Evolutionary Programming Applied to the Development of Quantitative Structure−Activity Relationships and Quantitative Structure−Property Relationships

Brian T. Luke

International Business Machines Corporation, MLM/078, Neighborhood Road, Kingston, New York 12401

Received June 7, 1994[⊗]

In developing a quantitative structure−activity relationship (QSAR), or a quantitative structure−property relationship (QSPR), one attempts to create a tool that predicts a biological activity, or chemical property, from a small set of experimentally and/or theoretically determined (structure-based) descriptors. If one were developing QSAR/QSPR predictors for different properties of a group of molecules, such as the melting and boiling points for a group of compounds, it would be advantageous to determine a series of accurate predictors. This would allow the researcher to see if a single set of descriptors can be used to predict good (thought maybe not the best) values for all properties. Unfortunately, regression analysis or the use of neural networks only yields a single predictor for a given set of descriptors and activities/properties. Rogers and Hopfinger recently showed how the genetic function approximation (GFA) can be used to develop multiple predictors. This paper uses the same data sets as the GFA paper to show how the method of evolutionary programming (EP) can also generate multiple predictors. EP, as it is applied here, is able to very quickly generate a series of different predictors and, in direct comparisons, finds good QSARs that were missed by the GFA.

INTRODUCTION

An *a priori* prediction of a molecule's biological activity, or a given property, from a set of experimentally or theoretically determined (structure-based) descriptors is very useful in screening possible compounds for further study. The standard procedure is to take a relatively small set of molecules with a known activity/property and determine a large number of descriptors. These can be simple structural descriptors such as the volume, surface area, or moments of inertia; physical descriptors such as the boiling point, melting point, or log(P); or electronic descriptors such as partial atomic charges or higher multipoles and their derivatives. The predictions are generated from a quantitative structure−activity relationship (QSAR),[1] or a quantitative structure−property relationship (QSPR), which use these descriptors.

It is very easy to generate a large number of descriptors, as demonstrated by Seiwood *et al.*[2] In their study, 53 descriptors were generated for each of a set of 31 molecules; yielding an overdetermined set of data (53 descriptors for only 31 activities). Since, in principle, each descriptor can be used in a polynomial series, it is easy to see why there is no real problem in fitting the data as closely as desired.

The real goal in developing a QSAR, or QSPR, is to predict the activity, or property, of a molecule not included in the original data set. Mathematically, this is equivalent to an interpolation of the initial data, since the value of each descriptor used in QSAR/QSPR for the new molecule should lie within the range of values spanned by the initial set of data.

Unfortunately, as the number of terms increases, the predictive (interpolative) ability of the QSAR/QSPR may decrease. This is often the case in a simple polynomial fit to data. As the number of polynomial terms increases the error in the fit decreases, but the polynomial may begin to oscillate wildly between data points.

What is wanted is a predictor that contains only a few terms. This will result in a "smoother" predictor and yield better interpolative results. Extending this, it would be nice if multiple good predictors can be generated from such an overdetermined set of data. Each of these relationships can be used to predict the activity, or property, of a new compound. If they yield quite different values for the activity, or property, the predictive ability of these relationships should be questioned. The activity/property of this new compound could then be measured and included in the initial set used to generate new predictors. This would be superior to using only a single relationship to predict the activity/property, since the latter would yield no *a priori* indication of a possible problem.

In addition, the situation may arise where one would like to predict more than one activity/property for a set of compounds using the same set of descriptors. By generating multiple predictors for each activity/property, one can attempt to locate a set of descriptors that yield good predictive relationships for all activities/properties, though they may not yield the "best" prediction for some or all.

Standard regression analysis, or the use of a neural network, only produces a single predictive model for the data for a given number of descriptors. Rogers and Hopfinger[3] recently presented an extension of the genetic algorithm,[4,5] called genetic function approximation (GFA), and showed how this method can be used to generate multiple predictors. This study shows how the evolutionary programming (EP) method[6−8] can also be used to generate multiple predictors.

The EP method is easy to understand and program. The next section outlines the basic structure of the EP method as it applies to QSAR/QSPR development. This is followed by an analysis of the same data sets used in the GFA paper.[3]

EVOLUTIONARY PROGRAMMING METHOD

The description of how the EP method can be applied to the QSAR/QSPR problem will use the Selwood *et al.*[2] data

[⊗] Abstract published in *Advance ACS Abstracts,* September 15, 1994.

0095-2338/94/1634-1279$04.50/0

mentioned above. This data contains the activity, log(IC_{50}), and 53 properties of 31 analogues of antifilarial antimycin A_1. The form of a particular QSAR used here is as follows.

$$\text{PREDICTOR} = C_0 + \sum_{i=1}^{53} C_i \times (\text{DESCR}_i)^{n_i} \quad (1)$$

The restrictions imposed by this form of the QSAR are that each descriptor can only be used once (though this will be circumvented in a later test) and has an integer exponent, n_i. Each possible predictor over the Selwood data is composed of a string of 53 integers defining each n_i. If a particular n_i, is zero, that descriptor is not used in the expression and the corresponding C_i is not determined, since it is folded into C_0. If a given n_i is nonzero, the corresponding descriptor is used (with that exponent), and the C_i is determined. The number of nonzero integers is denoted nterm and the (neterm + 1) coefficients in eq 1 are determined by a least-squares fit to the 31 activities.

The general form of the EP method[6-8] is outlined in the following five steps.

1. Start with a random population of N predictors. In other words, start with N sets of 53 integers, n_i.
2. Determine the fitness, or weakness, of each predictor.
3. Allow each predictor to create a new predictor. Determine the fitness of this new predictor and add it to the population.
4. Order the $2N$ predictors from best to worst fitness and remove the n least fit predictors from the population. This restores the population size to N and completes a cycle.
5. If the number of cycles is less than a user-defined number, return to step 3.

The initial population of predictors for all tests presented in this study is generated as follows. Each initial animal can have nterm ranging from 1 to 4. If a random number between 0.0 and 1.0 is less than 0.1, nterm is set to 1. If this random number is less than 0.5, 0.8, and 1.0, nterm is set to 2, 3, and 4, respectively. The program randomly chooses nterm descriptors (positions along the n_i vector) and randomly sets the exponent between the input values MIN and MAX. All other positions along the n_i vector are set to zero.

Given a set of nterm nonzero n_i's, the C_i's in eq 1 are determined using least-squares regression. Since the predictor is trying to minimize the error, or COST, the fitness is defined as the inverse of the COST. The COST is defined by the following expression.

$$\text{COST} = \text{RMS\_ERROR} \times \text{XVAL}^{|\text{ITERM} - \text{nterm}|} \times \prod_{i=1}^{\text{nterm}} \text{WEIGHT}(n_i) \quad (2)$$

The first term in the COST expression is simply the root-mean-square (RMS) error between the values predicted by eq 1 and the measured values. The second term is present to push the best solution toward a given number of terms, ITERM. For example, if XVAL is set at 2.0 and ITERM to 2, a three-term (or one-term) solution will have to yield one-half the RMS_ERROR before it will have the same cost. The last term controls the range of exponents that a given predictor can have. WEIGHT(1) is always set to 1.0 in these tests, which means that a given descriptor can be present in the QSAR expression with an exponent of 1 without increasing the COST. If WEIGHT(2), or WEIGHT(−1), is set to 2.0, the RMS_ERROR will have to be reduced by a factor of 2 before a descriptor can have an exponent of 2, or −1, and not increase the COST. In these tests, WEIGHT(1) is 1.0, WEIGHT(2) is either 1.0 or 2.0, and all other WEIGHTs are 100.0.

The next step is to discuss the way a new predictor is created from an existing one. This is controlled by two user-defined variables, CUT_1 and CUT_2. If a random number between 0.0 and 1.0 is less than, or equal to, CUT_1 some nonzero n_i is increased or decreased by 1. This may reduce the number of descriptors used in eq 1 and therefore nterm in eq 2. If the random number is greater than CUT_1 and less than, or equal to, CUT_2, a zero n_j is increased or decreased by 1. This will always increase the value of nterm by 1. Finally, if the random number is greater than CUT_2, a nonzero n_i and a zero n_j are switched. This will not affect the value of nterm, but will use a different descriptor in eq 1. The COST of this new predictor will only differ from the COST of its "parent" by the RMS_ERROR term in eq 2.

All results presented here are generated by setting CUT_1 to 0.2 and CUT_2 to 0.5. This means that approximately 50% of the new offspring are generated by simply switching the position of a nonzero exponent, which samples different combinations of the same number of descriptors. In addition, a slightly greater emphasis is placed on adding a descriptor (increasing nterm) than removing one (decreasing nterm).

To ensure that the process does not converge to multiple copies of the same (fit) predictor, each new predictor is compared against all other predictors currently present in the population (old and new). If it is different, its COST is calculated. If not, it is discarded and a new predictor is generated from that parent. (Note that this does not require every animal present in the final population to be unique, since the check is not made of the initial, random population. If two initial animals are the same and their COST is small enough, they may be able to survive until the end.)

From the above discussion, it should be clear that evolutionary programming is modeled after a population of organisms that produce offspring asexually. In this analogy, the vector of n_i exponents represents the genes of each organism. In particular, N organisms constitute the initial population. Each organism generates a unique offspring by transferring a copy of its genes and then allowing for a change through mutation. This new organism is added to the population and the N most-fit organisms survive to the next generation. This process is repeated for a given number of generations.

Though evolutionary programming[6-8] is similar to a simple genetic algorithm,[4,5] they are really quite different strategies for solving this type of problem. In a genetic algorithm, two parents are used to create an offspring by taking genes from one or the other parent. This is called a mating operation. In all cases, at least one of the parents is chosen based on its fitness.

A one-point crossover is often used as the mating operator. Here, n_1−n_k would be taken from one predictor, and n_{k+1}−

n_{53} would be taken from the second parent to generate a new offspring. This can be generalized to a two-point crossover and on to a uniform crossover operator,[12] where each n_i is randomly taken from one parent or the other. A mutation can still be applied to this offspring, but most applications use mating as the primary means of creating an offspring. In contrast, Nettleton and co-workers[13] recently presented good results by employing a genetic algorithm that had a high mutation-to-crossover ratio and is therefore intermediate between genetic algorithms and evolutionary programming.

In this study, evolutionary programming is chosen over a genetic algorithm for the simple reason that the latter may have problems when the number of descriptors increases in a QSAR/QSPR investigation. Though the Selwood data has 53 descriptors, it is not hard to imagine a set of data containing 1000 descriptors. If a researcher wants to find good predictors that only use three of the descriptors (nterm = 3), the n_i vector contains 997 zeroes and three nonzero values. If both parents have nterm = 3, it is very likely that nterm will not be 3 in an offspring. (If a uniform crossover is used, it is most likely that the offspring will have nterm = 0). Similarly, in a standard genetic algorithm, the offspring (or best of multiple offspring)[14] replaces the least-fit predictor in the set (or lesser-fit parent), whether or not the offsprings fitness is better than the solution it is producing. Therefore, there is no guarantee that the overall fitness of the population will increase from generation to generation in the case where a large number of descriptors are present. The evolutionary programming method described here does not allow nterm to change by more than one when generating an offspring and the overall fitness of the population cannot increase from one generation to the next.

RESULTS

Data Set 1. The first data set examined by Rogers and Hopfinger in their GFA paper[3] comes from the work of Selwood *et al.*[2] Since the number of descriptors (53) is large, a population size of 40 is used throughout. In addition, the run is allowed to continue for 300 generations. The initial population is generated using MIN = 1 and MAX = 2, which means that the selected descriptors will randomly be used either linearly or quadratically. At most, the best five QSAR equations will be presented for each run.

In the first run, a two-descriptor equation is wanted with all exponents at 1. Therefore, ITERM is set to 2 and WEIGHT(2) to 2.0 in eq 2. The results of this run are presented at the top of Table 1. The best two-descriptor equation involves the partition coefficient, **LOGP**, and the moment of inertia about the z-axis, **MOFI_Z**. The second best two-descriptor equation uses **LOGP** and the y-component of the moment of inertia, **MOFI_Y**. This was the best two-descriptor equation found by Rogers and Hopfinger[3] (the fifth best overall).

The GFA work only used the descriptors linearly, but allowing for quadratic descriptors can be done by setting WEIGHT(2) to 1.0. The five best two-descriptor linear/quadratic equations are shown in Table 1 under "ITERM = 2, WEIGHT(2) = 1.0". All five if these equations yield a smaller RMS_ERROR than any of the two-descriptor linear equations. It is interesting to note that the best four predictors using linear/quadratic descriptors are the same as the best four using linear descriptors, in a different order, with the second descriptor squared.

The five best linear, three-descriptor predictors are found by setting ITERM = 3 and WEIGHT(2) = 2.0 in eq 2. The two best descriptors listed in Table 1 are identical to the two best results found using the GFA. The third and fifth best predictors found here were not found in the GFA study.[3] Their fourth best linear, three-descriptor (eighth best overall) is the seventh best found here. The five listed in Table 1 and a predictor using **LOGP**, **ESDL3**, and **VDWVOL** are all found to be better.

Of the 40 unique solutions found at the end of this EP run, 33 are three-descriptor, linear equations. The remaining 7 are four-term, linear equations that have a small enough RMS_ERROR to surmount the factor of 1.2 penalty (XVAL). All 33 of these three-descriptor equations yield smaller RMS_ERRORs than the QSAR model proposed by Selwood *et al.*[2] or Wikel and Dow.[9]

The best three-descriptor, linear/quadratic predictors can be found by setting WEIGHT(2) = 1.0. The results in Table 1 again show that all five predictors yield a smaller RMS_ERROR than the best linear, three-descriptor equation. The four best can be generated from the best and third best linear, three-descriptor equations by squaring one or two of the terms.

The five best linear, four-descriptor predictors are found using ITERM = 4 and WEIGHT(2) = 2.0. The results are shown in Table 1 and are quite different from the GFA results.[3] Both studies agree in the best four-descriptor, linear equation, but the GFA paper did not find the second best predictor. Their second four-descriptor equation is the third best found here, and they did not find the fourth or fifth best predictor. In fact, their third best four-descriptor, linear equation is the ninth best predictor found with the EP method.

Finally, the five best four-descriptor, linear/quadratic predictors are shown in Table 1. Again, all five equations yield a smaller RMS_ERROR than the best four-descriptor, linear equation.

Two points need to be raised about this analysis of the Selwood *et al.* data.[2] The first deals with the resulting equations shown in Table 1. It is clear that **LOGP** is the most important single descriptor since it is present in 39 of the 40 equations. The second deals with the efficiency of the EP method. The longest runs are for the ITERM = 4 jobs; WEIGHT(2) = 2.0 running slightly faster than WEIGHT(2) = 1.0. Even so, these jobs finished in approximately 17.9 and 18.7 CPU-s, respectively, on a single node of an SP1 Scalable Parallel machine (which would be the same as a desktop RS/6000, Model 370).

Data Set 2. The second set of data analyzed in the GFA paper comes from the work of Cardozo *et al.*[10] where a set of 17 acetylcholinesterase inhibitors were examined. They used molecular decomposition–recomposition (MDR) to reduce the data to three descriptors: the energy of highest occupied molecular orbital (HOMO), **HOMO**; the HOMO (π-orbital) coefficient on ring atom 4, $\mathbf{C_4}$; and the total dipole moment, $\mathbf{U_t(D)}$. They generated a QSAR using $\mathbf{C_4}$, $\mathbf{U_t(D)^2}$, **HOMO**, and $\mathbf{HOMO^2}$.[10] This predictor has an RMS_ERROR of 0.3858 when all 17 data points are used.

Their QSAR[10] cannot be generated by the EP program as outlined above since each descriptor can be used only once, though this problem can be quickly remedied. Before the EP program is run, the data file is expanded. In this case,

Table 1. QSARs Determined by the EP Method Using the Data of Selwood *et al.*[2]

ITERM = 2, WEIGHT(2) = 2.0

$-\log(IC_{50}) = 0.677\,87*\mathbf{LOGP} - 8.2681 \times 10^{-5}*\mathbf{MOFI\_Z} - 2.2157$
RMS-Error = 0.5080
$-\log(IC_{50}) = 0.693\,53*\mathbf{LOGP} - 8.4123 \times 10^{-5}*\mathbf{MOFI\_Y} - 2.1483$
RMS-Error = 0.5130
$-\log(IC_{50}) = 0.607\,63*\mathbf{LOGP} - 0.206\,24*\mathbf{PEAX\_X} - 0.226\,16$
RMS-Error = 0.5156
$-\log(IC_{50}) = 0.666\,57*\mathbf{LOGP} - 0.013\,827*\mathbf{SURF\_A} + 1.6439$
RMS-Error = 0.5165
$-\log(IC_{50}) = 4.0749*\mathbf{ATCH4} - 37.310*\mathbf{ATCH6} - 9.2888$
RMS-Error = 0.5456

ITERM = 2, WEIGHT(2) = 1.0

$-\log(IC_{50}) = 0.682\,68*\mathbf{LOGP} - 8.2681 \times 10^{-5}*\mathbf{SURF\_A}^2 - 1.0937$
RMS-Error = 0.4874
$-\log(IC_{50}) = 0.606\,91*\mathbf{LOGP} - 6.0052 \times 10^{-3}*\mathbf{PEAX\_X}^2 - 1.9158$
RMS-Error = 0.4975
$-\log(IC_{50}) = 0.612\,88*\mathbf{LOGP} - 1.4284 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 2.7780$
RMS-Error = 0.4990
$-\log(IC_{50}) = 0.62372*\mathbf{LOGP} - 1.3694 \times 10^{-9}*\mathbf{MOFI\_Y}^2 - 2.7802$
RMS-Error = 0.5023
$-\log(IC_{50}) = 0.51626*\mathbf{LOGP}^2 - 1.6352 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 0.97856$
RMS-Error = 0.5061

ITERM = 3, WEIGHT(2) = 2.0

$-\log(IC_{50}) = 0.583\,60*\mathbf{LOGP} + 1.5136*\mathbf{SUM\_F} - 7.4882 \times 10^{-5}*\mathbf{MOFI\_Y} - 2.5008$
RMS-Error = 0.4294
$-\log(IC_{50}) = 0.568\,46*\mathbf{LOGP} + 0.810\,01*\mathbf{ESDL3} - 0.013\,020*\mathbf{SURF\_A} + 2.8716$
RMS-Error = 0.4309
$-\log(IC_{50}) = 0.568\,44*\mathbf{LOGP} + 1.4521*\mathbf{SUM\_F} - 7.2818 \times 10^{-5}*\mathbf{MOFI\_Z} - 2.5345$
RMS-Error = 0.4315
$-\log(IC_{50}) = 0.589\,37*\mathbf{LOGP} + 0.736\,40*\mathbf{ESDL3} - 7.6949 \times 10^{-5}*\mathbf{MOFI\_Y} - 0.804\,56$
RMS-Error = 0.4442
$-\log(IC_{50}) = 0.574\,43*\mathbf{LOGP} + 0.693\,85*\mathbf{ESDL3} - 7.4692 \times 10^{-5}*\mathbf{MOFI\_Z} - 0.929\,88$
RMS-Error = 0.4475

ITERM = 3, WEIGHT(2) = 1.0

$-\log(IC_{50}) = 0.525\,68*\mathbf{LOGP} + 1.5876*\mathbf{SUM\_F} - 1.2478 \times 10^{-9}*\mathbf{MOFI\_Y}^2 - 3.1154$
RMS-Error = 0.4054
$-\log(IC_{50}) = 0.515\,13*\mathbf{LOGP} + 1.5494*\mathbf{SUM\_F} - 1.2919 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 3.0925$
RMS-Error = 0.4070
$-\log(IC_{50}) = 0.519\,58*\mathbf{LOGP} + 1.3485*\mathbf{SUM\_F}^2 - 1.3452 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 2.7012$
RMS-Error = 0.4191
$-\log(IC_{50}) = 0.589\,37*\mathbf{LOGP} + 1.3690*\mathbf{SUM\_F}^2 - 1.2945 \times 10^{-9}*\mathbf{MOFI\_Y}^2 - 2.7085$
RMS-Error = 0.4201
$-\log(IC_{50}) = 0.574\,60*\mathbf{LOGP} + 1.3255*\mathbf{SUM\_F} - 1.5514 \times 10^{-5}*\mathbf{SURF\_A}^2 - 1.5112$
RMS-Error = 0.4232

ITERM = 4, WEIGHT(2) = 2.0

$-\log(IC_{50}) = 0.499\,84*\mathbf{LOGP} + 2.8075*\mathbf{ATCH4} + 0.84222*\mathbf{ESDL3} - 0.199\,60*\mathbf{PEAX\_X} + 1.790\,78$
RMS-Error = 0.3861
$-\log(IC_{50}) = 0.483\,51*\mathbf{LOGP} + 3.5934*\mathbf{ATCH4} + 9.7021*\mathbf{ATCH5} - 5.5592 \times 10^{-5}*\mathbf{MOFI\_Z} - 1.7516$
RMS-Error = 0.3882
$-\log(IC_{50}) = 0.485\,96*\mathbf{LOGP} + 3.4443*\mathbf{ATCH4} + 10.124*\mathbf{ATCH5} - 5.5073 \times 10^{-5}*\mathbf{MOFI\_Y} - 1.7490$
RMS-Error = 0.3912
$-\log(IC_{50}) = 0.441\,91*\mathbf{LOGP} + 3.8178*\mathbf{ATCH4} + 9.2566*\mathbf{ATCH5} - 0.14201*\mathbf{PEAX\_X} - 0.321\,43$
RMS-Error = 0.4442
$-\log(IC_{50}) = 0.553\,24*\mathbf{LOGP} + 2.3341*\mathbf{ATCH4} + 0.834\,17*\mathbf{ESDL3} - 7.6230 \times 10^{-5}*\mathbf{MOFI\_Z} - 0.208\,48$
RMS-Error = 0.3974

ITERM = 4, WEIGHT(2) = 1.0

$-\log(IC_{50}) = 0.503\,68*\mathbf{LOGP} + 4.3013*\mathbf{ATCH4} + 77.681*\mathbf{ATCH5}^2 - 1.1212 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 2.2123$
RMS-Error = 0.3532
$-\log(IC_{50}) = 0.596\,81*\mathbf{LOGP} + 2.0996*\mathbf{SUM\_F} + 15.132*\mathbf{SUM\_R}^2 - 1.3747 \times 10^{-9}*\mathbf{MOFI\_Y}^2 - 4.3441$
RMS-Error = 0.3542
$-\log(IC_{50}) = 0.459\,63*\mathbf{LOGP} + 3.8634*\mathbf{ATCH4} + 9.6863*\mathbf{ATCH5} - 1.0513 \times 10^{-9}*\mathbf{MOFI\_Z}^2 - 2.1636$
RMS-Error = 0.3557
$-\log(IC_{50}) = 0.682\,75*\mathbf{LOGP} + 1.8710*\mathbf{SUM\_F} + 17.828*\mathbf{SUM\_R}^2 - 1.8037 \times 10^{-5}*\mathbf{SURF\_A}^2 - 4.3441$
RMS-Error = 0.3567
$-\log(IC_{50}) = 0.507\,23*\mathbf{LOGP} + 4.1792*\mathbf{ATCH4} + 80.309*\mathbf{ATCH5}^2 - 1.0533 \times 10^{-9}*\mathbf{MOFI\_Y}^2 - 2.2393$
RMS-Error = 0.3573

the three descriptors are used to generate a new table with 15 descriptors. These 15 descriptors represent each of the original descriptors, each of them squared, each of them cubed, the inverse of each descriptor, and the three pairwise products of the descriptors.

This new table of $-\log(IC_{50})$ values and 15 descriptors for the 17 compounds is now used in the EP program. To make sure that the matrix used to calculate the coefficients in eq 1 is not singular, each descriptor must either not be used or be used with an exponent of one. The generation

of an offspring is also adjusted to make sure that each n_i can only have the value of 0 or 1. If a random number between 0.0 and 1.0 is less than, or equal to, CUT_1 a nonzero n_i is changed to 0. This will always reduce nterm in eqs 1 and 2. If the random number is greater than CUT_1 and less than, or equal to, CUT_2, a zero n_j is changed to 1. This will always increase the value of nterm by 1. Finally, if the random number is greater than CUT_2, a nonzero n_i and a zero n_j are switched, leaving nterm unchanged. The initial population is generated with MIN = MAX = 1.

Instead of using each descriptor to a different power to create new columns in the data table, any function of one or more descriptors could be used. This includes truncated power splines used by Rogers and Hopfinger[3] in the GFA paper. Though this leads to an excellent fit, it was not done here. The reason is that the knots used in these splines are specific to a given set of data, only affect a subset of a column of data, and selectively improve the fit for a small number of outliers. The descriptors generated here have nonzero values for all 17 compounds. Since no truncated power splines are used here, a direct comparison to the GFA results cannot be made.

The data will be used two ways with the EP method. In the first case, a filter is used so that only the descriptors and the descriptors squared are used. The second set uses all 15 descriptors. The first set is used with ITERM set to 3 or 4 and is run for 100 generations with a population size of 20. Since the second set has many more descriptors, the population size and number of generations are increased to 30 and 200, respectively.

The five best QSARs for a given number of descriptors with each data set is presented in Table 2. For the three-descriptor QSAR using only linear and quadratic terms, it is interesting to note that each descriptor is used only once. When all 15 descriptors are used to generate a three-descriptor QSAR, only $\mathbf{C}_4$, the inverses of $\mathbf{U}_t(\mathbf{D})$ and **HOMO**, and the three products of descriptors are used.

Four of the five four-descriptor predictors that use linear or quadratic descriptors are better than the three-descriptors QSARs. When all 15 descriptors are used to generate QSARs, the RMS–ERRORs drop substantially. All five of the predictors listed in Table 2 have a much smaller RMS–ERROR than the five-descriptor QSAR generated by Cardozo *et al.*[10] (0.3858).

Using sets of five of the 15 descriptors generates QSARs that had lower RMS–ERRORs. Increasing the value of ITERM to 6 did not cause as large a drop in the RMS–ERROR.

Again, a key advantage of the EP method is its speed. The ITERM = 6 job with a population size of 30 that ran for 200 generations only took 8.8 CPU-s to finish on a single node of an SP1.

Data Sets 3 and 4. The last two data sets examined by the EP method are grouped together since they are obtained from the QSPR study of Koehler and Hopfinger.[11] In particular, these authors used seven descriptors to predict the glass, T_g, and melting, T_m, transition temperatures for sets of 35 and 30 polymers, respectively. These data sets were re-examined by Rogers and Hopfinger[3] using GFA with truncated power splines.

This data set will be treated like the first set in that each descriptor will only be used once, either linearly or quadratically. Extra descriptors could have been added, as was done with the second set, but this would have made comparisons to the results of Koehler and Hopfinger more difficult.

All used a population size of 20 and ran for 100 generations. The initial population is generated with MIN = 1 and MAX = 2.

The resulting of the T_g predictors are presented in Table 3 for ITERM ranging from 2 to 5 and WEIGHT(2) either 2.0 (looking for linear fits) or 1.0 (allowing for quadratic fits). The T_m QSPRs generated by the EP method are listed in Table 4. When an attempt is made to find two-descriptor, linear predictors, for T_g, only three of the final 20 solutions contain two descriptors. Though all 20 are linear predictors, the other 17 contained between three and five descriptors. Their RMS–ERRORs are small enough to overcome the XVAL penalty, given as the second term in eq 2, and produce a small enough overall COST. When WEIGHT(2) is reduced to 1.0, the situation becomes even worse. Only one of the final 20 QSPRs contains two descriptors; the best is the two-descriptor, linear predictor. The remaining 19 results contain more than two descriptors. A two-descriptor solution could have been pushed harder if XVAL is made larger than 1.2, but these results are interesting in that they **strongly** suggest that more than two descriptors should be used.

Table 4 shows that a two-descriptor QSPR is slightly better for T_m. When linear and quadratic terms are allowed, four of the 20 final predictors contain two descriptors; the other 16 contain more.

Of particular note are the ITERM = 5, WEIGHT(2) = 2.0 results for T_g and T_m, since five-descriptor, linear QSPRs were developed by Koehler and Hopfinger.[11] They used the same five descriptors for each QSPR; $\mathbf{S}_B$, $\mathbf{M}_B$, $\mathbf{S}_S$, $\mathbf{E}_+$, and $\mathbf{E}_-$. The results shown in Tables 3 and 4 suggest that this was a reasonably good choice. This set of descriptors yields the third best QSPR for T_g and the second best for T_m. The best T_g QSPR is only the fifth best for T_m, the second best T_g QSPR is only the fourth best for T_m, and the fifth best T_g QSPR is only the third best for T_m. Note that the fourth best QSPR for T_g and the best QSPR for T_m did not have the same set of descriptors in the top five predictors of the other.

The results in these tables also show that a four-descriptor, linear QSPR using $\mathbf{S}_B$, $\mathbf{S}_S$, $\mathbf{E}_+$, and $\mathbf{E}_-$ yields results that are only a little worse than the five-descriptor equations, and represent the second best QSPR for both T_g and T_m.

As expected from the small number of descriptors, the EP jobs finished very rapidly. The ITERM = 5, WEIGHT(2) = 1.0 results are generated in 4.0 and 3.6 CPU-s for T_g and T_m, respectively, on a single node of an SP1 computer.

CONCLUSIONS

The evolutionary programming method is shown to be a very effective computational method for generating a series of accurate QSAR/QSPRs. In the one data set where these results can be directly compared to the predictors generated by the genetic function approximation, the EP method found several good QSARs that were missed by the GFA. The results presented here also show the EP method to be an extremely fast computational procedure for generating predictors.

Though the EP method found good QSARs that were missed by the GFA with the Selwood data set, these results

Table 2. QSARs Determined by the EP Method Using the Data of Cardozo *et al.*[10]

ITERM = 3, ONLY LINEAR AND QUADRATIC TERMS

$-\log(IC_{50}) = 9.6813 + 2.6120*\mathbf{C_4} + 0.763\,49*\mathbf{U_t(D)} - 0.056\,904*\mathbf{HOMO^2}$
RMS-Error = 0.4333

$-\log(IC_{50}) = 14.758 + 2.6153*\mathbf{C_4} + 0.762\,36*\mathbf{U_t(D)} + 1.0748*\mathbf{HOMO}$
RMS-Error = 0.4339

$-\log(IC_{50}) = 9.6578 + 4.7295*\mathbf{C_4^2} + 0.833\,35*\mathbf{U_t(D)} - 0.056\,157*\mathbf{HOMO^2}$
RMS-Error = 0.4394

$-\log(IC_{50}) = 14.660 + 4.7356*\mathbf{C_4^2} + 0.832\,37*\mathbf{U_t(D)} - 1.0600*\mathbf{HOMO}$
RMS-Error = 0.4399

$-\log(IC_{50}) = 10.715 + 2.4860*\mathbf{C_4} + 0.121\,68*\mathbf{U_t(D)^2} - 0.054\,958*\mathbf{HOMO^2}$
RMS-Error = 0.4438

ITERM = 3, ALL TERMS

$-\log(IC_{50}) = 13.605 - 19.440*\mathbf{C_4} + 7.2971*\mathbf{C_4}*\mathbf{U_t(D)} + 0.230\,65*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.4010

$-\log(IC_{50}) = 11.679 + 5.7314*\mathbf{C_4}*\mathbf{U_t(D)} + 1.5442*\mathbf{C_4}*\mathbf{HOMO} + 0.164\,98*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.4068

$-\log(IC_{50}) = -3.2259 - 6.4518*\mathbf{U_t(D)^{-1}} - 11.600*\mathbf{HOMO^{-1}} - 0.295\,22*\mathbf{C_4}*\mathbf{HOMO}$
RMS-Error = 0.4083

$-\log(IC_{50}) = -3.0778 - 3.6196*\mathbf{U_t(D)^{-1}} - 10.580*\mathbf{HOMO^{-1}} + 0.912\,28*\mathbf{C_4}*\mathbf{U_t(D)}$
RMS-Error = 0.4086

$-\log(IC_{50}) = -2.5595 + 2.8075*\mathbf{C_4} - 6.4346*\mathbf{U_t(D)^{-1}} - 10.960*\mathbf{HOMO^{-1}}$
RMS-Error = 0.4090

ITERM = 4, ONLY LINEAR AND QUADRATIC TERMS

$-\log(IC_{50}) = -797.73 + 2.6615*\mathbf{C_4} + 1.0643*\mathbf{U_t(D)} - 168.76*\mathbf{HOMO} - 8.8835*\mathbf{HOMO^2}$
RMS-Error = 0.3876

$-\log(IC_{50}) = -833.29 + 4.8975*\mathbf{C_4^2} + 1.1558*\mathbf{U_t(D)} - 176.17*\mathbf{HOMO} - 9.2697*\mathbf{HOMO^2}$
RMS-Error = 0.3902

$-\log(IC_{50}) = -854.56 + 2.5338*\mathbf{C_4} + 0.179\,92*\mathbf{U_t(D)^2} - 180.91*\mathbf{HOMO} - 9.5147*\mathbf{HOMO^2}$
RMS-Error = 0.3952

$-\log(IC_{50}) = -897.40 + 4.6571*\mathbf{C_4^2} + 0.195\,64*\mathbf{U_t(D)^2} - 189.86*\mathbf{HOMO} - 9.9810*\mathbf{HOMO^2}$
RMS-Error = 0.3972

$-\log(IC_{50}) = 7.2541 + 2.8240*\mathbf{C_4} + 3.0500*\mathbf{U_t(D)} - 0.388\,04*\mathbf{U_t(D)^2} - 0.067\,193*\mathbf{HOMO^2}$
RMS-Error = 0.4191

ITERM = 4, ALL TERMS

$-\log(IC_{50}) = -40.531 + 50.113*\mathbf{U_t(D)} - 17.278*\mathbf{U_t(D)^2} + 1.9498*\mathbf{U_t(D)^3} + 0.694\,02*\mathbf{C_4}*\mathbf{U_t(D)}$
RMS-Error = 0.3469

$-\log(IC_{50}) = 52.990 - 5.5800*\mathbf{U_t(D)^2} + 0.944\,70*\mathbf{U_t(D)^3} - 64.036*\mathbf{U_t(D)^{-1}} + 0.694\,35*\mathbf{C_4}*\mathbf{U_t(D)}$
RMS-Error = 0.3482

$-\log(IC_{50}) = 96.801 - 23.690*\mathbf{U_t(D)} + 0.461\,06*\mathbf{U_t(D)^3} - 93.775*\mathbf{U_t(D)^{-1}} + 0.697\,43*\mathbf{C_4}*\mathbf{U_t(D)}$
RMS-Error = 0.3498

$-\log(IC_{50}) = -42.560 + 2.0841*\mathbf{C_4} + 51.555*\mathbf{U_t(D)} - 17.648*\mathbf{U_t(D)^2} + 1.9878*\mathbf{U_t(D)^3}$
RMS-Error = 0.3500

$-\log(IC_{50}) = -42.939 + 52.006*\mathbf{U_t(D)} - 17.818*\mathbf{U_t(D)^2} + 2.0084*\mathbf{U_t(D)^3} - 0.217\,17*\mathbf{C_4}*\mathbf{HOMO}$
RMS-Error = 0.3512

ITERM = 5, ALL TERMS

$-\log(IC_{50}) = 5.6108 + 3.7610*\mathbf{U_t(D)^2} - 93.688*\mathbf{U_t(D)^{-1}} - 10.488*\mathbf{HOMO} + 0.679\,05*\mathbf{C_4}*\mathbf{U_t(D)} + 3.5398*\mathbf{U_t}*\mathbf{HOMO}$
RMS-Error = 0.2843

$-\log(IC_{50}) = 6.3716 + 3.8160*\mathbf{U_t(D)^2} - 95.779*\mathbf{U_t(D)^{-1}} - 10.527*\mathbf{HOMO} - 0.216\,54*\mathbf{C_4}*\mathbf{HOMO} + 3.5728*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2844

$-\log(IC_{50}) = 5.8638 + 2.0613*\mathbf{C_4} + 3.8167*\mathbf{U_t(D)^2} - 95.844*\mathbf{U_t(D)^{-1}} - 10.586*\mathbf{HOMO} + 3.5743*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2850

$-\log(IC_{50}) = 55.779 + 3.7400*\mathbf{U_t(D)^2} - 93.575*\mathbf{U_t(D)^{-1}} + 0.545\,75*\mathbf{HOMO^2} + 0.681\,65*\mathbf{C_4}*\mathbf{U_t(D)} + 3.5273*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2875

$-\log(IC_{50}) = 56.713 + 3.7933*\mathbf{U_t(D)^2} - 95.630*\mathbf{U_t(D)^{-1}} + 0.547\,44*\mathbf{HOMO^2} - 0.217\,21*\mathbf{C_4}*\mathbf{HOMO} + 3.5587*\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2878

ITERM = 6, ALL TERMS

$-\log(IC_{50}) = -128.14 - 0.023\,579*\mathbf{C_4^{-1}} + 4.8219*\mathbf{U_t(D)^2} - 113.99*\mathbf{U_t(D)^{-1}} - 20.287*\mathbf{HOMO} - 614.32*\mathbf{HOMO^{-1}} + 4.4529\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2608

$-\log(IC_{50}) = -35.198 - 0.024\,027*\mathbf{C_4^{-1}} + 4.8794*\mathbf{U_t(D)^2} - 115.82*\mathbf{U_t(D)^{-1}} + 1.0903*\mathbf{HOMO^2} - 645.58*\mathbf{HOMO^{-1}} + 4.5165\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2612

$-\log(IC_{50}) = -3.7607 - 0.024\,408*\mathbf{C_4^{-1}} + 4.9178*\mathbf{U_t(D)^2} - 117.33*\mathbf{U_t(D)^{-1}} - 0.077\,383*\mathbf{HOMO^3} - 665.73*\mathbf{HOMO^{-1}} + 4.5643\mathbf{U_t(D)}*\mathbf{HOMO}$
RMS-Error = 0.2633

$-\log(IC_{50}) = -793.94 - 49.908*\mathbf{U_t(D)} + 5.8644*\mathbf{U_t(D)^2} - 130.05*\mathbf{U_t(D)^{-1}} - 197.73*\mathbf{HOMO} - 10.382*\mathbf{HOMO^2} - 0.18424*\mathbf{C_4}*\mathbf{HOMO}$
RMS-Error = 0.2644

$-\log(IC_{50}) = -479.85 - 49.577*\mathbf{U_t(D)} + 5.8272*\mathbf{U_t(D)^2} - 129.18*\mathbf{U_t(D)^{-1}} - 98.660*\mathbf{HOMO} - 0.362\,63*\mathbf{HOMO^3} - 0.184\,97*\mathbf{C_4}*\mathbf{HOMO}$
RMS-Error = 0.2646

should not be taken to imply that the EP method will always do so. Since GFA is based on the genetic algorithm, it has more user-defined (or defaulted) parameters that must be set than the EP method.

For example, the GFA described in ref 3 uses a one-point crossover mating operator. This will group contiguous functions together from parent to offspring. If a given parent uses function 5 but not function 6, for example, it will have (1,0) in the fifth and sixth positions of the n_i vector. If the "best" solution has (0,0) in the fifth and sixth positions, and the other parent has (0,1) in these positions, the only way to obtain (0,0) is to cut the n_i vectors between the fifth and

Table 3. QSARs Determined by the EP Method Using the T_g Data of Koehler and Hopfinger[11]

ITERM = 2, WEIGHT(2) = 2.0	
$T_g = 334.60 - 39.705*\mathbf{S}_B - 35.647*\mathbf{E}_+$ RMS-Error = 29.23	$T_g = 156.97 - 53.811*\mathbf{E}_D - 28.814*\mathbf{E}_+$ RMS-Error = 35.00
$T_g = 279.00 - 27.242*\mathbf{S}_B + 2.5697*\mathbf{M}_B$ RMS-Error = 34.86	
ITERM = 2, WEIGHT(2) = 1.0	
$T_g = 334.60 - 39.705*\mathbf{S}_B - 35.647*\mathbf{E}_+$ RMS-Error = 29.23	
ITERM = 3, WEIGHT(2) = 2.0	
$T_g = 300.59 - 29.526*\mathbf{S}_B - 37.400*\mathbf{E}_+ - 33.829*\mathbf{E}_-$ RMS-Error = 23.69	$T_g = 283.96 - 29.732*\mathbf{S}_B + 1.4414*\mathbf{M}_B - 28.429*\mathbf{E}_+$ RMS-Error = 26.11
$T_g = 267.91 - 28.445*\mathbf{S}_B - 25.949*\mathbf{E}_D - 29.534*\mathbf{E}_+$ RMS-Error = 25.27	$T_g = 338.75 - 39.727*\mathbf{S}_B - 1.0007*\mathbf{M}_S - 41.258*\mathbf{E}_+$ RMS-Error = 26.58
$T_g = 345.12 - 38.489*\mathbf{S}_B - 9.8059*\mathbf{S}_S - 33.581*\mathbf{E}_+$ RMS-Error = 25.62	
ITERM = 3, WEIGHT(2) = 1.0	
$T_g = 300.59 - 29.526*\mathbf{S}_B - 37.400*\mathbf{E}_+ - 33.829*\mathbf{E}_-$ RMS-Error = 23.69	$T_g = 345.12 - 38.489*\mathbf{S}_B - 9.8059*\mathbf{S}_S - 33.581*\mathbf{E}_+$ RMS-Error = 25.62
$T_g = 267.91 - 28.445*\mathbf{S}_B - 25.949*\mathbf{E}_D - 29.534*\mathbf{E}_+$ RMS-Error = 25.27	$T_g = 309.57 - 32.191*\mathbf{S}_B + 0.017\,109*\mathbf{M}_B^2 - 30.492*\mathbf{E}_+$ RMS-Error = 26.10
$T_g = 300.19 - 30.489*\mathbf{S}_B + 4.7865*\mathbf{E}_D^2 - 30.331*\mathbf{E}_+$ RMS-Error = 25.41	
ITERM = 4, WEIGHT(2) = 2.0	
$T_g = 256.29 - 22.783*\mathbf{S}_B - 19.433*\mathbf{E}_D - 32.531*\mathbf{E}_+ - 28.211*\mathbf{E}_-$ RMS-Error = 21.16	$T_g = 307.87 - 30.958*\mathbf{S}_B - 0.613\,16*\mathbf{M}_S - 40.594*\mathbf{E}_+ - 29.116*\mathbf{E}_-$ RMS-Error = 22.61
$T_g = 313.57 - 30.156*\mathbf{S}_B - 7.3115*\mathbf{S}_S - 35.595*\mathbf{E}_+ - 28.720*\mathbf{E}_-$ RMS-Error = 21.39	$T_g = 291.20 - 30.176*\mathbf{S}_B - 7.2073*\mathbf{S}_S - 19.899*\mathbf{E}_D - 29.441*\mathbf{E}_+$ RMS-Error = 23.30
$T_g = 268.58 - 23.696*\mathbf{S}_B + 1.0432*\mathbf{M}_B - 31.937*\mathbf{E}_+ - 29.214*\mathbf{E}_-$ RMS-Error = 21.82	
ITERM = 4, WEIGHT(2) = 1.0	
$T_g = 279.04 - 23.911*\mathbf{S}_B + 3.6940*\mathbf{E}_D^2 - 33.040*\mathbf{E}_+ - 28.850*\mathbf{E}_-$ RMS-Error = 21.04	$T_g = 268.58 - 23.696*\mathbf{S}_B + 1.0432*\mathbf{M}_B - 31.937*\mathbf{E}_+ - 29.214*\mathbf{E}$ RMS-Error = 21.82
$T_g = 256.29 - 22.783*\mathbf{S}_B - 19.433*\mathbf{E}_D - 32.531*\mathbf{E}_+ - 28.211*\mathbf{E}_-$ RMS-Error = 21.16	$T_g = 253.34 - 4.8019*\mathbf{S}_B^2 + 4.64626*\mathbf{E}_D^2 - 32.146*\mathbf{E}_+ - 29.612*\mathbf{E}_-$ RMS-Error = 21.87
$T_g = 313.57 - 30.156*\mathbf{S}_B - 7.3115*\mathbf{S}_S - 35.595*\mathbf{E}_+ - 28.720*\mathbf{E}_-$ RMS-Error = 21.39	
ITERM = 5, WEIGHT(2) = 2.0	
$T_g = 275.46 - 24.672*\mathbf{S}_B - 5.5849*\mathbf{S}_S - 15.375*\mathbf{E}_D - 32.169*\mathbf{E}_+ - 25.481*\mathbf{E}_-$ RMS-Error = 19.81	$T_g = 253.08 - 21.873*\mathbf{S}_B + 0.459\,60*\mathbf{M}_B - 14.654*\mathbf{E}_D - 31.322*\mathbf{E}_+ - 27.559*\mathbf{E}_-$ RMS-Error = 20.93
$T_g = 265.56 - 24.450*\mathbf{S}_B - 0.48573*\mathbf{M}_S - 17.898*\mathbf{E}_D - 35.446*\mathbf{E}_+ - 24.921*\mathbf{E}_-$ RMS-Error = 20.43	$T_g = 313.85 - 30.352*\mathbf{S}_B - 6.7768*\mathbf{S}_S - 0.103\,69*\mathbf{M}_S - 36.267*\mathbf{E}_+ - 28.296*\mathbf{E}_-$ RMS-Error = 21.37
$T_g = 288.61 - 26.007*\mathbf{S}_B + 0.716\,56*\mathbf{M}_B - 5.6354*\mathbf{S}_S - 32.257*\mathbf{E}_+ - 26.721*\mathbf{E}_-$ RMS-Error = 20.55	
ITERM = 5, WEIGHT(2) = 1.0	
$T_g = 293.16 - 25.464*\mathbf{S}_B - 5.6539*\mathbf{S}_S + 2.9934*\mathbf{E}_D^2 - 32.471*\mathbf{E}_+ - 25.843*\mathbf{E}_-$ RMS-Error = 19.62	$T_g = 269.37 - 5.3251*\mathbf{S}_B^2 - 6.6189*\mathbf{S}_S + 3.8320*\mathbf{E}_D^2\,31.306*\mathbf{E}_+ - 25.652*\mathbf{E}_-$ RMS-Error = 20.03
$T_g = 275.46 - 24.672*\mathbf{S}_B - 5.5849*\mathbf{S}_S - 15.375*\mathbf{E}_D - 32.169*\mathbf{E}_+ - 25.481*\mathbf{E}_-$ RMS-Error = 19.81	$T_g = 271.71 - 24.159*\mathbf{S}_B - 1.3357*\mathbf{S}_S^2 - 16.194*\mathbf{E}_D - 31.287*\mathbf{E}_+ - 26.571*\mathbf{E}_-$ RMS-Error = 20.24
$T_g = 290.51 - 24.973*\mathbf{S}_B - 1.3993*\mathbf{S}_S^2 + 3.1695*\mathbf{E}_D^2 - 31.505*\mathbf{E}_+ - 26.858*\mathbf{E}_-$ RMS-Error = 19.99	

sixth position and use the right substring from parent 1 and the left substring from parent 2. In Figure 2 of ref 3, Rogers and Hopfinger only show a single offspring from a crossover operation. It may be useful to generate both possible offspring, allow for mutation, and then only keep the most fit one.

This conservation of contiguous sections of the "chromosome" is both a strength and a weakness of the genetic algorithm. Again assume that (1,0) in the fifth and sixth positions yields a very good, but not the best, result. Many offspring will have this (1,0) combination since a predictor with it will have a high fitness and a high probability of being one of the parents. When the offspring is added to the population, the number of predictors with this (1,0) combination will increase from generation to generation. In the limiting case that all predictors in the population contain this (1,0) combination, a (0,0) predictor will never be generated unless a mutation operator is present that will either remove a descriptor or swap one descriptor for another in the equation. (Note that in ref 3 the mutation operators only add a function or change the position of a spline knot.)

The EP method is similar to a GA in that they both eventually rely on a "survival of the fittest" strategy. Good "genes" are passed from a parent to its offspring, and the offspring can improve on its fitness through mutation.

Table 4. QSARs Determined by the EP Method Using the T_m Data of Koehler and Hopfinger[11]

ITERM = 2, WEIGHT(2) = 2.0

$T_m = 330.70 - 59.674*\mathbf{E}_+ - 108.93*\mathbf{E}_-$
RMS-Error = 56.47
$T_m = 304.12 + 3.2682*\mathbf{M}_B - 106.35*\mathbf{E}_-$
RMS-Error = 68.46
$T_m = 391.09 - 18.919*\mathbf{S}_S - 125.08*\mathbf{E}_-$
RMS-Error = 70.47

$T_m = 426.08 - 20.266*\mathbf{S}_B - 119.72*\mathbf{E}_-$
RMS-Error = 72.35
$T_m = 312.73 - 37.787*\mathbf{E}_D - 113.81*\mathbf{E}_-$
RMS-Error = 73.01

ITERM = 2, WEIGHT(2) = 1.0

$T_m = 330.70 - 59.674*\mathbf{E}_+ - 108.93*\mathbf{E}_-$
RMS-Error = 56.47
$T_m = 337.94 + 32.317*\mathbf{E}_+^2 - 96.223*\mathbf{E}_-$
RMS-Error = 60.17

$T_m = 355.60 - 0.262\,32*\mathbf{M}_S^2 + 60.000*\mathbf{E}_+^2$
RMS-Error = 66.30
$T_m = 364.63 - 5.5592*\mathbf{M}_S + 55.713*\mathbf{E}_+^2$
RMS-Error = 66.30

ITERM = 3, WEIGHT(2) = 2.0

$T_m = 348.19 - 15.255*\mathbf{S}_S - 56.976*\mathbf{E}_+ - 101.99*\mathbf{E}_-$
RMS-Error = 52.46
$T_m = 342.87 - 2.5512*\mathbf{M}_S - 62.961*\mathbf{E}_+ - 100.89*\mathbf{E}_-$
RMS-Error = 52.67
$T_m = 356.47 - 8.9463*\mathbf{S}_B - 57.107*\mathbf{E}_+ - 103.24*\mathbf{E}_-$
RMS-Error = 55.79

$T_m = 308.48 - 15.069*\mathbf{E}_D - 57.779*\mathbf{E}_+ - 101.25*\mathbf{E}_-$
RMS-Error = 56.04
$T_m = 319.89 - 0.655\,72*\mathbf{M}_B - 55.672*\mathbf{M}_S - 104.91*\mathbf{E}_+$
RMS-Error = 56.22

ITERM = 3, WEIGHT(2) = 1.0

$T_m = 349.13 - 4.6524*\mathbf{S}_S^2 - 55.689*\mathbf{E}_+ - 102.15*\mathbf{E}_-$
RMS-Error = 52.03
$T_m = 348.19 - 15.255*\mathbf{S}_S - 56.976*\mathbf{E}_+ - 101.99*\mathbf{E}_-$
RMS-Error = 52.46
$T_m = 352.28 - 3.8100*\mathbf{M}_S + 39.210*\mathbf{E}_+^2 - 78.076*\mathbf{E}_-$
RMS-Error = 52.64

$T_m = 342.87 - 2.5512*\mathbf{M}_S - 62.961*\mathbf{E}_+ - 100.89*\mathbf{E}_-$
RMS-Error = 52.67
$T_m = 338.09 - 0.111\,11*\mathbf{M}_S^2 - 66.016*\mathbf{E}_+ - 101.26*\mathbf{E}_-$
RMS-Error = 53.05

ITERM = 4, WEIGHT(2) = 2.0

$T_m = 382.15 - 13.160*\mathbf{S}_B - 2.8400*\mathbf{M}_S - 59.557*\mathbf{E}_+ - 91.619*\mathbf{E}_-$
RMS-Error = 51.15
$T_m = 378.88 - 10.457*\mathbf{S}_B - 15.759*\mathbf{S}_S - 53.886*\mathbf{E}_+ - 95.120*\mathbf{E}_-$
RMS-Error = 51.47
$T_m = 346.97 - 9.6749*\mathbf{S}_S + 1.0858*\mathbf{M}_S - 59.362*\mathbf{E}_+ - 101.11*\mathbf{E}_-$
RMS-Error = 52.30

$T_m = 325.75 - 2.4963*\mathbf{M}_S - 11.431*\mathbf{E}_D - 61.453*\mathbf{E}_+ - 95.239*\mathbf{E}_-$
RMS-Error = 52.41
$T_m = 340.80 - 14.888*\mathbf{S}_S - 4.7272*\mathbf{E}_D - 56.446*\mathbf{E}_+ - 99.751*\mathbf{E}_-$
RMS-Error = 52.41

ITERM = 4, WEIGHT(2) = 1.0

$T_m = 382.15 - 13.160*\mathbf{S}_B - 2.8400*\mathbf{M}_S - 59.557*\mathbf{E}_+ - 91.619*\mathbf{E}_-$
RMS-Error = 51.15
$T_m = 376.17 - 9.3433*\mathbf{S}_B - 4.6859*\mathbf{S}_S^2 - 52.979*\mathbf{E}_+ - 96.163*\mathbf{E}_-$
RMS-Error = 51.23
$T_m = 367.34 - 18.451*\mathbf{S}_B^2 - 4.8701*\mathbf{S}_S^2 - 52.805*\mathbf{E}_+ - 96.007*\mathbf{E}_-$
RMS-Error = 51.35

$T_m = 379.35 - 13.934*\mathbf{S}_B + 0.128\,00*\mathbf{M}_S^2 - 62.982*\mathbf{E}_+ - 91.246*\mathbf{E}_-$
RMS-Error = 51.38
$T_m = 378.88 - 10.457*\mathbf{S}_B + 15.759*\mathbf{S}_S^2 - 53.886*\mathbf{E}_+ - 95.120*\mathbf{E}_-$
RMS-Error = 51.47

ITERM = 5, WEIGHT(2) = 2.0

$T_m = 449.96 - 23.773*\mathbf{S}_B - 2.0223*\mathbf{M}_B - 3.6582*\mathbf{M}_S - 69.909*\mathbf{E}_+ - 94.696*\mathbf{E}_-$
RMS-Error = 49.79
$T_m = 430.32 - 18.062*\mathbf{S}_B - 1.5709*\mathbf{M}_B - 18.932*\mathbf{S}_S - 60.731*\mathbf{E}_+ - 98.476*\mathbf{E}_-$
RMS-Error = 50.60
$T_m = 382.31 - 12.351*\mathbf{S}_B + 6.0680*\mathbf{S}_S - 1.9032*\mathbf{M}_S - 57.508*\mathbf{E}_+ - 92.325*\mathbf{E}_-$
$\mathrm{RMS}_{\mathbf{E}}\mathbf{rror} = 51.00$

$T_m = 399.50 - 15.531*\mathbf{S}_B + 2.9216*\mathbf{M}_S + 7.4121*\mathbf{E}_D - 59.993*\mathbf{E}_+ - 93.809*\mathbf{E}_-$
RMS-Error = 51.08
$T_m = 411.48 - 14.236*\mathbf{S}_B - 17.009*\mathbf{S}_S + 13.754*\mathbf{E}_D - 54.310*\mathbf{E}_+ - 99.159*\mathbf{E}_-$
RMS-Error = 51.23

ITERM = 5, WEIGHT(2) = 1.0

$T_m = 449.96 - 23.773*\mathbf{S}_B - 2.0223*\mathbf{M}_B - 3.6582*\mathbf{M}_S - 69.909*\mathbf{E}_+ - 94.696*\mathbf{E}_-$
RMS-Error = 49.79
$T_m = 413.10 - 20.545*\mathbf{S}_B - 0.024\,519*\mathbf{M}_B^2 - 3.4263*\mathbf{M}_S - 66.349*\mathbf{E}_+ - 93.742*\mathbf{E}_-$
RMS-Error = 49.96
$T_m = 410.22 - 21.593*\mathbf{S}_B - 0.024\,722*\mathbf{M}_B^2 - 0.156\,00*\mathbf{M}_S^2 - 70.604*\mathbf{E}_+ - 93.184*\mathbf{E}_-$
RMS-Error = 50.19

$T_m = 382.55 - 12.741*\mathbf{S}_B - 2.9541*\mathbf{S}_S^2 - 0.076\,420*\mathbf{M}_S^2 - 58.012*\mathbf{E}_+ - 91.618*\mathbf{E}_-$
RMS-Error = 50.23
$T_m = 439.62 - 23.674*\mathbf{S}_B - 1.8233*\mathbf{M}_B - 0.160\,48*\mathbf{M}_S^2 - 73.170*\mathbf{E}_+ - 93.996*\mathbf{E}_-$
RMS-Error = 50.25

Again, it is impossible to state whether or not the EP method is superior to a GA since all possible mutation operators (and mating operators in a GA) have not been tried with a variety of probabilities. On the other hand, the structure of the EP method is better suited to this problem when the number of descriptors grows very large but the number of descriptors wanted in the final predictor remains small.

ACKNOWLEDGMENT

The author would like to thank Dr. David Rogers for a preprint of his GFA paper,[3] for helpful suggestions on this

manuscript, and for supplying the data set of Selwood *et al.*[2]

REFERENCES AND NOTES

(1) Hansch, C.; Fujita, T. ρ-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *J. Am. Chem. Soc.* **1964**, *86*, 1616.

(2) Selwood, D. L.; Livingstone, D. J.; Comley, J. C.; O'Dowd, A. B.; Hudson, A. T.; Jackson, P.; Jandu, K. S.; Rose, V. S.; Stables, J. N. Structure–Activity Relationships of Antifilarial Antimycin Analogues: A Multivariate Pattern Recognition Study. *J. Med. Chem.* **1990**, *33*, 136–142.

(3) Rogers, D.; Hopfinger, A. J. Application of Genetic Function Approximation to Quantitative Structure–Activity Relationships and Quantitative Structure–Property Relationships. *J. Chem. Inf. Comput. Sci.* **1994**, in press.

(4) Bledsoe, W. W. The use of biological concepts in the analytical study of systems. Paper presented at ORSA-TIMS National Meeting, San Francisco, CA, 1961.

(5) Holland, J. H. *Adaptation in Natural and Artifical Systems*; The University of Michigan Press: Ann Arbor, MI, 1975.

(6) Fogel, D. B. Applying Evolutionary Programming to Selected Traveling Salesman Problems. *Cybern. Syst. (USA)* **1993**, *24*, 27–36.

(7) Fogel, D. B.; Fogel, L. J.; Porto, V. W. Evolutionary Methods for Training Neural Networks. IEEE Conference on Neural Networks for Ocean Engineering (Cat. No. 91CH3064-3) **1991**, 317–327.

(8) Fogel, D. B.; Fogel, L. J. Optimal Routine of Multiple Autonomous Underwater Vehicles Through Evolutionary Programming. Proceedings of the Symposium on Autonomous Underwater Vehicle Technology. AUV '90 (Cat. No. 90CH2856-3) **1990**, 44–47.

(9) Wikel, J. H.; Dow, E. R'. The Use of Neural Networks for Variable Selection in QSAR. *Bioorg. Medicinal Chem. Lett.* **1993**, *3*, 645–651.

(10) Cardozo, M. G.; Iimura, Y.; Sugimoto, H.; Yamanishi, Y.; Hopfinger, A. J. QSAR Analysis of the Sibstituted Indanone and Benzylpiperidine Rings of a Series of Indanone-Benzylpiperidine Inhibitors of Acetylcholinesterase. *J. Med. Chem.* **1992**, *35*, 584–589.

(11) Koehler, M. G.; Hopfinger, A. J. Molecular Modelling of Polymers: 5. Inclusion of Intermolecular Energetics in Estimating Glass and Crystal-melt Transition Temperatures. *Polymer* **1989**, *30*, 116–126.

(12) Srinivas, M.; Patnaik, L. M. Genetic Algorithms: A Survey. *Computer* **1994**, June, 17–26.

(13) Nettleton, D. J.; Garigliano, R.; Siemens Plessey Defence Systems, Large Ratios of Mutation to Crossover: The Example of the Travelling Salesman Problem *Proc. SPIE–Int. Soc. Opt. Eng.* **1993**, *1962*, 110–119.

(14) Luke, B. T. Substructure Searching Using Genetic Methods. *J. Chem. Inf. Comput. Sci.*, submitted for publication.