LA-UR-97-3675

Title:

Fuzzy Logic vs. Niched Pareto Multiobjective Genetic Algorithm Optimization: Part I: Shaffer's F2 Problem

Author(s):

Brian J. Reardon

Submitted to:

Los Alamos
NATIONAL LABORATORY

Form No. 836 R5
ST 2629 10/91

Fuzzy Logic vs. Niched Pareto Multiobjective Genetic Algorithm Optimization: Part I: Schaffer's F2 Problem

Brian J. Reardon, Los Alamos National Laboratory, Los Alamos, NM 87545

Abstract
A new multiobjective selection procedure for Genetic Algorithms based on the paradigms of fuzzy logic is introduced, discussed, and compared to the niched Pareto selection procedure. In the example presented here the fuzzy logic procedure optimized the parameter of Schaffer's F2 problem in a manner of comparable efficiency to that of the niched Pareto approach. The main advantage, explicitly shown in this report, that the fuzzy logic approach has over the niched Pareto approach is that the experimental error or 'uncertainty' in the values to which a function is being optimized towards can be accounted for and thus used to better refine the optimal parameter range.

1.0 Introduction

1.1 Justification For Work

Numerous problems in materials science require the optimization of nonlinear (Fraga, 1996), multiparameter (Duggirala, 1994), multiobjective (Ozyurt, 1996) functions. In addition, many such materials science optimization problems have numerous minimums (or maximums) (Morgan, 1996) where all such points need to be located. Such problems are typified by atomic structure determination of proteins (Bush, 1995), of clusters (Zeiri, 1995), of small molecules (Judson, 1993), of alloys (Sutton, 1994), and of spin glasses (Smith, 1992). Other problems include, potential function parameter optimization (Skinner, 1995), x-ray diffraction pattern recognition (Paszkowicz, 1996), curve fitting (Ahonen, 1997; Karr, 1995), and production scheduling (Swinehart, 1996). Unfortunately, such problems are not easily tractable to such methodologies as Newton - Cauchy (Nazareth, 1995) or maximum entropy (Gzyl, 1995) and thus more robust search procedures are needed.

The requirements of an optimization method that can handle the above mentioned conditions are that it (i) account for the fact that there may exist an entire range of feasible solutions and it should determine what that range is, (ii) it must be capable of multiobjective (or attribute) optimization, (iii) it must be able to conduct comparisons solely on the final difference between experiment and theory and thus aim to minimize that difference, (iv) must obtain reasonable answers in a timely manner, thus out performing a simple random or Monte Carlo search. At present, the only optimization technique that efficiently attempts to fulfill all of these requirements is the genetic algorithm (GA) (Goldberg, 1989; Holland, 1975).

1.2 Generic GA Description

Generally speaking, one can formulate any optimization problem into a single standard of measurement - a cost function or a fitness function - that determines the performance of a decision and then recursively improves the performance by selecting from the most feasible of alternatives. For example, the square difference between an experimental quantity, *O*, and its theoretical value *f*(*x*) could define a fitness function such as $(O - f(x))^2$, where *x* represents the independent variable which must be optimized to minimize $(O - f(x))^2$. Traditional deterministic optimization techniques require the use of gradient or higher order statistical analysis of the cost function (Bazaraa, 1979). These methods find optimal solutions exponentially fast. Unfortunately, the solutions are usually locally optimal and insufficient for applied engineering problems (Anderson, 1986).

Darwinian evolution is an intrinsically robust search and optimization procedure. Evolved biota have optimized solutions to complex problems at every level of organization, from the cell up to the population. The problems that biota have solved and continue to improve upon, are typified by chaos, chance, temporality, nonlinearity, and multidimensionality. Such problems have proven to be intractable to deterministic optimization techniques, especially in situations where heuristic solutions are not available.

A GA falls into the much broader category of evolutionary algorithms. These algorithms attempt to simulate the processes of evolved biota in optimization. The essence of such a simulation lies in the expression of a solution to a problem not as a single value but as a string of fundamental building blocks (genes) that can be manipulated in much the same way as an extant species will manipulate its gene pool through selection and mating to produce more optimal offspring for the current environment. For example, consider x_1, which is a member of a population of feasible solutions to a problem but not necessarily the optimal solution. The real value of x_1 is expressed as a string of binary digits, e.g.: 101101110, that is L digits long. Each digit or bit is called an allele. This binary string is mapped to a real value of x_1 such that the string 11111111 corresponds to x_{max} and 00000000 corresponds to x_{min}. x_{max} and x_{min} define the upper and lower bounds respectively of the range of x that is being searched. The real value of x_1 is commonly referred to as a phenotype, whereas the binary string that defines x_1 is referred to as a chromosome and the genetic information contained in the chromosome is defined as a genotype. If a function requires the optimization of more than one variable, $f(x,y)$, then the total chromosome for a specific member is formed by placing the binary digits defining x and y back to back in one string. For example if x_1=001100 and y_1 = 110001 then the chromosome for member #1 would be: 001100110001.

Manipulation of these strings occurs in much the same way as extant species manipulate chromosomes. First, competition among members of the population determines who is most fit or optimal. Second, the most optimal members are allowed to reproduce. Reproduction involves slicing the chromosomes of two members of the populations and then exchanging the segments:

$$\begin{matrix} X_1 : 10100011 \\ X_2 : \underline{11110111} \end{matrix} \rightarrow \begin{matrix} \tilde{x}_1 : 1010\underline{0111} \\ \tilde{x}_2 : \underline{1111}0011 \end{matrix}$$

$\tilde{x}_1$ and $\tilde{x}_2$ are the resulting progeny and will be placed in the next generation. The actual crossover site is selected randomly with some probability, p_c. Third, mutation occurs, which in a positively entropic system ensures genetic diversity in the subsequent generation. Mutation involves flipping the value of a randomly selected allele with some probability, p_m. The new population that evolves from the selection, crossover, and mutation operators is defined as a generation. This cycle is repeated for a number of generations as specified by the user.

Numerous references are available regarding the formal structure of genetic algorithms and the theoretical foundation of their success (Fogel, 1994; Goldberg, 1989; Chambers, 1995).

1.3.0 Multiobjective Selection Procedures

1.3.1 Summation Of Least Squares

The simultaneous optimization of multiple objectives or attributes presents a real challenge. Consider a situation in which there are three functions ($f_1(x,y)$, $f_2(x,y)$, and $f_3(x,y)$) whose independent variables x and y need to be optimized. In such a situation the values of x and y that provide an optimal value of f_1 may or may not correspond to the optimal values of f_2 and f_3. Consequently a number of approaches have been adopted to extract a measure of fitness for the multiobjective system as a whole.

The first and easiest approach is to define and optimize a single function that incorporates each of the objectives (Swaragi, 1985). Such a function would look like:

$$F = \sum_{i=1}^{N=3} A_i \left(O_i - f_i(x,y)\right)^2 \qquad \text{Eq. 1.}$$

where f_i is one of the objective functions, O_i is the value to which f_i is being optimized towards, and A_i is a weighting factor used to ensure that one objective does not dominate the total fitness, F.

This approach has a number of draw backs. First, the weighting factors, A_i, have a significant impact on the ability of the GA to optimize all the functions. Second, the presence of the square is necessary to properly define the fitness function as a minimization problem, however, this results in a loss of information that could otherwise be used by the GA.

1.3.2 Vector Evaluated GA

Another approach to multiobjective optimization was Schaffer's vector evaluated genetic algorithm (Schaffer, 1985; Richardson, 1989). This approach involved evolving subpopulations along each of the objectives in question by allowing selection to occur independently for each criterion while allowing mating to occur between subpopulations. This approach has a limitation, however, in that there is a significant bias against middling individuals. In other words, individuals that were most optimum for one particular objective were prefered over individuals that represented a genuine trade off between objectives.

1.3.3 Niched Pareto GA

The next approach most commonly used today is the niched Pareto genetic algorithm. The methodology and efficiency of such a method has been elucidated elsewhere (Horn, 1994; Ritzel, 1994) but a brief introduction will be presented here.

The Pareto front is defined as the boundary along which all members of a population lie in a non-dominated environment. This means that, although members may have different attribute values, there is no member who is clearly better fit in all attributes. This environment, or frontier, represents the best of all possible tradeoffs between multiple attributes. Thus, the goal in a niched Pareto GA is to allow an entire population to evolve towards a non-dominated Pareto optimal frontier. Such an approach has practical use when one considers that a unique solution to a problem may not be attainable.

Traditional GAs (Goldberg, 1989) use binary tournament selection in which two members of a population are selected at random and compared. The better fit of the two members (the dominating member) is then saved for reproduction. Pareto domination tournaments differ slightly from binary tournament selection. Here, selection is accomplished by choosing 2 individuals (candidates), i = 1,2, at random from the current population. In addition, a set of j individuals is chosen, also at random, from the current population. This is the comparison set. Assuming the goal is to minimize the absolute difference between experimental and calculated attributes, the ith member is considered dominated by the comparison set if for any single j and for all k attributes $\left((O_k - f_{ki})^2 \geq (O_k - f_{kj})^2\right)$ is true and for the same j and at least one k attribute $\left((O_k - f_{ki})^2 > (O_k - f_{kj})^2\right)$ is true.

After the comparison, there are three possible situations: one candidate is dominated by the comparison set and the other is not, or neither is dominated, or both are dominated. In the first case, the non-dominated candidate is selected for reproduction. In the second and third case there is no clear winner and selection is conducted via simultaneous continuously updated phenotypic niche counting. The individual with the lower niche count in the partly filled next generation is selected for reproduction, since its genetic material is least represented in the gene pool.

Phenotypic niche counting is conducted by determining the distance, d_{ij}^{Ph}, separating members i and j in variable space. This separation distance is calculated through:

$$d_{ij}^{Ph} = \sqrt{\sum_{k=1}^{V}\left(\frac{x_{ik} - x_{jk}}{x_{k\max} - x_{k\min}}\right)^2}, \qquad \text{Eq. 2.}$$

where the summation over k refers to the V variables that constitute a single member of a population. x_{kmax} and x_{kmin} are the maximum and minimum allowed values of the variable space being searched. If d_{ij}^{Ph} is less than some scaling factor, σ, then the interaction is counted in the crowding factor:

$$C_i^{Ph} = \sum_j 1 - \frac{d_{ij}^{Ph}}{\sigma}. \qquad \text{Eq. 3.}$$

Note that the distances are normalized in the definition above. This normalization ensures that a single variable being optimized over a small search range does not dominate the crowding factor. In the Pareto tournament selection, the least crowded member will be selected for reproduction, since its genetic material is least represented in the gene pool.

This approach to multiobjective optimization has proven useful in a number of practical applications. However, it also has some drawbacks. The first drawback lies again in the defining the fitness of each attribute as the square value of the difference between the experimental value and the calculated value. This results in a net loss of information that could otherwise be useful to the GA in the optimization process.

The second drawback lies in the Pareto selection method itself. Namely, one can intuitively see that as the number of objectives being optimized goes up, the likelihood that one randomly selected member dominates another goes down. Thus, the likelihood that a member is selected due to actual fitness superiority rather than simple niche counting is dependent on the number of objectives.

An increase in the prominence of niching over Pareto selection leads to a third problem in the selection procedure. Namely, the influence of the scaling factor. If the selection procedure depends primarily on niching then the scaling factor will force the population to occupy sites in the variable space that are separated by a distance that is a function of the scaling factor. The evolution of such a population is thus not so much optimal in fitness as it is most uniformly distributed throughout the search space.

Another important point to consider in many engineering problems is that optimization of a function towards an experimentally derived value should in principle account for some degree of error in the experiment. All of the multi-objective optimization procedures described above assume that the value one is optimizing towards is an absolute. Furthermore, the above procedures do not allow the user to make adjustments for the fact that some objectives may be known experimentally to a high degree of accuracy while others will be known to a lower degree.

These drawbacks reveal a need for a revision to the niched Pareto procedure. The revision proposed in this work involves a rudimentary application of fuzzy logic as a selection method.

1.4 Fuzzy Logic Selection Methodology

A quick browse through the literature will show that fuzzy sets and fuzzy logic have achieved an important role in day-to-day engineering. The strength of fuzzy logic lies in its ability to define decision rules based on vaguely defined concepts rather than crisply expressed mathematical constructs (Bandemer, 1995). The use of fuzzy logic in conjuction with genetic algorithms is not new (Taring, 1997; Sanchez, 1997). However, the vast majority of papers on the subject deal with the use of Genetic Algorithms to help evolve optimal fuzzy sets. Surprisingly, little work has been done on using fuzzy logic as a selection method within a multiobjective genetic algorithm. The innovative work that has been done (Sakawa, 1997) overlooks a few of the distinct advantages that fuzzy logic offers a multiobjective genetic algorithm. These advantages will be focused on here and in subsequent reports.

Multiobjective optimization using fuzzy logic, as presented here, is fundamentally a hybrid of the least squares selection method and the niched Pareto selection method. The whole procedure can be summarized in two steps. First, a single fitness value that incorporates the values of all the objectives is calculated using fuzzy rules. Second, as in the Pareto selection procedure, two randomly selected members are compared to a comparison set. If one member has a fuzzy fitness value that dominates the set and the other does not then the dominating member is selected. Otherwise, continuously updated phenotypic niching is incorporated.

The key to the fuzzy logic approach lies in the definition of the fitness function and its corresponding fuzzy rules:

$$F = \frac{1}{N}\sum_{i=1}^{N} f'\left(f_i\right) \qquad \text{Eq. 4}$$

which is essentially an average over the N objectives in question. *f’* is a fuzzy logic rule set that scales the objective, f_i, according to how far away it is from the experimentally optimal solution. A typical fuzzy set would have the form:

$$\text{if } f_i \le \left(O_i - E_i\right) \rightarrow f'\left(f_i\right) = \left(\frac{S_{\min}}{f_{i\min} - \left(O_i - E_i\right)}\right)\left(f_i - \left(O_i - E_i\right)\right) \qquad \text{Eq. 5a}$$

if $\left(O_i - E_i\right) \le f_i \le \left(O_i + E_i\right) \to f'\left(f_i\right) = 0$ Eq. 5b

if $f_i \ge \left(O_i + E_i\right) \to f'\left(f_i\right) = \left(\frac{-S_{\max}}{\left(O_i + E_i\right) - f_{i\max}}\right)\left(f_i - \left(O_i + E_i\right)\right)$ Eq. 5c

where O_i is the ith experimental value that the ith function, f_i, is being optimized towards, E_i is the error or accepted uncertainty in O_i, $S_{min(max)}$ is a scaling parameter for values below (above) the accepted value, $f_{imin(imax)}$ is the smallest (largest) value of all the *i*th objectives in the population. Figure 1 is a graphic representation of the above functions.

Defining a multiobjective fitness function in such a way has a number of practical advantages. First, experimental uncertainty in the values to which a function is being optimized can be accounted for since all the calculated values within a certain range have the same fitness. Second, values less than or greater than the optimal value will each have their own distinct fitness and thus the GA will in effect have more information with which to optimize. Third, as will be shown explicitly in a subsequent report, the ability of the GA to select a member will not be influenced by the number of objectives as is the case in the niched Pareto method. Fourth, while one could argue that S_{max} and S_{min} are user defined parameters, their net influence on the final optimization procedure is not as great or as unpredictable as that of the weighting factors in the sum of least squares approach since it is a relatively simple matter to set S_{max} and S_{min} to the same values for all objectives.

A fifth idiosyncrasy of this procedure is that the fuzzy fitness is a function of the least optimal members of a population (f_{imax}, f_{imin}). Thus, the fitness of a particular member in one generation will not necessarily be equivalent to its fitness in a subsequent generation. The ramifications of such a dynamic definition of fitness will be not be discussed here but in subsequent papers.

2.0 Sample Problem: Schaffer's F2

A standard functional test of the ability of a GA to handle multiobjective optimization is Schaffer's F2 function which involves the simultaneous minimization of $f12(x) = x^2$ and $f22(x) = (x-2)^2$. This problem was addressed by Schaffer (Schaffer, 1985) using his vector evaluated GA and by Horn et al. (Horn, 1994) using the niched Pareto GA. Horn et al.'s work clearly showed the ability of the niched Pareto GA to find the optimal range of x, $0<x<2$ and to maintain an even distribution of the population throughout that range for a large number of generations. The application of the fuzzy fitness function to an optimization problem such as this involves an added dimensionally since the degree to which the two functions will be minimized must also be specified.

2.1 Procedure

The procedure will involve two steps in which Horn et al.'s results will first be approximately reproduced. The second step will involve the implementation of the fuzzy logic selection criteria. The parameters of the optimization used in the niched Pareto approach are identical to that of Horn et al.'s with the exception of a more tightly defined niche scaling parameter (Table I). Table II lists the additional fuzzy fitness parameters needed. The 'accepted error' in the minimum values will be varied to quantify the effect of this parameter on the final population distribution. Setting the accepted error to 4.0 for each function effectively forces the fuzzy fitness GA to find the same range of x as the niched Pareto GA.

3.0 Results And Discussion

3.1 The Niched Pareto Selection Criterion

Figure 2 shows the variation in the type of selection used by the niched Pareto GA as a function of generation number. While the number of Pareto selected members does flucuate with generation there seems to be an average range of Pareto selections that remains constant. This, by default, means that the average number of niche selected members is also a relative constant with generation. However, if we break the total niching into niche selection due to both members being non-dominated and both members dominated by the comparison set we see that initially there is a significant difference. Namely, for the first few generations there are very few situations in which both members dominate the comparison set and a large number where they are dominated by the comparison set. After approximately 15 generations the system equilibrates.

Since the number of niche selections is fundamentally constant the number of randomly selected individuals is also relatively constant. The number of randomly selected members is actually considerably higher than that observed by Horn et al. This is because of the use of a smaller scaling parameter in the niche counting scheme. The smaller scaling parameter results in a larger number of niche counts of zero which accounts for 99% of all the randomly selected instances. It is important to note that even with this relatively high degree of randomness incorporated into the optimization the results still closely match that of Horn et al.'s (figures 3a-b).

Figures 3a shows the population distribution at generation 0 and 200. Figure 3b shows the phenotype distribution as a function of generation number upto generation 50. Note the even distribution of the population in the optimal phenotype range even upto 200 generations (figure 3a) and a large percentage of random selections. Also note from figure 3b that the population as a whole actually reaches equilibrium quickly (within 15 generations).

3.2 The Fuzzy Logic Selection Criterion

Figure 4 is the same as figure 2 except that the fuzzy selection criteria was used instead of the niched Pareto selection criteria. The error in the objective values is 4.0 so figure 4 is directly comparable to figure 2. The difference in the selection process is remarkable. Compared to the Pareto method there are few selections due to fitness and most members are in fact selected in the niche process involving members who both dominate the comparison set. Very few members are niche selected due to being dominated by the comparison set, whose size is 4. Even with these differences the system reaches equilibrium, as seen in figures 5a-b, in a shorter number of generations.

The behavior of the selection criteria (Pareto vs. niched) does not change significantly when the estimated error changes. As is expected, however, the optimal range for x does change significantly. Figures 6 and 7 show the *Ei*=2.0 and *Ei*=6.0 data respectively after 200 generations. The *Ei*=2.0 data has a considerably narrower range than the *Ei*=4.0 populations where as the *Ei*=6 is slightly wider. It is important to point out that even with this effective change in optimal area, the fuzzy logic selection criteria was able to optimize the population to an equilibrium in a shorter number of generations than the niched Pareto GA.

4.0 Conclusions

This work shows that the fuzzy logic selection criteria and the niched Pareto selection criteria are comparable in efficiency at optimizing a multiobjective problem. While the fuzzy logic approach seems to optimize more quickly, the niched Pareto approach does maintain a more even distribution of the population in the optimal frontier. The main advantage demonstrated here, however, of the fuzzy approach over the niched Pareto GA is that one can assume an ‘error’ in the values of the functions that one is optimizing towards and thus one has greater control over the range in phenotypic space that the population evolves towards.

Tables

Table I. The parameters of the niched Pareto optimization of $f(x) = x^2$ and $f(x) = (x-2)^2$

pc	0.9
pm	0.01
Populations size	30
comparison set size	4
Niching scale	0.01
range of x	-6.0 to 6.0
chromosome length	14
generations	200

Table II. The additional parameters needed in the fuzzy fitness optimization of $f(x) = x^2$ and $f(x) = (x-2)^2$

the experimental minimums	0.0
errors in experimental minimums	2.0, 4.0, 6.0
Fuzzy scaling: Smin,Smax	1.0,1.0

Figures

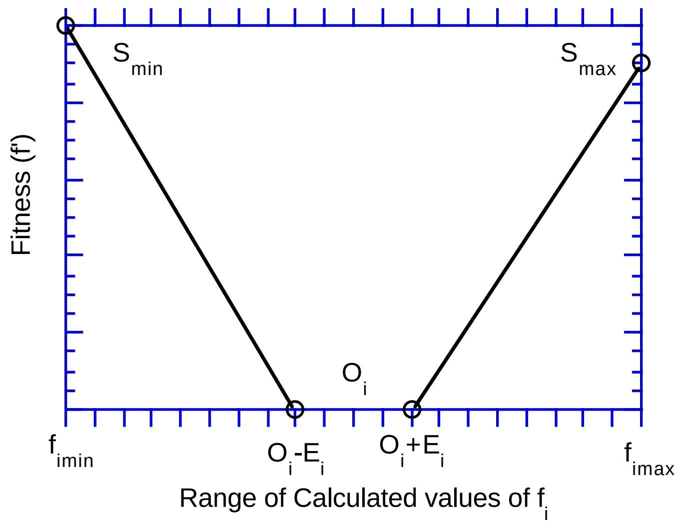

Figure 1) The fuzzy logic fitness as a function of one of the objective values of the *i*th member of a population.

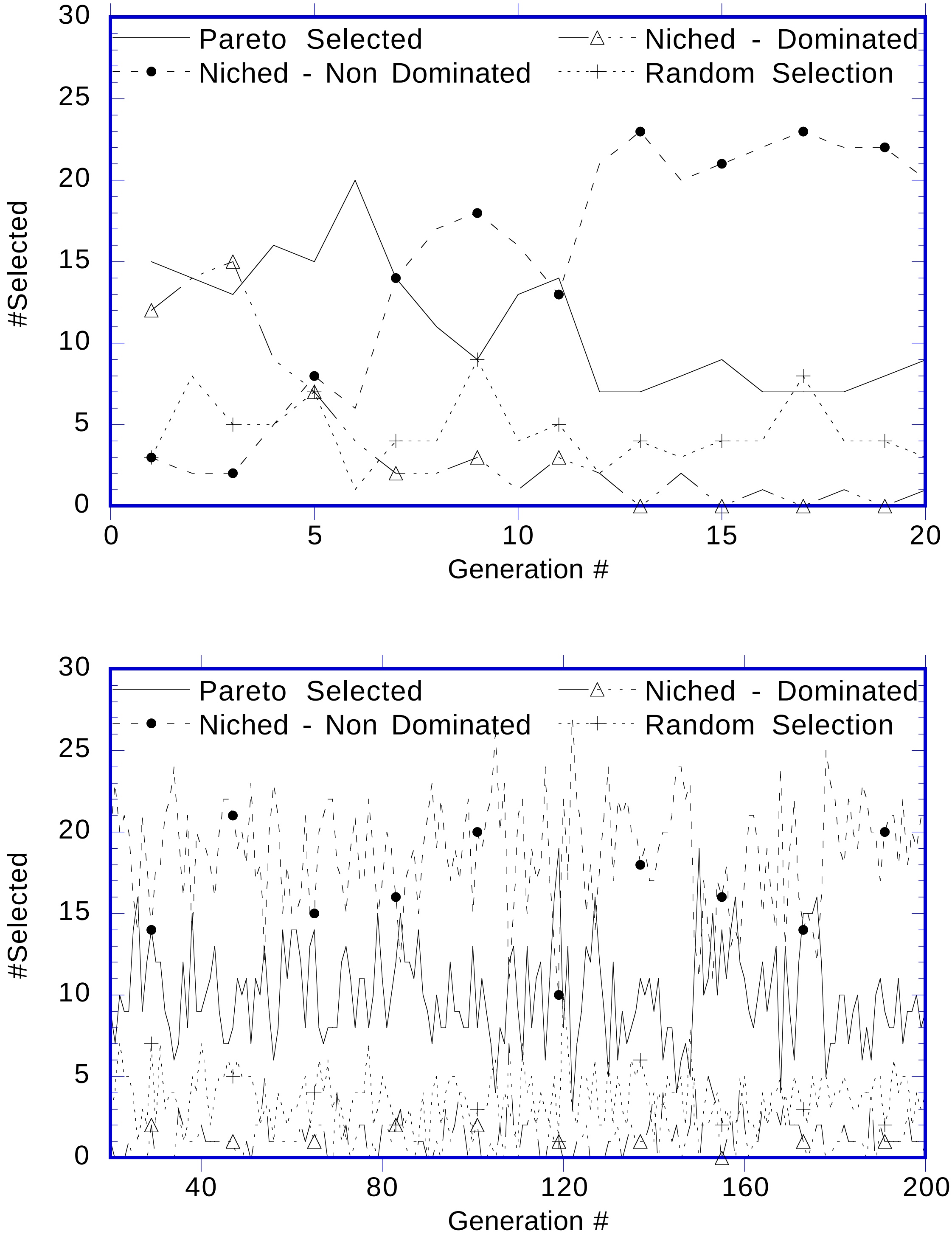

Figure 2) Selection type vs. generation number for the niched Pareto Genetic Algorithm.

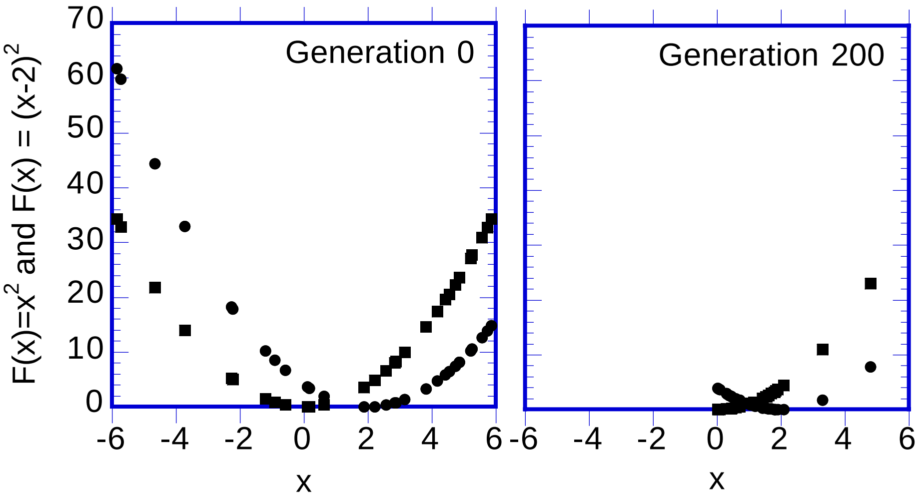

Figure 3a) Distribution of population at generation 0 and 200 using niched Pareto GA. n*f(x) = x²*; l*f(x) = (x-2)²*

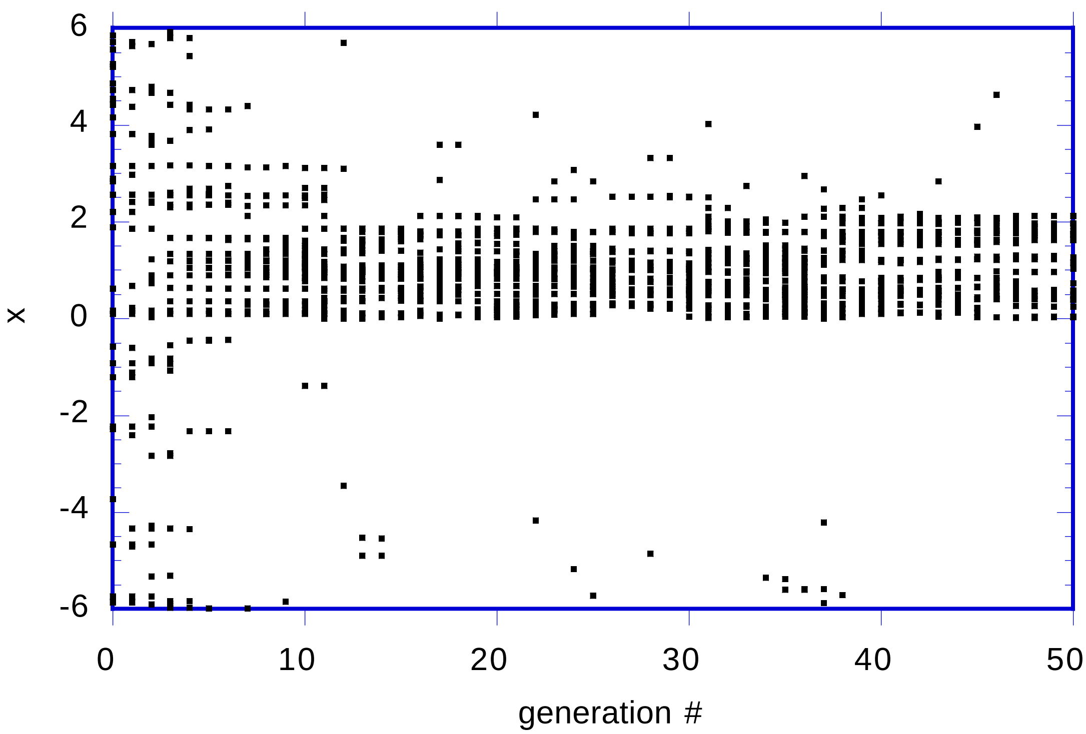

Figure 3b) Phenotypic distribution vs. generation number of niched Pareto GA.

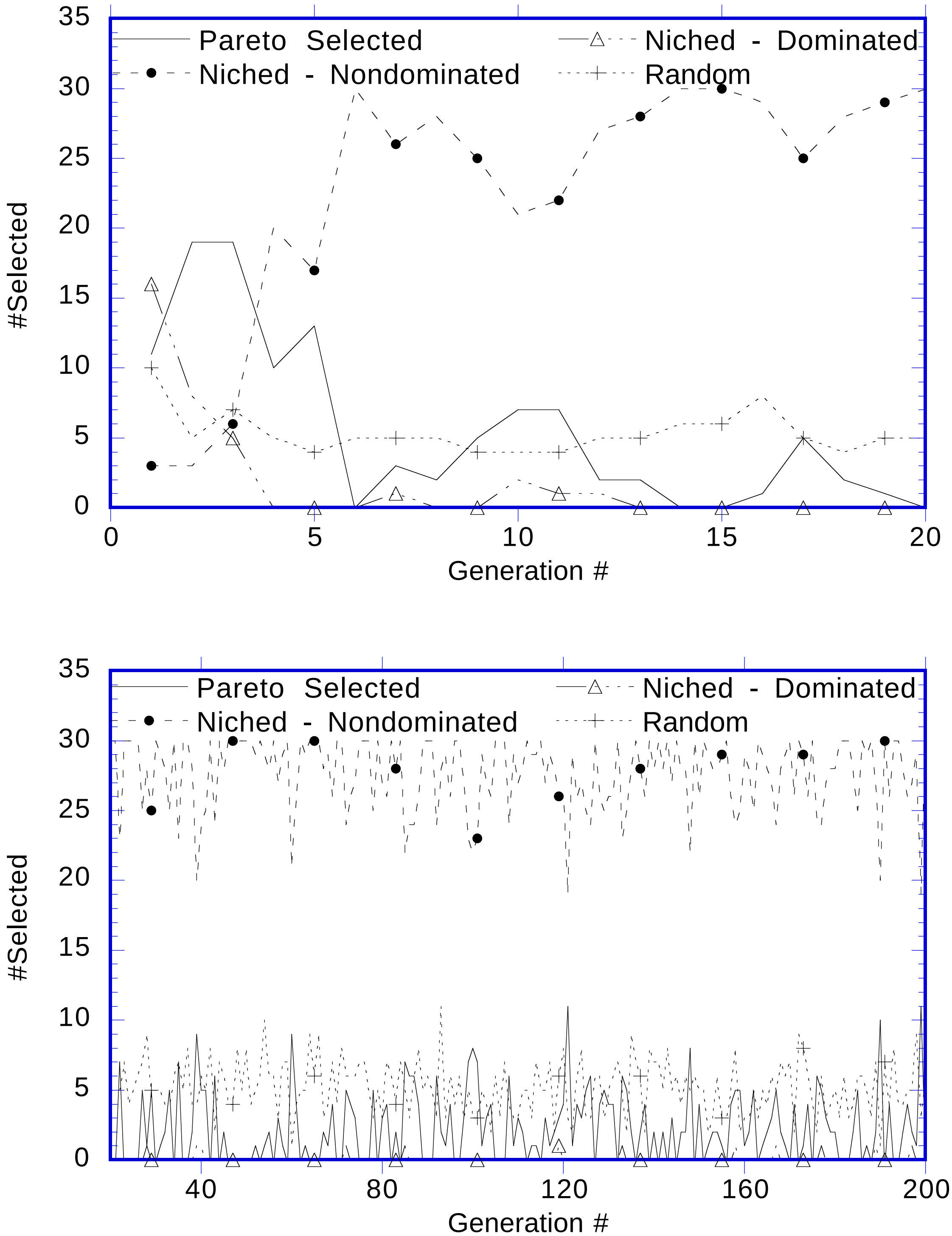

Figure 4) Selection type vs. generation number using the fuzzy logic based genetic algorithm with an excepted error of *Ei*=4.0.

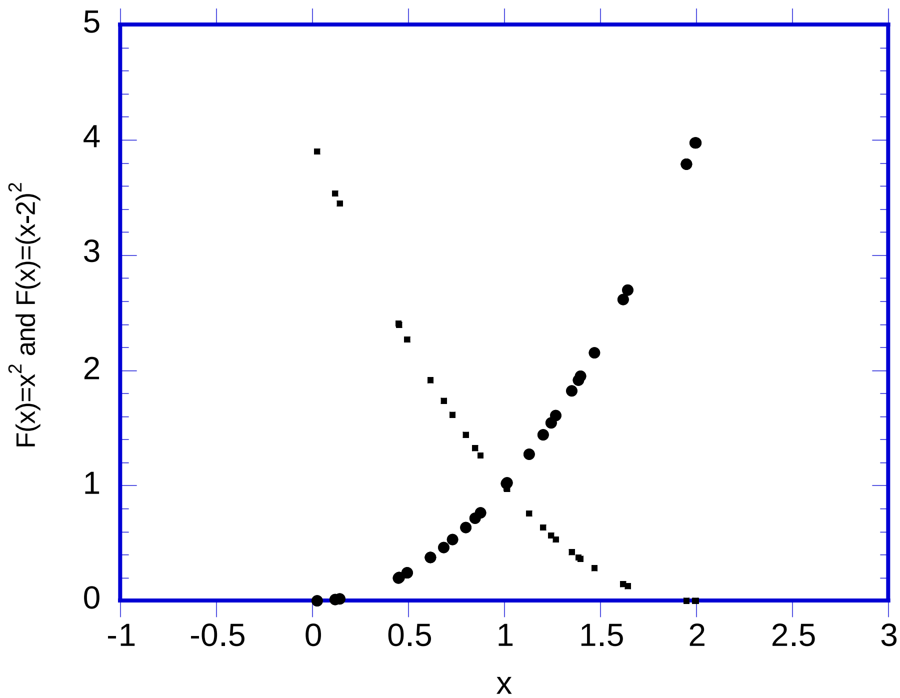

Figure 5a) Population with *Ei*=4.0 at generation 200. l$f(x) = x^2$; n$f(x) = (x-2)^2$

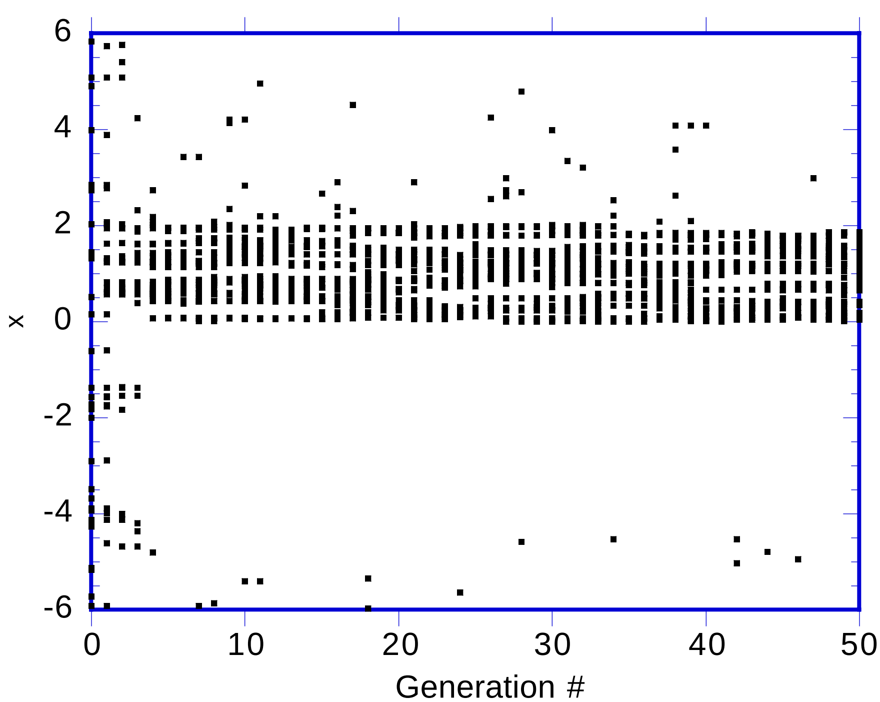

Figure 5b) Phenotypic distribution vs. generation number of fuzzy logic GA.

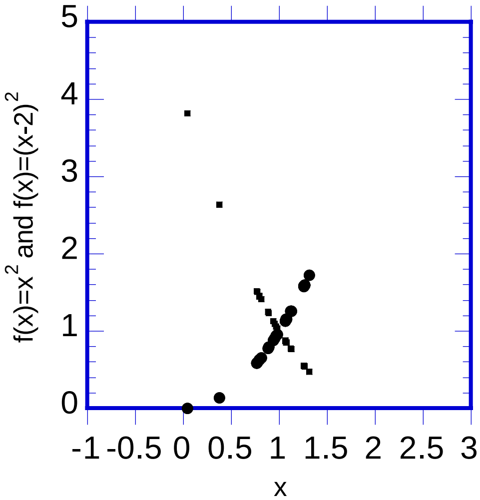

Figure 6) Population with *Ei*=2.0 at generation 200. l*f(x)* = x^2; n*f(x) = (x-2)*

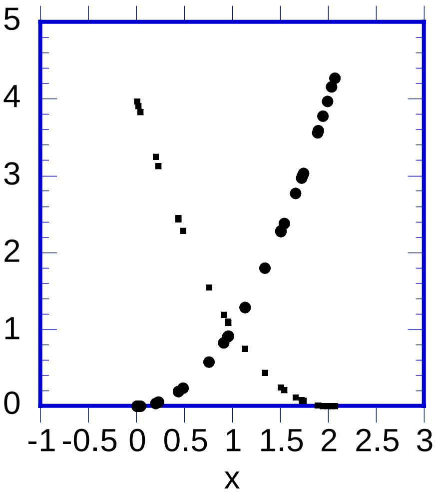

Figure 7) Population with *Ei*=6.0 at generation 200. l*f(x)* = x^2; n*f(x) = (x-2)*

References

Ahonen, H., Desouza, P. A., Garg, V. K., A Genetic Algorithm For Fitting Lorentzian Line-Shapes In Mossbauer-Spectra, Nuclear Instruments & Methods In Physics Research Section B-Beam, Interactions With Materials And Atoms, 124, 4, 633-638, 1997.

Anderson, B. D. O., Bitmead, R. R., Johnson, C. R., Kokotovic, P. V., Kosut, R. L., Mareels, I. M. Y., Praly, L., Riedle, B. D., Stability Of Adaptive Systems: Passivity And Averaging Analysis, M I T Press, Cambridge, Ma, 1986.

Bandemer, H., Gottwald, S., Fuzzy Sets, Fuzzy Logic, Fuzzy Methods With Applications, John Wiley & Sons, New York, 1995.

Bazaraa, M. S., Shetty, C. M., NonLinear Programming, John Wiley, New York, 1979.

Bush, T. S., Catlow, C. R. A., Battle, P. D., Evolutionary Programming Techniques For Predicting Inorganic Crystal-Structures, Journal Of Materials Chemistry, 5, 8, 1269-1272, 1995.

Chambers, L., Practical Handbook Of Genetic Algorithms: New Frontiers, Volume I I, C R C Press, New York, 1995.

Duggirala, R., Shivpuri, R., Kini, S., Ghosh, S., Roy, S., Computer-Aided Approach For Design And Optimization Of Cold-Forging Sequences For Automotive Parts, Journal Of Materials Processing Technology, 46, 1-2, 185-198, 1994.

Fogel, D. B., An Introduction To Simulated Evolutionary Optimization, I E E E Transactions On Neural Networks, 5, 1, 3, 1994.

Fraga, E. S., Matias, T. R. S., Synthesis And Optimization Of A Nonideal Distillation System Using A Parallel Genetic Algorithm, Computers & Chemical Engineering, 20, Sa, S79-S84, 1996.

Goldberg, D. E., Genetic Algorithms In Search, Optimization, And Machine Learning, Addison - Wesley Pub. Co. Inc., New York, 1989.

Gzyl, H., The Method Of Maximum Entropy, World Scientific, River Edge, N. J, 1995.

Holland, J. H., Adaptation In Natural And Artificial Systems,, University Of Michigan Press,, Ann Arbor, MI, 1975.

Horn, J., Nafpliotis, N., Goldberg, D. E., A Niched Pareto Genetic Algorithm For Multiobjective Optimization, Proceedings Of The First I E E E Conference On Evolutionary Computation, Institute Of Electrical And Electronics Engineers; Ieee Neural Networks Council, I E E E World Congress On Computational Intelligence, Piscataway, N J, 1, 1, 82-87, 1994.

Judson, R. S., Jaeger, E. P., Treasureywala, A. M., Peterson, M. L., Conformational Searching Methods For Small Molecules. II. Genetic Algorithm Approach, Journal Of Compuational Chemistry, 14, 11, 1407-1414, 1993.

Karr, C. L., Weck, W., Massart, D. L., Vankeerberghen, P., Least Median Squares Curving Fitting Using A Genetic Algorithm, Engineering Applications Of Artificial Intelligence, 8, 2, 177-189, 1995.

Morgan, S. R., Higgs, P. G., Evidence For Kinetic Effects In The Folding Of Large R N A Molecules, Journal Of Chemical Physics, 105, 16, 7152-7157, 1996.

Nazareth, J. L., The Newton-Cauchy Framework : A Unified Approach To Unconstrained Nonlinear Minimization, Springer-Verlag, New York, 1994.

Ozyurt, B., Mogili, P., Mierau, B., Sunol, S. G., Sunol, A. K., A Hierarchical Approach To Simultaneous Design Of Products And Processes, Computers & Chemical Engineering, 20, Sa, S73-S78, 1996.

Paszkowicz, W., Application Of The Smooth Genetic Algorithm For Indexing Powder Patterns : Tests For The Orthorhombic System, Materials Science Forum, 228, 1&2, 19-24, 1996.

Richardson, J. T., Palmer, M. R., Liepins, G., Hilliard, M., Some Guidelines Of Genetic Algorithms With Penalty Functions, Proceedings Of The Third International Conference On Genetic Algorithms, J. D. Schaffer, Morgan Kauffman, New York, 191-197, 1989.

Ritzel, B. J., Eheart, J. W., Ranjithan, S., Using Genetic Algorithms To Solve A Multiple Objective Groundwater Pollution Containment Problem, Water Resources Research, 30, 5, 1589-1603, 1994.

Sakawa, M., Kato, K., Sunada, H., Shibano, T., Fuzzy Programming For Multiobjective 0-1 Programming Problems Through Revised Genetic Algorithms, European Journal Of Operations Research, 97, 1, 149-158, 1997.

Sanchez, E., Shibata, T., Zadeh, L. A., Genetic Algorithms And Fuzzy Logic Systems : Soft Computing Perspectives, World Scientific, River Edge, N J, 1997.

Schaffer, J. D., Multiple Objective Optimization With Vector Evaluated Genetic Algorithms, Proceedings Of The First International Conference On Genetic Algorithms And Their Applications : July 24-26, 1985 At The Carnegie-Mellon University, Pittsburg, Pa / Sponsored By Texas Instruments, Inc., Naval Research Laboratory, J. Grefenstette, Lawrence Erlbaum Associates, Publishers, Hillsdale, N. J., 1, 1, 93-100, 1988.

Skinner, A. J., Broughton, J. Q., Neural Networks In Computational Materials Science : Training Algorithms, Modelling And Simulation In Materials Science And Engineering, 3, 3, 371-390, 1995.

Smith, R. W., Energy Minimization In Binary Alloys Models V Ia Genetic Algorithms, Computer Physics Communications, 71, 1, 134-146, 1992.

Sutton, P., Hunter, D. L., Jan, N., The Ground State Energy Of The ±J Spin Glass From The Genetic Algorithm, J. Phys. I France, 4, 1281-1285, 1994.

Swaragi, Y., Nakayama, H., Theory Of Multiobjective Optimization, Academic Press, Orlando, Florida, 1985.

Swinehart, K., Yasin, M., Guimaraes, E., Applying An Analytical Approach To Shop-Floor Scheduling : A Case-Study, International Journal Of Materials & Product Technology, 11, 1-2, 98-107, 1996.

Tarng, Y. S., Nian, C. Y., Kao, J. Y., Automatic Synthesis Of Membership Functions For The Force Control Of Turning Operations, Journal Of Materials Processing Technology, 65, 1-3, 80-87, 1997.

Zeiri, Y., Prediction Of The Lowest Energy Structure Of Clusters Using A Genetic Algorithm, Physical Review E, 51, 4, 2769, 1995.