International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 1

# Tackling Real-Coded Genetic Algorithms

**M. Nishidhar Babu**
Dept. Mech. Engg., RVR & JCCE,
Gunrtur, Andra Pradesh,India

**Y. Kiran**
Dept. Mech. Engg., KHIT,
Gunrtur, Andra Pradesh,India

**A. Ramesh**
Dept. Mech. Engg., VIT, Vellure, Tamil Nadu, India

**V. Rajendra**
Dept. Mech. Engg., CSU, Melbourne, Australia

## ABSTRACT

Genetic algorithms play a significant role, as search techniques for handling complex spaces, in many fields such as artificial intelligence, engineering, robotic, etc. Genetic algorithms are based on the underlying genetic process in biological organisms and on the natural evolution principles of populations. These algorithms process a population of chromosomes, which represent search space solutions, with three operations: selection, crossover and mutation.

Under its initial formulation, the search space solutions are coded using the binary alphabet. However, the good properties related with these algorithms do not stem from the use of this alphabet; other coding types have been considered for the representation issue, such as real coding, which would seem particularly natural when tackling optimization problems of parameters with variables in continuous domains. In this paper we review the features of real-coded genetic algorithms. Different models of genetic operators and some mechanisms available for studying the behavior of this type of genetic algorithms are revised and compared.

***Keywords:*** *Genetic algorithms, real coded genetic algorithm*

## INTRODUCTION

Genetic algorithms (GAs) are general purpose search algorithms which use principles inspired by natural genetic populations to evolve solutions to problems. The basic idea is to maintain a population of chromosomes, which represent candidate solutions to the concrete problem that evolves over time through a process of competition and controlled variation. Each chromosome in the population has an associated fitness to determine which chromosomes are used to form new ones in the competition process, which is called selection. The new ones are created using genetic operators such as crossover and mutation. GAs has had a great measure of success in search and optimization problems. The reason for a great part of their success is their ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces, i.e., their adaptation. This is their key feature, particularly in large, complex, and poorly understood search spaces, where classical search tools (enumerative, heuristic, :::) are inappropriate, offering a valid approach to problems requiring efficient and effective search techniques. Fixed-length and binary coded strings for the representation of the solutions have dominated GA research since there are theoretical results that show them to be the most appropriate ones and as they are amenable to simple implementation. But the GA's good properties do not stem from the use of bit strings (29;22). For this reason, the path has been lain toward the use of non-binary representations more adequate for each particular application problem. One of the most important ones is the real number representation, which would seem particularly natural when optimization problems with variables in continuous search spaces are tackled. So a chromosome is a vector of floating point numbers whose size is kept the same as the length of the vector, which is the solution to the problem. GAs based on real number representation is called real-coded GAs (RCGAs). The use of real coding initially appears in specific applications, such as in (23) for chemo metric problems, and in (26) for the use of met operators in order to find the most adequate parameters for a standard GA. Subsequently, RCGAs have been mainly used for numerical optimization on continuous domains (30; 26; 22; 30). Until 1991 specific theoretical studies about RCGA operation weren't done and so the use of these algorithms was controversial; researchers familiar with fundamental GA theory didn't understand the great success of RCGAs since this suggested that binary coding should be more effective than coding based on large alphabets. Later, tools for the theoretical treatment of RCGAs were proposed (see(30; 22; 20)) and so their power was corroborated. Other evolution algorithms based on real coding are the Evolution Strategies (ES). The similarity between RCGAs and ES allows some of their genetic operators to be exchanged (see (28)). The main objective of this paper is to deal with the RCGAs. To do that, first we study the binary coding and its advantages and drawbacks. Then we study the main issues related with RCGAs and the tools for the analysis of the RCGAs.

We need to highlight that this paper is advisable and interesting for people working on GAs and people that need a power search procedure for problems with large, complex, and poorly understood search spaces. RCGAs demonstrate to be ones of the most appropriate search methods on problems with these features where continuous variables are implied. We set up the paper as follows. In Section 2 we attempt to describe the principal aspects of GAs, thinking of people that are not familiarized with them. Then, in Section 3 we expose the particular features of binary-coded GAs (BCGAs) and show the reasons argued for preferring binary alphabet, which has been the most used through GA history. We also point out the principal problems that appear when BCGAs are applied. In Section 4 we attempt the RCGAs, we expose their advantages, we present and compare the crossover and mutation operators proposed for these algorithms in the literature, we deal with the application of RCGAs for handling convex spaces, and finally, we treat the hybridization of RCGAs with other search methods. In Section 5 we report tools that allow the behavior of RCGAs to be studied from a theoretical point of view. In Section 6 some conclusions are pointed out.

## REAL CODED GENETIC ALGORITHM(RCGA)

We have already noted the complicated proceed=ss of encoding and decoding used in binary GA.in real coded GA ,the design variables represented as floating point numbers. If a problem has n design variables, then the design vector can be represented exactly in the same form as used for gradient-based method.

$X=\{x_1,x_2,x_3,\ldots,x_n\}$

Note that the binary GA, the design vector was represented by a string and the elements $x_1,x_2,x3,\ldots.x_n$ were each represented as binary substrings of m bits each. In real GA, we use n floating point numbers while in binary GA, we use m * n bits to represents the design vector. The real GA can use single- or double-precision arithmetic depending on the computer. Since real GA is search algorithm which starts from a population of values, like for binary GA, these move limits have the form $x_i^l \leq x_i \leq x_i^U$,i =1 to n. The fitness value of the function is calculated using

$f(x) = f(x_1 ,x_2 ,x_3,\ldots\ldots,x_n)$.

Real GA therefore saves us from the complexity of using the encoding and decoding operations of binary GA.

### Starting population

The starting population is created by taking a floating point number within the design space for each variable. Thus, the starting population with all design variables lying between 1 and 10 may be expressed in the form using random numbers with up to eight decimal places.

**Member 1 :**

$\underbrace{2.34527849}$ $\underbrace{1.34272192}$ $\underbrace{1.23972398}\ldots$ $\underbrace{7.23582302}$

$x_{11}$ $x_{12}$ $x_{13}$ $x_{1n}$

**Member 2 :**

$\underbrace{7.54989200}$ $\underbrace{3.04011890}$ $\underbrace{2.18998363}\ldots$

5.82990872

$x_{21} x_{22}$ $x_{23}$ $x_{2n}$

:

:

:

**Member N :**

$\underbrace{9.82019902}_{x_{N1}\ x_{N2}}\ \underbrace{4.02384002}_{x_{N3}}\ \underbrace{5.77810282}_{x_{Nn}}\ldots\ \underbrace{0.12927494}$

The initial population is therefore a matrix of real numbers of size N*note that random sampling can also be enhanced by uniform sampling to make sure that all parts of the design space are adequately sampled. In addition, a complement approach could be used by subtracting the population of each design variable from its upper bound.

Just as in binary GA, it is advantageous to have a larger initial population compared to the population value used in the GA generations. Thus, if we start with a population of 1.2 N-2 N, the N best points can be selected for further operations such as reproduction, crossover rand mutation operations. However, in real GA, some of those operations need to be defined in a new manner compared to binary GA.

## Reproduction

Roulette wheel selection can be used for real GA. The fitness function and cumulative probability can then be calculated in exactly the same manner as for binary GA. Just as in binary GA, we can select the mating pool members as 1,3,2,1 and3 .the mates can then be paired as (1,3) and(2,1) assuming 80 per cent crossover. The only difference in the reproduction operations for real GA is that string i is replaced by design variable i which is the design vector $\{x_{i1},x_{i2},x_{i3},\ldots x_{in}\}$.we also see that scaling correction and tournament selection can be applied to real GA .the reproduction operator does not create any new points.

## Crossover

The crossover operator used for real GA is different from that used binary GA. A simple approach we may follow is to take two sites along the parent as the crossover site and then exchange the variables inside the crossover sites. For example, if there are two six-variables parents as given below

Father $=\{x_{f1},x_{f2},x_{f3},x_{f4},x_{f5},x_{f6}\}$,
Mother$=\{x_{m1},x_{m2},x_{m3},x_{m4},x_{m5},x_{m6}\}$,
We select two crossover sites randomly and get 1 and 3.the children are then as shown in the following.
Father $= \{ x_{f1}, x_{f2}, x_{f3}, x_{f4}, x_{f5}, x_{f6}\}$,
Mother$= \{ x_{m1}, x_{m2}, x_{m3}, x_{m4}, x_{m5}, x_{m6}\}$,

The above strategy well with binary GA where the variables are coded strings, however in real GA the swapping of design variables does not introduced any new information. We need a method to create new design variables .one approach for creating new design variables is to use the blending method and define the children.

$x^{(1)}_{new} =\beta x_{mn} + (1 -\beta)x_{fn}$ ,
$x^{(2)}_{new} = \beta x_{fn} + (1 - \beta)x_{mn}$

In the above ,$\beta \in (0,1)$,$x_{mn}$ is the nth design variable in the mother design vector and xfn is the nth design variable in the father design vector .the limiting case occurs when $\beta =0 =x^{(1)}_{new}= x_{fn},x^{(2)}_{new}=x_{mn}$ and when $\beta=0.5$,the two children are the average of the two-parent design variables and are essentially identical twins.in general ,it is a good idea to take a random number $\beta\in(0,1)$ and find the values of the two children. This method is also called the blending method since it combines information from both children. This method is also called the blending method since combines information from both parents to get the children and therefore simulates nature. However, the blending method described in now interpolates between the parent values and is not capable of extrapolating into the design space.

Blending approaches to crossover which extrapolate have also been proposed by some researchers. One such approach is called the linear crossover where there children are created using two parents as follows:
$x^{(1)}_{new}= 0.5x_{mn}+0.5_{fn}$,
$x^{(2)}_{new}=1.5x_{mn}-0.5x_{fn}$,
$X^{(3)}_{new}=-0.5x_{mn}+1.5x_{fn}$,

Here the first child is interpolated while the second and third children are extrapolated. A problem with extrapolation is that sometimes a child may go outside the bounds of the design variable. In such a case, the child is not selected. The best two children among the three are selected for further operations. as an example, consider $x_{mn}=1,x_{fn}=2$.Then,the children are given by $x^{(1)}_{new}=1.5$, $x^{(2)}_{new}=0.5$, $x^{(3)}_{new} =2.5$.thus we see that the linear crossover both interpolates and extrapolates using the parent values .

A further generalization of the concept of the linear crossover in needed to ensure that more than three children can be created in case more than one need to be discarded because they lie outside the move limits for the design variable .we can define any number of children of two parents using heuristic crossover.

$x_{new} = \beta(x_{mn}-x_{fn}) + x_{mn}.$

This approach allows generation of children both inside and outside the parent range depending on the value of the random number β ϵ (0, 1). Heuristic crossover also introduces an element of randomness in the crossover process which is absent in linear crossover.

As an example of implementing the two-point heuristic crossover consider the two-point design vectors, shown below:
Father= {2.762, 4.384, 1.236, 0.524},
Mother= {7.310, 8.236, 5.426, 4.316},

Here each design variable has abound of (0, 10) as the lower and upper limit. Using a random number generator, we pick two crossing sites, 2 and 3. Again, we generate three random numbers β ϵ (0, 1) and get 0.1783, 0.8264 and 0.3123, using these values and heuristic crossover, we get three children:

$x^{(1)}_{new}$ =0.1783(8.236-4.384)+8.236=8.9228,
$x^{(2)}_{new}$ =0.8264(8.236-4.384)+8.236=38.0745,
$x^{(3)}_{new}$ = 0.3123(8.236-4.384)+8.236=9.4389.
The second child is outside the movie limit of the design variable and is not selected, we then take the other two values to farm the children:
Father= {2.762, 4.384, 1.236, 0.524},
Mother= {7.310, 8.236, 5.426, 4.316},

The tow point heuristic crossover is good approach to use for real coded GA and is recommended for applications.

**Mutation**
As binary GA, mutation is needed in real GA to ensure that the algorithm does get stuck or coverage to a local minimum. To apply mutation with probability $p_m$. We change $n\times N\times p_m$ design variables in a random manner .recall that $n\times N$ is the number of real numbers in the population .as an example, consider design vectors of size n =6 and N=10 to make up the population. Taking a mutation probability ,$p_m$=0.05,we need to change three design variables , that is (6×10×0.05=3).to do this, we randomly select three variables from the population and replace them by a random number lying between the move limits corresponding to these variables.

In summary, we point out that many problems involving a low level of discretization can be solved using binary GA, even when the variables are real. However, as the desired accuracy of the optimal point increases, real GA becomes advantageous in terms of storage requirements. Recent research literature shows a growing popularity of real GA over binary GA.

## PRESSURE VESSEL PROBLEM

The Problem is to design a compressed air storage tank with a working pressure of 1000 psi and a minimum volume of 750 ft$^3$. The schematic of a pressure vessel is shown in Fig.7.1. The cylindrical pressure vessel is capped at both ends by hemispherical heads. Using rolled steel plate (SAEJ 2340 TYPE 830 R), the shell is to be made in two halves that are joined by two longitudinal welds to form a cylinder. Each head is forged and then welded to the shell. Let the design variables be denoted by the vector

$\boldsymbol{X}=[x_1, x_2, x_3, x_4]$

Where
$x_1$ is the spherical head thickness, $x_2$ is the shell thickness, $x_3$and $x_4$ are the radius and length of the shell, respectively.
The objective in this Project is to minimize the manufacturing cost of the pressure vessel. The manufacturing cost of the pressure vessel is a combination of material cost, welding cost and forming cost. That can be refer in Sandgren (1990) for more details on how cost is determined.
The constraints are set in accordance with respective ASME codes. Here the main objective is to reduce the cost by reducing weight of Pressure Vessel. So the objective function

$$f(x) = 0.6224\, x_1 x_2 x_3 + 1.7781\, x_1^2 x_3 + 3.1661\, x_2 x_4^2 + 19.84\, x_4 x_1^2$$

$x_1 = R =$ Radius of the shell
$x_2 = L =$ Length of the shell
$x_3 = T_s =$ Thickness of the shell
$x_4 = T_b =$ Thickness of the dish end

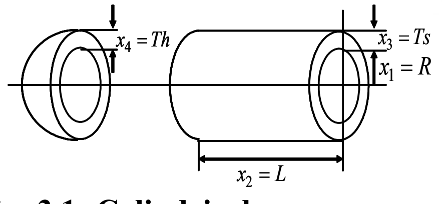

**Fig: 3.1: Cylindrical pressure vessel**

**Design constraints**

The four important constraints under consideration are

1. Hoop stress≤ Allowable stress

$$g_1(x) = 0.0193 \quad x_1 - x_4 \le 0$$

2. Longitudinal stress≤ Allowable stress

$$g_2(x) = 0.00954 \quad x_1 - x_3 \le 0$$

3. Volume≤750×1728inch$^3$

$$g_3(x) = 750 \times 1728 - \frac{4}{3}\pi x_1{}^3 - \pi x_1{}^2 x_2 \le 0$$

4. Length

$$g_4(x) = x_2 - 240 \le 0$$

**Variable bounds**

The upper and lower bounds on two design variables are

$25 \le x_1 \le 150$

$25 \le x_2 \le 240$

$0.0625 \le x_3 \le 1.25$

$0.0625 \le x_4 \le 1.25$

**Note: All are in inch**

## RESULTS AND DISCUSSIONS

The values of best design variables and the constraints for the 500 iteration obtained after running the program for Real coded Genetic Algorithm written in the C-language is given below.

**Table: 5.1: Programming Results**

| S.No | f(x)in $ | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 115339.8198 | 6.1190 | 5.8798 | 46.4278 | 197.4326 | 5.222965 | 5.436842 | 460144.3 | 42.5674 |
| 2 | 112644.9246 | 6.1243 | 5.8850 | 46.4623 | 187.5280 | 5.227579 | 5.441793 | 395894.7 | 52.4720 |
| . | . | . | . | . | . | . | . | . | . |
| 22 | 115339.8198 | 6.1190 | 5.8798 | 46.4278 | 197.4326 | 5.22965 | 5.436842 | 406144.3 | 42.5674 |
| 23 | 103103.1814 | 4.3550 | 1.0431 | 86.2090 | 193.6479 | 2.691221 | 0.220628 | 5908975 | 463521 |
| . | . | . | . | . | . | . | . | . | . |
| 155 | 15161.8598 | 1.5539 | 195.28 | 0.775 | 0.894 | 0.760824 | 0.928576 | 27686.69 | 45.2360 |
| 156 | 13590.1317 | 44.167 | 173.78 | 0.537 | 0.894 | 0.361261 | 1.584072 | 27686.69 | 45.2360 |
| . | . | . | . | . | . | . | . | . | . |
| 1993 | 6064.0452 | 0.8121 | 0.3924 | 41.0906 | 190.2364 | 0.019067 | 0.00037 | 3670.959 | 49.7636 |
| 1994 | 6006.5315 | 0.8041 | 0.3924 | 41.0906 | 190.2364 | 0.01103 | 0.00037 | 3670.959 | 49.7636 |
| . | . | . | . | . | . | . | . | . | . |
| 3655 | 5985.6915 | 0.8014 | 0.3924 | 41.0906 | 190.1739 | 0.008309 | 0.00037 | 333.146 | 49.8261 |
| 3656 | 5946.7682 | 0.7959 | 0.3924 | 41.0906 | 190.1739 | 0.002848 | 0.00037 | 3339.146 | 49.8261 |
| . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . |
| 10499 | 5940.3633 | 0.7959 | 0.3924 | 41.0906 | 189.8874 | 0.002848 | 0.00037 | 1819.803 | 50.1126 |

**Table 5.2: Comparison of RCGA with other Optimization Methods.**

| | Sandgren | RVR |
|---|---|---|
| Method | Penalty | RCGA |
| R[inch] | 47 | **41.0906** |
| L[inch] | 117.701 | **189.8874** |
| Ts[inch] | 1.125 | **0.7959** |
| $T_h$[inch] | 0.625 | **0.3924** |
| g1 | -0.194 | **-0.00285** |
| g2 | -0.0283 | **-0.0003956** |
| g3 | -0.0510 | **-1849.0951** |
| g4 | 0.054 | **-50.1126** |
| Objective[$] | 8129.800 | **5940.3633** |

Where

Penalty: Penalty Approach

RCGA : Real coded Genetic Algorithm

From above table it is clear that the Real coded GA gives the best results hence it can have beneficially used for evaluating the cost of the pressure vessel.

## CONCLUSION AND FUTURE SCOPE

In the present work parameters such as thickness of the shell, and dish end, length and radius of the pressure vessel are optimized by making use of Real coded genetic algorithm powerful non-traditional optimization method and these results are compared with other Optimization Methods.

1. It is found that the results obtained from RCGA are better as its search is for global optimum as against the local optimum in traditional search methods.
2. Sandgrenhas solved the problem using different algorithms such as Penalty Approach, But it is found that the results obtained by using proposed algorithm is better optimized than any other earlier solutions reported.
3. It can be concluded that by applying RCGA, the optimal design parameters for the pressure vessels are obtained and the objective minimization of cost by reducing weight of Pressure vessel is achieved.
4. In the present study the application of RCGA has been shown for a Pressure vessel problem with four variables and four design constraints.

**Future Scope:**

- In the proposed study the application of RCGA can be extended for pressure vessels with more than four variables and constraints (including Thermal Stresses).

- The present problem can also be extended for the use of the composite materials for weight minimization.

**REFERENCES**

1) L. Zhu and J. T. Boyle, "Optimal Shapes for Axisymmetric Pressure Vessels : A Brief," J. Press. Vessel Technol., vol. 122, no.November, pp. 443–449, 2000.

2) B. Abdi, H. Mozafari, A. Ayob, and R. Kohandel, "Optimum Size of a Ground-Based Cylindrical Liquid Storage Tank under Stability and Strength Constraints Using Imperialist Competitive Algorithm," Appl. Mech. Mater., vol. 110–116, pp. 3415–3421, 2011.

3) R. C. Carbonari, P. a. Muñoz-Rojas, E. Q. Andrade, G. H. Paulino, K. Nishimoto, and E. C. N. Silva, "Design of Pressure Vessels using Shape Optimization: An integrated approach," Int. J. Press. Vessel. Pip., vol. 88, no. 5–7, pp. 198–212, 2011.

4) S. H. Nasseri, Z. Alizadeh, and F. Taleshian, "Optimized Solution of Pressure Vessel Design Using Geometric Programming," J. Math. Comput. Sci., vol. 4, no. 3, pp. 344–349, 2012.

5) J. J. Proczka, K. Muralidharan, D. Villela, J. H. Simmons, and G. Frantziskonis, "Guidelines for The Pressure and Efficient Sizing of Pressure Vessels for Compressed Air Energy Storage," Energy Convers. Manag., vol. 65, pp. 597–605, 2013.

6) S. Hassan, K. Kumar, C. D. Raj, and K. Sridhar, "Design and Optimization of Pressure Vessel Using Metaheuristic Approach," Appl.Mech. Mater., vol. 465, pp. 401–406, 2014.

7) R. Talebitooti, M. H. Shojaeefard, and S. Yarmohammadisatri, "Shape Design Optimization of Cylindrical Tank Using b-Spline Curves," Comput. Fluids, vol. 109, no. October, pp. 100–112, 2015.

8) P. V. V Saidpatil and P. A. S. Thakare, "Design & Weight Optimization of Pressure Vessel Due to Thickness Using Finite Element Analysis," Int. J. Emerg. Eng. Res. Technol., vol. 2, no. 3, pp. 1–8, 2014.

9) K. S. Raju and S. S. Rao, "Design Optimization of a Composite Cylindrical Pressure Vessel Using FEA," Int. J. Sci. Res. Publ., vol. 5, no. 12, pp. 522–530, 2015.

10) Blachut, J., Eschenauer, H.A., 2001. Emerging Methods for Multidisciplinary Optimization. Springer, Wien, New York.

11) Schmitt, L.M., 2001. Theory of genetic algorithms. *Theoretical Computer Science*, **259**(1-2):1-61. [doi:10. 1016/S0304-3975(00)00406-0]

12) Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science*, **220**(4598):671-680. [doi:10.1126/science.220.4598.671]

13) de Vicente, J., Lanchares, J., Hermida, R., 2003. Placement by thermodynamic simulated annealing. *Physics Letters A*, **317**(5-6):415-423.

*14)* Peng-fei LIU, Ping XU, Shu-xin HAN, Jin-yang ZHENG. "Optimal design of pressure vessel using an improved genetic algorithm", *J Zhejiang UnivSci A 2008 9(9):1264-1269.*

15) Park JH, Hwang JH, Lee CS, Hwang W. Stacking sequence design of composite laminates for maximum strength using genetic algorithms. Compos Struct 2001;52(2):217e31.

16) Soremekun G, Gu¨ rdal Z, Kassapoglou C, Toni D. Stacking sequence blending of multiple

composite laminates using genetic algorithms. Compos Struct 2002;56(1):53e62.

17) Lee YJ, Lin CC. Regression of the response surface of laminated composite structures. Compos Struct 2003;62(1):91e105.

18) Kim CU, Kang JH, Hong CS, Kim CG. Optimal design of filament wound structures under internal pressure based on the semi-geodesic path algorithm. Compos Struct 2005;67(4):443e52.

19) Holland, J.H., Holyoak, K.J., Nisbett, R.E. &Thagard, P.R. (1986). Induction. Processes of Inference, Learning, and Discovery. The MIT Press, Cambridge

20) EshelmanL.J.&Schaffer J.D. (1993). RealCoded Genetic Algorithms and IntervalSchemata. Foundation of Genetic Algorithms 2, L.Darrell Whitley (Ed.) (Morgan Kaufmann Publishers, San Mateo), 187–202.

21) Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encoding constraint. Proc. of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.) (Morgan Kaufmann Publishers, San Mateo), 86–91.

22) Radcliffe N.J. (1992). NonLinear Genetic Representations. Parallel Problem Solving from Nature 2, R.M¨anner and B. Manderick (Ed.) (Elsevier Science Publichers, Amsterdam), 259–268.

23) Lucasius, C. B. &Kateman G. (1989). Applications of genetic algorithms in chemometrics. Proc. of the Third International Conference on Genetic Algorithms, J. David Schaffer (Ed.) (Morgan Kaufmann Publishers, San Mateo), 170–176.

24) Davis, L. (1989). Adapting Operator Probabilities in Genetic Algorithms. Proc. of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.) (Morgan Kaufmann Publishers, San Mateo), 61–69.

25) Davis, L. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York.

26) Wright, A. (1991). Genetic Algorithms for Real Parameter Optimization. Foundations of Genetic Algorithms 1, G.J.E Rawlin (Ed.) (Morgan Kaufmann, San Mateo), 205–218.

27) Herrera, F, HerreraViedma, E., Lozano, M. &Verdegay, J.L. (1994). Fuzzy Tools to Improve Genetic Algorithms. Proc. Second European Congress on Intelligent Techniques and Soft Computing, 1532–1539.

28) Herrera, F, Lozano, M. &Verdegay, J.L. (1995). Tuning Fuzzy Logic Controllers by Genetic Algorithms. International Journal of Approximate Reasoning 12, 299–315.

29) Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encodingconstraint. *Proc. of the Third Int. Conf. on Genetic Algorithms*, J. David Schaffer (Ed.)(Morgan Kaufmann Publishers, San Mateo), 86–91.

30) Wright, A. (1991). Genetic Algorithms for Real Parameter Optimization. *Foundations of Genetic Algorithms 1*, G.J.E Rawlin (Ed.) (Morgan Kaufmann, San Mateo), 205–218.