VLSI DESIGN
1996, Vol. 4, No. 3, pp. 207–215
Reprints available directly from the publisher
Photocopying permitted by license only

On Generating Optimal Signal Probabilities for Random Tests: A Genetic Approach

M. SRINIVAS[a,*] and L. M. PATNAIK[b,†]

[a]*Motorola India Electronics Limited, 33A, VLSoor Road, Bangalore 560042, INDIA;* [b]*Microprocessor Applications Laboratory, Indian Institute of Science, Bangalore 560 012, INDIA*

Genetic Algorithms are robust search and optimization techniques. A Genetic Algorithm based approach for determining the optimal input distributions for generating random test vectors is proposed in the paper. A cost function based on the COP testability measure for determining the efficacy of the input distributions is discussed. A brief overview of Genetic Algorithms (GAs) and the specific details of our implementation are described. Experimental results based on ISCAS-85 benchmark circuits are presented. The performance of our GA-based approach is compared with previous results. While the GA generates more efficient input distributions than the previous methods which are based on gradient descent search, the overheads of the GA in computing the input distributions are larger.

To account for the relatively quick convergence of the gradient descent methods, we analyze the landscape of the COP-based cost function. We prove that the cost function is unimodal in the search space. This feature makes the cost function amenable to optimization by gradient-descent techniques as compared to random search methods such as Genetic Algorithms.

Keywords: Testing; Random Test Vectors; Signal Probabilities; Genetic Algorithms; COP testability measure

1. INTRODUCTION

Test techniques for detecting faults in combinational digital circuits may be broadly classified into two categories—deterministic methods and random methods. Deterministic methods make use of algorithms that exploit detailed circuit information to generate the test vectors, while random test methods employ randomly generated vectors and fault simulation to locate test vectors. The distributions of the 0s and 1s in the randomly generated vectors are determined from relatively more macroscopic features of the circuit.

Fault simulation is a key component of random test methods. A fault simulator is used to simulate the behaviour of the circuits with and without the presence of faults. Based on the output responses of the fault-free circuit and the circuit with the faults, the randomly generated vector is classified as a test vector.

A decade ago, random test methods were inefficient in comparison to deterministic methods, prima-

*Phone: 91-80-5598615. Email: msrini@turning.miel.mot.com.
†Phone: 91-80-3342451. Fax: 91-80-3341683. Email: lalit@micro.iisc.ernet.in.

rily due to two reasons : (i) high costs of fault simulation and (ii) prohibitively long test sequences for 'resistant' circuits. Recent advances have mitigated the effects of both factors. The costs of fault simulation have been steadily decreasing until date. Also the use of non-uniform distributions and multiple distributions for generating the random vectors has considerably decreased the number of vectors to be simulated for attaining maximal fault coverage.

Research on improving the performance of random test methods dates back to the seventies [2][1][16][19]. Agrawal et.al. [4] have demonstrated that the number of vectors to be simulated to detect all detectable faults in a circuit can be significantly reduced by using inequiprobable 0s and 1s in the test vectors. Schurmann et al. [19] use adaptively varying input distributions (signal probabilities) to reduce the length of test sequences. Wunderlich [22] and Lisanke [14] have formulated the problem as one of optimizing a 'cost' associated with the set of signal probabilities. In both cases, *gradient descent*-based optimization techniques have been utilized. More recently Wunderlich [24] has discussed the advantages of employing multiple distributions for generating the test vectors; both theoretical and experimental evidence point to the utility of the approach. Other related work includes [12][13][15][17][23].

In this paper we consider the problem of optimizing input distributions (of 0s and 1s) for generating random test vectors. We demonstrate the application of Genetic Algorithms (GAs), powerful search and optimization techniques, to determine the optimal input signal probabilities. Genetic Algorithms have evolved as robust optimization and search procedures for a wide range of complex problems. In several cases, GAs have located solutions better than traditional problem-specific algorithms. The task of optimizing the input signal probabilities for generating random vectors is a complex one, primarily due to the complexity of the circuits and the non-trivial interactions among the input signals.

Essential to every optimization problem is a cost that may be associated with each solution that reflects the quality of the solution. The cost function that we employ to evaluate the signal probabilities is based on the COP testability measure [5], and formulated by Lisanke et al. [14]. The efficacy of the GA in optimizing the input signal probabilities is verified by actual fault simulation of the circuits, using random vectors generated from the optimal signal probabilities. Experimental results for ISCAS-85 benchmark circuits are presented. The performance of the GA is compared with previous approaches.

To analyze the performance of the GA, we study the profile of the COP-based cost function. We prove that the COP-based cost function is unimodal, and that this feature favours gradient descent-based optimization techniques.

2. OVERVIEW OF GENETIC ALGORITHMS

Genetic Algorithms are probabilistic search methods that employ a search technique based on ideas from natural genetics and evolutionary principles. They were conceived by Holland [10] in 1975, and since then, they have emerged as general purpose, robust optimization techniques (see [7], [9] [10]).

Genetic Algorithms employ a random, yet directed search for locating the globally optimal solution. They are superior to gradient descent techniques as the search is not biased towards the locally optimal solution. On the other hand, they differ from random sampling algorithms due to their ability to direct the search towards relatively 'prospective' regions in the search space.

Typically a Genetic Algorithm is characterized by the following components:

- a genetic representation (or an encoding) for the feasible solutions to the optimization problem
- a population of encoded solutions
- a fitness function that evaluates the optimality of each solution
- genetic operators that generate a new population from the existing population
- control parameters

The GA may be viewed as an evolutionary process wherein the population of feasible solutions to the optimization problem evolves over a sequence of generations. During each generation, the fitness of each solution is evaluated, and solutions are *selected* for reproduction based on the relative fitness values. *Selection* embodies the principle of *Survival of the fittest*. 'Good' solutions are selected for reproduction while 'bad' solutions are eliminated. The 'goodness' of a solution is determined from its fitness value. The selected solutions then undergo reproduction under the action of the *crossover* and *mutation* operators. It has to be noted that the genetic representation may differ considerably from the natural form of the parameters of the solutions. While fixed-length and binary encoded strings for representing solutions have dominated GA research since they provide the maximum number of schemata, more efficient encodings based on the nature of the parameters of the problem are also common. [8] provides some theoretical perspectives for the choice of low cardinality alphabets for encoding the solutions.

Crossover causes the exchange of 'genes' between two randomly selected 'parents' to form new 'offspring'. The crossover occurs only with some probability (the crossover rate), and when the solutions are not subjected to crossover, they remain unmodified. The intuition behind crossover points to the possibility of structured exchange of 'genes' between 'good' solutions to form 'better' offspring. The notable crossover techniques include single-point, two-point, and uniform types [21].

Mutation involves the modification of the values of each 'gene' of a solution with some probability (the mutation rate). Though the traditional role of mutation in GAs has been that of restoring lost or unexplored genetic material into the population, recent reports [18] have shown that mutation can play an important role in the working of GAs. Apart from selection, crossover, and mutation, various other auxiliary operations are common in GAs. Of these, *scaling* mechanisms are widely used. Scaling involves a readjustment of fitness values of solutions to sustain a steady selection pressure in the population, and to prevent the premature convergence of the population to suboptimal regions in the search space.

Our Implementation

In the GA proposed by Holland [10], the feasible solutions are encoded using binary strings. The crossover and mutation operators are specifically designed for operating on binary strings. Recent GA research has explored the possibility of more efficient operators and encodings (see [6]). For problems defined on continuous spaces, real valued variables (genes) implemented using floating point arithmetic use less CPU time, and also provide a more natural mapping of the problem parameters.

In our implementation of the GA, we use real valued variables to represent the input signal probabilities. We utilize an improvised uniform crossover operator [21], wherein the variables of the offspring are obtained by averaging the corresponding variables of the parent solutions. Note that not all variables are averaged. The crossover occurs with a probability p_c. When solutions are not subjected to crossover, the variables are passed on to the offspring unmodified. The mutation operator modifies the value of a variable by a randomly generated number in the range 0-1. Mutation of a variable also occurs with a probability p_m. We employ adaptive variation of p_c and p_m to simultaneously improve the exploratory and exploitative capacities of the GA [20]. p_c and p_m are evaluated for each solution based on its fitness in the following fashion:

$$p_c = k_1(f_{max} - f')/(f_{max} - \bar{f}), \quad f' \geq \bar{f} \tag{1}$$

$$p_c = k_3, \quad f' \leq \bar{f} \tag{2}$$

and

$$p_m = k_2(f_{max} - f)/(f_{max} - \bar{f}), \quad f \geq \bar{f}, \tag{3}$$

$$p_m = k_4, \quad f \leq \bar{f}, \tag{4}$$

where

f is the fitness of a the solution
$\bar{f}$ is the average fitness of the population
f' is the higher of the fitnesses of the two solutions undergoing crossover
f_{max} is the maximum fitness in the population

We have employed the following values for the constants: $k_1 = k_3 = 0.5$, $k_2 = k_4 = 1.0$. We have arrived at these values for the constants after extensive experimentation. More details, together with a detailed analysis of the adaptive approach for varying p_c and p_m may be obtained in [20].

3. COP BASED COST FUNCTION

Essential to any optimization problem is a *cost* that is associated with each feasible solution. The choice of the cost function is critical as the cost should reflect the *quality* of the solution as truly as possible. The optimization problem under consideration is that of finding the optimal set of input signal probabilities that minimize the total CPU time required to detect all faults.

Lisanke et al. [14] have devised a cost function that estimates the total time required for detecting all faults as a function of the input signal probabilities. The cost function is based on the COP testability measure [5]. COP assesses the testability of the circuit by determining the controllabilities and observabilities at each fault site in the circuit. For more details about COP, please refer to [5].

The COP based cost function is derived as follows. For a specific test vector, the probabilities of detection of sa0 (stuck-at-zero) and sa1 (stuck-at-one) faults at a signal i are given respectively by:

$$Pd_{i/0} = C_i O_i \tag{5}$$

and

$$Pd_{i/1} = (1 - C_i)\, O_i \tag{6}$$

where C_i and O_i are the 1-controllability and observability of the signal i.

The probability of detecting a fault j after applying n random vectors is given by

$$CPd_j(n) = 1 - (1 - Pd_j)^n \tag{7}$$

The percentage of all faults detected (the fault coverage) after applying n vectors is,

$$FCE(n) = \frac{1}{M} \sum_{j=1}^{M} CPd_j(n) \tag{8}$$

where M is the number of undetected faults in the circuit.

Since the total time required for fault simulation for a single vector is proportional to the number of undetected faults, $1 - FCE$, the total time required to detect all possible faults is proportional to

$$T = \sum_{n=0}^{\infty} 1 - FCE(n) \tag{9}$$

The above expression reduces to the following form:

$$T = \frac{1}{M} \sum_{n=0}^{\infty} \sum_{n=0}^{M} (1 - Pd_j)^n \tag{10}$$

Equation (10) may be further simplified to

$$T = \frac{1}{M} \sum_{n=0}^{M} \frac{1}{Pd_j} \tag{11}$$

4. EXPERIMENTS AND RESULTS

For all our experiments, we have used 9 circuits from the ISCAS benchmarks [11]. First we present results that demonstrate the decrease in costs achieved by the GA. Next the results of actual fault simulation of the circuits using random vectors that have been generated from the optimal signal probabilities are presented.

Table I gives the average cost of solutions in the initial population of the GA, and the cost of the best solutions in the 50th, 100th, and the 200th generations.

Some interesting observations may be made from Table I. The *ease* with which a circuit may be tested with test vectors generated from random signal probabilities is given by the average cost of the initial population. It may noticed that c6288 is an easy circuit, while c2670 and c3540 are resistant to random testing. The most significant decrease in the cost is seen for c880 and c2670, wherein a hundredfold reduction is evident. With the optimized signal probabilities, c880 becomes the second most easily testable circuit after c6288.

Worthwhile noting is the steady decrease in the cost for c2670, while for other circuits the improvement in cost is less significant. This indicates that the cost may be further decreased by running the GA for a longer period. After 500 generations the cost for c2670 decreased to 83.5. We may conclude from the experimental evidence that circuits that are *resistant* to test vectors with equiprobable 0s and 1s may be easily tested with vectors with nonuniformly distributed 0s and 1s. Also notable is the negligible variation in the cost for c1355.

The COP testability measure is not an exact measure of the testability of the circuit, as it is based on the assumption that the probability of detecting a fault is linearly dependent on its observability and controllability. This may not be true always. A detailed discussion of this topic is available in [3]. To make a more accurate assessment of the quality of the optimal signal probabilities, we have generated random vectors using the optimal signal probabilities, and we have performed fault-simulation of the circuits for detecting the faults. Table II gives the number of random vectors that have been generated to locate all detectable faults. We also present comparative results from two previous approaches due to Lisanke et al. [14], and Wunderlich [24]. Lisanke et al. use a gradient descent technique to obtain the optimal signal probabilities. Wunderlich has employed multiple distributions to improve the testability of random vectors.

Table II demonstrates that the performance of the GA is comparable with the other techniques. Except for c499, our results are consistently better than Lisanke's and Wunderlich's methods.

The drawback of the GA-based approach, however, is the computing overhead imposed by the GA. Since the GA works by sampling the solution space to focus its search, a considerable amount of effort is devoted to optimizing the signal probabilities. On the other hand, Lisanke's method exploits the quick convergence properties of a gradient descent technique, and an efficient way of computing the gradient of the cost function, to impose minimal computing overheads in optimizing the signal probabilities. Details of the overheads in computing the multiple distributions by Wunderlich's method are not given in [24]. However the author indicates that the overheads are substantial and needing up to one hour of CPU time. Table III presents a comparison of the CPU overheads for our GA-based approach and Lisanke's method.

Our experience is that the GAs demonstrate superior performance over gradient descent techniques

TABLE I Variation of cost with number of generations

Circuit	Ini. Avg.	50 gens.	100 gens.	200 gens.
c432	1724.13	26.999	22.55	20.65
c499	1135.71	25.67	25.46	25.28
c880	1408.45	21.81	16.11	14.19
c1355	15551.22	4325.27	4217.98	4189.36
c1908	1016.26	83.08	76.24	71.43
c2670	64935.06	2607.0	1402.42	575.11
c3540	11173.18	534.31	432.79	402.60
c5315	2702.71	79.12	67.55	58.93
c6288	28.57	8.50	8.27	8.21

TABLE II Number of vectors required for detecting all detectable faults

Circuit	Coverage	GA	Lisanke	Wunderlich
c432	99.23 %	245	320	512
c499	99.89 %	758	732	539
c880	100.00 %	186	160	260
c1355	99.49 %	2122	2784	2244
c1908	99.52 %	3120	3916	2308
c2670	95.48 %	6335	6400	10766
c3540	96.00 %	4021	4352	12220
c5315	98.89 %	843	1024	1316
c6288	99.59 %	65	—	109

when the landscape of the cost function is complex and multimodal. On simpler, unimodal landscapes, gradient descent techniques outperform the GA as the global optimum coincides with the local optimum. The relatively quick convergence of Lisanke's gradient descent method to the optimal solutions indicates that the search space is a smooth and a unimodal one. To our knowledge, no characterization of the COP based cost function has been made earlier in the literature. In the next section we analyze the cost function to determine the nature of its landscape.

5. ANALYSIS OF THE COP BASED COST FUNCTION

The COP-based cost function may be expressed as,

$$T(s_0, s_1, s_2, \ldots, s_I) = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{Pd_j} \tag{12}$$

TABLE III Cost overheads of the optimization technique: Cost of fault simulation is 1 unit

Circuit	GA	Lisanke
c432	3.934	0.504
c499	3.944	0.438
c880	4.352	0.465
c1355	3.774	0.088
c1908	3.535	0.101
c2670	4.039	0.082
c3540	4.071	0.124
c5315	3.319	0.070
c6288	1.3735	—

where $s_i s$ are the signal probabilities and I is the number of primary inputs.

We are interested in characterizing the minima of T. Hence, we evaluate the first and the second partial derivatives of t with respect to s_i.

Consider a single fault j, and the cost associated with j:

$$T_j = \frac{1}{O_j C_j}, \quad j: sa0 \tag{13}$$

$$T_j = \frac{1}{O_j(1.0 - C_j)}, \quad j: sa1 \tag{14}$$

Since we need to evaluate $\frac{\partial T_j}{\partial s_i}$, we also need to evaluate $\frac{\partial C_j}{\partial s_i}$ and $\frac{\partial O_j}{\partial s_i}$. Table IV gives the expressions for controllabilities and observabilities of the various gates as a function of their input controllabilities and output observabilities.

To evaluate the partial derivatives of T with respect to s_i, we may assume s_k, $k \neq i$, to be constants. Consequently the controllabilities for the gates connected directly to the primary inputs can be expressed in the following form:

$$C_j = \alpha_j + \beta_j s_i \tag{15}$$

where α_j and β_j are specific to the gate.

Consider a gate j that is not connected directly to the primary inputs. The signal s_i can propagate only to one input of j. The controllabilities of the other inputs do not vary with s_i, and thus may be treated as

TABLE IV Expressions for Controllabilities and Observations

Gate	C_i	O_i
AND	$\Pi_{k \in inputs} C_k$	$O_{out} \Pi_{k \in inputs, k \neq i} C_k$
NAND	$1 - \Pi_{k \in inputs} C_k$	$O_{out} \Pi_{k \in inputs, k \neq i} C_k$
OR	$1 - \Pi_{k \in inputs} (1 - C_k)$	$O_{out} \Pi_{k \in inputs, k \neq i} (1 - C_k)$
NOR	$\Pi_{k \in inputs} (1 - C_k)$	$O_{out} \Pi_{k \in inputs, k \neq i} (1 - C_k)$
NOT	$-C_{in}$	O_{out}
XOR	$C_{in1}(1 - C_{in2}) + C_{in2}(1 - C_{in1})$	O_{out}

constants. Thus we again get a linear relationship between C_j and C_{in}, the controllability of the input to which the signal propagates:

$$C_j = \alpha'_j + \beta'_j C_{in} \tag{16}$$

We may apply Equation (15) recursively until C_j may be expressed directly in terms of s_i. Consequently the cost C_j for any fault varies linearly with s_i. Equation (15) represents this relation.

On similar lines, it may be shown that the observability O_j is also linearly dependent on s_i. We represent this linear relationship as follows:

$$O_j = \gamma_j + \delta_j s_i \tag{17}$$

It follows that

$$T_j = \frac{1}{(\gamma_j + \delta_j s_i)(\alpha_j + \beta_j s_i)}. \tag{18}$$

It may be noted that T_j is a positive quantity.

The second partial derivative of T_j with respect to s_i simplifies to the following expression:

$$\frac{\partial^2 T_j}{\partial s_i^2} = T_j \left[\frac{\beta_j^2}{(\alpha_j + \beta_j s_i)^2} + \frac{\beta_j \delta_j}{(\alpha_j + \beta_j s_i)(\gamma_j + \delta_j s_i)} + \frac{\delta_j^2}{(\gamma_j + \delta_j s_i)^2} \right] \tag{19}$$

The right hand side of Equation (19) may be further simplified to

$$T_j \frac{\left[\left(\frac{\beta_j}{(\alpha_j + \beta_j s_i)} \right)^3 - \left(\frac{\delta_j}{(\gamma_j + \delta_j s_i)} \right)^3 \right]}{\left[\frac{\beta_j}{(\alpha_j + \beta_j s_i)} - \frac{\delta_j}{(\gamma_j + \delta_j s_i)} \right]} \tag{20}$$

In the above expression, T_j is positive. Also, the numerator and denominator of the second term have the same sign. Consequently $\partial^2 T_j / \partial s_i^2$ is always positive.

It follows that

$$\frac{\partial^2 T}{\partial s_i^2} = \sum_{j=1}^{M} \frac{\partial^2 T_j}{\partial s_i^2} \tag{21}$$

is also positive.

Let T have a minimum at $s_i = s'_i$. It implies that $\partial T / \partial s_i = 0$ at $s_i = s'_i$. Since $\partial^2 T / \partial s_i^2 \leq 0$, the following expressions are true:

$$s_i < s'_i \Rightarrow \frac{\partial T}{\partial s_i} < 0 \tag{22}$$

$$s_i > s'_i \Rightarrow \frac{\partial T}{\partial s_i} > 0 \tag{23}$$

Expressions (22) and (23) imply that T can only have one minimum when s_i is varied. The same conclusion is true for any s. It follows that T has only one optimum in the search space, since the above analysis does not make any assumptions about the values of s_k, $k \neq i$.

The above observation has also been verified experimentally. For c432 with all signal probabilities held at 0.5, we have plotted the variation of the cost with each signal probability. We have observed that in each case, either the cost increases or decreases monotonically, or has a single minimum.

6. CONCLUSIONS

In this paper we have proposed a Genetic approach to optimize the input signal probabilities for generating random test vectors. We have simulated the circuits from ISCAS-85 benchmarks using random vectors generated from the optimal signal probabilities. Our GA-based approach produces solutions that are in general better than the previous methods, but imposes higher computing overheads than the gradient descent methods. Recent research has established that GAs are efficient in searching highly convoluted and multimodal search spaces, whereas traditional gradient descent techniques get stuck at local optima. However, on smooth unimodal functions, gradient descent techniques use the gradient information efficiently to quickly converge to the optimum. In this work, the superior performance of the gradient descent method over the GA was unexpected, and this motivated us to investigate the landscape of the COP-based cost function. The analysis clearly shows that the COP based cost function varies smoothly and that it is unimodal.

This paper also opens up the issue of the choice of cost functions for various optimization problems. The experience presented in this paper clearly shows that even for complex problems, the right choice of a cost function could simplify the task of locating the optimal solution to a great extent. Also, we observe that the choice of the optimization algorithm should be based on the characteristics of the cost function. One simple approach is to look at the landscape of the function, and choose a search-based technique or a gradient-based technique accordingly. Future work needs to be carried out in this area, to realize the possibility of significant improvements in the performance of optimization methods tailored specifically to the problem under consideration.

References

[1] P. Agrawal and V. Agrawal, 'Probabilistic Analysis of Random Test Generation Method for Irredundant Combinational Logic Circuits', IEEE Transactions on Computers, Vol. C-24, July 1975, pp. 691–695.

[2] P. Agrawal and V. Agrawal, 'On Monte Carlo Testing of Logic Tree Networks', IEEE Transactions on Computers, Vol. C-25, June 1976, pp. 664–667.

[3] V. Agrawal, 'Testability Measures—What do they tell us', Proc. of the International Test Conference, 1982, pp. 391–396.

[4] V. Agrawal, 'When to use Random Testing', IEEE Transactions on Computers, Vol. C-27, Nov. 1978, pp. 1054–1055

[5] F. Brglez, 'On Testability Analysis of Combinational Networks', IEEE International Symposium on Circuits and Systems, 1984, pp. 221–225.

[6] L. Davis *Handbook of Genetic Algorithms*, Van Nostrand Reinhold Publishers, 1991.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[8] D.E Goldberg, 'Real-Coded Genetic Algorithms, Virtual Codes, and Blocking', University of Illinois at Urbana Champaign, Technical Report No. 90001, 1990.

[9] J. J. Grefenstette, 'Optimization of Control Parameters for Genetic Algorithms', IEEE Transactions on Systems, Man, and Cybernitics, Vol. SMC-16, No.1, pp. 122–128, Jan./Feb. 1986.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor : University of Michigan Press, 1975.

[11] "Special Session: Recent Algorithms for Gate-Level ATPG with Fault Simulation and their Performance Assessment", Proc. of the 1985 IEEE Int. Symp. Circuits and Systems (ISCAS), June 1985, pp. 663–698.

[12] B. Krishnamurthy et.al 'Improved Techniques for Estimating Signal Probabilities', Proc. of the International Test Conference, 1986, pp. 244–251.

[13] K. J. Lieberherr, 'Parameterized Random Testing', Proc. of the 21st Design Automation Conference, 1984, pp. 510–516.

[14] Lisanke et al., 'Testability Driven Random Test-Pattern Generator', IEEE Transactions on CAD, Vol. CAD-6, Nov. 1987, pp. 1082–1087.

[15] V. Muradali, V.K Agrawal, 'A New Procedure for Weighted Random BIST', Proc. of International Test Conference, 1990, pp. 661–669.

[16] K. P. Parker and E.J. McCluskey, 'Probabilistic Treatment of General Combinational Networks', IEEE Transactions on Computers, Vol. C-24, June 1975, pp. 668–670.

[17] J. Savir et al., 'Random Pattern Testability', IEEE Transactions on Computers, Vol. C-33, Jan. 1984, pp. 79–90.

[18] J. D. Schaffer et.al., 'A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization' Proc. of the Third International Conf. Of Genetic Algorithms, 1989, pp. 51–60.

[19] H. D. Schurmann, 'The Weighted Random Pattern Test Generator', IEEE Transactions on Computers, Vol. C-24, July 1975, pp. 695–700.

[20] M. Srinivas and L. M. Patnaik, 'Adaptive probabilities of Crossover and Mutation in Genetic Algorithms', IEEE Trans. on Systems, Man and Cybernetics, Vol. 24, No. 4, pp. 17-26, April 1994.
[21] G. Syswerda, 'Uniform Crossover in Genetic Algorithms', in Proc. of the Third International Conf. on Genetic Algorithms, 1989, pp. 2–9.
[22] H. J. Wunderlich, 'PROTEST : A Tool for Probabilistic Testing Analysis', Proc. of the ACM IEEE 22nd DAC, 1985, pp. 204–211.
[23] J. Waicukauski, E. Lindbloom, 'Fault Detection Effectiveness of Weighted Random Patterns', Proc. of the International Test Conference, 1988, pp. 245–250.
[24] H. J. Wunderlich, 'Multiple Distributions for Biased Test Patterns', IEEE Transactions on Computers, Vol. C-9, June. 1990, pp. 584–593.

Authors' Biographies

L. M. Patnaik obtained his Ph.D in 1978 in the area of Real-Time Systems and D.Sc. in 1989 in the area of Computer Systems and Architectures. He has published over 120 papers in refereed International Journals and over 130 papers in refereed International Conference Proceedings, in the areas of Computer Architecture, Parallel and Distributed Computing, VLSI CAD, Neural Networks, Real-Time Systems, and Genetic Algorithms. He is a Fellow of the IEEE. He serves on the Editorial Boards of several International Journals including VLSI Design, The Computer Journal, International Journal of High Speed Computing, Parallel Algorithms and Applications, and Computer-Aided Design.

M. Srinivas graduated from *Indian Institute of Technology*, Madras, INDIA, in 1989. From July 1989 to July 1990, he was employed with the *Centre for Development of Advanced Computing*, Bangalore. From August 1990 to July 1993, he has been a Ph.D student at the Department of Computer Science and Automation, *Indian Institute of Science*, Bangalore. He is currently employed with Motorola India Electronics Limited. His research interests include Theory and Design of Genetic Algorithms, Neural Networks, Stochastic Optimization, and Optimization in VLSI CAD Algorithms.