process itself could then be altered to take account of these characteristics.

ACKNOWLEDGMENT

We thank GlaxoSmithKline for funding and Daylight Chemical Information Systems Inc. for software support. The Krebs Institute for Biomolecular Research is a designated Biomolecular Sciences Centre of the Biotechnology and Biological Sciences Research Council.

REFERENCES AND NOTES

(1) Bohm, H.-J.; Schneider, G.; Eds. *Virtual Screening for Bioactive Molecules*; Wiley-VCH: Weinheim, 2000.
(2) Dean, P. M.; Lewis, R. A.; Eds. *Molecular Diversity in Drug Design*; Kluwer: Dordrecht, 1999.
(3) Willett, P.; Ed. Computational Methods for the Analysis of Molecular Diversity. *Perspect. Drug Discovery Des*. **1997**, *7/8*.
(4) Martin, E. J.; Crichlow, R. W. Beyond mere diversity: tailoring combinatorial libraries for drug discovery. *J. Comb. Chem*. **1999**, *1*, 32−45.
(5) Valler, M. J.; Green, D. Diversity screening versus focussed screening in drug discovery. *Drug Discovery Today* **2000**, *5*, 286−293.
(6) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3−25.
(7) Gillet, V. J.; Willett, P.; Bradshaw, J. Selecting combinatorial libraries to optimize diversity and physical properties. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 167−177.
(8) Zheng, W.; Hung, S. T.; Saunders: J. T.; Seibel, G. L. PICCOLO: A tool for combinatorial library design via multicriterion optimization. In *Pacific Symposium on Biocomputing 2000*; Atlman, R. B., Dunkar, A. K., Hunter, L., Lauderdale K., Klein, T. E., Eds.; World Scientific: Singapore, 2000; pp 588−599.
(9) Brown, J. D.; Hassan, M.; Waldman, M. Combinatorial library design for diversity, cost efficiency, and drug-like character. *J. Mol. Graph. Model*. **2000**, *18*, 427−437.
(10) Rassokhin, D. N.; Agrafiotis, D. K. Kolmogorov-Smirnov statistic and its application in library design. *J. Mol. Graph. Model*. **2000**, *18*, 368−382.
(11) Martin, E. J.; Blaney, J. M.; Siani, M. S.; Spellmeyer, D. C.; Wong, A. K.; Moos, W. H. Measuring diversity−experimental design of combinatorial libraries for drug discovery. *J. Med. Chem*. **1995**, *38*, 1431−1436.
(12) Gillet, V. J.; Willett, P.; Bradshaw, J. The effectiveness of reactant pools for generating structurally diverse combinatorial libraries. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 731−740.
(13) Gillet, V. J.; Nicolotti, O. Evaluation of reactant-based and product-based approaches to the design of combinatorial libraries. *Perspect. Drug Discov. Des.* **2000**, *20*, 265−287.
(14) Jamois, E. A.; Hassan, M.; Waldman, M. Evaluation of reagent-based and product-based strategies in the design of combinatorial library subsets. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 63−70.
(15) Sheridan, R. P.; Kearsley, S. K. Using a genetic algorithm to suggest combinatorial libraries. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 310−320.
(16) Sheridan, R. P.; SanFeliciano, S. G.; Kearsley, S. K. Designing targeted libraries with genetic algorithms. *J. Mol. Graph. Model*. **2000**, *18*, 320−334.
(17) Brown, R. D.; Martin, Y. C. Designing combinatorial library mixtures using a genetic algorithm. *J. Med. Chem*. **1997**, *40*, 2304−2313.
(18) Agrafiotis, D. K. Stochastic algorithms for molecular diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841−851.
(19) Good, A. C.; Lewis, R. A. New methodology for profiling combinatorial libraries and screening sets: cleaning up the design process with HARPick. *J. Med. Chem.* **1997**, *40*, 3926−3936.
(20) The World Drug Index is available from Derwent Information, 14 Great Queen St., London WC2B 5DF, UK.
(21) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol*. **1997**, *267*, 727−748.
(22) Bravi, G.; Green, D. V. S,; Hann, M. A.; Leach, A. R. PLUMS: A program for the rapid optimization of focused libraries. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1441−1448.
(23) Fonseca, C. M.; Fleming, P. J. An overview of evolutionary algorithms in multiobjective optimization. In *Evolutionary Computation*; De Jong, K., Ed.; Massachusetts Institute of Technology: Cambridge, 1995; Vol. 3, No. 1, pp 1−16.
(24) Fonseca, C. M.; Fleming, P. J. Genetic algorithms for multiobjective optimization: formulation, discussion and generalisation. In *Genetic Algorithms: Proceedings of the Fifth International Conference*; Forrest, S., Ed.; Morgan Kaufmann: San Mateo, CA, 1993; pp 416−423.
(25) Coello Coello, C. A. An undated survey of GA-based multiobjective optimization techniques. *ACM Comput. Surveys* **2000**, *32*, 109−143.
(26) Handschuh, S.; Wagener, M.; Gasteiger, J. Superposition of three-dimensional chemical structures allowing for conformational flexibility by a hybrid method. *J. Chem. Inf. Comput. Sci*. **1998**, *38*, 220−232.
(27) Gillet, V. J.; Khatib, W.; Willett, P.; Fleming, P. J.; Green, D. V. S. Multiobjective approach to combinatorial library design. *Abstracts of Papers, Part 1*, National Meeting of the American Chemical Society, Apr 1, 2001; American Chemical Society: Washington, DC; S 221: 75-COMP.
(28) International Patent Application No. PCT/GB01/05347.
(29) The SPRESI database is distributed by Daylight Chemical Information Systems, Inc., Mission Viejo, CA.
(30) The Available Chemicals Directory is available from MDL Information Systems, Inc., 146000 Catalina Street, San Leandro, CA 94577.
(31) Leach, A. R.; Bradshaw, J.; Green, D. V. S.; Hann, M.; Delany, J. J. III. Implementation of a system for reagent selection, library enumeration, profiling and design. *J. Chem. Inf. Comput. Sci*. **1999**, *39*, 1161−1172.
(32) Mason, J. S.; Pickett, S. D. Partition-based selection. *Perspect. Drug DiscoveryDes*. **1997**, *7/8*, 85−114.
(33) Cerius^2 is available from Accelrys Inc., 9685 Scranton Road, San Diego, CA 92121.
(34) Gillet, V. J.; Willett, P.; Fleming, P.; Green, D. V. S. Designing focused libraries using MoSELECT. *J. Mol. Graph. Model*. In press.

CI010375J